

PROGRAMMEZ!

le magazine du développeur

www.programmez.com

NOUVELLE
FORMULE

Java 8 arrive enfin !

Google Glass

Le futur
de l'informatique

Oculus Rift
La réalité virtuelle
à portée de codes !

Antiquité
Amiga 500
vs Atari ST

Systèmes embarqués,
objets connectés :

les architectures, les modèles
de développement, votre futur job ?



Le langage
DART

Créer et utiliser
une Web API

Web
Le Responsive
Design

Utiliser
Hadoop

© ISTOCK - DOLGACHOV

Mensuel n°172 - Mars 2014



Printed in EU - Imprimé en UE - BELGIQUE 6,45 €
SUISSE 12 FS - LUXEMBOURG 6,45 € DOM Surf 6,90 €
Canada 8,95 \$ CAN - TOM 940 XPF - MAROC 50 DH

Une infinité d'applications une seule plate-forme

avec l'environnement de conception de systèmes le plus complet qui soit



NII LabVIEW est le seul environnement de développement qui offre autant de possibilités d'intégration matérielle ; au point de vous permettre de relever n'importe quel défi en matière d'applications de mesure et de contrôle. LabVIEW est au cœur de ce que l'on appelle la conception graphique de systèmes : une approche qui combine les avantages d'une plate-forme logicielle ouverte et d'un matériel reconfigurable à l'infini, pour que vous développiez votre système en un minimum de temps.

Le logiciel de conception de systèmes LabVIEW offre une intégration matérielle inégalée et vous permet de programmer de la façon dont vous pensez : graphiquement.



>> Concevez beaucoup plus rapidement vos systèmes en visitant ni.com/labview-platform/

01 57 66 24 24

NATIONAL INSTRUMENTS France • 2 rue Hennepin - 92735 Nanterre Cedex, France • Tél. : (0)1 57 66 24 24 Fax : (0)1 57 66 24 14 • Société de droit américain - capital social 1.000,00 dollars • US - 11500 N Mopac Expwy, Austin-Texas USA - 10056236 - 344 497 649 - RCS Nanterre - SIRET B 344 497 649 00022 - APE 516J - N.I.I. FR 57344497649

©2012 National Instruments. Tous droits réservés. National Instruments, NI, LabVIEW, et ni.com sont des marques de National Instruments. Les autres noms de produits et de sociétés mentionnés sont les marques ou les noms de leurs propriétaires respectifs. Pour plus d'informations concernant les marques de National Instruments, veuillez vous référer à la partie Terms of Use sur le site ni.com/legal. 08371

 **NATIONAL
INSTRUMENTS™**



Et si 2014 était en réalité 1984

«... vous verrez pourquoi 1984 ne sera pas comme 1984 ». C'était la conclusion de la mythique publicité Apple réalisée pour le Macintosh par Ridley Scott. Le « Big Brother » de 1984 (roman de G. Orwell) ne s'est pas réalisé, enfin, pas selon le récit. Mais il faut garder à l'esprit cette idée. Les récentes affaires de piratages de données, PRISM et autres fichiers plus ou moins légaux / illégaux doivent faire prendre conscience que nous sommes surveillés et que nos données, nos informations ne nous appartiennent plus. Et nous sommes partiellement consentants.

Le « wearable computing » est le maître-mot d'une partie de l'industrie. Il s'agit de l'informatique (matérielle et logicielle) embarquée sur des lunettes, vêtements, bagues, montres, etc. Bref, tout ce qui nous touche au quotidien, en dehors du PC, du téléphone. Les Google Glass en sont un parfait exemple. Le wearable computing cherche à compléter, étendre l'Homme, ses capacités, ses accès à l'information. Le potentiel est bien plus important que les objets connectés, eux aussi promis à un avenir radieux.

A Programmez !, nous nous sommes posé la question de la place et du rôle du développeur. Les Google Glass, et tout autre objet wearable, ne sont rien s'il n'y a pas d'usages, d'applications, d'interactivité. Google a dévoilé plusieurs SDK pour pouvoir étoffer l'offre logicielle qui reste encore très restreinte (un peu normal, les lunettes ne sont pas encore en vente). Mais encore faut-il que le développeur ait une liberté d'action, ce qui n'est pas forcément le cas. Le potentiel existe, cela ne fait aucun doute mais le plus difficile sera de passer du marché potentiel au marché réel. Et le pari est loin d'être gagné. Nous pourrions dire un peu la même chose des lunettes Oculus Rift. Impressionnant pour la réalité augmentée et la 3D immersive, il faudra dépasser l'aspect gadget. Un des marchés phares : le jeu vidéo qui prendra une toute autre dimension.

Nous ne nous arrêterons pas à ces lunettes ! Ce mois-ci, nous vous proposons un contenu très riche : responsive design, développement embarqué, GPS sur Windows Phone 8, développement API, asynchronisme, langage DART (à ne pas louper !), Hadoop... Et à quelques semaines de sa sortie officielle, nous faisons le point sur Java 8.

Pour conclure ce numéro, nous vous réservons une battle d'enfer : Atari ST vs Amiga 500 ! Cela va réveiller de nombreux souvenirs et des débats sans fin entre les partisans des deux plateformes. Je vous préviens, j'étais un Atari ST pur jus !

J'ai failli oublier. Notre nouveau site web est en ligne depuis le 17 février : ouverture, communauté, partage !

Enjoy !

François Tonic

Directeur de la publication & rédacteur en chef
ftonic@programmez.com

sommaire

21	Matériel	32	Systèmes embarqués et objets connectés		
39	Un Hadoop sinon rien	52	Oculus Rift	4	Tableau de bord
24	Java 8	77	Responsive Design	47	Carrière
22	Le développeur du mois	68	Langage DART	49	Google Glass
19	Hacking	55	GPS et Windows Phone 8		
8	Une vie à la Silicon Valley #5				
12	Progress Software	63	Asynchronisme en C#	72	ASP.Net MVC 5
10	Journal d'un développeur	74	IntuiFace	58	API
		81			Atari ST vs Amiga 500

À LIRE DANS LE PROCHAIN NUMÉRO

n° 173 en kiosque le 29 mars 2014

Développeur indépendant

Pourquoi ne pas créer sa propre société de développement ? Avantages et inconvénients d'être son propre patron !

Unity3D vs Unreal Engine

Il existe de nombreux moteurs 3D. Mais il n'est pas toujours facile de choisir...

Refactoring

Le refactoring est une fonction de base que le développeur doit maîtriser. Où et comment l'utiliser

Office sur iPad :

viendra, viendra pas, Microsoft semble hésiter de plus en plus. Office est l'un des arguments pour les tablettes Surface...

4K :

les premières cartes SD pour stocker le 4K arrivent chez Panasonic et SanDisk. La pertinence de ces cartes reste à démontrer surtout qu'elles sont chères (entre 84 et 360 € selon la capacité).

Le code source trahit les nouveaux modèles des constructeurs...

Où comment lancer une rumeur avec une référence produit inconnue qui peut être vraie ou simplement une blague des développeurs...

Tizen :

malgré la lenteur des constructeurs à sortir les smartphones compatibles, de nouveaux partenaires arrivent (Baidu, Sprint, ZTE...)

Android

domine le marché des montres connectées selon Strategy Analytics. Il faut dire que le marché est à peine émergeant et aucun grand concurrent n'est encore présent.

Salesforce

annonce un 2e datacenter en Europe. Il sera en France

La technologie iBeacon

va arriver en France. NFC tremble-t-il ?

(Android) KitKat ne plait pas :

1,8 % du parc...

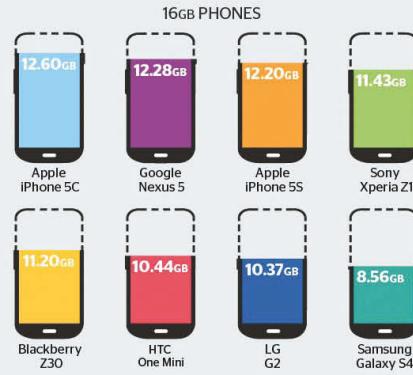
Miroir, miroir, quel est le langage le plus utilisé ?

L'index TIOBE donne chaque mois les langages les plus utilisés par les développeurs. Il s'agit d'un indicateur basé sur les recherches web. Des % à manipuler avec précaution.

Février 2014	Février 2013	Tendance	Langage	%	Evolution (en %)
1	2	↑	C	18.334 %	+1.25%
2	1	↓	Java	17.316 %	-1.07%
3	3	↑	Objective-C	11.341 %	+1.54%
4	4	↓	C++	6.892 %	-1.87%
5	5	↓	C#	6.450 %	-0.23%
6	6	↓	PHP	4.219 %	-0.85%
7	8	↑	(Visual) Basic	2.759 %	-1.89%
8	7	↓	Python	2.157 %	-2.79%
9	11	↑	JavaScript	1.929 %	+0.51%
10	12	↑	Visual Basic .NET	1.798 %	+0.79%

Le trio de tête reste identique : C, Java et Objective-C. Ce dernier connaît une petite progression. La plupart des autres langages subissent une baisse, exceptés VB et JavaScript.

How much storage space does your phone really have?



Le stockage réel sur mon smartphone...

La polémique sur la capacité de stockage réellement utilisable sur son smartphone n'a jamais cessé, mais elle est un peu moins violente que sur les tablettes, rappelez-vous de l'affaire Surface RT... Dommage il manque les Nokia Lumia pour compléter le graphique...

Quel bilan pour Windows 8 ?

Windows 8 sera-t-il le nouveau Windows Vista de Microsoft ? Microsoft annonce 200 millions de licences vendues après 15 ans de disponibilité, soit 40 millions de moins que Windows 7. Windows 8.1 est là pour rebooster la plate-forme qui plafonne à 11 % des utilisateurs PC...

Déchiffrement à la volée

Plus une entreprise est privée de ses données, plus le coût est élevé. La nouvelle technologie de déchiffrement automatique Kroll Ontrack permet aux ingénieurs de cibler uniquement les zones utilisées du disque dur tout en automatisant le processus de déchiffrement - réduisant ainsi à quelques heures, au lieu de quelques jours, le temps d'une récupération.



Les tablettes en France

Les analystes de GTK ont livré une étude sur les tablettes en France :

- 6,2 millions de tablettes vendues (et seulement 4,8 millions de PC)
- le prix moyen chute lourdement : 240 €
- Apple tient toujours la tête, Samsung n'est plus très loin
- Les Surface (Microsoft) sont (toujours) très confidentielles
- Android domine le marché avec une offre très nombreuse. iOS est logiquement 2e, Windows 8 / Windows RT sont toujours très loin derrière

Fragmentation Android

Version	API	Distribution
2.2 Froyo	8	1.3%
2.3.3 -		
2.3.7 Gingerbread	10	20.0%
3.2 Honeycomb	13	0.1%
4.0.3 -		
4.0.4 Ice Cream Sandwich	15	16.1%
4.1.x Jelly Bean	16	35.5%
4.2.x	17	16.3%
4.3	18	8.9%
4.4 KitKat	19	1.8%

Source : Google

WORLDWIDE MOBILE PHONE SALES TO END USERS BY VENDOR IN 2013 (THOUSANDS OF UNITS)

Company	2013 Units	2013 Market Share	2012 Units	2012 Market Share
Samsung	444,444.2	24.6%	384,631.2	22.0%
Nokia	250,793.1	13.9%	333,938.0	19.1%
Apple	150,785.9	8.3%	130,133.2	7.5%
LG	69,024.5	3.8%	58,015.9	3.3%
ZTE	59,898.8	3.3%	67,344.4	3.9%
Huawei	53,295.1	2.9%	47,288.3	2.7%
TCL	49,531.3	2.7%	37,176.6	2.1%
Lenovo	45,284.7	2.5%	28,151.4	1.6%
Sony	37,595.7	2.1%	31,394.2	1.8%
Yulong	32,601.4	1.8%	18,557.5	1.1%
Others	613,710.0	34.5%	609,544.9	34.9%
Total	1,806,964.7	100.0	1,746,175.6	100.0

Source: Gartner (February 2014)

Le marché mondial du téléphone portable

En 2013, Samsung a vendu 444 millions de téléphones (du plus basique aux smartphones) soit presque 25 % du marché mondial ! Nokia est 2e avec 250 millions mais en encaissant une sévère chute : - 80 millions de téléphones vendus. Et 3e, nous retrouvons Apple, qui avec uniquement l'iPhone, réussit à vendre 150 millions de téléphones (8,3 % du marché). Derrière, les concurrents se contentent des miettes et le marché est de plus en plus morcelé.

Tu veux 5 To sur un disque dur ?

Toshiba livre les premiers disques durs 5 To pour tout le monde !

Google et VMware virtualisent Windows... :

grâce à la plate-forme de virtualisation VMware Horizon, Google veut booster les fonctionnalités de ChromeOS en proposant... Windows !

Mais Microsoft pourrait faire mieux

en proposant les applications Android sur Windows Phone !

Langage R :
Amazon Web Service aime aussi le langage R

2,5 kg de bonbons
ont été distribués durant les TechDays 2014 sur le stand de Programmez !

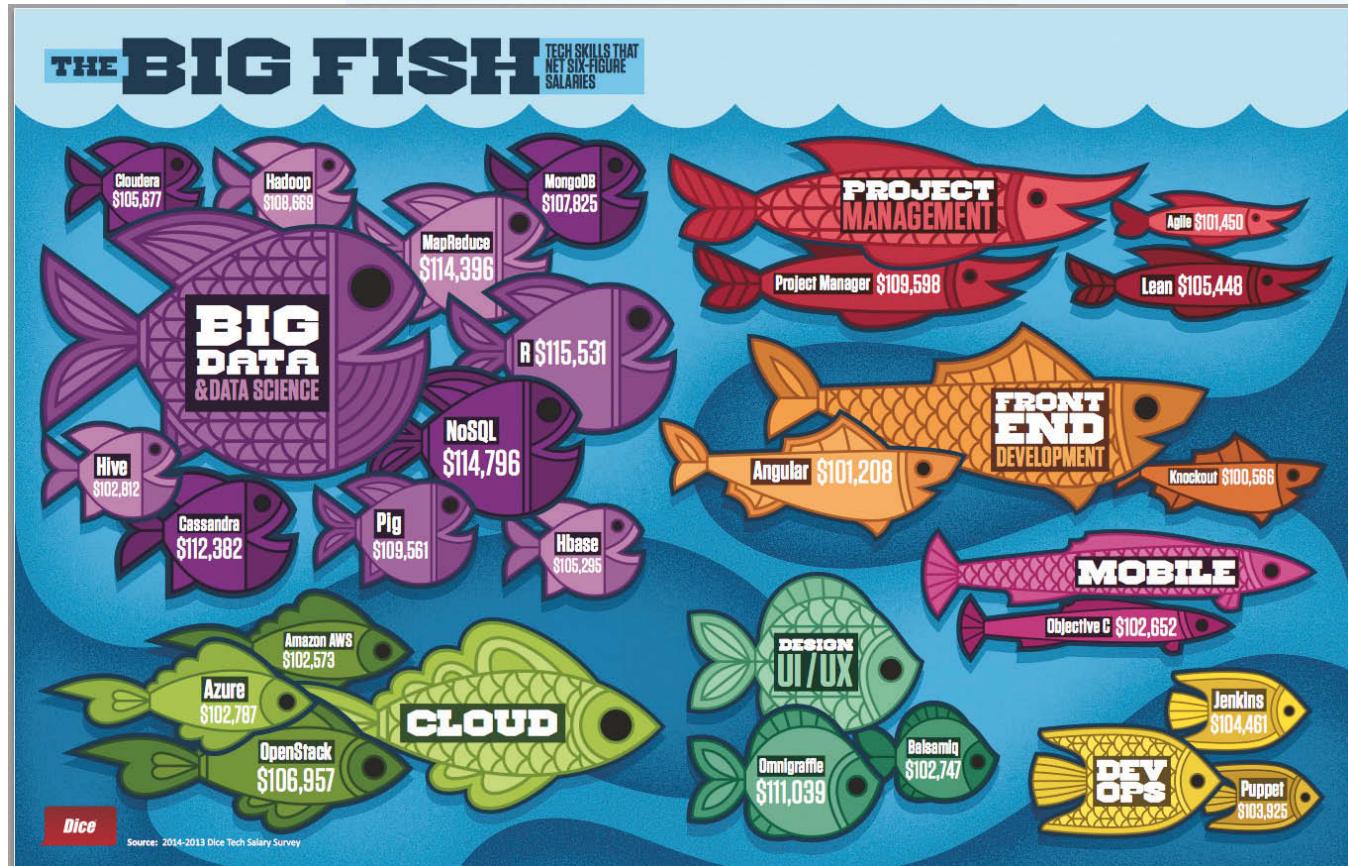
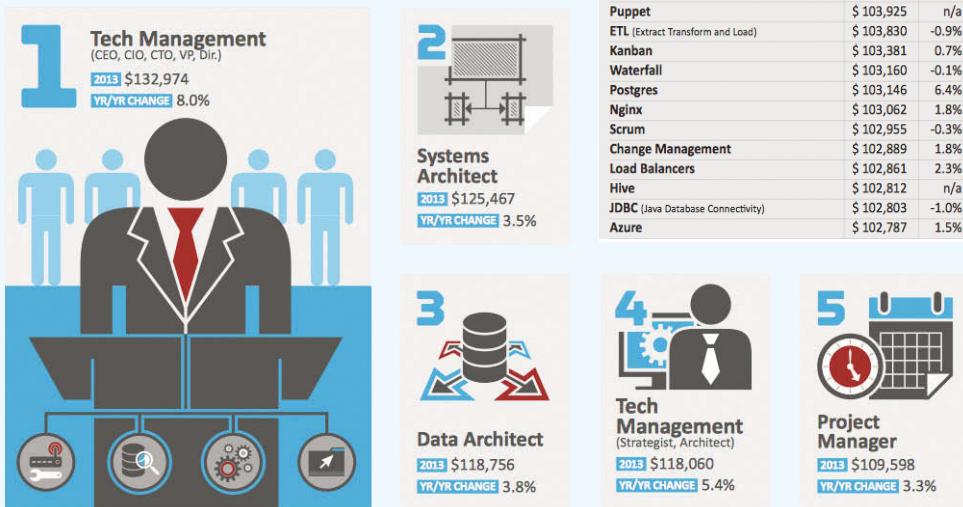
Les salaires des développeurs aux Etats-Unis

Dice a sorti une étude complète sur les salaires des développeurs américains. Premier constat : si tu habites à la Silicon Valley, tu seras mieux payé qu'à Saint Louis (108 603 \$ contre 76 220 \$).

Bien entendu, les profils influencent beaucoup le salaire. L'ingénieur développeur arrive à 97 000 \$, un top manager pourra atteindre 132 000 \$. La technologie sera elle aussi un argument pour gagner, ou moins. Par exemple, un expert langage R peut espérer 115 000 \$, un développeur Java risque de se contenter de 97 000 \$...

AVERAGE SALARY FOR High Paying Skills and Experience

SKILL	2013	YR/YR CHANGE
R	\$ 115,531	n/a
NoSQL	\$ 114,796	1.6%
MapReduce	\$ 114,396	n/a
PMBok	\$ 112,382	1.3%
Cassandra	\$ 112,382	n/a
Omnigraffle	\$ 111,039	0.3%
Pig	\$ 109,561	n/a
SOA (Service Oriented Architecture)	\$ 108,997	-0.5%
Hadoop	\$ 108,669	-5.6%
Mongo DB	\$ 107,825	-0.4%
SOX (Sarbanes-Oxley)	\$ 107,697	4.8%
Jetty	\$ 107,406	0.4%
UML (Unified Modeling Language)	\$ 107,387	4.7%
OpenStack	\$ 106,957	n/a
Big Data	\$ 106,542	-6.3%
CMMI (Capability Maturity Model Integration)	\$ 106,508	2.1%
3Par	\$ 106,432	7.4%
FCoE (Fibre Channel Over Ethernet)	\$ 106,062	2.7%
Cloudera	\$ 105,677	n/a
Lean	\$ 105,448	1.8%
Hbase	\$ 105,295	n/a
Solaris	\$ 104,710	3.7%
Jenkins	\$ 104,461	n/a
Puppet	\$ 103,925	n/a
ETL (Extract Transform and Load)	\$ 103,830	-0.9%
Kanban	\$ 103,381	0.7%
Waterfall	\$ 103,160	-0.1%
Postgres	\$ 103,146	6.4%
Nginx	\$ 103,062	1.8%
Scrum	\$ 102,955	-0.3%
Change Management	\$ 102,889	1.8%
Load Balancers	\$ 102,861	2.3%
Hive	\$ 102,812	n/a
JDBC (Java Database Connectivity)	\$ 102,803	-1.0%
Azure	\$ 102,787	1.5%



Le développeur d'apps en 2014

VisionMobile a dévoilé sa 6e étude « Developer Economics » sur les dernières tendances du développement d'application. Quelles sont les tendances ? Quelles plateformes dominent ?

Cette étude est menée sur 7 000 développeurs d'applications dans 127 pays. En 2013, les apps ont pesé 68 milliards \$, en 2016, le marché pourrait monter à 143 milliards \$. Plus que jamais, Android et iOS dominent outrageusement : 94 % des ventes de smartphones. iOS reste dynamique en Europe et en Amérique du Nord mais se fait distancer en Asie du Sud, Amérique du Sud, Afrique et Moyen Orient, HTML 5 y est 2e... Les développeurs iOS demeurent les plus fidèles à la plateforme. Second constat : HTML 5 est l'environnement prioritaire... Les tablettes attirent 83 % des développeurs d'apps mais demeurent une plateforme encore peu ciblée pour un déploiement en premier. En Europe, l'étude estime à 680 000 les développeurs d'apps, contre + 760 000 en Asie.

Les plateformes de développement

Le constat est sans appel pour le 1er trimestre 2014 (niveau global) :

- ▶ 71 % pour Android : tendance stable
- ▶ 55 % pour iOS : en légère baisse
- ▶ 52 % pour HTML 5 : tendance stable. A noter que HTML 5 inclut le développement pur HTML 5 et le développement hybride (HTML 5 + natif)
- ▶ Les systèmes Windows pèsent 47 % mais attention, l'étude sépare Windows 8 et Windows Phone 8 car Microsoft propose deux systèmes mobiles alors que Android et iOS ne proposent qu'un unique système.

Individuellement, Windows 8 et Windows Phone 8 sont très loin des 3 premières plateformes. Mais Microsoft a tout de même réussi à devenir une plateforme qui compte pour le développeur d'apps. C'est sans doute la plateforme qui a le potentiel le plus important à court terme. Les autres (Tizen, FirefoxOS) sont anecdotiques et ont tout à prouver. Sans surprise, HTML 5 est devenu une plateforme incontournable pour le développement mobile. L'approche hybride est en pleine explosion.

Il est intéressant de noter que Windows Phone / Windows 8 (20 % et 18 %) sont les plateformes qui ont la plus forte attractivité d'adoption par un développeur d'app. Là encore, la dualité joue. Cela monte aussi le potentiel de Windows sur la mobilité. 10 % (total) des développeurs interrogés disent vouloir abandonner ces plateformes dans le futur contre 2 % pour iOS,

Android et HTML 5. Ces trois plateformes continuent à attirer de nouveaux développeurs (16 - 15 %).

Petite surprise, Mozilla peut se targuer d'une belle curiosité pour son Firefox OS même si le marché est inexistant à l'heure actuelle. Par contre BlackBerry subit une nouvelle érosion et le ratio adoption / abandon est négatif. Pour le moment, Ubuntu Touch est inexistant.

Si nous regardons la géographie, l'Europe de l'Ouest reste très attachée à Android, iOS, HTML 5 (tendance globale). Même si là encore, on observe la percée de Windows / Windows Phone. Mais le « iOS first » et le « Android first » demeurent une réalité sur l'ensemble des marchés.

Courant 2014, Windows 8 / Windows Phone 8 devraient remettre en cause cela.

La partie économique est une composante très importante. Et là, iOS conserve son avance. Le revenu moyen généré par la plateforme est de 750 \$ contre 150 pour Android et 25 pour Windows Phone.

Quid des tablettes ?

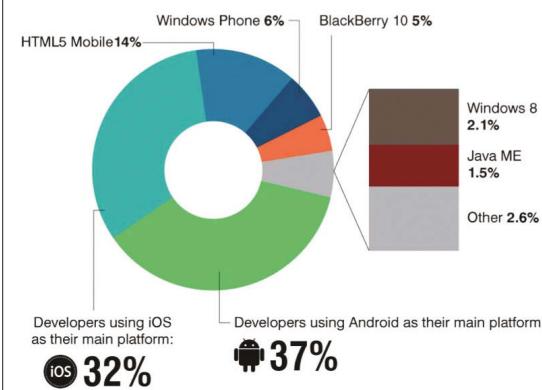
Le développement sur tablettes demeure en retrait par rapport à celui sur smartphones. Le marché y est moins développé. Et de nombreuses applications mobiles fonctionnent malgré tout sur les deux types de

terminaux. Actuellement, les développeurs iOS sont plus volontiers développeurs sur iPad que ceux sur Android. Mais là encore, le marché évolue. Les tablettes Android dominent désormais le marché grâce aux nombreuses tablettes à bas prix.

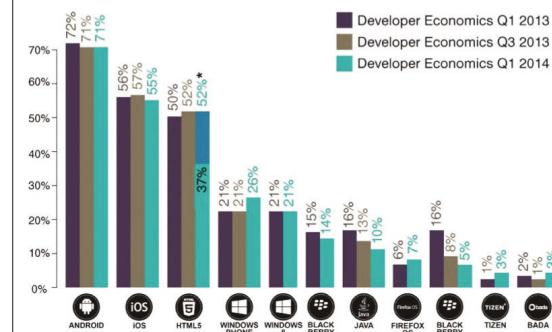
Et les outils ?

Un développeur mal outillé fera un mauvais développement. C'est encore plus vrai pour les

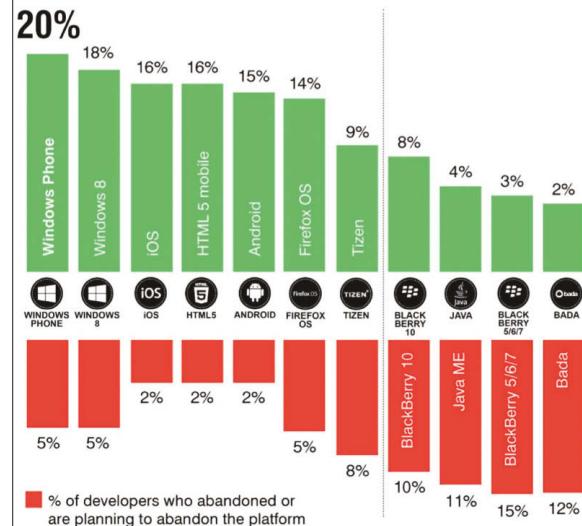
DEVELOPER LOYALTY SPLIT BETWEEN IOS & ANDROID
% of developers using each platform as their main platform (n=6,311)



MOBILE DEVELOPER MINDSHARE, Q1 2014
% of developers using each mobile platform (n=6,311)



% of developers who intent to adopt the platform



© VisionMobile, Developer Economics Q1 2014

apps. Une tendance lourde qui se dessine est l'usage des environnements pour créer des apps multiplateformes (PhoneGap, Xamarin, Appcelerator). 30 % des développeurs disent les utiliser. Les moteurs et outils dédiés aux jeux sont eux aussi très utilisés.

Etude complète :

<http://www.developereconomics.com/reports/q1-2014/>

La rédaction

SERVEURS DÉDIÉS

PREMIÈRE MONDIALE CHEZ 1&1



TOUT NOUVEAU ET DÉJÀ CHEZ 1&1 :

INTEL® ATOM™
1&1 SERVEUR DÉDIÉ A8i

À partir de

39,99
€ HT/mois
47,99 € TTC*

NOUVEAU : 1&1 SERVEUR DÉDIÉ A8i AVEC 30 % DE PERFORMANCE EN PLUS

- Intel® Atom™ C2750
- 8 Coeurs et 8 Go de RAM
- 2 x 1 To SATA HDD
- Parallels® Plesk Panel 11
- Linux, Windows ou Clé-en-main
- Bande passante 100 Mbps
- Architecture 64 bits
- System-on-Chip (SoC) :
30 % de performance supplémentaire



0970 808 911
(appel non surtaxé)



1and1.fr

* Le serveur dédié A8i est à partir de 39,99 € HT/mois (47,99 € TTC) pour un engagement de 24 mois. Également disponible avec une durée d'engagement de 12 mois ou sans durée minimale d'engagement. Frais de mise en service : 49 € HT (58,80 € TTC). Conditions détaillées sur 1and1.fr. Intel, le logo Intel, Intel Atom et Intel Inside sont des marques commerciales d'Intel Corporation aux États-Unis et/ou dans d'autres pays.

Chapitre 5 : Loi de Moore et Singularité Technologique, bullshit marketing ?!

Au travers des précédents chapitres, nous avons eu l'occasion de parcourir les différents domaines clés ou montants de la Silicon Valley, rappelez-vous les ruptures parcourues : l'automobile avec Tesla. Nous avons ensuite parcouru l'impression 3D rendant le pouvoir à la créativité unitaire à chaque individu... Cette seconde rupture étant d'ailleurs de plus en plus liée à la première citée juste avant, avec un volume de plans de pièces prêtes à imprimer et disponibles sur simple téléchargement depuis les communautés.

Nous avons également fait un rapide passage par les terres du Global Brain ou encore la GAI (General Artificial Intelligence), cette quête semblant être menée par Google qui ne cesse de donner des indicateurs forts en ce sens, réels, légitimes, possibles, voulus ou désirés... IBM aussi veut sa part de gâteau avec Watson mais la GAI naîtra-t-elle réellement un jour et surtout naîtra-t-elle des dinosaures ou mastodontes de la technologie ? Pas certain...

Rappelez-vous de l'épisode de la photo numérique et de la descente aux abysses les plus profondes de Kodak alors que Instagram se faisait racheter pour le milliard de dollars ... il y a de quoi (rêver) se poser des questions ! Les règles ont changé, les frontières se sont déplacées et continuent de se déplacer ... C'est peut-être vous, oui, vous, sur votre bureau, occupé à pianoter un nouvel algorithme de classifieur naïf Bayésien révolutionnaire de data mining ... « Black Swan Theory » quand tu nous tiens !!!

Le Big Data et le Machine Learning ont également été l'une de nos investigations afin de comprendre le mouvement et les profils recherchés au sein de la Valley ! Et autant vous dire que, vous, développeur, si vous hésitez à prendre l'avion pour venir vivre l'aventure Californienne, alors arrêtez, ce qu'il vous manque ce n'est pas du temps mais des décisions !!! Et la meilleure des décisions est d'agir, le taux d'emploi des ingénieurs en programmation est proche du maximum ! Ciblez les technologies modernes telles que Hadoop et le Pig Map Reduce, MongoDB, R, Python, Scala, Go, ... sans oublier de passer à l'ère de la sémantique et de l'ontologie. Et nous avons enfin conclu sur le monde du médical et de la robotique où les enjeux sont les plus forts actuellement ! Les domaines pourraient être encore très nombreux, la Génétique et les Nanotechnologies ou encore l'industrialisation des exosquelettes avec la médiatisation toute prochaine au Brésil quand un adolescent handicapé moteur lancera le coup d'envoi du match d'ouverture de la coupe du

monde de football 2014, debout avec un exosquelette ... la mort des fauteuils roulants et la complète liberté des personnes handicapées moteurs comme le présente l'infographie suivante : Fig.1. Mais ce monde et cette accélération dans lesquels nous vivons actuellement n'ont de cesse de progresser et d'accélérer telle une courbe exponentielle ! Tels des Christophe Colomb, tentons de comprendre les origines de ce monde que nous découvrons et subissons chaque jour ...

En route vers une croissance exponentielle !

Mon arrivée à la Valley n'est pas anodine, et pour cause, Geek dès mon plus jeune âge, je me passionne pour cette accélération constante des technologies et leurs usages. Accélération excessivement bien expliquée par Gordon Moore au travers de sa réputée « Loi de Moore » Fig.2.

En 1965, un des 3 co-fondateurs de la Compagnie Intel, Gordon Moore, écrivait pour le magazine "Electronics" que le nombre de transistors sur une puce double chaque année. La question majeure à nous poser, tout comme dans le domaine de la mode et des tendances de couleurs de mode, est-ce ces conjectures qui guident les constructeurs ou est-ce les constructeurs qui vérifient ces conjectures ?!? Depuis mon premier ordinateur en 1982, en passant par le beeper, les premiers mobiles, le Psion Serie 5mx et je vous passe encore pas mal d'autres étapes par manque de place... une histoire qui doit vous être bien familière !!!! Que d'évolutions, que de ruptures ... et tous ça en « 30 ans » !!! Tout est dit, à l'heure où nous nous faisons cérébralement dépasser par le flot d'in-



Fig.1

Source : walk again project.

formations et d'actualités sans cesse croissant, pour imaginer et construire notre futur, il nous faut nous arrêter et « prendre le temps » de regarder dans le miroir les origines de cette vague déferlante de technologie qui nous conduira à son point extrême appelé couramment point de Singularité !

Conjectures et Loi de Moore

Effectivement, c'est ainsi que l'on peut les appeler puisqu'il s'agit bien plus de suppositions émises par Moore que réellement de règles pleinement établies ou de lois. Bien que celles-ci se vérifient en grande partie suite à leur proposition. Reprenons-les rapidement :

1. Première loi de Moore en 1965 (19 Avril) : Moore énonce une croissance exponentielle (x2) annuelle des semi-conducteurs. (voir l'article original de Moore "Cramming more components onto integrated circuits" : http://download.intel.com/museum/Moores_Law/Articles-Press_Releases/Gordon_Moore_1965_Article.pdf)

2. Seconde loi de Moore :

Ajustement de la première loi en posant que le nombre de transistors des microprocesseurs sur une puce de silicium double tous les deux ans (en réalité, nous serons plus sur un coefficient de 1,96 année entre 1971 et 2001)... Fig.3. Une des premières conséquences de ces conjectures se ressent directement dans l'évolution du matériel informatique que nous achetons annuellement et fait clairement l'affaire de tous les revendeurs.

3. Troisième loi de Moore :

Extension de la loi de Moore à toutes choses (puissance, capacité, vitesse, ...) pouvant ainsi doubler tous les 18 mois !

Ne nous arrêtons pas en si bon chemin et pour-

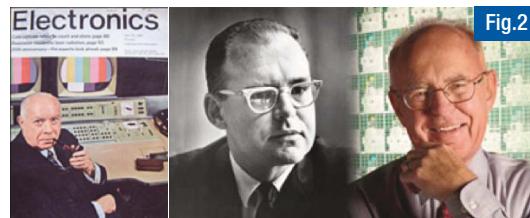


Fig.2

Fig.3

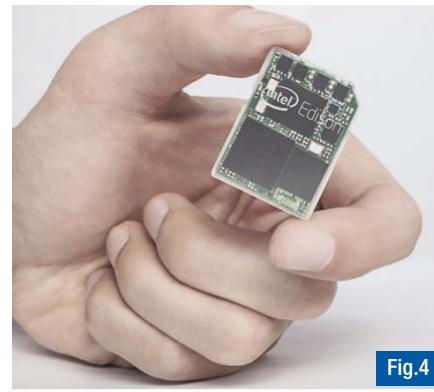
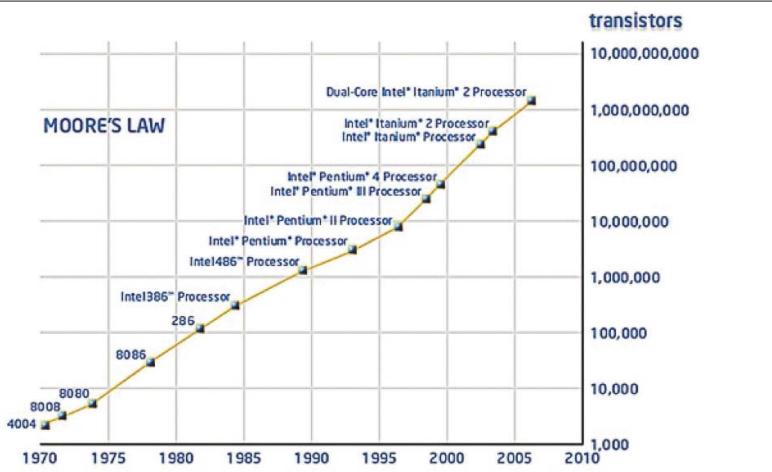
Source : Intel - <http://www.intel.com>

Fig.4

l'évolution des technologies au regard de l'évolution de l'humanité ! La technologie, Internet, les smart softwares (Google Now, Siri), le Big Data et le machine learning, les objets connectés, la robotique, ... « l'informationnel » suit une croissance exponentielle amenant la civilisation humaine à rencontrer une croissance technologique qu'elle ne peut (ou difficilement) prédire à ce jour tant elle est déjà et sera de plus en plus importante... les constats de cet article et chapitres précédents ne font que renforcer les faits ! Et que dire de l'Intelligence Artificielle ?

“Si les vitesses de calcul doublent tous les deux ans, qu’arrive-t-il quand des IA informatisées font la recherche ?” – Eliezer Yudkowsky

Afin de mieux comprendre le vaste domaine de la singularité technologique, je vous invite à parcourir l'article de Eliezer S. Yudkowsky quant aux 3 écoles majeures sur le domaine : « Accelerating Change », « Event Horizon » et « Intelligence Explosion » (<http://yudkowsky.net/singularity/schools/>). Vous y trouverez les noms des avocats de chacune des écoles; si le sujet vous passionne au plus haut point, dirigez-vous sans plus attendre sur le site de la Singularity University : <http://singularityu.org/>

Alors, Bullshit marketing ou réalité ?

Des origines de la loi de Moore à la Singularité Technologique et après lecture de tous les faits apportés lors de ces premiers chapitres d'une vie à la Silicon Valley, à la fin de la journée, la question d'une réalité ou d'un Bullshit Marketing se clarifie! Les faits sont clairs, les mots sont forts : « rupture, fracture, ... ». L'accélération ne semble pas trouver de limite à court ou moyen terme, sachant que les cycles ne font que se raccourcir ! Peter H. Diamandis vient de publier le livre « Abundance » apportant une forme d'explication sur non pas comment faire face à ces challenges, mais bien comment effectuer le savant mélange de cette accélération technologique avec l'être humain et le capital disponible afin de résoudre les grands challenges de notre humanité ou de ce qui l'anime !

Gregory Renard [Redo] – Geek in Silicon Valley
<http://gregoryrenard.wordpress.com>

suivons cette extension en tentant de l'appliquer à nos éléments numériques du quotidien : iPad, iPhone, Google Car, Google Now, Petman, Siri, mais aussi d'autres domaines comme l'informatique cognitive, le Big Data, l'IA, le social mining, les nanotechnologies,... Tous subissent à ce jour une évolution exponentielle de par l'héritage des performances de calcul de l'informatique guidée par cette hypothétique Loi de Moore. Mais quelle est la limite à cette croissance et existe-t-elle ?

Une fin annoncée de la loi de Moore ? Une crise économique ou une singularité technologique ?

Intel a pris conscience que cette loi rencontreraient trop rapidement son propre point de singularité technologique, et, de ce fait, investit maintenant d'autres domaines d'innovation. *“Innovation happens at many levels. We will continue to innovate at the transistor and individual chip level, but now our job is to work with the extended ecosystem of developers, software designers, systems engineers, service providers and others to innovate at the platform level.”* – Craig Barrett, Intel CEO – IDF 2005. Source : http://download.intel.com/museum/Moores_Law/Articles-Press_Releases/Press_Release_Mar2005.pdf

Effectivement, depuis 2005, les annonces de fin de la Loi de Moore (dit “The Wall”) se font de plus en plus fréquentes et Gordon Moore en personne relativise ses lois avec une perspective à 2020 (Conformément aux études menées par Intel en ce sens) :

“Selon Gordon Moore l'industrie n'est pas loin des contraintes énoncées quelques années plus tôt par le physicien Stephen Hawking à savoir la vitesse de la lumière et la nature atomique de la chose.” Source : <http://www.clubic.com/actualite-80182-loi-moore-caduque-15-ans.html>

De plus, la Loi de Moore se trouve également confrontée à d'autres lois telles que :

► La loi de Wirth ou loi de Reiser (obésiciel) selon laquelle le logiciel ralentit plus vite que le matériel n'accélère ne favorisant ainsi pas la

notion de singularité technologique.

Source : http://fr.wikipedia.org/wiki/Loi_de_Wirth
► La loi de Rock, du nom de son découvreur Arthur Rock, stipule que le coût d'une fonderie de semi-conducteurs double tous les 4 ans car le procédé de fabrication, la photolithographie, utilisé depuis une quarantaine d'années se rapproche toujours plus de ses limites physiques. Source : http://fr.wikipedia.org/wiki/Loi_de_Rock

Et pourtant, Intel ne cesse de se réinventer, le CES 2014 fut une très belle surprise en ce sens avec les annonces comme l'Intel Edison, un nouvel ordinateur au format d'une carte SD, basé sur la technologie Intel Quark, doté de capacités sans-fil intégrées et supportant de multiples systèmes d'exploitation Fig.4.

Du prototype à la production, Intel Edison permettra d'accélérer le rythme de l'innovation et du développement de produits par les inventeurs, entrepreneurs et concepteurs. L'Intel Edison devrait être disponible cet été.

Mais après l'Intel Edison, que se passera-t-il ? Est-ce que la course à la miniaturisation toucheira à sa fin comme l'imagine son créateur à l'horizon 2020 ? Allons-nous vivre une crise économique mondiale suite au fameux Wall comme le propose l'article de « Economic Crisis » ou allons-nous nous diriger tout simplement vers le fameux point de singularité technologique ?

De Moore à la Singularité Technologique ?

La singularité technologique définit un caractère ou un point dit « singulier » de la technologie, une fracture ou une rupture. Cette notion est présente dans de nombreux domaines comme la physique avec la singularité gravitationnelle, en mathématiques avec par exemple $f(x) = 1/x$ admettant une singularité quand $x = 0$ et enfin en technologie avec l'hypothèse d'une évolution telle que cette dernière pourrait prendre le pas sur l'humanité !

Etablie en 1993 par Venor Vinge, la notion de singularité technologique apporte une vision de

La technologie n'est qu'un outil au service du développeur

Je ne sais pas si vous avez l'habitude de suivre cette rubrique, mais pour faire court, je travaille depuis un peu moins de deux ans chez Microsoft (Microsoft Corp à Seattle, NDLR). J'ai eu des opportunités incroyables, j'ai été « trimbalé » de team en team pour développer des fonctionnalités dans divers produits comme Surface, Windows ou XBOX.

Que ce soit UI, driver, data model, runtime/sdk, directx, en C#, C++ ou JavaScript, j'ai toujours apprécié mes missions et suivi avec une grande joie mon équipe. J'ai rencontré des personnes extrêmement intelligentes qui m'ont aidé à grandir. J'ai aussi pu avoir accès aux codes sources de tous les produits de ces équipes, ce qui m'a permis d'apprendre et comprendre comment tout cela fonctionne.

Puis, récemment un (nouveau) changement s'est offert à moi. Je viens d'intégrer Office Core, l'équipe qui crée les fondations de tous les logiciels Office (toutes plateformes confondues). Ce qui m'a poussé à « descendre » dans le core, est que ma vision de mon travail a évolué, je veux de la data-structure, des algorithmes de pointes, de la science. J'aime faire des choix techniques, jouer avec les dernières technologies, faire des expériences utilisateurs, mais je trouve que cette facette de notre métier ne représente qu'une toute petite partie de ce qu'on doit savoir faire.

Par manque de maturité et d'expériences, j'ai évangélisé des outils à de nombreux clients. J'ai répondu à leurs problèmes par des solutions technologiques. Lorsqu'on me posait une question telle que « Comment faire pour avoir un service type WebAPI performant pouvant stocker des données temporaires par milliers ? », je répondais machinalement « le cloud (merci la pub) ! Et tu fais ton service en .NET WCF et puis tu rajoutes du Data Services par-dessus, et un peu de Linq aussi et tu utilises la BDD noSQL de Azure ».

« ça dépend »

Non seulement je ne réponds pas réellement au problème mais j'oublie une chose fondamentale : il n'existe aucune réponse exacte. D'ailleurs plus on a de l'expérience et plus la réponse sera « ça dépend ». Ne croyez pas un mec qui vous sort une solution miracle. La solution est miracle



si elle prend en considération un panel illimité de variables : de l'historique de l'entreprise en passant par les compétences des développeurs de l'équipe...

Mais au-delà de ça, je réponds par de la techno au lieu de me baser sur ce qui compte le plus : les structures de données.

Nous sommes dans un monde illimité, nous souhaitons virtualiser un bout de ce monde dans un système limité : quelles sont les structures de données et algorithmes qui me permettront de répondre à ce problème. Après avoir répondu à ces questions, on a une bonne idée de l'architecture « globale » pour résoudre ce problème. C'est seulement au dernier moment qu'intervient la technologie.

La technologie est au service du développeur. La technologie évolue extrêmement vite, ce que vous écrivez aujourd'hui en HTML sera sûrement réécrit en FOOxml dans dix ans, si ce n'est pas dans 1 an. Toutes ces choses sont volatiles et nous éloignent de ce que nous sommes : des ingénieurs logiciels complets. S'il faut faire du JAVA et que vous détestez ça mais que ça fait le « job » et que ça le fait bien, alors OK. Si c'est du .NET 2.0, que vous êtes un technophile et que bosser sur cette antiquité vous fait mal au

ventre : changez votre façon de penser. Les challenges ne sont pas là où vous le croyez, ce n'est pas apprendre par cœur la documentation de la techno, non, c'est pouvoir voir plus loin, voir plus haut.

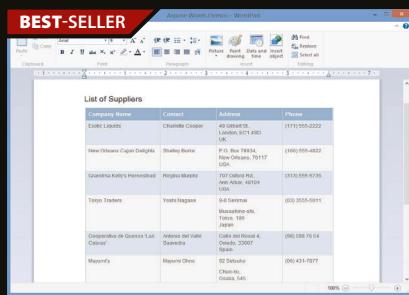
Je suis toujours le premier à faire des « trolls » sur les technologies, avec des dizaines/parfois centaines de réponses de fan tombant dans le panneau. Un peu comme les utilisateurs qui se font la guerre iPhone versus Android. C'est marrant, mais le problème arrive quand des jeunes étudiants te demandent « quelles technos dois-je apprendre pour être un bon développeur ». Il faut rester sur du générique, apprendre ce qu'est un service, du Cloud, de l'UX, de l'algo/data structure, de l'architecture logicielle, du distribué, de l'électronique, de la gestion de source, de l'écriture de documentation, des méthodologies,

des maths... Et c'est seulement lorsqu'on aura besoin de la techno qu'on s'y plongera. C'est ce que font tous les gens qui sortent de Stanford, Princeton ou MIT et qui vont travailler chez Amazon, Facebook, Apple ou Microsoft.

Les pro-technos ont UN argument, une technologie mal utilisée peut mener le projet à sa perte. C'est vrai. Sauf qu'assimiler une techno et un langage, cela prend combien de temps une fois qu'on a de très bonne fondation? Deux semaines? Eventuellement trois, pour connaître les best practices? C'est rien et des auteurs très qualifiés de blogs et de livres ont déjà couvert le sujet pour nous. C'est d'ailleurs pour ça que l'on a besoin d'experts en technologies. Mais dans ce cas-là, cet expert est consultant, pas un développeur chez un éditeur logiciel.

Si vous vous reconnaissiez dans ces lignes, demandez-vous si vous ne passez pas à côté de quelque chose. Et ne soyez pas trop démoralisé quand une RH ou un client vous demanderont d'être un « expert .NET ou JAVA », ces mots les rassurent. Si vous êtes bien préparé avec des bases solides, vous pourrez postuler à n'importe quel job qui touche l'ingénierie informatique. Car être ingénieur informatique, c'est déjà une expertise/spécialité.

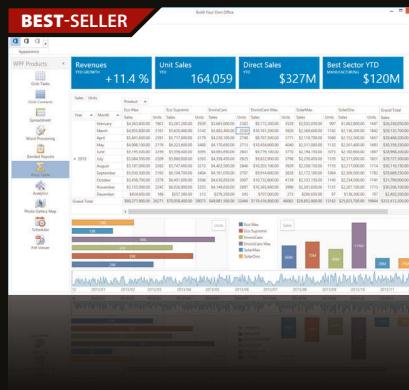
```
:quit<enter>
$>
```

**Aspose.Words for .NET**

à partir de € 732

**Lisez, modifiez et écrivez des documents Word sans Microsoft Word.**

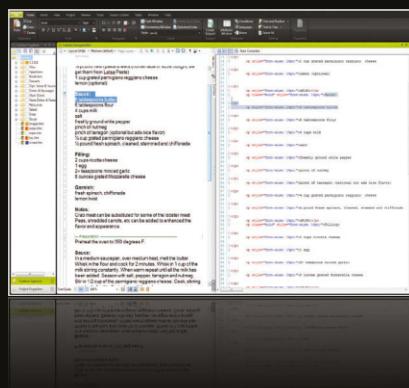
- Création de documents, manipulation du contenu/formatage, puissante capacité de fusion de courrier et exportation en DOC/HTML
- Accès détaillé à tous les éléments d'un document par programmation
- Support les formats de fichiers: DOC, DOCX, WordprocessingML, RTF, HTML, OOXML, OpenDocument, PDF, XPS, EMF et EPUB

**DevExpress Universal Suite**

à partir de € 1 612

**400+ outils et contrôles WinForms, ASP.NET, WPF, Silverlight et Windows 8.**

- Exploitez votre base de codes pour développer des applications tactiles multiplateformes
- Inclut un tableau de bord de visualisation interplateformes et un serveur de rapports
- Nouveaux contrôles DevExpress incluant Tableur et Carte
- Codage, débogage et refactorisation avec CodeRush pour Visual Studio
- Inclut la nouvelle galerie unifiée de modèles applicatifs DevExpress

**MadCap Flare**

à partir de € 366

**Produisez de la documentation pour tous les publics, toutes les langues ou formats.**

- Créez, gérez et publiez du contenu sous différents formats, ce qui inclut l'impression, en ligne, bureau et mobile
- Centralisez vos contenus et générez les formats voulus
- Création et publications natives pour les langues à écriture de droite à gauche
- Soignez le volet social de la documentation pour mieux retenir l'attention des clients

**SQL Diagnostic Manager**

à partir de € 1 754

**Suivi des performances SQL 24h/24, 7j/7, alertes et diagnostics.**

- Suivi des performances SQL Server physiques et virtuelles
- Analyse approfondie des requêtes pour identifier les délais et les consommations de ressource excessifs
- Planification de la capacité révélant les tendances de croissance des bases de données et réduisant l'utilisation du serveur

HP Proliant MicroServer Gen8

HP propose toute une gamme de serveurs, de taille réduite, dans la famille Proliant. Ce MicroServer Gen 8 ressemble à un gros NAS. Pour un peu moins de 400 €, on dispose d'un serveur, de 4 baies pour les disques durs et d'une connectique plutôt complète.

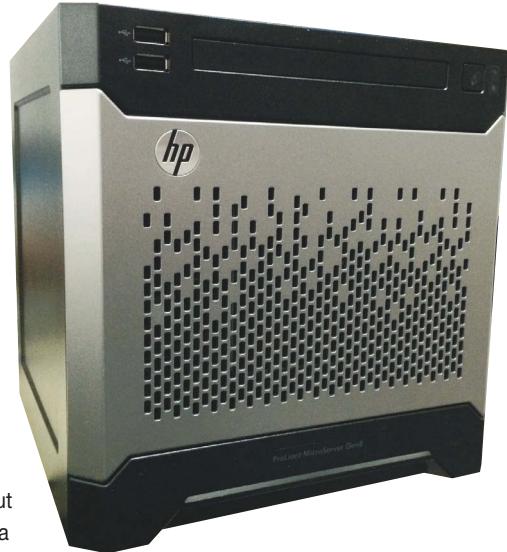
Après avoir déballé la bête et avoir branché les câbles nécessaires, il faut tout d'abord configurer et installer un système serveur. Une version de Windows Server 2012 Essential est livrée par défaut (mais non pré-installée). Préalablement, n'oubliez pas d'installer un ou plusieurs disques durs dans les baies. La version de base est livrée nue. Si vous n'avez pas de lecteur de DVD externe, le plus simple sera de créer une clé USB contenant l'ISO du système (Windows Server ou Linux). L'installation peut être assez longue surtout sous Windows Server qui nécessite plusieurs redémarrages pour pouvoir finaliser l'installation et configurer les services de base. A cela se rajoute la mise à jour du système (parfois 1 Go)...

Le bloc embarque 16 Go de mémoire vive (maximum), de l'éthernet 1 Gb, un socket processeur, une carte graphique Matrox, un contrôleur Smart Array, et, pour l'administration, une connexion iLo (integrated Lights Out). La configuration est relativement simple, à condition de savoir ce que l'on fait. Pour faciliter l'installation système, Intelligent Provisioning aide beaucoup. Bien entendu, on peut aller très loin dans la configuration, nous avons affaire à un serveur ! Le firmware se mettra à jour automatiquement pour être parfaitement à jour. iLo est un puissant outil d'administration qui permettra à tout administrateur de surveiller et de manager l'ensemble du serveur et des composants. L'interface graphique est plutôt

ergonomique. Sur la partie stockage, on peut monter à 12 To. Petit plus non négligeable, la présence d'un port microSD : idéal pour y mettre un système entier même si ce dernier n'est pas géré à chaud. Il faut noter que ce modèle ne gère pas, par défaut, le changement à chaud des disques, il faut rajouter une fonctionnalité Smart Array / SATA RAID. Cela peut être un défaut pour les utilisateurs avancés car la sécurité des données et des logiciels serveurs sera moindre. Par contre, l'ensemble est construit avec qualité. Et la compacité du serveur permet de le glisser sous un bureau, dans un placard... Nos tests ne montrent pas une chaleur excessive. En revanche, en démarrage, et même à chaud, il sera bruyant; en fonctionnement linéaire, ce Proliant est toutefois relativement silencieux. Avec Windows Server 2012 Essential, la machine est satisfaisante sur les performances même s'il ne faut pas s'attendre à des performances d'un haut de gamme. Le système connaît parfois des saccades et mieux vaut avoir un bon réseau pour assurer les débits.

Pour quel usage ?

Ce microserver peut avoir différents usages : serveur de fichiers, serveur de virtualisation, serveur de stockage, serveur web, etc. Il sera bien taillé pour supporter des infrastructures de petites tailles ; mais attention, il faut rajouter un



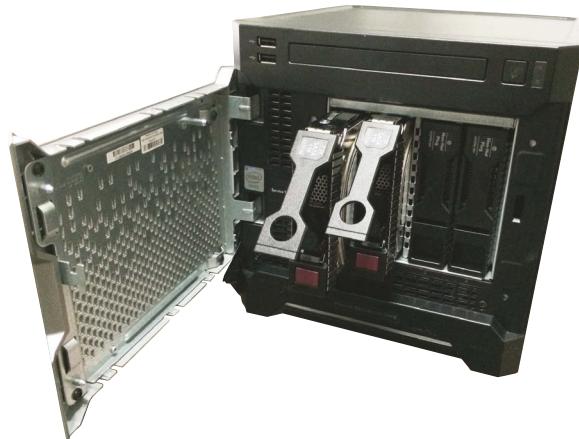
budget pour la licence iLo et les disques. Vous pouvez monter votre Cloud privé dessus si vous le souhaitez. Il se destine aux TPE et PME essentiellement. Mais un développeur exigeant pourra utiliser ce type de machine pour se monter son infrastructure, héberger ses propres sites et étendre les capacités sur un service de Cloud public (mode hybride). A choisir entre un « simple » NAS et ce Proliant, nous choisissons le microserver car plus polyvalent que le NAS. Il est disponible à partir de 396 € (tarif officiel).

Les +

- Les baies
- L'assemblage de qualité
- Les multiples ports USB
- Ethernet
- iLo 4
- port microSD
- port PCIe

Les -

- USB 3 non actif lors de l'installation
- Temps d'installation
- Le bruit parfois élevé
- Accès aux baies trop simple en production
- changement à chaud trop limité



Complétez votre collection

PROGRAMMEZ!
le magazine du développeur
www.programmez.com



Prix unitaire : 6 €
(Frais postaux inclus)

France
métropolitaine
uniquement.

nouveau

Tout Programmez! sur une clé USB

Tous les numéros de Programmez! depuis le n°100.



Clé USB 2 Go. Livrée dans sa boîte. Testé sur Linux, OS X, Windows. Les magazines sont au format PDF.



* tarif pour l'Europe uniquement. Pour les autres pays, voir la boutique en ligne

vous pouvez aussi commander directement sur notre site internet : www.programmez.com

- 168 : exemplaire(s)
- 169 : exemplaire(s)
- 170 : exemplaire(s)
- 171 : exemplaire(s)

- La Clé USB avec les numéros de Programmez depuis le n°100

29,90 € (Tarif pour Europe uniquement)

Commande à envoyer à :

Programmez!
7, avenue Roger Chambonnet
91220 Brétigny sur Orge

exemplaires x 6 € = € + 29,90 € soit au **TOTAL** = €

Prix unitaire : 6 € (Frais postaux inclus), France métropolitaine uniquement.

M. Mme Mlle Entreprise : Fonction :

Prénom : Nom :

Adresse :

Code postal : Ville :

Tél : (*Attention, e-mail indispensable*)

E-mail : @

Règlement par chèque à l'ordre de Programmez !

Progress Software : développer des applications métiers rapidement, sans code, et dans le nuage

Depuis 1981, Progress propose l'un des langages L4G les plus réputés pour le développement d'applications de gestion. Avec Progress Pacific, en plus d'être plus rapide et plus simple, et avec le minimum de code, le L4G, sa base de données et ses outils, se déclinent désormais dans le nuage.

Comment développer rapidement des applications métiers ? Au début des années 80, faisant suite à deux décennies de langages de type Cobol, et parallèlement au développement des langages de type C, émergent de nouveaux langages dits de quatrième génération, appelés L4G. Ils sont intéressants à plus d'un titre : ils permettent de développer plus rapidement des applications. Ils disposent d'une courbe d'apprentissage plus simple et rapide qu'un langage plus classique. Ils sont généralement complétés d'une syntaxe déclarative basée sur SQL afin de manipuler les bases de données ; les risques d'erreurs s'en trouvent réduits. Ces langages sont plus particulièrement adaptés à la conception de programmes structurés, comme les logiciels de gestion. C'est sur cette base que, dès sa création en 1981, Progress propose une plateforme de développement d'applications de gestion qui affiche une philosophie : aller plus vite ! AGL, son L4G, ne s'apprend pas à l'école, et ne dispose pas d'instances de normalisation comme la plupart des langages propriétaires. En revanche, il permet de développer quatre fois plus vite qu'en Java ou C. Disposant d'une courbe d'apprentissage rapide, et réputée assez facile, il est accessible aux pratiquants de ces langages plus complexes. Et au fil des années, AGL a été complété de nombreux outils et fonctionnalités, développés en interne par Progress ou fruits d'acquisitions.

C'est ainsi que depuis son origine, AGL est accompagné d'une base de données. La réunion de ces deux composants de base forme la plateforme Progress OpenEdge. Cette solution a fait l'objet de nombreuses évolutions, en particulier ces dernières années avec un investissement important de plus d'un milliard de dol-

lars en R&D. La solution supporte plus de performances, le stockage, le multi-tenant, et un partitionnement horizontal ou vertical de la base de données. L'horizontal permet d'assurer l'étanchéité logique, une problématique classique, ce qui réduit le travail pour la communauté des développeurs, réduit la redéfinition des tables, des identifiants, des partitions, etc. Intégré nativement, l'horizontal est important pour supporter le Cloud, le SaaS et le partitionnement logique entre clients différents d'une même application SaaS. Le vertical est quant à lui destiné au partitionnement des tables sur un même serveur.

Retours d'expériences

Notre présence à Progress Exchange, qui s'est déroulée à Boston en octobre 2013, nous a permis de rencontrer des utilisateurs de la plateforme Progress et d'évoquer leur retour d'expérience.

- Le président de PUG France

« Progress fournit un ensemble cohérent d'environnements de développement pour des applications orientées business, des composantes totalement intégrées au niveau du langage et qui permettent de faire des choses aussi larges que



Gilles Querret,
Président de PUG
France,
Président de River
Side Software

des écrans anciens ou de la mise à disposition de services web et REST », nous a affirmé Gilles Querret, président de PUG France, également président de River Side Software, expert indépendant sur OpenEdge et l'administration des bases de données. « Tout cela nous est proposé dans

un environnement cohérent et simplifié, qui ne nécessite pas de chercher dans des sources multiples et variées, car la recherche d'informations est souvent longue et pénible. Tout est regroupé au sein d'une même plateforme, avec un langage simple d'utilisation et flexible sur le code produit. Le langage L4G est essentiel, il autorise un requêtage de manière simple et plus visible que le SQL, et de manière naturelle. L'ensemble est rapide dans la logique métier. »

- Nearshoring en Roumanie

Développant en Roumanie pour des clients européens, allemands, belges et néerlandais, Wayfare est une société hollandaise de nearshoring qui pratique OpenEdge depuis une quinzaine d'années. Pour son CEO, Wim Martens, « Progress était le langage le plus simple à apprendre. Même si, quand Windows est arrivé, c'est devenu plus compliqué, avec la nécessité d'enrichir l'environnement. Progress nous permet en particulier de réaliser des développements 'low cost'. De plus en plus d'applications migrent sur Progress, la plateforme permet une décision et une migration rapides. Aujourd'hui, nos clients sont moins intéressés par le choix des applications que par leur fiabilité et notre capacité de créer des applications directement fonctionnelles. Progress affiche également une capacité à tourner sur une longue période, avec un coût de maintenance réduit. »

- Editeur RH français

Anthony Cross, Directeur avant-vente de Progress Software, évoque également pour nous une société française, qui a souhaité rester anonyme, qui a réécrit avec RollBase son application ressources humaines destinée aux grands

L4G

Apparus au début des années 80, les langages de programmation dits de quatrième génération (L4G ou 4GL en anglais) se veulent proches des langages naturels. Composés de macro-commandes et de mini programmes, ils permettent l'écriture de programmes même complexes avec moins de lignes de code, donc plus rapidement et avec moins d'erreurs.

Portrait de Progress Software

C'est en 1981 que des étudiants diplômés du MIT (Massachusetts Institute of Technology) créent Data Language Corporation. La jeune société développe et commercialise un langage L4G: Progress 4GL. La société prend le nom de Progress Software en 1987. L'éditeur fournit une plateforme de développement pour des applications de gestion de type ERP, CRM, finance, industrie, distribution, télécoms, etc. Une centaine d'éditeurs dans le monde

exploitent la technologie Progress. En France nous citerons Proginov, éditeur d'ERP en mode ASP, et l'intégrateur Sopra. Le nombre d'utilisateurs finaux n'est pas connu, ils seraient entre 2 et 3 millions, avec, par exemple en France, les 20 000 licences de la CNAM qui a développé une application RH sur mesure, jugée plus efficace à customiser qu'un progiciel. PSA, la Caisse d'Allocations Familiales, des banques y figurent également.

Une année pleine de technologies et de codes avec

PROGRAMMEZ!
le magazine du développeur

www.programmez.com

1 an 11 numéros

49€
seulement (*)

2 ans 22 numéros

79€
seulement (*)

Spécial étudiant
39€
1 an 11 numéros

réservée à la France
Métropolitaine

(*) Tarifs France métropolitaine

Toutes nos offres sur www.programmez.com

Oui, je m'abonne

ABONNEMENT retourner avec votre règlement à
Programmez, 17, route des Boulanger 78926 Yvelines cedex 9

- Abonnement 1 an au magazine :** 49 € (au lieu de 65,45 €, prix au numéro)
 Abonnement 2 ans au magazine : 79 € (au lieu de 130,9 €, prix au numéro)
 Abonnement spécial étudiant 1 an au magazine : 39 €

Photocopie de la carte d'étudiant à joindre

Tarifs France métropolitaine

Offre spéciale : abonnement + clé USB Programmez!

- 1 an (11 numéros) + clé USB : 60 €
 2 ans (22 numéros) + clé USB : 90 €

Clé USB contenant tous les numéros de Programmez! depuis le n°100, valeur : 29,90 €

Tarifs France métropolitaine

M. Mme Mlle Entreprise : _____ Fonction : _____

Prénom : _____ Nom : _____

Adresse : _____

Code postal : _____ Ville : _____

Tél : _____ (*Attention, e-mail indispensable pour les archives sur internet*)

E-mail : _____ @ _____

Je joins mon règlement par chèque à l'ordre de Programmez !

Je souhaite régler à réception de facture

I love Progress !

Invités à rencontrer l'éditeur lors du dernier Progress Exchange, la grand-messe de l'écosystème de la plateforme de développement qui s'est déroulée à Boston en octobre dernier, quelle n'a pas été notre surprise de découvrir des développeurs, des partenaires et des clients qui s'affichaient ouvertement heureux. Allant jusqu'à nous affirmer en plusieurs occasions « *I love Progress* » ! Le phénomène est suffisamment rare pour que nous menions notre enquête...

La simplicité du langage ABL, la présence d'une base de données performante et d'outils de programmation, l'ensemble fourni sous le nom d'OpenEdge, participent bien évidemment à créer ce climat de satisfaction. Mais ce que l'écosystème Progress apprécie également, c'est la transparence et la proximité de l'éditeur. Ainsi Phil Pead, son CEO (*Chief Executive Officer*), n'a pas hésité, lors de son discours d'ouverture à faire un surprenant mea culpa, dénonçant les erreurs passées de la société; elle s'est en particulier fourvoyée un temps dans le RPM (*Responsive Process Management* ou gestion de la réactivité opérationnelle), oubliant ses fondamentaux, au risque de délaisser son positionnement et ses clients historiques.

L'erreur stratégique, qui a fragilisé son écosystème, est aujourd'hui oubliée. De judicieuses acquisitions et le développement à vitesse accélérée de nouveaux outils intégrés à la plateforme sont venus renforcer l'image d'un retour aux sources de Progress. L'intégration du langage Java, un BPM (*Business Project Management*), un outil de règles (Corticon), des interfaces et templates (RollBase), un framework d'intégration des données



Phil Pead, CEO de Progress + fond du keynote Progress Exchange

(DataDirect) sont autant de signes positifs au service des développeurs.

Mais le message le plus porteur d'avenir pour l'écosystème Progress est certainement venu de Pacific, qui ouvre à la communauté des développeurs un accès dans le nuage. Pacific est ni plus ni moins qu'une plateforme PaaS (*Platform-as-a-Service*) qui accueille les développeurs pour le prototypage des projets, mais également s'ils le souhaitent pour l'hébergement dans le Cloud de leurs solutions multi-tenant.

Quant à l'avenir de Progress Software, il s'annonce ambitieux et porte sur deux axes : le Big Data et la mobilité. John Goodson, CPO (*Chief Product Officer*) de Progress, a été peu prolix quant à la stratégie Big Data de l'éditeur, se contentant d'anticiper l'évolution de Pacific vers une plateforme analytique unique qui embrasse le Big Data, supporte le No-SQL et offre plus de connexions au Cloud avec des mécanismes sécurisés.

Plus concrète a été la présentation de Ope-

nEdge Mobile, très attendue par l'écosystème Progress qui peut désormais intégrer la mobilité dans ses développements. La plateforme de développement dédiée aux environnements mobiles, intégrée à Progress Developer Studio, dispose d'un ensemble de modules : Publish Web Server pour la publication des services et apps mobiles, Build Apps Server pour la consolidation des services mobiles. Il y a également Configure pour les connexions RDBMS aux bases de données, Launch HTML5 pour le run des tests des applications mobiles, ainsi que des émulateurs pour les tests.

PUG France

La communauté française des utilisateurs

Progress a son club, PUG
(*Progress User Group*)
France. Site : pug-france.fr



PUG France, le club utilisateurs Progress



Antony Cross,
Directeur avant-
vente de Progress
Software

comptes, initialement développée en C/C++. « Ils souhaitaient re-développer rapidement leur application pour disposer d'une version 'light' et SaaS, dans le Cloud afin d'être immédiatement accessible aux clients. D'où leur choix de Progress Pacific. En prenant en compte

leur expérience passée dans les RH, ils ont pu refaire l'application en moins d'un an. Ce qui est quatre fois plus rapide avec un L4G va dix fois plus vite avec Pacific et RoleBase. »

Développer dans le nuage

Progress Exchange a été l'occasion d'évoquer l'évolution du développement d'applications. Progress entend développer la notion de 'citizen

developer'. Derrière cette expression, le 'développeur citoyen', se cache la vision de la démocratisation du développement d'applications. Elle place l'utilisateur au centre d'une approche plus personnalisée du développement, dont les capacités de création individuelle se limitent habituellement à des outils comme Excel. Les langages de quatrième génération et les plateformes qui les hébergent représentent une opportunité pour créer lui-même les applications dont il a besoin. Reportée sur l'écosystème Progress, l'approche 'citizen developer' fait plutôt référence à la capacité de développer simplement et rapidement des applications. Qui plus est pour des usages plus tournés vers les métiers, voire éventuellement personnalisés. Pour Anthony Cross, « Le développement de la notion de 'citizen developer' est important dans les logiciels de gestion, où la micro seconde

n'est pas importante, qui ne font pas nécessairement appel au temps réel, et qui sont embarqués dans des systèmes. Nous allons vers le Saint Graal où l'on a presque plus de code à écrire, où tout se fait par configuration, clic, drag-and-drop, saisie du nom de l'objet, etc. Et en particulier le SaaS, que nous considérons comme le deuxième grand changement de la programmation. »

Cette approche du développement d'applications se traduit par Progress Pacific, plateforme PaaS qui associe OpenEdge (le langage L4G et la base de données), à Corticon (règles business), RollBase (plateforme PaaS et templates) et DataDirect (déploiement). Pour créer les applications simplement et plus rapidement, la solution s'appuie principalement sur RollBase, plateforme de développement d'applications business dans le Cloud, acquise par l'éditeur en

Quelle interopérabilité entre mes différents fournisseurs Cloud ?

Avec Aruba Cloud,

vous avez l'assurance de ne pas être prisonnier d'un fournisseur. Nos services sont intégrés au **driver DeltaCloud** et compatibles **S3**. De plus, vous pouvez utiliser des formats standards d'images de machines virtuelles, **avec VHS et VMDK**, ainsi que des modèles personnalisés provenant éventuellement d'autres sources.



3
hyperviseurs



6 datacenters
en Europe



APIs et
connecteurs



70+
templates



Contrôle
des coûts

*MON PAYS, MON CLOUD.

“ Nous avons choisi Aruba Cloud car nous bénéficions d'un haut niveau de performance, à des coûts contrôlés et surtout car ils sont à dimension humaine, comme nous. Xavier Dufour - Directeur R&D - ITMP

Contactez-nous!

0810 710 300

www.arubacloud.fr



Cloud Public

Cloud Privé

Cloud Hybride

Cloud Storage

Infogérance

MY COUNTRY. MY CLOUD.*

juin dernier. RollBase est un outil d'implémentation de règles complexes dans le moteur de règles. Il permet de développer à 80 % sans code, tout en respectant la mise en oeuvre de règles toujours plus compliquées et spécifiques à l'entreprise, et généralement en JavaScript. A terme, l'objectif de Progress est d'aller vers une absence totale de code dans le processus de développement des applications métiers. Les cinématiques et développements simples sans code, mais complexes, sont réalisés dans le moteur de règles, ce qui permet de valider qu'elles sont correctes, et de faire la pré-recette fonctionnelle au moment de la création et non pas au final, et ainsi de gagner beaucoup de temps ! Et les développements sont documentés. Pour créer une application, Progress Software propose donc trois voies : la traditionnelle et historique, avec son langage ABL, la générique avec le langage JavaScript, et l'innovante avec ses règles business.

Progress Pacific : un L4G cloud

Avec l'arrivée de sa plateforme Cloud Pacific, Progress voit désormais deux étapes dans l'accès à ses solutions. La première est historique, reprise par ses partenaires qui profitent de leurs acquis dans la programmation pour former rapidement, en moins d'un mois, leurs développeurs au L4G, puis au L4G avancé, et sur des modules spécifiques selon les besoins applicatifs et métiers. La seconde étape est liée à la simplification du développement par l'absence de beaucoup de code. « C'est une étape très riche pour les développeurs Cobol, qui sont fonctionnellement qualifiés, mais qui souffrent de la migration vers d'autres langages de programmation, constate Anthony Cross. Le citizen developer peut développer sans code avec Pacific. C'est une approche du développement qui apporte de nouvelles compétences fonctionnelles adaptées aux développeurs parfois dépassés par la technique mais pas par le métier. L'apprentissage est quasi immédiat, car ils sont technophiles, pour développer rapidement des applications métiers. »

Progress Pacific permet de ne plus coder sur des applications fortement tactiques, et souvent moins stratégiques, et répond aux besoins spécifiques des créateurs d'applications plus que des développeurs. La démarche est sensible dans les grandes entreprises, qui ont tendance à homogénéiser leurs outils de gestion au travers de gros progiciels du sol au plafond. En faisant en particulier le choix stratégique de prendre un numéro un mondial, avec tout ce qui est embarqué dans la solution, et de la customiser. « La violence de la customisation, c'est son coût en jours/hommes. C'est pourquoi les entre-

prises ont une réflexion globale sur l'usage de petites applications tactiques pour développer facilement et rapidement, tout en limitant les erreurs. C'est pourquoi elles vont aller vers des plateformes SaaS. »

Le Cloud fait à minima gagner un mois de projet

Pacific est également une solution proposée aux sociétés nouvelles qui ne disposent pas des fonds nécessaires pour s'équiper des machines, des licences, des hébergeurs, etc. Elles peuvent ici faire le choix d'une location mensuelle. Mais que se passe-t-il lorsque le projet fonctionne et que l'entreprise doit faire face à l'augmentation des comptes en SaaS ? Progress permet la migration de la plateforme sur un Cloud privé et des serveurs spécifiques. « Quand l'entreprise marche bien, les coûts du SaaS augmentent lourdement. Avec Progress Pacific, il est possible de la migrer sur un Cloud privé, tout en conservant la richesse de la plateforme. Notre leitmotiv est 'any data', de pouvoir se connecter sur n'importe quelle source de données, à partir de n'importe quel objet connecté. C'est pour cela que nous offrons la capacité de conserver l'historique, y compris le mode caractère. » L'association d'une platefor-

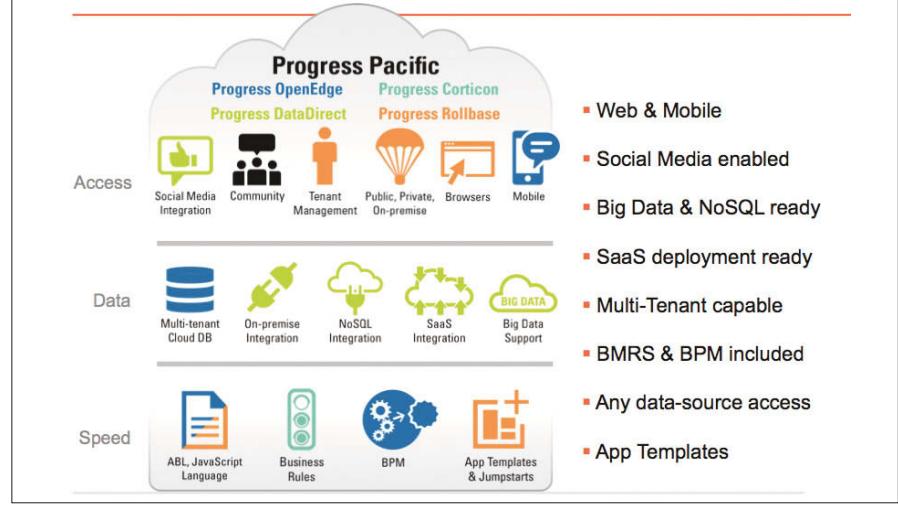
me de développement L4G et du Cloud, à l'image de Progress Pacific, présente un nombre singulier d'avantages. L'un des plus importants porte sur l'accélération des délais de développement. Prenons l'exemple du développement d'une application tactique en Java : la création et la réunion du comité d'architecture, du comité de sécurité, du comité de base de données pour justifier le modèle sont chronophages et augmentent les délais. Avec une plateforme cloud validée par l'éditeur et l'hébergeur, qui prennent en charge les serveurs, la création dynamique des tables, la sécurité, etc., le chef de projet sous pression gagne déjà un mois de projet. « Avant, avec Turbo Pascal, C, etc., des logiciels avaient la prétention de coder à la place des développeurs, mais la qualité du code produit n'y était pas ! », conclut Anthony Cross. « Aujourd'hui, l'approche est différente. Plus de mise à disposition d'outils qui génèrent du code à la place du développeur, mais une logique d'outils suffisants en eux mêmes pour créer des objets qui correspondent aux objets métiers, selon un modèle de briques pour construire une application. Nous sommes au début d'un nouveau mode de pensée applicative au sein d'une DSi ou d'un éditeur de logiciels. »

Yves Grandmontagne

Composition de la plateforme Progress Software

- ▶ ABL (Progress 4GL), langage de quatrième génération, est au cœur de l'offre de Progress Software.
- ▶ OpenEdge offre le langage ABL, une base de données, et un ensemble d'outils de programmation. Cette base est complétée d'outils développés ou acquis :
 - ▶ Java ;
 - ▶ un moteur de BPM (Business Project Management) ;
 - ▶ Corticon, un moteur de règles business ;
 - ▶ RollBase, pour les interfaces SaaS et les templates ;
 - ▶ DataDirect, un framework d'intégration des données, qu'elle soit on-premise, No-SQL, SaaS, Big Data, ou encore des réseaux sociaux.
- Et d'une nouvelle plateforme dans le Cloud :
- ▶ Progress Pacific est une plateforme PaaS, association dans le Cloud de la plateforme de développement OpenEdge, avec Corticon, RollBase et DataDirect.

Progress Pacific



mars

Next Game Frontier : une conférence technique sur le web gaming



Le campus de Microsoft accueille une conférence sur le jeu sur Internet le 13 mars prochain. Au menu de cette journée :

- ▶ Jeux 3D avec WebGL et Babylon.js
- ▶ Comment créer des ensembles 3D pour les jeux web et mobiles, les problématiques du designer
- ▶ Les améliorations HTML5 avec WebCL
- ▶ Présentation des frameworks three.js et minko.io

Une conférence à ne pas rater ! site : <http://lanyrd.com/2014/ngf2014/>



Le 29 mars prochain, une journée Windows Azure aura lieu en France, à Paris et à Lyon. Le Global Windows Azure est un événement mondial avec des conférences, des sessions techniques, des rencontres. Objectif : montrer Azure et les possibilités techniques et de développement. En France, c'est la communauté ZeCloud qui l'organise.

Les participants devront apporter leur environnement de développement et les outils nécessaires pour participer à la journée... Cloudmagazine.fr

est partenaire de la journée, ainsi que Programmez !.

Site : <http://goo.gl/TJlw4t>

avril

Journée française des tests logiciels

Le 1er avril se tient la journée française des tests logiciels à Paris. Il s'agit de la 6e année. 12 conférences se succèderont toute la journée : gestion des exigences, test et valeur ajoutée métier, retours d'expériences, perspectives du marché des tests, les tests et l'agilité, le modèle TMMI, etc. Cette année encore, la journée va tenter d'expliquer le bon usage des tests, pourquoi il faut en faire et comment les faire. Site : <http://www.jftl.org>

Cloud Computing World Expo

Les 9 et 10 avril, le Cloud Computing sera à l'honneur à Paris. De nombreuses conférences seront organisées sur les deux jours : Open Cloud Forum, le poste connecté, comment choisir son Cloud, Opex ou Capex, le forum OW2... Site : <http://www.cloudcomputing-world.com>

Devoxx France 2014

L'événement du monde Java se déroulera du 16 au 18 avril prochains à Paris. Cette année les principaux thèmes seront : Java, HTML 5, mobile, Cloud, Big Data, NoSQL, agilité, DevOps, les langages alternatifs, les startups et aussi le future Devoxx. Avec l'arrivée de Java 8, Java sera bien entendu à l'honneur mais Devoxx s'ouvre à d'autres horizons. La première journée est dite université. Ce sont des conférences de 3h et des ateliers. En fin de journée, des sessions courtes se succèderont. Les 17 et 18 avril, c'est la conférence proprement dite avec les multiples conférences. Site : <http://www.devoxx.fr>

ScrumDay 2014

L'événement phare de l'agilité et de la méthode Scrum en France aura lieu cette année les 10 et 11 avril à Disneyland Paris. Le Scrum User Group France est toujours le maître d'œuvre. Site : <http://www.scrumday.fr>

INRIA
INVENTEURS DU MONDE NUMÉRIQUE

VOUS PENSEZ
QUE L'INFORMATIQUE
PEUT CHANGER LE MONDE ?
CHANGEONS-LE ENSEMBLE.

if (l == tab [c-a])

int a;
while (c-a>=l)
{
}

if (l == tab [c-a])

if (tab[c-a])
b++;
else
a++;

INGÉNIEURS CONFIRMÉS ET SPÉIALISTES (H/F)

A l'interface des sciences informatiques et mathématiques, les ingénieurs d'Inria collaborent, au sein des 8 centres de recherche, avec leurs collègues chercheurs à l'avancée des sciences et des technologies numériques de demain. Ingénieurs expérimentés en R&D, informatique et mathématiques appliquées, Inria vous propose des opportunités. Mettez vos talents au service de notre institut, venez participer à la création de solutions numériques concrètes en interaction avec la société et le monde économique dans un environnement collaboratif, international, multiculturel et stimulant.

Découvrez nos offres et nos modalités de candidature sur inria.fr - Campagne de recrutement ouverte jusqu'à fin septembre 2014.

BadBios, le virus qui n'existe pas...

Le 31 octobre dernier, Dan Goodin, dans les colonnes d'Ars Technica, relate l'aventure extraordinaire du virus BadBios découvert par le chercheur Dragos Ruiu.

Un virus rootkit capable de se cacher dans l'espace adressable de la mémoire BIOS (un rootkit donc, puisque préexistant avant le lancement du noyau), multiplateforme (Windows/OSX/BSD) et capable de se propager de machine en machine sans même qu'un câble réseau ne relie les ordinateurs entre eux. Dragos Ruiu désactive les cartes WiFi, supprime les accès Bluetooth, vérifie qu'il ne traîne pas une quelconque extension infrarouge... le virus se propage toujours de machine en machine. Mais la diffusion de l'infection semble s'arrêter lorsque le chercheur débranche les haut-parleurs et micros des stations de test.

La communauté est en émoi, car de telles constatations impliquent des choses peu banales.

- ▶ BadBios (c'est son surnom) aurait donc en mémoire un nombre d'exploits distants assez important pour infecter plusieurs types de noyaux. Ce qui en soi n'est pas impensable, surtout si ces exploits visent des applications elles-mêmes multiplateformes, genre Java ou Acrobat Reader,
- ▶ BadBios serait multivectoriel, capable de se transporter via clef USB ou liaison sans fil, peut-être via audio dans le domaine du sub-audible, ou plus simplement Ethernet cuivre... ce serait un opportuniste patenté. Là encore, rien d'inimaginable, Stuxnet, entre autres, a prouvé que tcp/ip n'était pas un point de passage obligatoire, et le coup de la disquette infectée a encore de beaux jours à vivre,
- ▶ Badbios (et c'est là la chose la plus inquiétante) possèderait une bibliothèque d'exploits « driver level » capables de convaincre une machine saine de se mettre en « état de réception » quel que soit le niveau de correctif appliqué et sans qu'il y ait eu intrusion préalable. C'est une série d'attaques driver level qui a ouvert les hostilités lors du premier

« MOAB », ou Month of Apple Bug lancé en décembre 2006 par un chercheur en sécurité de talent, Kevin Finisterre.

Reste enfin la question du « virus bios », longtemps une légende urbaine destinée à effrayer les enfants turbulents et les informatico-crédules. Ce vecteur d'attaque existe réellement, plusieurs preuves de conception ont été développées par d'autres chercheurs en sécurité. L'on pourrait citer l'équipe Jonathan Brossard (Toucan System)/Florentin Demetrescu (Cassidian) et de leur rootkit furtif Rakshasa, qui voyage de l'espace Bios à la zone de boot PXE.

Il en va de même pour l'hypothèse de cette transmission dans le domaine audio ... Il n'a pas fallu deux semaines pour que deux autres chercheurs de l'Institut Allemand Fraunhofer, Michael Hanspach et Michael Goetz, parviennent à prouver que deux ordinateurs pourraient dialoguer de cette manière sans que cela se remarque, en exploitant les plages de fréquence situées au-delà du seuil d'audition (20 kHz et plus). Ce qui tend à prouver une seule chose : les haut-parleurs des ordinateurs utilisés, des Lenovo T400, sont de bonne qualité et leurs micros très fiables... mais en aucun cas les Allemands ne sont parvenus à coder un exploit capable de déclencher à distance une réception de binaire en mode audio. Le PoC fait plouf.

Après un suspense insoutenable, Drago Ruiu déclarait le 22 novembre dernier sur Tweeter qu'il s'était trompé sur plusieurs points et que le super-virus BadBios n'existe pas. Déconvenue dans le petit cercle d'auteurs de romans de technoscience.

Il n'en demeure pas moins que si BadBios comportait de gros morceaux de fantasmes et de vent, le concept même de bootkit Bios et de vecteur de propagation non conventionnelle n'appartient pas au domaine de l'imaginaire. Il en va de même pour l'exploitation de failles mul-

tiplateformes (il suffit de lire quelques beaux CVE Adobe ou Oracle pour s'en convaincre). Quant aux attaques au niveau driver ou noyau par des ports conventionnels (Wifi, Firewire etc), cela n'a rien d'exceptionnel. Chacun de ces exploits pris séparément est plausible et a déjà fait l'objet de recherches publiées, voire de développements et de mises en application. Mais un cocktail réunissant tous ces ingrédients reste d'une complexité élevée, et ne peut intéresser que des amateurs d'attaques ciblées. Il n'y a quasiment rien à craindre pour l'heure d'une invasion de BadBios aussi destructrice qu'un Conficker/Downadup.



LE TOP 5 DES HACKERS ARRÊTÉS AUX ETATS-UNIS



1 mafiaboy

1,2 à 7,5 milliards \$ de préjudices.
Exploit datant de l'an 2000 à l'âge de 15 ans.
Cibles : Yahoo, CNN, ebay, etc.

2 J4Guar

200-400 millions \$ de préjudices.
Exploit : 2006-2008, à l'âge de 25 ans. 20 ans de prison.
Cibles : OfficeMax, Forever 21, etc.

3 RTM

97 millions \$ de préjudices.
Exploit : 1988, à l'âge de 23 ans.
Cible : 6000 PC

4 KWYJIBO

80 millions \$ de préjudices.
Exploit : 1999, à l'âge de 31 ans.

5 Vladimir Levin

10,7 millions \$ de préjudices.
Exploit : 1994, à l'âge de 23 ans.

Source : SimpliSafe

Lien : <http://goo.gl/iw1WPk>

Faille « zero day » dans Internet Explorer 10

Fireeye a découvert une faille dans IE 10. L'exploit est référencé CVE-2014-0322. Il semblerait que l'attaque soit partie des Etats-Unis. Les chercheurs à l'origine de la découverte disent que l'attaque est classique et utilise un code ActionScript. Microsoft avait corrigé 15 failles dans un patch de sécurité, sorti quelques jours auparavant.

Pour en savoir plus : <http://www.fireeye.com/blog/?p=4514>



Le ransomware pourrait bien être la principale menace en 2014

Connus sous le nom de ransomware, ces logiciels malveillants sévissent depuis plus de 10 ans. Ils empêchent le fonctionnement normal de l'ordinateur généralement en configurant le système d'exploitation pour se substituer à l'interface principale, par exemple l'explorateur Windows, ou en chiffrant certains fichiers importants.

L'utilisateur ciblé peut reprendre le contrôle de son ordinateur en s'acquittant d'une somme d'argent auprès du pirate qui lui enverra le mot de passe ou la clé de chiffrement lui permettant de restaurer le bon fonctionnement du système. Parfois, le ransomware impose à la cible de réagir plus rapidement en signalant la présence de fichiers illégaux sur la machine et qu'une amende doit donc être payée pour restaurer l'ordinateur. À titre d'anecdote, le FBI a déclaré avoir reçu des centaines d'appels de la part de victimes qui proposaient de payer une amende lors de la dernière campagne du ransomware Reveton en 2012, code malveillant qui a utilisé le nom du FBI pour exécuter certaines menaces.

Avec la diffusion du logiciel malveillant Cryptolocker cette année, le ransomware prend de nouvelles proportions en termes de sophistication technique, d'évolutivité et de qualité d'exécution. Début septembre, Cryptolocker a été envoyé à des utilisateurs d'ordinateurs sous la forme d'un fichier joint à un courrier électronique. Puis, une fois la pièce jointe ouverte, le logiciel malveillant a commencé à chiffrer des documents et autres fichiers utilisateur au moyen d'une puissante clé de chiffrement AES symétrique et d'une cryptographie à clé publique. Ce ransomware a ensuite demandé une rançon en Bitcoins, la crypto-monnaie dont la popularité ne cesse de croître depuis six mois. Mais beaucoup de victimes ont rencontré

des problèmes pour réunir les 2 bitcoins exigés dans un délai de 72 heures. En effet, une bonne connaissance et du temps sont indispensables à l'installation de l'infrastructure permettant d'acheter des Bitcoins. Les auteurs de Cryptolocker se sont donc adaptés et ont lancé un service de déchiffrement Cryptolocker qui porte à 10 jours l'échéance, mais en échange d'une rançon plus élevée, à savoir 10 Bitcoins. Des milliers de personnes sont affectées par ce fléau et nombre d'entre elles choisissent de payer, le seul autre remède connu étant la restauration du système à partir d'une sauvegarde. Les victimes sont aussi bien des particuliers que des entreprises, d'où parfois le chiffrement d'unités entières de disque réseau. On rapporte même qu'un service de police local s'est résolu à payer la rançon pour pouvoir restaurer son environnement d'exploitation. Une fois la rançon acquittée, Cryptolocker télécharge la clé de déchiffrement via le réseau TOR et lance un processus de récupération long, à condition que votre antivirus n'ait pas empêché le fonctionnement de cette clé en tentant de la neutraliser.

Se défendre contre Cryptolocker s'avère étonnamment difficile. Et pourtant, Cryptolocker s'exécute comme un simple programme utilisateur et il n'exige aucun privilège spécial pour chiffrer vos fichiers utilisateur. Quant à nos traditionnelles suites de protection des équipements, elles se contentent de détecter des

variantes connues de ces logiciels malveillants alors que les pirates peuvent très facilement modifier leur code pour les rendre indécelables.

La plupart des fournisseurs de services de sécurité insistent sur la formation des utilisateurs pour éviter l'exécution de logiciels malveillants. Ils suggèrent en outre le lancement de sauvegardes fréquentes et fiables pour faciliter la restauration des fichiers chiffrés, une parade somme toute très faible et peu satisfaisante.

Je crains que Cryptolocker aura le même impact, mais dans la catégorie des logiciels ransomware, que le virus Conficker qui a sévi 2008 et inspiré une nouvelle génération de logiciels malveillants aux fonctionnalités évoluées. En effet, son intérêt financier attirera à lui seul de nombreux développeurs de logiciels malveillants qui se joindront au mouvement et amélioreront leurs programmes. Se prémunir contre cette menace sera le plus gros défi de 2014 en termes de sécurité des équipements.



Wolfgang Kandek
CTO Qualys Inc. Plus de vingt ans d'expérience dans la sécurité, le développement et la gestion des systèmes d'informations. <https://community.qualys.com/blogs/laws-of-vulnerabilities>



Abonnement **PDF**
30 € par an soit 2,73 € le numéro
www.programmez.com



A 38 ans, Fabrice est aujourd'hui Chief Technical Officer Intranet & Extranet dans une entreprise internationale dont le siège est à Genève. Il occupe ce poste depuis 2007, après avoir été développeur .NET, DBA SQL Server, puis consultant spécialisé sur les technologies Microsoft. Sa formation de base est assez longue : un BAC S (obtenu avec un peu de mal :)), puis un BTS en Agroalimentaire (ENSMIC), suivi d'une école d'ingénieur dans le même milieu (IST). Il a quitté cette école en cours de route pour rejoindre une école d'ingénieur en informatique (ESGI, option réseau & Système). Ce diplôme en poche, il a travaillé quelques années sur la région parisienne avant de venir sur Genève où il a repris ses études en 2011 pour un Executive MBA en Risk Management (Université de Genève & HEC), suivi d'une année en Corporate Internal Audit (HEG). Tout ceci permet de réaliser que la formation ne s'arrête pas avec le diplôme d'ingénieur ; on peut tous reprendre nos études à tout moment, il suffit d'avoir la motivation et le projet.

De l'agroalimentaire à l'informatique : tous les chemins mènent aux codes

Comment es-tu tombé dans l'informatique et plus spécialement dans le développement ?

L'informatique est venue à moi très tôt, comme beaucoup de ma génération, avec un Amstrad CPC 464, que je programmais petit, en suivant les lignes de codes de livres disponibles à l'époque (en BASIC). Puis j'ai laissé cela de côté jusqu'à mes études supérieures, où j'ai replongé avec les premières machines sous Windows 3.1, ensuite je suis monté en compétences seul (création de site Web, bricolage de matériel, dépannage d'amis et famille, ...), pour enfin décider d'en faire mon métier en suivant une école d'ingénieur spécifique (ESGI).

Le développement est venu en parallèle, car j'ai toujours mené les deux de front, avec une spécialisation en Réseau & Système et tous mes stages pratiques en développement WEB. L'aboutissement fut alors de devenir développeur .NET, administrateur système Windows Server et DBA SQL Server, ce que je fais encore aujourd'hui. Cependant, le développement est pour moi mis de côté, sauf pour les scripts PowerShell, que je crée pour des besoins business ou techniques. Tout ceci fut aussi accompagné d'une participation active au sein de différentes communautés de développeurs sur les plateformes Microsoft, ce qui m'a permis d'être reconnu Microsoft MVP (Most Valuable Professional) depuis 2003. Tout d'abord MVP ASP.NET, puis MVP SQL Server pour maintenant être MVP SharePoint architecture.

Pour toi, qu'est-ce qui fait que l'on aime toujours et encore le développement, la technique ?

Le développement en tant que tel possède une grande part de créativité et d'innovation, car il demande toujours à son créateur de trouver une solution afin de résoudre une problématique ou un besoin. Même si je ne fais plus de réel développement, je comprends bien cette logique et cela donne une certaine

élégance à ce métier. Le développeur au fond de la cave est de plus en plus dépassé au profit de la star qui créera le futur Facebook : les TechDays de Paris l'ont encore prouvé cette année.

Tu as gardé un regard très geek : gadget, veille techno. C'est important pour ton job et ta passion ?

J'aime bien me tenir au courant et je suis toujours à l'affût de solutions techniques qui peuvent me rendre un service personnel ou professionnel. En revanche, je suis moins sensible aux gadgets, car je raisonne beaucoup trop avant un achat pour qu'un gadget inutile passe ce filtre :) Pour mon job, les gadgets sont plus un moyen d'entrer en relation sociale avec d'autres personnes, dans d'autres métiers que l'IT.

Etre développeur n'est pas toujours facile : pression, évolution constante, frustration des projets et des «chefs», c'est quoi pour toi d'être développeur aujourd'hui ? Le job a-t-il changé depuis tes débuts ?

Depuis ma sortie d'école, le marché a beaucoup changé et les acteurs se sont concentrés, demandant de plus en plus de rentabilité.

C'est une situation générale et non liée uniquement à l'IT.

En revanche, ce métier est passé d'un mode artisanal à un mode de production industriel très (voire trop) rapidement.

On voit donc dans ce milieu, différents modes de fonctionnement, qui sont aussi issus de négociations de plus en plus dures avec les clients (le prix d'un Jour/homme est de plus en plus bas) et d'une concurrence mondiale de plus en plus forte.

Ainsi, certaines entreprises misent sur la qualité et le service sur-mesure, ce qui laisse plus de marge de manœuvre au développeur, tandis que d'autres cherchent à maintenir une rentabilité en mettant une pression de plus en plus forte sur le développeur en bout de



LE BUREAU DE FABRICE

Mon bureau est implanté dans un open-space important, ce qui améliore les échanges avec mes collègues, même si la concentration en pâtit un

peu. Dans ce cadre, je travaille souvent avec le casque, écoutant soit des podcasts musicaux (Tiesto, Armin Van Buuren, ...), soit des pod-

casts d'émissions de radio (certaines émissions de BFM Business comme « Goût de Luxe », « Chine Hebdo » ou encore « l'atelier numérique »). Ma

machine de travail est principalement un laptop Lenovo, car tout le reste est sur des serveurs dont je prends le contrôle à distance.

chaîne. Quoi qu'il en soit, le monde est devenu logiciel, et les bons développeurs seront toujours recherchés, il faut juste faire bien attention à suivre la technologie dans le bon timing, ni trop tôt, ni trop tard.

Et en dehors du boulot, qu'est-ce que tu aimes faire ? Comment trouves-tu l'équilibre entre travail, vie privée, passion, famille ?

Je faisais beaucoup de sport avant, mais l'âge avançant, je me consacre à d'autres activités aujourd'hui, en profitant des lieux d'exceptions avec ma femme.

Peux-tu nous présenter ton quotidien en quelques mots ?

Mon quotidien est destiné à gérer un ensemble

de fermes Microsoft SharePoint (toutes les versions aujourd'hui), soit directement, soit en support haut niveau. Cela revient à gérer plus ou moins 50 serveurs déployés à travers le monde. A cela, il faut ajouter la gestion des serveurs de bases de données Microsoft SQL Server qui sont dans des configurations variées (toutes les versions sont présentes). Enfin, je fournis de plus en plus de consulting interne, pour des besoins ponctuels sur des cadres techniques spécifiques ou plus généraux.

Comment vois-tu ton job évoluer ?

Après mes différentes formations, MBA Risk Management et Audit Interne, mon travail risque plus d'évoluer sur la partie consulting interne pour aider les lignes de business à résoudre des

situations complexes, ce qu'on appelle généralement les due diligence.

Des conseils aux étudiants et développeurs qui nous lisent ?

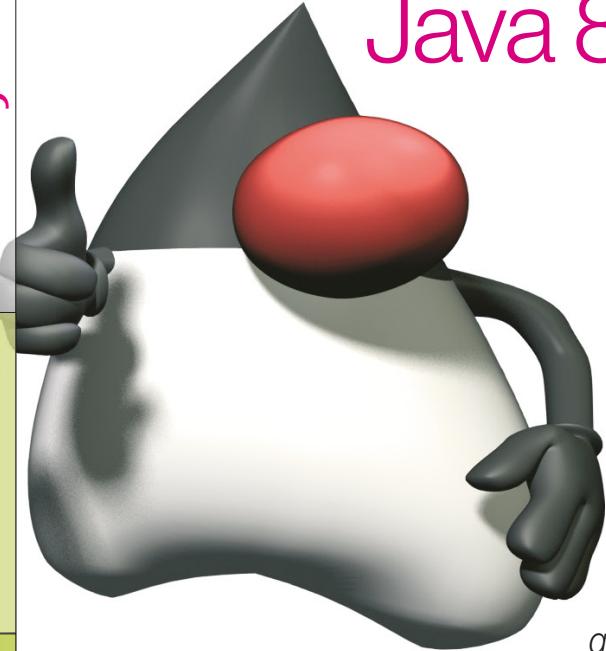
L'informatique est une porte d'entrée vers tout, mais il ne faut pas s'enfermer dans ce milieu, sous peine de se déconnecter du monde réel. Il faut s'ouvrir à toute opportunité qui peut se présenter, que ce soit via le milieu associatif, via une formation spécifique (ou générale) ou encore simplement avec des amis (ou de la famille). Il faut rester curieux de tout ce qui se passe autour et rester ouvert sur les changements à venir. Facebook, Twitter, Google ou Criteo sont des entreprises qui n'existaient pas 30 ans auparavant.

PROGRAMMEZ!
le magazine du développeur
www.programmez.com

Toute l'actualité des technologies et du développement sur www.programmez.com



Java 8 : est-il toujours un langage actuel ?



Il faut toujours se méfier des articles dont le titre est une question. Le plus souvent, le but est d'inciter le lecteur à lire l'article pour y trouver la réponse, alors que la plupart du temps, la réponse se trouve déjà dans la question. Se demander si, avec la sortie de cette nouvelle version, Java est toujours un langage actuel pourrait laisser supposer que non; Java est un langage d'un autre âge, conçu à une autre époque (ce qui est vrai), et qui a peut-être eu des problèmes dans son évolution, ce qui n'est probablement pas faux. Peut-être est-il

intéressant de se pencher sur l'histoire de ce langage, de son écosystème, afin de mieux comprendre où nous en sommes, et de comprendre les enjeux de cette nouvelle version, la 8ème.

Du projet Oak à Java 8

Java est un langage conçu dans la première moitié des années 90, il y a donc environ 25 ans, sous le nom de « projet oak ». La première idée des équipes qui conçoivent ce nouveau langage est qu'il soit portable. Non pas portable au sens du C, que l'on peut compiler sans modification du code source sur tous les OS, (même si la réalité des choses est un peu plus problématique), mais au sens où un code Java déjà compilé peut s'exécuter n'importe où. Leur objectif annoncé est de proposer cette technologie aux câblo-opérateurs américains, qui pourront ainsi déployer leurs applications sur les nombreux modèles de décodeurs dont ils équipent leurs clients.

Et effectivement, les premières versions expérimentales de Java fonctionnent sur une *set top box* graphique, qui exhibe un petit personnage triangulaire blanc et noir au nez rouge, devenu la fameuse mascotte Duke. Cette utilisation de Java viendra, mais beaucoup plus tard.

Son succès industriel ne vient pas immédiatement. Cela vient aussi du fait que les premières versions diffusées sont encore probablement un peu expérimentales. La première version réellement utilisable est la 2, sortie en 1998, il y a donc 16 ans. Durant ces 16 dernières années, les versions se succèdent jusqu'à la 8, sortie prévue le 18 mars 2014.

Sun réussit à maintenir un rythme de sortie d'environ une version majeure tous les deux ans. Et puis à la version 6 en 2006, l'annonce du rachat de Sun par Oracle vient perturber les choses.

Au premier degré, le rachat perturbe le rythme des sorties : la version 7 ne sort qu'en 2011, soit après cinq longues années d'attente. Techniquement la période n'est pas la bonne : en 2005 le débat « comment intégrer les lambdas expressions » en Java s'ouvre.

La question n'est pas d'intégrer ou non cette technologie, mais porte bien sur le comment. Le délai dans la sortie de la version 7 met en sommeil ce débat, et cette intégration également. Rétrospectivement, ce retard n'est peut-être pas une si mauvaise chose, cela aura probablement permis de ne pas tomber dans certains écueils.

Le rachat perturbe aussi les esprits. Sun Microsystems jouit depuis long-

temps d'une excellente image auprès des développeurs open source. Aux débuts de Java, c'est Sun qui permet de conclure les développements de la première JVM disponible sur Linux, au travers du projet Blackdown. Linux est encore un système émergent à cette époque, et n'est pas dans la liste des OS officiellement supportés.

Oracle ne jouit pas de cette image. Le fait qu'il se retrouve propriétaire de Java provoque plutôt la défiance de la part de cette communauté, un peu turbulente, il est vrai. D'autant que les premières décisions d'Oracle vis-à-vis du JCP, instance qui gouverne les évolutions de Java, ne vont pas dans le sens attendu. On se rappelle du départ tonitruant de Doug Lea, professeur à la *State University of New York* à Oswego, éminent spécialiste de la programmation concurrente, et auteur des principales API Java dans ce domaine.

En 2010, Oracle se retrouve donc avec un problème à résoudre : quelle sera la prochaine version majeure de Java ? Et surtout : que mettra-t-on dans cette version ? Voilà quatre ans qu'il n'y a pas eu de version majeure, le projet Lambda est dans l'ornière, et les rumeurs commencent à courir : « Java est en train de devenir le nouveau Cobol », c'est-à-dire un langage du passé, condamné à disparaître à plus ou moins long terme.

La réponse d'Oracle a été de sortir une version 7, avec ce qui était alors prêt à être publié. Donc une version 7 sans le projet Lambda. Cette version contient malgré tout de nombreux ajouts très intéressants : quelques modifications du langage, des mises à jour des API existantes, une nouvelle API I/O, un framework de calcul parallèle. Mais l'attente est déçue : les lambdas sont reportés pour la version 8, plus tard. La réaction de la blogosphère est lapidaire : Java 7 sans les lambdas déçoit.

Un deuxième événement vient perturber les choses. 2012 devait être l'année de l'intégration du projet Lambda dans Java 8. Elle sera l'année de la correction de problèmes de sécurité dans Java 6 et 7. Ces problèmes arrivent jusque dans la presse grand public, l'image de Java est atteinte, et Oracle doit consacrer toutes les ressources disponibles aux correctifs. Java 8 prend du retard, la date de sortie annoncée est repoussée de 6 mois. C'est donc en 2014 que Java 8 sort. On peut considérer que l'inté-

gration du projet Lambda, initialement prévue probablement pour 2008 a donc pris 6 ans de retard. Considérable pour un langage informatique.

L'introduction des lambdas : quel impact ?

En 1998, le JDK de Java 2 compte environ 1500 classes. Pour Java 4, dernière version avant les génériques, ce total s'élève à 2700 classes. Java 5 voit arriver les génériques, modification majeure du langage, et compte environ 3300 classes. Java 6, dernière version publiée par Sun microsystems, compte 3500 classes. En 8 ans, le nombre de classes du JDK augmente donc, le langage subit une modification majeure. La JVM, elle, ne change pas, en tout cas pas dans la façon dont elle exécute le code Java. Le bytecode Java ne change pas pendant plus de 10 ans.

Java 7 est une rupture de ce point de vue. Le langage évolue un peu (sans grande révolution toutefois), le nombre de classes passe à 7400, et une nouvelle instruction de bytecode est introduite. Cette augmentation du volume de l'API en fait l'une des plus grosses mises à jour du JDK.

De ce point de vue, Java 8 est une mise à jour modeste : on passe à 7700 classes. En revanche, le nombre de classes modifiées par l'introduction des lambdas, est supérieur aux nombres de classes qui avaient été modifiées par l'introduction des génériques en Java 5. Rares sont les parties du JDK à ne pas être touchées par l'arrivée des lambdas.

Mais encore plus que le nombre de classes, la JVM ou les API, ce sont les patterns de code que l'arrivée des lambdas bouleverse.

Le pattern Iterator

Tous les développeurs sont nourris, parfois sans le savoir, par un livre de *design patterns* fondamental : *Elements of Reusable Software* du célèbre *gang of four*. Ce livre a été publié pour la première fois en 1994, c'est-à-dire avant que la première version de Java ne soit diffusée. Il est toujours disponible en librairie, ce qui, pour un ouvrage d'informatique, est une réelle performance. Il contient des design patterns fondamentaux, dont le pattern Iterator. Ce pattern est au cœur du framework Collection, lui-même au cœur de toutes les applications Java existantes. Le framework Collection est apparu dans la version 2 de Java, en 1998 donc. La plupart des développeurs Java sont nés (du moins professionnellement) avec le pattern Iterator dans leur biberon.

L'introduction des lambdas en Java 8, rend d'une certaine façon ce pattern obsolète. Ce sont donc les habitudes de travail de millions de développeurs Java qui peuvent s'en trouver changées. Pourquoi changer un pattern qui a tant fait ses preuves ? Pour une raison simple. Ce pattern est au cœur de tous les traitements de données en Java. Ces traitements de données sont de plus en plus complexes, traitent des volumes de données de plus en plus importants, et ont besoin du parallélisme pour rester performants. La programmation de ces traitements devient donc de plus en plus complexe, au point que s'appuyer sur des implémentations optimisées des principaux patterns algorithmiques devient indispensable.

Et c'est précisément l'objet de l'introduction des lambdas en Java. Simplifier l'écriture des traitements complexes et s'appuyer sur des API optimisées notamment pour les traitements de type *map / filter / reduce*.

Premières expressions lambdas

Une première expression lambda

Qu'est-ce qu'une expression lambda en Java ? La réponse est simple : une façon alternative d'écrire une instance de classe anonyme. Le code suivant est écrit en Java 7, et permet de sélectionner les fichiers portant l'extension .java d'un répertoire dir.

```
File dir = new File(<<D:/tmp>>) ;
String[] fileNames = dir.list(new FilenameFilter() {
```

```
    public boolean accept(File dir, String name) {
        return name.endsWith(<<.java>>) ;
    }
}) ;
```

Une façon alternative d'écrire l'instance de classe anonyme passée en paramètre de la méthode list() est la suivante.

```
File dir = new File(<<D:/tmp>>) ;
String[] fileNames =
    dir.list((File dir, String name) ->
name.endsWith(<<.java>>)) ;
```

Le code suivant est une expression lambda de Java 8, implémentation de l'interface FilenameFilter.

```
((File dir, String name) -> name.endsWith(<<.java>>))
```

Construction d'une expression lambda

Comment une telle expression est-elle construite ? Et tout d'abord, comment le compilateur Java 8 reconnaît-il l'implémentation de cette interface ? Le compilateur Java 8 sait reconnaître le type d'une expression lambda par son contexte. Quand il rencontre le code suivant :

```
String[] fileNames =
    dir.list((File dir, String name) ->
name.endsWith(<<.java>>)) ;
```

Il sait que la méthode list() prend en paramètre une instance de FilenameFilter, et en déduit que l'élément passé en paramètre de cette méthode doit être de ce type. Ensuite, il constate que cette interface ne comporte qu'une seule méthode. L'expression lambda doit donc être une implémentation de cette méthode. Si l'interface implémentée comporte plus d'une méthode, une erreur de compilation est générée : une expression lambda ne peut pas être une instance d'une interface comportant plus d'une méthode.

Il ne reste plus au compilateur qu'à associer les éléments constitutifs de l'expression lambda qu'il analyse avec la méthode à implémenter. Quels sont ces éléments ? Ils sont au nombre de trois : les paramètres de la méthode, son type de retour, et éventuellement les exceptions qu'elle peut jeter.

Le compilateur doit donc reconnaître ces éléments :

- les paramètres sont déclarés en premier dans l'expression lambda ;
- suit la petite flèche : -> ;
- suit le corps de la méthode implémentée, qui retourne éventuellement quelque chose.

Évidemment, si l'un de ces éléments n'est pas correct, une erreur de compilation sera générée.

La plupart du temps, il est même possible d'omettre les déclarations de type des paramètres : le compilateur est capable de les deviner tout seul.

Cette première écriture :

```
(File dir, String name) -> name.endsWith(<<.java>>)
```

Peut donc devenir la suivante, plus légère, et tout aussi lisible :

```
(dir, name) -> name.endsWith(<<.java>>)
```

Un peu plus de détails

Une expression lambda est donc une autre façon d'écrire une implémentation d'une interface, à condition qu'elle ne possède qu'une unique

méthode. Formellement, ces interfaces particulières prennent donc un statut spécial en Java 8. On les appelle dorénavant des *interfaces fonctionnelles*. Une interface fonctionnelle en Java 8 est une interface qui ne possède qu'une unique méthode abstraite.

On peut l'annoter avec une nouvelle annotation du JDK : `@FunctionalInterface`. Ce point n'est pas obligatoire. Cela dit, si cette annotation est présente, alors le compilateur peut vérifier que l'interface annotée est bien fonctionnelle. Si elle ne l'est pas, il va générer une erreur, dans l'IDE, donc à un moment où il est facile de corriger les choses.

Que faire des lambdas ?

Une nouvelle syntaxe dans un langage est toujours une opportunité pour écrire de nouveaux patterns de code. L'introduction des expressions lambdas a pour objectif, entre autres, de faciliter l'écriture d'opérations map / filter / reduce en Java.

De quoi dispose-t-on en Java 7 pour cette implémentation ? Essentiellement du pattern iterator, ce qui est un peu maigre. La façon de programmer une opération de type map / filter / reduce jusqu'en Java 7 consiste à itérer sur les éléments de la collection qui les porte, et à appliquer les différentes opérations. En fait, cette façon de faire n'a pas changé depuis l'introduction de l'API Collection en 1998. Il était peut-être temps, effectivement, de revoir un peu les choses, et d'enrichir les fonctionnalités du JDK sur ce point.

Map / filter / reduce

Introduction

Ce traitement (on pourrait parler de pattern algorithmique) se retrouve depuis plusieurs années au centre des traitements de données. On se rend compte que cette modélisation est en fait un cadre, une forme d'abstraction très générale, dans laquelle entrent nombre de traitements spécifiques.

Cette abstraction comporte trois étapes.

La première est l'étape de *mapping*. Un mapping consiste à transformer une liste d'un certain type, en une liste d'un autre type. Par exemple, un mapping peut transformer une liste d'objets de type `Personne`, en une liste d'entiers, où chaque entier est l'âge de chaque personne. Un mapping ne change pas la taille de la liste en entrée, ce qui fait que l'on a autant d'âges que l'on avait de personnes. Un mapping fonctionne avec une *fonction de transformation*, appliquée à chacun des éléments de la liste en entrée.

La deuxième étape est le filtrage. Le filtrage est fondamentalement différent d'un mapping. Là où le mapping change le type de la liste en entrée, le filtrage ne le change pas. Le résultat du filtrage d'une liste d'âges, est donc aussi une liste d'âges. En revanche, alors que le mapping conserve le nombre d'éléments de la liste en entrée, le filtrage diminue ce nombre. Un filtrage fonctionne avec un prédictat. Un prédictat prend en paramètre un élément de la liste, et retourne vrai si l'élément considéré doit être conservé dans la liste en sortie, faux si l'élément doit en être retiré.

La dernière étape est la réduction. En première approche, une réduction correspond aux fonctions d'agrégations du SQL : `sum`, `avg`, `max`, `min`, etc. On peut aussi conceptualiser l'étape de réduction plus largement : une réduction consiste à accumuler les éléments de la liste en entrée dans un container, au sens le plus large possible. Si les éléments en entrée sont des entiers, ce container peut être lui-même un entier, et l'opération d'accumulation, tout simplement la somme. Si ces éléments sont des chaînes de caractères, ce container peut être aussi une chaîne de caractères, et l'opération d'accumulation, la concaténation. Enfin, si les éléments en entrée sont des objets quelconques, le container peut être une liste, et l'opération d'accumulation l'ajout de ces éléments dans la liste.

Ce dernier point de vue permet d'envisager des opérations de réduction

très sophistiquées, permettant de construire des tables de hachage complexes par exemple.

Si la modélisation d'un mapping ou d'un filtrage ne pose pas vraiment de problème, celle de la réduction est un peu plus délicate. De quels éléments a-t-on besoin pour spécifier une réduction ?

Pour répondre à cette question, il faut comprendre comment la réduction fonctionne exactement, et comment elle peut être optimisée.

Paralléliser ses traitements

La première optimisation dont on a besoin est probablement la parallélisation. En 2005, une révolution a eu lieu en électronique numérique : les processeurs ont cessé de voir leur fréquence de fonctionnement augmenter, comme c'était le cas auparavant. Cette période bénie dispensait les développeurs d'optimiser leurs traitements, puisqu'une application peu performante, se retrouvait rapide, comme par magie, par le simple déploiement sur un processeur plus rapide.

2005 marque la fin de cet âge d'or : la fréquence des processeurs cesse d'augmenter. Comment la nouvelle puissance des processeurs s'exprime-t-elle ? Par la multiplication de leurs coeurs. Aujourd'hui un processeur compte une petite dizaine de coeurs, demain il pourra en compter des dizaines, des centaines, des milliers.

Comment exploiter cette puissance ? La réponse est simple : en parallélisant les traitements. La parallélisation des traitements massifs de données est la principale façon de diminuer les temps de ces traitements. Jusqu'en version 6, rien dans le JDK ne permet de paralléliser les traitements de façon simple. En JDK 7, le framework Fork / Join apporte une solution technique. Cette solution reste complexe à mettre en œuvre, délicate à mettre au point, à tester, et éventuellement à corriger. Jusqu'en Java 7, paralléliser des traitements de données reste une affaire de développeurs chevronnés, connaissant parfaitement la programmation concurrente.

Le calcul parallèle n'est malheureusement pas qu'une affaire de technique de programmation. Des problèmes purement algorithmiques se posent également. Paralléliser des traitements reste, et restera toujours, une tâche délicate, à mener avec précaution. Paralléliser des traitements de façon naïve peut mener à une dégradation des performances d'une application, allant même jusqu'à l'effondrement complet d'un serveur.

Intégrer ces traitements dans les API est donc une façon de décharger les développeurs de ce genre de tâche, et donc de rendre accessible le parallélisme, entre autres, à la plupart des applications. Paralléliser un traitement en Java 8 est, du moins du point de vue technique, aussi simple qu'appeler une méthode `parallel()` sur le bon objet. Les problèmes algorithmiques, eux, restent entiers.

Mapper et filtrer en parallèle

Le cas du mapping et du filtrage ne posent quasiment pas de problème à paralléliser. Il faut simplement prendre garde à ce que l'opération de filtrage soit sans état pour que les choses se passent bien.

Prenons un exemple de mapping simple : la transformation d'une liste d'objets de type `Person` en leurs âges, par application de l'opération `p.getAge()`.

```
List<Person> list = ... ;
List<Integer> ages = list.stream()
    .map(p -> p.getAge())
    .collect(Collectors.toList()) ;
```

L'appel à la méthode `collect()` permet juste de stocker le résultat dans une liste. En première approche, on peut se dire que la parallélisation d'un tel traitement est immédiate, et ne posera pas de problème. Effecti-

vement, distribuer la liste de personnes sur les cœurs du processeur qui effectue le traitement semble immédiat, la détermination de l'âge d'une personne ne dépend que de l'objet personne passé en paramètre, donc tout doit bien se passer.

Il n'en est en fait rien. Une liste en Java a une sémantique précise : elle retient l'ordre dans lequel ses objets lui ont été ajoutés. L'ordre des personnes en entrée a donc son importance : il y a un premier, puis un deuxième etc... Cette sémantique doit être respectée, et donc l'ordre des âges de la liste retournée doit être le même que l'ordre des personnes qui ont ces âges. Autrement dit, le premier âge de la liste retournée doit être l'âge de la première personne en entrée, et ainsi de suite. Cette information implicite doit être maintenue en cas de parallélisation. Elle doit donc être partagée et mise à jour par chacun des cœurs : elle va constituer un point de contention. Le même traitement sur une collection plutôt qu'une liste ne pose pas ce problème.

D'autres méthodes posent ce genre de problème dans l'API Stream, et vont se paralléliser de façon délicate. Paralléliser un traitement de façon judicieuse est un point délicat, même sur des opérations qui semblent simples.

Réduire en parallèle

L'opération de réduction, de ce point de vue, est un peu plus problématique. La réduction d'un ensemble d'objets est spécifiée de façon unitaire, sur deux objets, et appliquée récursivement sur toute la collection.

```
List<Person> list = ... ;
List<Integer> ages = list.stream()
    .map(p -> p.getAge())
    .reduce(0, (i1, i2) -> i1 + i2) ;
```

Une opération de réduction doit pouvoir être appliquée dans n'importe quel ordre, et toujours retourner le même résultat.

Pour cela, elle doit posséder une propriété que les collégiens connaissent bien : l'associativité. Être associatif signifie que réduire deux éléments i_1 et i_2 , puis réduire le résultat avec un troisième élément i_3 donne le même résultat que réduire tout d'abord i_2 et i_3 , puis le résultat avec i_1 . La propriété peut paraître évidente, il n'empêche que ni le compilateur ni la JVM ne mettront personne en garde si une réduction non associative est exécutée sur une liste d'objets.

Les choses deviennent encore plus délicates en parallèle. Une réduction non associative appliquée en série à une liste donnée d'objets, retournera toujours le même résultat. On peut relancer le traitement plusieurs fois, le résultat est parfaitement reproductible. Il n'en est rien en parallèle. Une même réduction non associative, appliquée plusieurs fois sur une même liste d'objets a toutes les chances de retourner des résultats différents à chaque exécution. Encore une fois, il n'est du ressort ni du compilateur, ni de la JVM, d'informer le développeur du fait que la réduction est non associative.

Optimisation des réductions

Certaines opérations de réduction ont besoin de balayer toutes les données disponibles pour déterminer leur résultat. C'est le cas des opérations courantes, telles que `max()`, `min()`, `sum()`, etc... Ce n'est pas le cas de toutes les opérations de réduction. L'opération `allMatch()`, par exemple, retourne vrai si le prédictat passé en paramètre est vrai pour tous les éléments traités. Ce qui signifie, à l'inverse, que, dès que l'on rencontre un élément pour lequel le prédictat est faux, le retour du `allMatch()` sera faux. L'itération peut alors s'interrompre avant que toutes les données ne soient consommées.

Il y a une conséquence à ce constat : la bonne façon d'appliquer des opé-

rations de type map / filter / reduce à un ensemble de données, consiste à les appliquer en une seule itération plutôt qu'en une itération par opération. Cette approche est qualifiée de *lazy*, puisqu'elle peut s'arrêter dès que le résultat final est connu, dans certains cas avant d'avoir consommé toutes les données.

API Stream

Un nouveau concept dans le JDK

Le JDK 8 introduit un nouveau concept sous forme d'une interface : Stream. C'est sur cette interface que le pattern map / filter / reduce est disponible, et non pas sur l'interface Collection comme on aurait pu s'y attendre.

Qu'est-ce qu'un Stream ? Techniquement il s'agit d'une interface paramétrée. On parle donc de « stream de chaînes de caractères », ou de « stream de personnes ». Cette nouvelle notion est construite sur quatre idées simples.

- ▶ Un stream ne porte pas de données. Une collection porte ses données, ce qui rend sa duplication éventuellement coûteuse.
- ▶ Un stream est construit sur une source de données. Une source de données peut être une collection ou un tableau, mais aussi une entrée / sortie ou un itérateur. Une source de données peut ne pas être bornée, ou tout du moins, la quantité de données qu'elle va effectivement produire peut ne pas être connue *a priori*. Un stream n'a pas le droit de modifier la source de données sur laquelle il est construit. S'il le fait, cela pourra avoir des conséquences non prévisibles sur les résultats des traitements. Cette règle rend la parallélisation des traitements possible.
- ▶ Nous venons de voir que la source sur laquelle un stream est construit peut ne pas être bornée, mais nos temps de traitements, eux, doivent s'exécuter en temps fini ! Il nous faut donc un mécanisme qui puisse nous garantir ce point.
- ▶ Enfin, un stream applique les opérations sur les données de la source de façon *lazy*. Nos trois opérations de base map / filter / reduce sont donc appliquées en une seule itération sur les éléments de la source d'un stream, plutôt qu'une par une. Cela permet d'optimiser les choses, notamment pour les opérations de réduction qui n'ont pas besoin de s'appliquer à toutes les données d'une source. La contrepartie est qu'il nous faut un mécanisme pour déclencher les traitements déclarés sur un stream.

On voit sur ces quatre idées qu'une collection n'est pas un stream. Notamment une collection porte ses données, ce qui n'est pas compatible.

Construction d'un Stream

Il existe de nombreux patterns pour construire un stream. On peut construire des streams sur des collections, sur des tableaux, sur des sources I/O, sur des itérateurs, etc...

Des nouvelles méthodes ont également été introduites dans les classes existantes du JDK, qui retournent des streams. Par exemple, une méthode `chars()` a été créée sur la classe `String`, qui retourne un stream, construit sur les lettres constitutives de cette chaînes de caractères. Une méthode `lines()` a été créée sur la classe `BufferedReader`, qui retourne un stream construit sur les lignes du fichier sur lequel ce `reader` a été ouvert.

Opérations sur un Stream

Les méthodes map / filter / reduce sont donc définies sur l'interface Stream. Techniquement, les méthodes définies sur cette interface sont de deux types.

- ▶ Les méthodes qui définissent des opérations dites *intermédiaires* sur ce stream. Une opération intermédiaire ne déclenche aucun traitement,

nous sommes toujours dans la phase de déclaration des opérations. Typiquement, map et filter sont des opérations intermédiaires.

- ▶ Les méthodes qui définissent des opérations dites *terminales*. Une opération terminale est une opération qui déclenche les traitements. Déclencher les traitements signifie reprendre toutes les opérations intermédiaires qui ont été déclarées sur ce stream, et les appliquer aux données en une seule fois, avec l'opération terminale. Les opérations de réduction sont des opérations terminales.

On ne peut invoquer qu'une seule opération terminale sur un stream. La déclaration d'une seconde opération terminale se traduit par une exception. Un stream est donc un objet éphémère, utilisable une seule fois. Si l'on veut exécuter deux réductions différentes sur un même jeu de données, il faut donc créer deux streams différents. Cela n'est pas un problème, puisqu'un stream est un objet léger : entre autres, il ne porte pas ses données.

Exécution en parallèle

Les streams ont été créés pour supporter les traitements parallèles simplement. Exécuter les traitements sur un stream en parallèle est simple : il suffit d'invoquer la méthode `parallel()` de ce stream. Encore une fois, écrire cet appel de méthode est très simple, mais ne doit pas être fait à la légère. Un traitement non adapté au parallélisme peut réellement dégrader les performances d'une application, et totalement effondrer un serveur. Il faut garder présent à l'esprit que les traitements parallèles en Java sont construits sur les principes de la programmation concurrente, et que le moindre point de contention conduira à des dégradations de performance.

La réduction

Deux types de réduction

La réduction est probablement l'étape la plus complexe du pattern map / filter / reduce. L'API Stream définit deux types de réduction : la réduction simple et la réduction dite *mutable*, dans un container.

La réduction simple est la réduction classique, telle qu'on l'entend en général. On peut penser aux fonctions d'agrégation du SQL. La somme, le max, le min, la moyenne, sont autant de réductions simples.

La réduction mutable est construite sur le même concept, généralisé. On peut voir une somme comme étant l'accumulation d'entiers dans un container, en fait un entier lui-même, par opération d'addition. Vue sous cet angle, une réduction est donc une succession d'accumulations d'éléments d'un type donné, dans un container pouvant accueillir ces éléments.

Une réduction mutable est donc définie sur deux opérations.

- ▶ La création du container d'accueil. S'il s'agit d'un entier cette création est triviale.
- ▶ L'ajout d'un élément donné à ce container, par l'opération d'accumulation.

La somme entre dans ce cadre sans problème. Mais aussi la concaténation de chaînes de caractères, ou encore l'ajout des éléments dans une liste, ou dans une table de hachage. Dans ce dernier cas l'opération d'accumulation comporte aussi le calcul d'une clé, utilisée pour ranger l'élément dans la table destination.

Il nous manque toutefois un élément pour définir complètement la notion de réduction mutable : le support du parallélisme. Paralléliser une réduction mutable va consister à distribuer les traitements sur plusieurs coeurs. Chaque cœur va calculer un résultat partiel.

Ces résultats partiels devront ensuite être rassemblés en un résultat unique. Dans le cas de la concaténation par exemple, nous avons donc besoin d'une opération qui concatène deux chaînes, calculées sur deux coeurs différents. Dans le cas général, nous avons besoin d'une opération

capable de fusionner deux containers partiellement remplis par chacun des coeurs, dans le cas où la réduction est menée en parallèle.

La réduction mutable a donc besoin d'une troisième opération : la fusion de deux containers partiellement remplis.

Exemples de réductions

Écrivons un premier exemple de réduction mutable simple : la concaténation de chaînes de caractères. Comme on vient de le voir, cette réduction fonctionne avec trois opérations.

- ▶ La création du container d'accumulation : il s'agit ici de la création d'une chaîne de caractères.
- ▶ L'ajout d'un élément au container : cela se traduit par la concaténation de la chaîne résultat avec une chaîne élément.
- ▶ La fusion de deux containers partiellement remplis : cette opération est la même que la précédente.

On peut donc écrire le code suivant.

```
persons.stream()
    .map(p -> p.getName())
    .collect(
        new String(),
        (s1, s2) -> s1.concat(" " ).concat(s2),
        (s1, s2) -> s1.concat(" " ).concat(s2)
    );
```

En quoi ce code consiste-t-il ? Tout d'abord, on construit un stream sur une liste de personnes, sur lequel on déclare une opération qui retourne un stream de chaînes de caractères, composé des noms de ces personnes. On réduit ensuite ce stream dans une chaîne de caractères, dont la construction est le premier argument de l'appel à la méthode `collect()`. Le deuxième argument est l'opération qui accumule un élément (`s2`) dans un container existant (`s1`).

On ajoute un caractère de séparation entre les noms, pour que le résultat soit plus joli. Le dernier argument est la fusion de deux containers : `s1` et `s2`. Il se trouve que les deuxième et troisième opérations sont en fait les mêmes.

Écrivons un deuxième exemple :

l'accumulation de ces noms dans une liste.

```
persons.stream()
    .map(p -> p.getName())
    .collect(
        new ArrayList(),
        (liste, s) -> liste.add(s),
        (liste1, liste2) -> liste1.addAll(liste2)
    );
```

On reconnaît les trois mêmes opérations : la création du container d'accumulation, l'ajout d'un élément à un container existant, et la fusion de deux containers partiellement remplis.

On peut simplifier l'écriture de ce code en utilisant une syntaxe appelée *method reference*.

```
persons.stream()
    .map(Person::getName)
    .collect(
        ArrayList::new,
        ArrayList::add,
        ArrayList::addAll
    );
```

Utilisation de collectors

Dans la pratique, exprimer une réduction mutable de cette façon est assez lourd, et peu adapté aux opérations complexes. Le JDK 8 nous propose donc une classe particulière, `Collectors`, qui encapsule ces trois éléments pour de nombreux cas applicatifs courants.

Dans le cas de la concaténation de chaînes de caractères, on peut en fait écrire simplement.

```
persons.stream()
    .map(Person::getName)
    .collect(
        Collectors.joining(" ", " "))
    ;
```

Dans le cas de l'accumulation dans une liste, le code n'est pas beaucoup plus compliqué.

```
persons.stream()
    .map(Person::getName)
    .collect(
        Collectors.toList())
    ;
```

Cette classe `Collectors` possède de nombreuses méthodes. Le but ici n'est pas de toutes les présenter. Montrons juste un petit aperçu.

Cet appel permet de construire une table de hachage, dans laquelle les clés sont les âges des personnes de notre liste, et les valeurs, la liste des personnes qui ont cet âge.

```
persons.stream()
    .collect(
        Collectors.groupingBy(
            Person::getAge,
            Collectors.toList())
    )
    ;
```

Construire un histogramme des âges devient très simple, plutôt que d'accumuler les personnes par âges dans une liste, il suffit de les compter.

```
persons.stream()
    .collect(
        Collectors.groupingBy(
            Person::getAge,
            Collectors.counting())
    )
    ;
```

Mener ces traitements en Java 7 aurait imposé d'utiliser le pattern Iterator, ce qui aurait mené à l'écriture d'un code plus complexe, moins lisible, et surtout moins performant. La classe `Collectors` propose de nouveaux patterns de traitements de données, qui auraient été implémentés à l'aide du pattern Iterator.

Java 8 : une version pour l'avenir

Pourquoi les lambdas ?

On peut se poser la question du pourquoi de l'introduction des lambdas en Java 8. Une mauvaise réponse consisterait à dire « pour se mettre à niveau par rapport à la concurrence ». Quelle est la concurrence ? Probablement les autres langages objet pour programmer les mêmes applica-

tions que Java : C++ et C# notamment. Effectivement, les expressions lambdas ont été introduites dans ces langages il y a quelques années déjà. De ce point de vue, Java rattraperait son retard par rapport à ces langages.

Je ne pense pas que l'enjeu soit là. Comme James Gosling l'écrivait il y a 17 ans, Java est un langage de cols bleus, conçus pour concevoir des API et des applications et pour résoudre des problèmes réels.

Java n'est pas un projet de recherche, ni un laboratoire d'expérimentations. Introduire une caractéristique (telle que les expressions lambdas) aussi importante dans ce langage, ne peut avoir qu'un seul but : amener de nouveaux patterns qui autorisent la conception de nouvelles API et applications, dans le but de résoudre des problèmes réels et actuels. Ce point a été très clairement affirmé par Brian Goetz, concepteur des expressions lambdas et de l'API Stream.

Une version toujours compatible

L'introduction des lambdas en Java 8 n'est donc pas une fin en elle-même, et n'aurait pas eu de sens sans l'introduction conjointe de l'API Stream. Les expressions lambdas et l'API Stream ouvrent de nouvelles opportunités pour écrire les applications, simples et performantes.

Elles ne remettent pas en cause la compatibilité ascendante : un code Java 8 peut appeler une librairie compilée en Java 7, en lui transmettant des lambdas en tant qu'implémentations d'interfaces fonctionnelles. Le challenge de la compatibilité ascendante a été relevé.

Migrer vers Java 8

Se pose alors la question de la migration des applications. Quel en sera le coût ? Dans la mesure où cette version de Java n'apporte pas d'incompatibilité, le coût d'une migration vers Java 8 ne sera pas plus important qu'une migration vers Java 7.

En revanche, une fois migré vers Java 8, les opportunités d'amélioration du code existant, d'optimisation des performances, de productivité pour l'implémentation des patterns map / filter / reduce notamment, seront incomparablement supérieures.

Utiliser ces opportunités ne sera pas gratuit : le premier coût sera certainement la formation des équipes de développement, le second probablement celui de l'audit. C'est aussi le prix à payer pour suivre les évolutions techniques des langages, et pouvoir exploiter les derniers raffinements qu'ils proposent.

En bref

On a pu effectivement se poser la question de l'avenir de Java, notamment dans la période 2008-2011, période de flottement pour le langage, son écosystème et probablement aussi sa gouvernance.

Cette version 8 remet sans aucun doute Java au premier plan des langages objet. L'introduction des lambdas expressions et de l'API Stream représente une avancée majeure de la plateforme, qui offre des opportunités très larges, y compris pour les versions futures.

Après cette période de flottement, Java est de nouveau sur les rails, de nouveau capable d'intégrer les dernières évolutions de langages modernes, dans l'unique but de pouvoir exploiter au mieux la puissance de traitement des ordinateurs actuels et à venir.

 José Paumard - Formateur & expert Java / Java EE, SOAT

Blog : blog.soat.fr - twitter : [@JosePaumard](https://twitter.com/JosePaumard)

Blog de l'auteur : blog.paumard.org

Maitre de conférence depuis 15 ans, José se passionne également pour l'éducation et la transmission des connaissances. Il tient un blog, Java le soir, source documentaire francophone sur les technologies Java pour des milliers de développeurs dans le monde.

Quid du support de Java 8 au niveau outils ?

Longtemps attendue, souvent repoussée, la version 8 de Java arrive enfin en mars 2014. Si les fonctionnalités apportées par cette nouvelle mouture promettent de changer profondément la façon de travailler des développeurs Java, l'interrogation principale est de savoir s'ils pourront bénéficier des avancées de Java SE 8 tant au niveau des outils qu'au niveau des frameworks de l'écosystème Java. Tentative de réponse.

La chimère Java 8 pointe enfin le bout de son nez, apportant avec elle une avancée considérable : les expressions Lambdas. Cette fonctionnalité qui met à portée de clavier des développeurs Java la programmation fonctionnelle, était attendue de longue date puisque déjà disponible dans la plupart des langages tournant sur le JVM. Oracle aura donc mis du temps à produire cette nouvelle mouture mais le résultat promet déjà beaucoup. Outre les Lambdas expressions, qui viennent modifier la syntaxe du langage, Java SE 8 apporte une API de manipulation du temps enfin digne de ce nom, les annotations sur les types Java, le moteur d'exécution JavaScript Nashorn, ainsi que les profiles pour le JDK. Loin d'être exhaustive, cette liste montre déjà l'intérêt de passer à Java SE 8 au plus tôt. Alors comment faire ? Le téléchargement du JDK 8 n'est évidemment pas suffisant. En effet, le JDK n'est jamais une fin en soi pour un développeur et son travail passe également par un certain nombre d'outils que ce soient des IDE ou des frameworks qui sont légions dans l'écosystème Java.

Du côté des IDE

La première problématique pour le développeur consiste donc à pouvoir bénéficier de Java SE 8 directement au sein de l'IDE. Développé par Oracle, NetBeans supporte naturellement les principales nouveautés de Java 8 depuis un certain temps et sa dernière version officielle la 7.4 permet dès à présent de bénéficier de la syntaxe des Lambdas expressions dans son code. Leur support ne se limite pas à l'auto-complétion puisqu'il concerne également le refactoring. En effet, NetBeans détecte les expressions pouvant être transformées en Lambdas expressions et propose au développeur d'effectuer ce travail. La figure 1 montre ainsi l'utilisation de la méthode statique invokeLater de SwingUtilities qui prend en entrée une instance de Runnable. Ce morceau de code est un candidat parfait pour un refactoring en une Lambda expression et l'on voit que NetBeans propose ce refactoring Fig.1.

L'IDE permet de sélectionner la transformation en Lambda expression ou bien d'ignorer la suggestion si l'on souhaite garder l'ancienne syntaxe. Une fois le refactoring effectué, nous obtenons le code présenté à la Fig.2. Bien entendu, cette aide est également proposée aux développeurs sur des codes plus complexes. Prenons en exemple le code suivant :

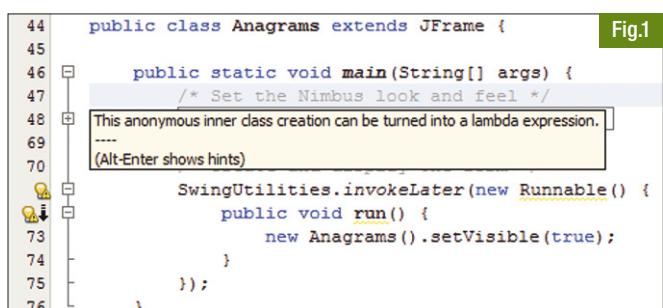


Fig.1

```

44 public class Anagrams extends JFrame {
45
46     public static void main(String[] args) {
47         /* Set the Nimbus look and feel */
48         This anonymous inner class creation can be turned into a lambda expression.
49         -----
50         (Alt-Enter shows hints)
51
52         SwingUtilities.invokeLater(new Runnable() {
53             public void run() {
54                 new Anagrams().setVisible(true);
55             }
56         });
57     }
58 }
```

Figure 1 : Proposition de refactoring de NetBeans

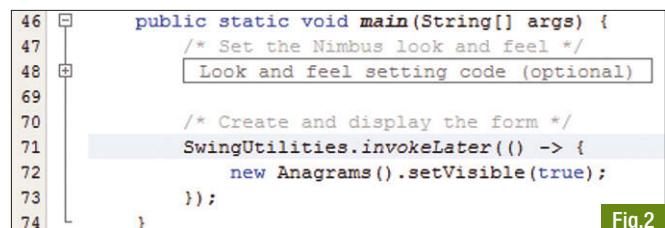


Fig.2

```

46     public static void main(String[] args) {
47         /* Set the Nimbus look and feel */
48         Look and feel setting code (optional)
49
50         /*
51          * Create and display the form
52          */
53         SwingUtilities.invokeLater(() -> {
54             new Anagrams().setVisible(true);
55         });
56     }
57 }
```

Figure 2 : Refactoring en Lambda expression

```
Collections.sort(personList, new Comparator<Person>() {
    public int compare(Person p1, Person p2){
        return p1.firstName.compareTo(p2.firstName);
    }
});
```

Ce code sera ainsi simplifié de la sorte avec NetBeans :

```
Collections.sort(personList, (Person p1, Person p2) -> p1.
    firstName.compareTo(p2.firstName));
```

Les streams de Java SE 8 sont également supportés par NetBeans 7.4 et de fait par NetBeans 8 qui sortira en même temps que la nouvelle mouture du JDK. Ainsi, l'IDE permet un refactoring automatique des boucles for-each classiques sous la forme d'expressions Lambdas au sein de la méthode forEach d'un stream. Prenons ainsi le code ci-dessous :

```
List<String> strings = new ArrayList<>();
for (String string : strings) {
    System.out.println("Content: " + string);
}
```

Ce morceau de code sera refactoré de la sorte :

```
List<String> strings = new ArrayList<>();
strings.stream().forEach((string) -> {
    System.out.println("Content: " + string);
});
```

L'avantage de ces streams est indéniable puisque cela permet d'appliquer facilement des opérations de filtrage ou de chaînage sur des collections Java notamment.

Nouveauté essentielle en vue de la mise en place du système de modules Jigsaw dans Java 9, les profils, ou "profiles", constituent des sous-ensembles d'API du JDK permettant aux applications de définir de quel profil elles ont besoin pour fonctionner. Avec Java SE 8, 3 profils ont été définis : Compact1, Compact2 et Compact3. Ils viennent se rajouter au JDK complet toujours à disposition. Chaque profil est un sous-ensemble du profil supérieur. Ainsi, Compact1 est un sous-ensemble de Compact2, Compact2 de Compact3 et Compact3 du JDK complet. Au niveau de Net-

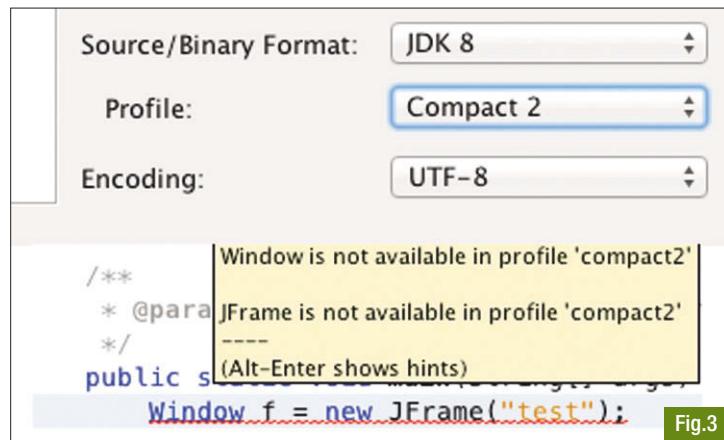


Figure 3 : Sélection d'un profil

Beans, il est possible de sélectionner le profil que l'on souhaite utiliser pour une application donnée. Une fois ce choix effectué, l'IDE nous alerte lorsque l'on recourt à des classes et méthodes d'une API non supportée par le profil choisi. Cette alerte se révélant évidemment bien utile comme le montre la figure 3 où l'on a choisi le profil Compact2 qui n'inclut pas les fonctionnalités de Swing Fig.3.

Outre NetBeans poussé par Oracle, le support de Java 8 avance à bonne vitesse dans Eclipse qui reste l'éditeur de référence des développeurs Java. La roadmap prévisionnelle table sur une mise à disposition générale du support Java 8 d'Eclipse pour le 18 Mars 2014 soit le même jour que la sortie de Java SE 8. Au programme, peu ou prou les mêmes fonctionnalités que celles de NetBeans pour les Lambdas expressions mais également pour la gestion des profils. Enfin, un mot sur l'IDE qui monte, à savoir IntelliJ Idea, de l'éditeur JetBrains. De plus en plus populaire, il supporte lui aussi la majorité des nouveautés de Java SE 8 depuis la sortie des premiers builds du JDK 8 courant 2013 et sera donc fin prêt pour la sortie de la nouvelle version.

Du côté des frameworks

Comme nous avons pu le voir, les développeurs pourront bénéficier des nouveautés de Java 8 au sein de leur IDE dès la sortie de la nouvelle mouture. Au niveau des frameworks, le constat est différent et il faudra donc étudier la question au cas par cas suivant les frameworks utilisés. Prenons l'exemple de Spring qui reste un des frameworks le plus populaires du monde Java. Pivotal, l'éditeur derrière Spring, a profité de l'occasion pour sortir la première version majeure du framework depuis 4 ans. La version 4.0 de Spring tire ainsi profit des Lambdas expressions en interne et offre leur support au sein de son expression language. En sus, l'API Date and Time de Java 8 est intégrée au framework qui se base désormais sur cette

dernière, permettant aux développeurs de tirer pleinement parti des fonctionnalités de celle-ci. Les améliorations apportées au package java.util.concurrent sont également de la partie.

Solution ORM de référence de la sphère Java, Hibernate ne tire pas encore parti des avancées de Java 8 tant au niveau des Lambdas expressions en interne que de l'API Date and Time qui n'a pas encore été prise en charge et qui ne sera donc pas disponible pour le développeur lors de la sortie officielle de Java SE 8. Les développeurs utilisant conjointement Spring et Hibernate devront donc s'adapter.

Bibliothèque utilitaire devenue indispensable dans la trousse à outils du développeur Java, Google Guava va voir une grande partie de son API devenir inutile puisque faisant doublon avec des nouveautés de Java 8. Pour ces cas précis, il sera préférable de basculer vers Java 8. On pense ainsi aux classes liées à la programmation fonctionnelle telles que Function, Supplier ou Predicate, à celles liées à la concurrence avec les nouvelles méthodes ajoutées à CompletableFuture ou encore à des méthodes et classes utilitaires avec StringJoiner par exemple. Néanmoins, l'existence de Guava garde tout son sens pour les développeurs travaillant dans des environnements limités à des versions antérieures du JDK ou utilisant un JDK open source. On pense ainsi aux développements Android qui bénéficient grandement de l'existence de Guava ou encore à GWT. La situation au niveau des frameworks est clairement plus compliquée quant au support de Java SE 8 dès sa sortie. Les 3 exemples remontés ici le montrent avec un support déjà prêt pour l'un ou pas encore programmé pour l'autre. Aux développeurs de rester attentif aux évolutions des frameworks et outils qu'ils utilisent et éventuellement de basculer vers d'autres solutions si l'utilisation des nouvelles fonctionnalités de Java 8 apporte un réel plus.

Conclusion

Les innombrables reports de la sortie de Java 8 couplés à la mise à disposition systématique des builds du JDK tout au long de son développement auront permis aux éditeurs des IDE de prendre en compte au plus tôt les nouveautés de cette nouvelle mouture et d'être prêts dès sa sortie. Ceci étant un pré-requis quant à une prise en compte massive du JDK 8. Au niveau des frameworks, la situation est plus contrastée et le développeur devra voir au cas par cas ce qu'il en est pour bénéficier des nouveautés de Java SE 8. Bien entendu, grâce à la compatibilité ascendante de Java, les frameworks fonctionnant sous Java 7 fonctionneront normalement sous Java 8. Pas de panique donc mais des arbitrages à effectuer quant à la nécessité de changer un framework ne supportant pas encore Java 8.

 Sylvain Saurel – Ingénieur d'Etudes Java / JEE
sylvain.saurel@gmail.com

Eclipse & Java 8

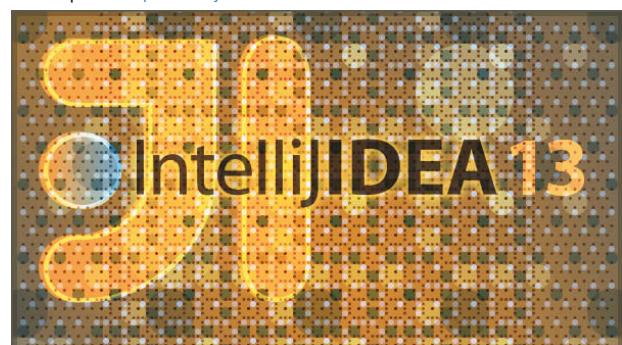
Une page dédiée au support de Java 8 dans l'environnement Eclipse est disponible. Elle décrit le support dans JDT Core de la version 8 du langage. Tout devrait être prêt pour le 18 mars 2014.

lien : https://wiki.eclipse.org/JDT_Core/Java8

Platforms	JRE	JDK
Windows (WinXP not supported)*	32-bit	exe (md5) 29.65 MB exe (md5) 32.53 MB
	64-bit	exe (md5) 109.61 MB exe (md5) 113.06 MB
Mac OS X	64-bit	dmg(md5) 56.59 MB
Linux	32-bit	tar.gz (md5) 55.43 MB
	64-bit	tar.gz (md5) 54.38 MB
Linux ARMv6/7 VFP, HardFP ABI	32-bit	tar.gz(md5) 83.40 MB
Solaris SPARC	64-bit	tar.gz (md5) 50.52 MB
	64-bit	tar.gz (md5) 47.96 MB
Solaris	64-bit	tar.gz (md5) 93.15 MB

IntelliJ IDEA

Les dernières versions de l'IDE IntelliJ supportent Java 8. Pour en savoir plus : <http://www.jetbrains.com/idea/whatsnew/>



Systèmes embarqués, objets connectés : pourquoi pas vous ?

Depuis quelques mois, on parle d'objets connectés. Un peu avant, c'était l'Internet des objets. Mais les deux se confondent peu ou prou. Les systèmes embarqués, ou enfouis, constituent un autre pan de l'industrie, souvent méconnu des développeurs et pourtant, toujours actifs.

On pourrait définir l'embarqué comme des systèmes installés dans des environnements, comme dans l'avionique (aviation), tous les terminaux et appareils ayant de l'électronique (grand public ou non), la robotique, les smartphones (qui est une forme d'embarqué), la domotique, les appareils électroménager, la voiture, etc. Bref, la notion d'embarqué est très large.

Deux types d'embarqué peuvent être considérés : les environnements contraints et non contraints. Par exemple, le militaire, la centrale nucléaire, l'avionique, le médical sont des environnements très contraints où l'électronique et l'informatique (matériel et logiciel) doivent être parfaitement déterminés, dans le sens, où le fonctionnement est prévisible et certain. Aucune panne n'est permise, ni dysfonctionnement. Et ce sont des systèmes qui évoluent très lentement et doivent fonctionner 10, 15, 20 ans. Ce sont des développés très spécifiques. D'autres systèmes sont moins contraints et les évolutions y sont plus rapides et constantes : la voiture intelligente, les appareils grand public, etc. Les développements et les systèmes sont plus souples, avec parfois des fonctionnements discontinus.

Les objets connectés ou l'Internet des objets, est le buzz du moment. Mais finalement, nous sommes dans la mouvance initiée depuis plusieurs années du Machine to Machine (M2M), c'est à dire, la possibilité que plusieurs appareils puissent interagir,

communiquer ensembles, sans ordinateur. Depuis quelques mois, les objets connectés se multiplient, plus ou moins réels. L'espérance sur ce marché est colossale. On nous annonce des milliards d'objets d'ici 2020. Si le marché, encore naissant, s'emballe, de nombreuses questions devront être résolues rapidement : la sécurité des objets et des données, standard des protocoles et les technologies, proposer des modèles de

développement clairs. Que se soit l'embarqué ou l'objet connecté, il existe de nombreuses plateformes matérielles et de développements. Le C++ demeure le langage de référence. Un langage de type Java ou C# ne sera pas adapté en environnement contraint. Le C revient aussi en force.

La rédaction.

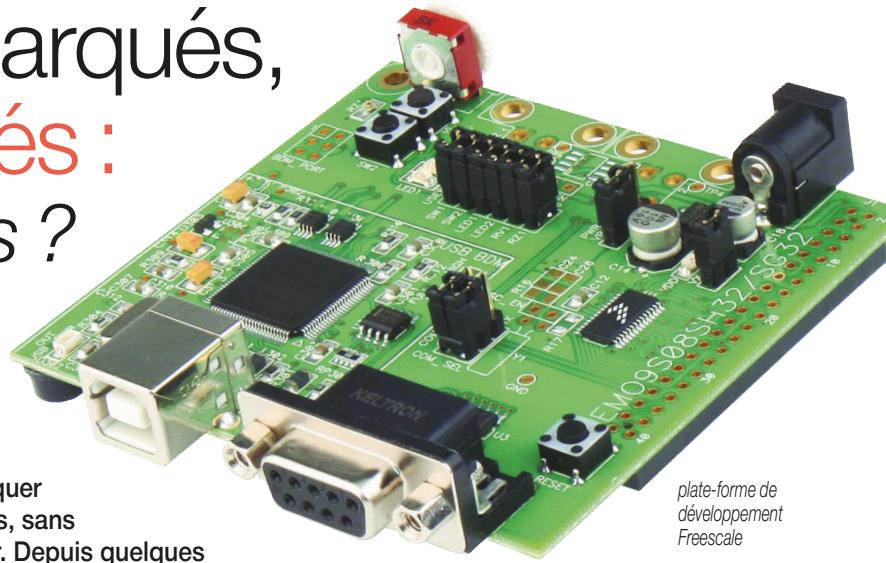


plate-forme de développement Freescale

Internet des objets et sécurité :

Il faut considérer le nombre croissant de tous ces « objets connectés » qui vont communiquer entre eux et avec le monde extérieur ; qu'il s'agisse de votre ordinateur, de votre voiture, de votre téléviseur voire même à moyen terme des vêtements que vous portez (concept de « wearable » technologies). Les entreprises et la société, d'une manière générale vont devoir se poser des questions essentielles liées au big data, à l'analytique, à la sécurisation des données mais également à l'identité. L'idée étant bien entendu que l'évolution vers cette ultra-connectivité soit bénéfique à tous – individus comme entreprises – mais maîtrisée pour éviter toute dérive.

Que se passera-t-il quand toutes ces données collectées, personnelles et professionnelles, ainsi que les appareils qui les produisent, seront victimes d'attaques par des hackers spécialisés capables de récupérer ces montagnes de données et de les analyser ? Que se passera-t-il si ces dernières tombent entre de mauvaises mains, ou pire, si certains hackers arrivent à prendre le contrôle à distance sur ces appareils connectés ? Si nous prenons l'exemple de la santé, secteur qui voit l'émergence de plus en plus d'appareils connectés visant à suivre l'état

de certains patients voire d'alerter les services médicaux en cas d'urgence, un hacking de ce type d'appareil connecté pourrait avoir des conséquences fatales en terme de vies humaines.

Au-delà du simple hacking des données, il est question de la sécurité au sens large du terme. Pas simplement de sécurité associée à une donnée ou un appareil connecté en particulier mais appliquée à la société dans son ensemble. C'est une problématique de sécurité, certes, mais au delà, d'éthique autour de l'Internet des objets. Dès demain, les entreprises seront non seulement capables de savoir où vous vous trouvez 24h/24 et 7 jours/7, mais également de connaître vos marques préférées, les horaires auxquels vous êtes chez vous et sur votre lieu de travail, quand vous avez l'habitude d'aller faire vos courses, le type de vêtement que vous aimez porter, le type de données qui vous intéressent à propos de vous-même et de votre santé... Cela peut aller jusqu'à la manière dont vous conduisez, à quels jeux vous jouez, quelles émissions de télévision vous regardez, vos différentes habitudes de consommation et d'attitudes selon que vous vous trouvez sur votre lieu de travail ou chez

Les tendances 2013 de l'embarqué

EmbeddedMarket forecaster publie chaque année une étude très complète sur l'embarqué et particulièrement sur les développeurs. L'étude 2013 porte sur 663 développeurs (développeurs, ingénieurs matériels, développeurs systèmes, architectes, ingénieurs en firmware...). Presque la moitié vit en Amérique du Nord. L'industrie "automation et de contrôles" pèse pour 18 %, en bas du marché : la robotique et les terminaux de jeux.

Il est intéressant à noter que le développeur embarqué s'appuie sur des composants de développements bien connus : bibliothèques standards, orienté objet, composants, design patterns, architecture de services. Le logiciel est largement présent : +40 % des développeurs travaillent sur le logiciel. Le secteur télécom pèse environ 17 % (logiciel) et 14 % (matériel). C'est le premier secteur d'activité de l'embarqué actuellement.

Sur les standards logiciels et matériels, on ne sera pas surpris de voir que Linux (= Linux temps réel) est le premier système embarqué (1 tiers du marché). Eclipse est l'IDE de référence. Mais cela ne signifie pas que Java soit le langage n°1 de l'embarqué. Il pèse finalement très peu (à peine 7 %

selon l'étude). Cependant, on constate qu'Android ne cesse de progresser. Autre surprise, iOS progresse mais très lentement (10 %). Son arrivée dans les voitures pourra booster son écosystème tout en sachant que seul Apple fait des terminaux iOS. Les systèmes Windows représente environ 20 % des systèmes utilisés mais avec une tendance à la baisse. Les autres systèmes temps réels de référence (Wind River, Lynux, QNX) stagnent ou baissent.

Côté IDE, comme nous l'avons dit, Eclipse est très largement utilisé (38 %) mais les IDE Microsoft arrive à 40 %. Nous retrouvons plusieurs IDE de constructeurs ou d'éditeurs de systèmes : Wind River, Borland, CodeWarrior, QNX, etc. Plusieurs se basent aussi sur Eclipse. Notons que l'adressage 32 bits est très largement utilisé dans l'embarqué, le multicoeur aussi (que ce soit sur processeurs Intel ou ARM). Le développeur embarqué avec une suite d'outils spécialisés : IDE, compilateur assembleur, debugger, oscilloscope, bibliothèques spécialisées, analyseur logique, outils de tests, outils de configuration. Les tests sont primordiaux dans l'embarqué ainsi que l'optimisation.

Pour un développeur, plusieurs critères sont considérés pour choisir un système embarqué :

- le prix est un élément crucial : certaines plates-formes ont un coût élevé et les suites de développement alourdissent parfois la facture.
- Le support des processeurs et des plates-formes embarqués est un argument crucial
- La disponibilité du code source du système
- Les performances temps réel : cela va dépendre du contexte du projet

Si nous regardons les langages les plus utilisés, le top 4 n'étonne pas :

- 1 : C
- 2 : C++
- 3 : langage assembleur
- 4 : Java

Il y a quelques années, des spécialistes et analystes prédisaient que Java allait s'imposer dans l'embarqué...

quels enjeux pour demain ?

vous, etc. Nous avons vu ces précédents mois plusieurs entreprises se faire pirater leurs données, être victimes de failles de sécurité, mettant en péril des données professionnelles mais aussi personnelles (exemple récent des données personnelles chez Orange, des comptes piratés de Dropbox, ou encore de SnapChat). L'humain étant de plus en plus au cœur des technologies, les conséquences d'une mauvaise gestion et d'une mauvaise sécurisation des données auront des conséquences de plus en plus dramatiques en cas de hacking.

Parmi les solutions proposées, peu sont encore au point, ou tout du moins prêtes à faire face à une gestion aussi étendue des problématiques de sécurité. Nous avons bien souvent d'un côté des sociétés proposant de protéger les appareils hardware et les softwares (qu'ils soient professionnels ou personnels), d'un autre celles proposant de protéger les données et les applications (qu'elles soient personnelles ou professionnelles), mais peu fournissent des solutions de sécurité de bout en bout efficaces et intégrant tous les nouveaux paramètres et habitudes des utilisateurs finaux : utilisation d'appareils informatiques mobiles en tous genres, utilisation d'appa-

reils et vêtements (« wearables ») connectés, mix entre les données personnelles et professionnelles, etc. Le nombre d'appareils connectés et les outils permettant de générer et stocker des quantités impressionnantes de données ne cessent d'augmenter, quelles solutions doivent être adoptées par les entreprises et par les individus ? Quel est le rôle des experts en sécurité dans ce nouveau contexte ? Comment responsabiliser les individus eux-mêmes, car ils seront seuls à maîtriser réellement certaines données et paramètres ? Comment impliquer les institutions gouvernementales dans ce débat qui va devenir une réelle problématique de société ?

Pour accéder à mes objets connectés je vais utiliser mon Smartphone et l'application fournie par l'éditeur de l'objet en question... et si en parallèle mon entreprise me permet d'utiliser ce même équipement pour accéder aux données de l'Entreprise, cela peut constituer une faille de sécurité pour cette dernière. De plus, si j'utilise par exemple mon compte Facebook pour accéder à cette application, et qu'en plus cette identité Facebook (« connect with Facebook ») me permet également d'accéder à un portail d'Entreprise, le fameux « BOYI », la surface d'exposition

devient très importante pour le responsable de la sécurité de l'entreprise. Il est donc important de contrôler non pas uniquement ce qui est installé sur les appareils mobiles privés des employés, mais l'étendue de son accès sur les données sensibles de l'entreprise ainsi que des identifiants autorisés.

Des solutions de type proxy qui permettent aux mobiles d'accéder au réseau de l'entreprise ou des passerelles vers des tierces d'authentification comme Facebook sont donc le minimum à mettre en place pour commencer à se protéger des intrus.

Depuis quelques années les solutions de gestion des identités et gestion des accès (WEB SSO) sont des bonnes pratiques dans les Entreprises, qui devraient être aujourd'hui renforcées par des solutions de Gouvernance des Identités, de Gouvernance des Accès, mais également de fédération des identités, de reporting sur l'activité des utilisateurs et plus généralement la mise en place de solutions de supervision de la sécurité, de type SIEM.

 Jean-Philippe Sanchez,
Consultant sécurité chez NetIQ

Linux Real-Time : entrez dans la nouvelle ère de la conception de systèmes embarqués

Dans le passé et malgré des essais, Linux n'était pas adapté aux exigences des applications temps réel et hautes performances embarquées. Pourtant, une récente étude consacrée aux systèmes embarqués a clairement montré que Linux représente actuellement plus de 50% des systèmes d'exploitation utilisés. Linux Real-Time était attendu !



Fig.3

LocalGrid Technologies a pu rapidement porter son application existante et apporter une valeur ajoutée considérable en tirant parti de l'écosystème Linux accessible avec le nouveau matériel cRIO-9068. Cette nouvelle technologie commerciale a permis à la société de développer une solution répondant au cahier des charges pour fournir une application de réseau électrique intelligent à Toronto Hydro.

Peu de temps après son introduction, le système d'exploitation Linux s'est largement répandu dans le domaine de la conception de systèmes embarqués. En «open source», supporté par une communauté, complémentaire et accessible, cet OS compte en effet plus d'un attrait. L'adoption de Linux dans la conception de systèmes embarqués n'a rien de surprenant, car il est incroyablement personnalisable, offre un support pour un large éventail d'architectures de processeurs (par exemple ARM, x86, etc.), et présente une communauté aisément accessible qui regorge d'applications et de code. Une récente étude consacrée aux systèmes embarqués a clairement montré que Linux représente actuellement plus de 50% des systèmes d'exploitation embarqués utilisés, bien qu'aucune distribution ne représente à elle seule plus de 20% du marché.

En dépit des nombreux avantages qu'il offre aux concepteurs de systèmes embarqués, Linux, dans le passé, n'était pas adapté aux exigences des applications temps réel et hautes performances. Dès le début, les développeurs de logiciels embarqués ont étudié toute une variété de techniques permettant d'améliorer les performances de Linux pour

une utilisation temps réel. Bien que les concepteurs de systèmes embarqués aient proposé un grand nombre de solutions, aucune approche pour les performances temps réel avec Linux n'a remporté de véritable adhésion. Au fil des années, une technique a cependant éveillé un intérêt particulier : il s'agit de l'approche hybride qui consiste à utiliser la technologie d'hyperviseur afin de combiner Linux avec un système d'exploitation temps réel (OS RT) dédié, dans le cadre d'applications embarquées avec des exigences temps réel.

Linux : référence du temps réel

Le marché de l'embarqué a été témoin d'un certain nombre de solutions basées sur un hyperviseur, qui permettaient aux utilisateurs d'avoir recours à Linux accompagné d'un système d'exploitation temps réel plus adapté. Parmi ces solutions, citons INTEGRITY Multivisor de Green Hills et CODEZERO Embedded Hypervisor de B Labs. Malheureusement, elles semblent avoir manqué leur objectif en étant incapables de satisfaire les raisons essentielles qui poussaient les concepteurs de systèmes embarqués à utili-

ser Linux. Bien au contraire, les solutions à base d'hyperviseur disponibles augmentent la complexité du système, le rendent moins convivial, et requièrent davantage de spécialisation de la part des équipes de conception de systèmes embarqués.

Ce n'est que depuis peu que les concepteurs de systèmes embarqués disposent d'une alternative. À mesure que l'utilisation de Linux sur le marché de l'embarqué a poursuivi sa progression, certains développeurs ont participé à son essor en ajoutant des caractéristiques permettant d'améliorer le déterminisme du noyau Linux lui-même. Avec l'arrivée récente de l'ensemble de correctifs PREEMPT_RT, la communauté Linux a unifié son approche pour obtenir des performances temps réel avec ce système d'exploitation. Ce qui fait la particularité de cette solution est qu'un seul et même système d'exploitation Linux temps réel permet aux développeurs de tirer avantage de performances temps réel tout en alliant l'accessibilité, la facilité d'utilisation et la communauté disponibles avec un système d'exploitation d'usage général. Un système d'exploitation basé sur Linux, construit avec l'ensemble de correctifs PREEMPT_RT, présente un jitter du même

ordre que celui des autres OS RT dédiés tels que VxWorks de Wind River et Phar Lap ETS de IntervalZeros. De même, en ce qui concerne la charge du processeur en fonction de la vitesse de boucle, un système d'exploitation temps réel Linux basé sur PREEMPT_RT offre des performances équivalentes voire supérieures à celles d'un autre OS RT dédié. Cette amélioration des performances au niveau des applications - mesurée par un gain plus faible dans l'utilisation du microprocesseur à des vitesses de boucles plus élevées - résulte en grande partie de la capacité de planification disponible sur un OS RT basé sur Linux. Contrairement à la plupart des systèmes d'exploitation temps réel dédiés, un OS RT basé sur Linux peut proposer à la fois un ordonnanceur temps réel pour les tâches critiques et un ordonnanceur complètement équitable (CFS) plus efficace pour toutes les tâches qui ne sont pas temps réel.

De plus en plus de personnes s'accordent à dire que l'ensemble de correctifs PREEMPT_RT permet à un système d'exploitation temps réel basé sur Linux d'offrir des performances comparables aux OS RT existants. Le fait de pouvoir atteindre ce niveau de performances est une excellente chose, aussi bien pour les utilisateurs de Linux qui souhaitent gérer davantage d'applications temps réel et hautes performances que pour ceux qui ont actuellement recours à un OS RT et qui souhaitent bénéficier de la stabilité, de la facilité d'utilisation et de la communauté qu'offre Linux. En effet, la grande collection d'IP et d'outils proposée par une importante communauté d'utilisateurs expérimentés permet aux concepteurs de gagner en autonomie pour augmenter les fonctionnalités de leur solution pour un coût de développement moindre. Il est possible de tirer rapidement parti des nombreuses améliorations apportées à l'ensemble du système d'exploitation ;



Fig.1 Le contrôleur conçu par logiciel cRIO-9068 de National Instruments est une solution «sur étagère» qui offre aux utilisateurs un système d'exploitation temps réel basé sur Linux dans une enceinte durcie.

par exemple, la couche réseau éprouvée et stable proposée par Linux supporte pleinement les cartes d'interface réseau (NIC) dual-DHCP et offre également un meilleur support pour les protocoles de communication IPv6 et SNMP.

L'importance du PREEMPT_RT

L'ensemble de correctifs PREEMPT_RT offre un certain nombre de facteurs de différenciation décisifs pour les concepteurs de systèmes embarqués. Il étend la souplesse offerte par Linux à des architectures matérielles diverses et variées et la conception unique du système d'exploitation facilite l'allocation des ressources. Par conséquent, les utilisateurs bénéficient d'une plus grande facilité d'utilisation, d'une capacité de programmation plus intuitive, ainsi que d'une gestion de système et de capacités de configuration optimisées. Au-delà de la flexibilité matérielle, un système d'exploitation temps réel Linux permet à des ingénieurs logiciels de se former et d'intégrer plus facilement des équipes de conception de systèmes embarqués en raison de la large adoption de Linux et du grand nombre d'utilisateurs et ressources Linux disponibles. En s'appuyant sur

un OS RT basé sur Linux à la fois familier et plus accessible, les équipes de conception de systèmes embarqués peuvent relever plus efficacement les défis avec des chaînes d'outils logiciels solides.

À mesure que l'ensemble de correctifs PREEMPT_RT s'affirme comme l'approche standard pour accomplir des fonctionnalités temps réel avec Linux, un nombre croissant de solutions «sur étagère» offrant un OS RT basé sur Linux commence à apparaître sur le marché. Le nouveau contrôleur conçu par logiciel cRIO-9068, qui offre un OS RT basé sur Linux, ouvert et bien maintenu, dans une enceinte durcie, en est un exemple. Ce contrôleur cRIO-9068 supporte le développement d'applications avec le logiciel de conception de systèmes NI LabVIEW et avec des outils C/C++, ce qui permet aux utilisateurs de réutiliser pleinement leur code C/C++ et de tirer parti de la communauté Linux tout en profitant de l'interface de développement intuitive offerte par LabVIEW **Fig.1**.

L'embarqué plus accessible

La conception de systèmes embarqués est désormais plus facilement accessible aux équipes de conception logicielle possédant une expertise matérielle limitée grâce aux solutions «sur étagère» basées sur un système d'exploitation temps réel Linux. Pour des équipes de conception plus expérimentées, ces solutions optimisent la mise sur le marché dans la mesure où les ressources peuvent être consacrées aux tâches spécifiques à l'application plutôt qu'aux tâches ordinaires comme concevoir et implanter des connecteurs périphériques sur un processeur ou écrire un middleware pour intégrer des composants périphériques. De plus, pour les équipes soucieuses de s'appuyer sur des composants logiciels en open-source,



Fig.2 La nouvelle plate-forme de surveillance de l'intégrité des structures de S.E.A. Datentechnik GmbH a pu voir le jour grâce au système d'exploitation temps réel Linux disponible sur le nouveau contrôleur conçu par logiciel cRIO-9068.

un OS RT Linux intégré à une solution «sur étagère», par un fournisseur attaché au maintien et à la mise à jour du système d'exploitation, est une garantie suffisante pour justifier un investissement. Les tout premiers utilisateurs sont déjà témoins de ces avantages.

“Le système d’exploitation temps réel basé sur Linux disponible sur le matériel cRIO-9068 change réellement la donne,” souligne Wolfram Koerver, directeur exécutif de S.E.A. Datentechnik GmbH. «Avec le contrôleur cRIO-9068, nous pouvons ajouter d’importantes fonctions et technologies à notre plate-forme d’enregistrement de données, ce qui était impossible auparavant. Les efforts de développement pour mettre en œuvre les fonctions et les caractéristiques requises ont été réduits d’au moins 80% par rapport aux solutions basées sur VxWorks.” Fig.2.

Dans la mesure où les systèmes d'exploitation temps réel Linux deviennent de plus en plus courants dans les solutions «sur étagère», les utilisateurs doivent à présent rechercher les architectures matérielles sous-jacentes qui offrent le plus de souplesse. Par exemple, le contrôleur cRIO-9068 intègre l'architecture LabVIEW RIO (E/S reconfigurables), qui permet aux utilisateurs

de gérer les exigences les plus variées. La polyvalence offerte par l'architecture LabVIEW RIO provient d'un processeur fonctionnant avec un FPGA avec accès aux E/S.

Les utilisateurs peuvent rapidement mettre sur le marché des solutions bien différenciées avec du matériel «sur étagère» flexible offrant un OS RT basé sur Linux. Cette nouvelle technologie commerciale leur permet de réutiliser le code existant et de profiter de performances accrues, comme le montre LocalGrid Technologies. *“Lorsque nous avons reçu le contrôleur cRIO-9068, il ne nous a fallu que 24 heures pour exécuter notre programme d’application LabVIEW existant sans rencontrer le moindre problème,”* souligne Bob Leigh, président et PDG de LocalGrid Technologies. *“Nous avons été impressionnés par la simplicité de la transition logicielle entre les systèmes CompactRIO et par l’amélioration remarquable des performances apportées par ce nouveau contrôleur.”*

Saisir les opportunités

Alors que le noyau Linux traditionnel continue d'améliorer des caractéristiques telles que l'isolation des coeurs et la gestion de mémoire, l'ensemble de correctifs PREEMPT_RT va

être utile à une solution encore plus attrayante pour les concepteurs de systèmes embarqués. S'adressant aussi bien au développeur travaillant avec un système d'exploitation temps réel, un système d'exploitation temps réel basé sur Linux offre un grand nombre d'opportunités – en particulier lorsqu'il est disponible sur une plate-forme durable, robuste et flexible «sur étagère». Avec cette approche, les concepteurs peuvent se consacrer à des tâches comme la sécurité ou l'intégration de systèmes, tout en se reposant sur les fournisseurs pour mettre à jour et maintenir le système d'exploitation, ce qui leur permet de rationaliser encore davantage leur conception. Les solutions «sur étagère» qui offrent un OS RT Linux vont permettre aux concepteurs de garder une longueur d'avance sur le marché, accélérant plus que jamais la mise sur le marché de conceptions innovantes et marquant ainsi l'entrée dans une nouvelle ère de la conception de systèmes embarqués.

● Sanjay Challa

Product Manager for Embedded Software chez National Instruments

Programmez! want you !

06-29-09 © nycto



*Rejoignez
la communauté
Programmez!.*

*Vous avez
un code génial,
des bonnes
pratiques,
des conseils
à partager ?*

*Envoyez-nous
vos meilleurs textes
et codes.*

redaction@programmez.com



Questions – réponses avec Thierry Joubert

Thierry JOUBERT est cofondateur et Directeur Technique de THEORIS, spécialisé dans le domaine du développement logiciel avec une expertise confirmée dans les Systèmes Embarqués. Depuis plusieurs années il consacre en plus une partie de son temps à la formation dans les écoles d'ingénieurs.

Qu'est-ce que l'on entend par embarqué ?

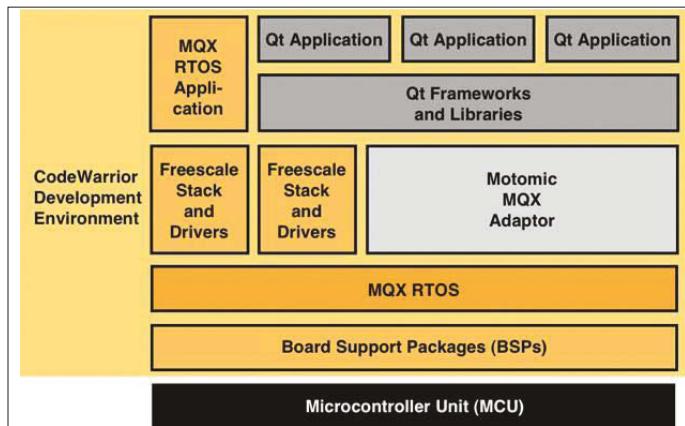
Comment le définir ?

Le terme "Embarqué" vient d'une mauvaise traduction en Français; le terme officiel est "enfoui", mais il est mal perçu par les gens qui communiquent sur le domaine. "Embarqué" sous-entend que le système est en mouvement, ce qui n'est vrai que pour une petite partie des systèmes concernés - beaucoup d'entre eux sont fixes (bâtiment intelligent, etc.). On

peut dire que le domaine de l'embarqué correspond à tous les équipements qui contiennent au moins un ordinateur pour réaliser tout ou partie de leurs fonctions. Aujourd'hui, c'est très vaste. A l'ère de la mobilité, on pourrait retourner la question en se demandant ce qui n'est pas aujourd'hui du domaine de l'embarqué? On touche ici une autre caractéristique du domaine, à savoir que l'utilisateur ne doit pas percevoir l'équipement comme un ordinateur. C'est vrai quand vous validez à l'entrée du métro ou que vous voyez la porte d'un magasin s'ouvrir devant vous ou encore quand vous louez un vélib. C'était vrai quand vous passiez un coup de fil dans les années 90, c'est moins vrai avec les smartphones d'aujourd'hui qui sont multifonctions et que l'on n'inclut plus dans le domaine embarqué. Par contre, quand vous passez un coup de fil, beaucoup de systèmes embarqués travaillent pour vous derrière les antennes. Une autre définition que j'aime bien pour les systèmes embarqués c'est "on le démarre en janvier 2014 pour l'arrêter en décembre 2024".

Qui dit embarqué dit souvent temps réel, système temps réel. Est-ce toujours le cas ? Et comment comprendre la notion de temps réel ?

Le temps réel, que l'on pourrait reformuler "déterminisme temporel", est une problématique délicate dans le cadre d'un système informatique car le temps n'est pas un des paramètres retenus par Von Neumann dans son automate (en un mot un ordinateur calcule très bien mais il ne sait pas être à l'heure). La notion de temps réel a longtemps caractérisé le domaine de l'embarqué sous la pression des vendeurs d'OS qui



l'utilisaient comme un argument de vente, dans la réalité les systèmes temps réel ne représentent qu'une fraction du domaine. Une borne Vélib' ou un automate bancaire ne sont pas dans le domaine des systèmes temps réel, un portique Ecotaxe qui dialogue avec les véhicules du trafic a de fortes contraintes temps réel.

C, C++, assembleur, c'est le trio des langages pour l'embarqué. Pourquoi ces langages demeurent-ils si importants ? Pour les performances ? La prédictibilité du code ? La robustesse ?

L'utilisation encore majoritaire de ce trio dans le domaine de l'embarqué est principalement imposée par la prédominance des problématiques hardware à toutes les étapes de la conception d'un système embarqué (disponibilité, RAM, performances de calcul, Entrées/Sorties, etc.). Pour configurer les registres d'un équipement d'entrée/sortie ou gérer une interruption, aucun niveau d'abstraction n'est souhaité et l'assembleur reste l'outil optimal, le C est tellement proche de l'assembleur qu'on peut en faire le même usage. Le langage C++ est utile pour disposer de meilleurs compilateurs mais dans beaucoup de projets on évite d'aller trop loin dans les classes virtuelles, la généricité ou le polymorphisme - c'est un vaste débat mais ces techniques peuvent vite provoquer une consommation mémoire mal contrôlée. Ce point ne doit pas masquer un usage de plus en plus important des langages Java ou C# qui permettent d'obtenir une bien meilleure productivité si on doit par exemple développer une IHM graphique; le travail doit toutefois être réalisé par un professionnel conscient des limites du

système. J'utilise souvent la métaphore suivante: un développeur classique sait poser un airbus sur l'aéroport de Roissy, dans l'embarqué il faut savoir se poser sur le pont d'un porte-avions (même l'avion doit être adapté!).

Google Glass, les objets connectés, les vêtements intelligents, la voiture connectée, l'embarqué explose et nous promet encore plus. Quelle est votre vision du marché et des tendances ?

C'est une équation d'abord économique. Comme le prix des processeurs chute, ils sont amenés à réaliser de plus en plus de fonctions, et plus il y en aura plus leur prix va baisser. Comme le processeur est programmable, c'est un composant idéal pour celui qui conçoit un objet car il introduit une "zone d'ajustement" par reprogrammation. Par conséquent on va en avoir partout, la principale limite c'est que le processeur est une pompe à énergie mais c'est aussi un domaine en constante progression.

Quid des métiers de l'embarqué : développeurs, ingénieurs matériels, etc. Quel métier et quelle carrière peut-on espérer faire dans l'embarqué ? Quel profil faut-il ?

C'est un domaine de spécialistes. La conception des circuits est un domaine immense et stratégique, mais il nécessite aujourd'hui des investissements colossaux qui ne sont plus à notre portée. Il nous reste la conception des cartes, même si beaucoup de la production est délocalisée. Le domaine dans lequel des carrières s'ouvrent est bien évidemment celui du logiciel où nous pouvons exceller comme on le fait dans un domaine connexe qui est celui du jeu vidéo (langage C majoritaire, ça vous dit quelque chose?). On peut aujourd'hui se procurer de minuscules ordinateurs pour quelques dizaines d'Euros, c'est l'occasion de se familiariser avec le monde de l'embarqué où on commence par comprendre comment démarrer la machine. Cette approche est très enrichissante dans la mesure où il me paraît essentiel de développer un profil hard/soft pour espérer acquérir la maîtrise de toutes les couches du système.

Exemple : Le Bloc d'Orange

L'opérateur Orange n'est pas étranger à l'univers des objets connectés. My Plug avait été une des premières excursions dans ce monde communiquant. Tout récemment, Orange a sorti le Bloc. Cet objet est un projecteur audio-vidéo, conçu avec Awox. Il fonctionne en Wifi et Bluetooth. Il permet de diffuser des chaînes (réservées aux abonnés Internet Orange TV), de la musique avec Deezer, ainsi que les contenus personnels. En complément, une application dédiée est disponible sur Android et iOS.

« Nous avons une politique d'ou-

verture, notamment avec les API, pour ces objets » précise Jean-Pierre Combe (directeur digital home management Orange). Le Bloc consomme diverses API : Deezer, Cloud d'Orange. Pour le moment aucune API du Bloc ne sont disponibles. « nous allons tester en interne des API mais nous n'avons pas décidé de les ouvrir ou non. Nous prenons la mesure du marché » poursuit Jean-Pierre Combe. L'un des intérêts des objets connectés est la possibilité de les intégrer entre eux et d'utiliser leurs services, via des API.



Le Bloc utilise plusieurs standards utilisés notamment en domotique (réseau et découverte de réseau) : UPnP et DLNA. Pour l'utiliser, il faut simplifier l'installation et le raccordement de l'objet au réseau et à son environnement. L'authentification se fait par des certificats qui authentifient chaque élément. Côté technique, le Bloc fonction-

ne sur Android 4.0 et sur un processeur ARM. Le modèle de développement est classique : C et Java.

LA SÉCURITÉ N'EST PAS UNE PRÉOCCUPATION DES CONSTRUCTEURS !

Le constat est sévère mais réaliste. « Chaque constructeur crée son objet sans que se soit forcément son métier » avertit Edouard Jeanson (directeur technique – security global line – Sogeti). Seb lance une cocotte-minute connectée, mais est-ce son métier de faire de l'informatique, de l'embarqué ? Non. Le monde connecté est fragmenté et morcelé. « Quelle protection de la vie privée ? Sait-on qui fait quoi ? » poursuit Edouard Jeanson. C'est une

question essentielle : car tout objet connecté est susceptible de récolter des informations, des données sur nos habitudes quotidiennes sans que l'on sache exactement comment les données sont collectées et comment elles sont stockées et utilisées. « Ces données ont une valeur financière » analyse notre expert. Il ne faut jamais oublier que ces données collectées vont permettre de personnaliser les offres (par exemple : savoir combien je consomme de

yaourt ou combien de fois j'utilise le four ou le lave-vaisselle par semaine et les fabricants peuvent ensuite m'envoyer des offres personnalisées), sans que l'utilisateur n'ait conscience de cela. « Il y a un problème de prise de conscience : il faut sensibiliser. » poursuit Edouard Jeanson. Les anglo-saxons se posent moins ce genre de question. La notion de vie privée n'est pas identique dans tous les pays.

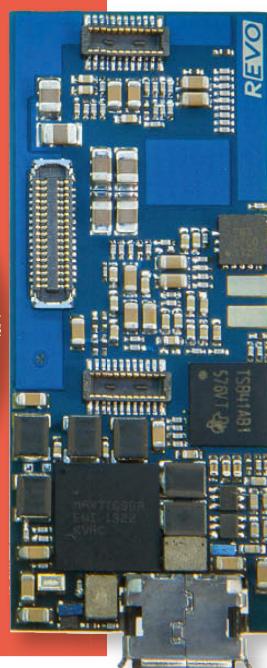
Par exemple, des enceintes connectées : pour mettre à jour les firmwares et logiciels, l'enceinte se connecte au site du constructeur, et ce dernier sait quelle enceinte se connecte

et est enregistrée. Si l'utilisateur refuse l'enregistrement, l'objet n'est pas à jour... Les attaques et les risques de sécurité sont identiques à ceux que l'on connaît déjà sur Internet. Il faut installer un firewall, définir des standards et utiliser les mêmes protocoles. Tant que chaque constructeur fera ce qu'il veut et à sa façon, le problème restera entier. Aujourd'hui, avec le marché qui explose, les constructeurs sont plus dans une logique de sortir rapidement un objet connecté que de se préoccuper de la sécurité.

Le Wearable computing : WaRP

Freescale s'intéresse beaucoup à un marché prometteur : l'informatique embarquée dans les vêtements et les accessoires (lunettes, montres, etc.). Cette plateforme matérielle est fournie par Revolution Robotics et la partie logicielle par Kynetics. Freescale apporte la partie ARM (32 bits) et certains capteurs. La plateforme est très compacte. La communication est une partie cruciale pour ce genre d'embarqué. WaRP supporte le WiFi, le Bluetooth, l'USB. Le système par défaut est Android 4.3. Le modèle de développement est donc assez standard. La volonté est de proposer une plateforme ouverte et des outils de développement ouverts. WaRP est sans licence d'utilisateur. Le matériel sera disponible durant le 2e trimestre de l'année, le prix estimé est de 149 \$.

Site : <http://www.warpboard.org>



Un Hadoop sinon rien



Big Data = Hadoop, Hadoop = Big Data.

Nous lisons (trop) souvent cette affirmation. Or, la réalité est loin de cette simplification. Oui, Hadoop sert à faire de le Big Data, mais non, on ne peut pas résumer le Big Data à Hadoop, et vice versa. Si déjà comprendre ce qu'est le Big Data ne va pas de soi, expliquer Hadoop demande encore plus d'explications. Car on confond beaucoup de notions.

Comme Didier Kirszenberg (HP France) nous l'a précisé : les entreprises pensent que ce n'est pas cher.

Le plus incroyable est que parfois, les responsables et DSI disent : il faut faire de l'Hadoop même s'ils ne savent pas pourquoi ! Autre exemple, traiter les données sur notre éléphant pour « seulement » 100 Go de données ! C'est comme si vous dites que je veux faire du SQL sur des bases NoSQL. Léger problème en perspective...

Hadoop est fait pour fonctionner sur des clusters massivement distribués sur des centaines de milliers de nœuds. Les données sont déposées sur un système de fichier adapté à ce fonctionnement distribué : HDFS. Hadoop est un écosystème en soi avec une multitude de projets annexes qui évoluent très

rapidement. On peut voir Hadoop comme un super ETL (extraire, transformer et charger des données).

Il est possible d'utiliser Hadoop pour transformer et analyser des données. Le projet Mahout est une collection d'algorithmes pour analyser et traiter les données. Mahout est capable d'apprendre de lui-même (notion de « machine learning ») et de s'adapter selon les données. Il permet de filtrer, de catégorisation. Mais attention, comme prévient Didier Kirszenberg, il faut des mathématiciens pour manipuler des outils comme Mahout. Car il faut définir les modèles mathématiques...

Le développeur, avant d'attaquer le moindre projet Hadoop, doit absolument connaître les points suivants :

- Savoir de quoi on parle,
- Connaître les projets, les outils Hadoop et de la communauté,
- Qu'est-ce que l'on veut faire ? Avant tout développement Hadoop, il faut impérativement avoir un objectif. Faire de l'Hadoop pour de l'Hadoop est le meilleur moyen pour échouer,
- Le quoi faire va conditionner le stockage, le formatage des données,
- Ne pas négliger l'impact de Java sur les performances,

D'autre part, il y a Hadoop et les distributions Hadoop packagées. Sauf si vous êtes un pur et dur, vous n'allez pas déployer Apache Hadoop mais une distribution packagée. Le choix ne manque pas : Cloudera, Hortonworks, Intel.

Hadoop est un projet qui évolue rapidement. Par exemple, au départ, la notion de scheduler n'existant pas, car il n'y en avait pas besoin. Mais les éditeurs et la communauté ont fait évoluer les choses.

La rédaction.





Hadoop : distribution, choisir, déployer

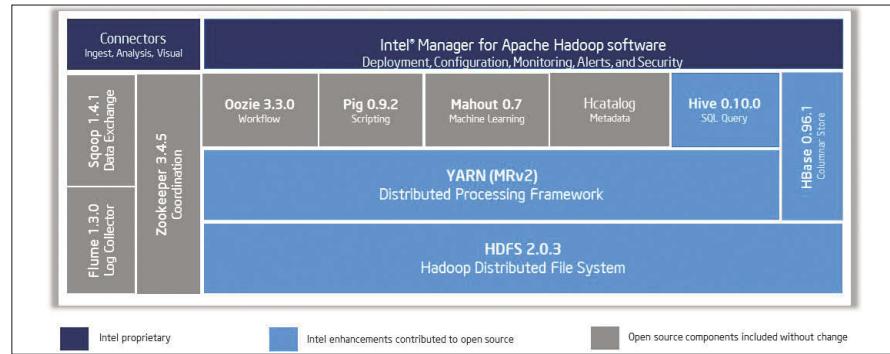
Hadoop recouvre des réalités très différentes. Parfois on se trompe même sur sa nature. Nicolas Maillard revient pour nous sur le fameux framework. Nicolas est Scalability Tech Lead chez fifty-five.

Comment choisir sa distribution Hadoop ?

Par simplification Hadoop est devenu synonyme de Big Data; ceci constitue un raccourci acceptable en ce sens qu'Hadoop est tout à la fois le porte étandard de ces nouvelles solutions et un composant central de son architecture. Pour autant, si l'on choisit Hadoop il faut garder à l'esprit le contexte métier et faire le choix de solutions en conséquence. Considérons que la stack

Hadoop se présente habituellement sous trois formes, par ordre croissant d'abstraction:

- ▷ Hadoop vanilla: le projet Hadoop open source lui-même, ainsi que l'ensemble de ses projets annexes, open source eux-mêmes.
- ▷ Hadoop Distribution: ensemble cohérent et packagé des technologies Hadoop prêtes à l'emploi.
- ▷ Big Data suite: outil packagé s'appuyant sur Hadoop pour répondre à un ensemble de besoins. Hadoop vanilla est un choix complexe qui force à réimplémenter l'ensemble des couches de monitoring de déploiement et à savoir harmonieusement marier les versions des différents projets. Plusieurs entreprises se sont constituées autour de l'idée de packager et d'élargir la base applicative Hadoop. Ces solutions permettent de mutualiser le premier



besoin d'une stack Hadoop, c'est à dire de pouvoir la déployer facilement et à volonté. Notons 3 principaux acteurs : Cloudera, HortonWorks et MapR. Les trois approches se ressemblent en ce sens qu'elles visent à simplifier le choix des versions, le déploiement, le monitoring et donc in fine à pouvoir se pencher sur le problème à résoudre au plus vite. Par ailleurs les trois distributions veulent répondre au problème de support commercial et, entre autres, de formations. La stack Hadoop peut constituer une nouvelle approche de la gestion des données ou data lake comme souvent présentée. Néanmoins, à moins d'avoir un nouveau problème sur lequel appliquer ces technologies il faut souvent intégrer ces solutions dans une architecture existante et repenser ses data workflow.

Souvent les "big data suites" peuvent être une façon de se doter d'un outil clef en main pour répondre au besoin avec complexité architecturale minimale.

La "big data suite" rend souvent la distribution hadoop transparente et offre en plus outils et workflow nécessaires à une logique particulière de tâche. Cette approche permet de simplifier le use-case en s'appuyant sur un best practice éprouvé et packagé. Ces solutions viennent souvent avec un ensemble de white paper sur des cas d'utilisation qui permettent de prendre un peu de recul et d'intégrer au mieux cette « boîte noire » dans l'architecture principale. Nous nous appuyons sur une connaissance fine des différentes distributions pour choisir l'architecture la plus adéquate en fonction des besoins identifiés chez les clients.

Les distributions

Cloudera : Le distributeur le plus ancien et le plus représenté.

Cloudera, représenté par Doug Cutting, l'un des papas de Hadoop, propose une distribution packagée et une suite d'outils impressionnantes pour administrer. Cloudera propose aujourd'hui l'outil d'installation le plus simple et complet du marché. L'une de ses

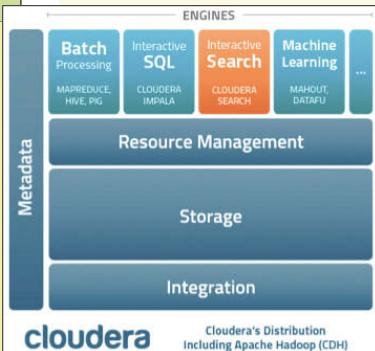
forces consiste à proposer une distribution en partie remaniée par leurs soins, ce qui permet d'upgrader ses versions de façon quasi transparente. Notons que cet avantage vient au prix d'être dépendant de leur cycle de release; intégrer des versions plus récentes de tel ou tel sous projet est très compliqué. Enfin, Cloudera bénéficie d'une communauté et mailing-list très active, ce qui est un point important lors des nombreux problèmes de configuration.

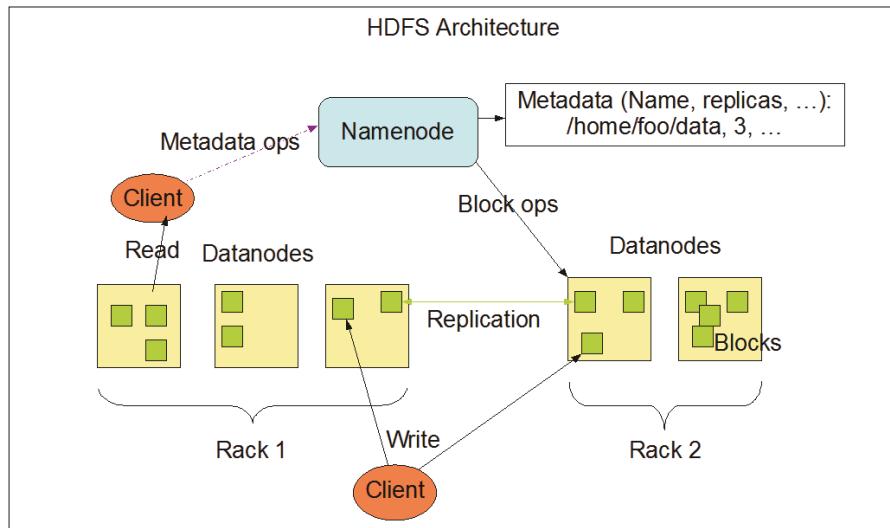
Hortonworks : Ce distributeur est une spinoff de Yahoo et a fait le choix de rester au plus près des projets open source, tous leurs efforts sont retranscrits dans les

projets open source correspondants. Cette philosophie leur permet de sortir des distributions au plus près des avancées de chacun des projets et permet donc de bénéficier des nouveautés de ces jeunes systèmes au plus vite. Par ailleurs cette plateforme permet une relative liberté quant aux choix des versions et des projets grâce à sa proximité avec chacun des projets initiaux. Leur support et mailing-list peuvent se confondre avec ceux de la communauté open source et reste très réactif.

MapR : Ce distributeur a fait le choix de proposer un service « entreprise ready ». Mapr est une distribution non open-source qui fait le choix de

redévelopper certains modules pour proposer un meilleur support et de meilleures performances. Cette approche leur permet de mieux contrôler leur écosystème, mais nécessite donc de connaître leur plateforme en particulier, puisque plus éloignée des projets open source initiaux. Ce choix plus orienté entreprise rend leur support et mailing-list moins réactifs, dans sa version ouverte ; les entreprises sont moins enclines à faire des retours d'expériences ou passer du temps à aider sur les forums d'entraide. La version actuelle du manage MapR est moins riche et distribuée que celle de Cloudera mais il faut noter que beaucoup d'efforts sont faits de ce côté.





Comment met-on en œuvre Hadoop ?

Hadoop est un changement de paradigme où le data product peut être reconstruit depuis la source à chaque instant et ne constitue qu'une vue particulière des données pour une utilisation ou un produit. Mettre en œuvre Hadoop consiste encore une fois à définir la fin. Un data product se définit en termes de finalité de support commercial et de service.

- ▷ Service : volumétrie, temps de réponse, sécurité
- ▷ Finalité : type de question et calcul mis en œuvre
- ▷ Commercial : disponibilité, évolution.

D'un point de vue commercial, refondre une architecture existante est coûteux, et peu compatible avec une offre stable et facilement évolutive. D'un autre point de vue, intégrer une solution aussi large que Hadoop permet d'être libre de son utilisation et de développer des produits annexes et spécifiques. En fonction de son core business le coût architectural initial d'une distribution Hadoop et de la montée en compétence des équipes techniques permettra de proposer des produits innovants et de maîtriser son destin technique.

Hadoop est une solution large et versatile avec un écosystème très riche plus ou moins intégré. Plus particulièrement la sortie d'Hadoop 2 en octobre permet de concevoir et d'intégrer des solutions et workflows bien plus complexes que le MapReduce initial. Il faut donc faire le point sur les structures de données et le type de calculateur les plus mûrs et leur intégration à la plateforme. De ce point de vue, une solution big data peut-être mieux adaptée au besoin qu'une construction « maison ». La stack Hadoop fut initialement conçue pour un but précis de batch processing. Le besoin était de faire tourner des process sur de larges volumes de données,

tâches qui pouvaient prendre plusieurs heures. Comme un exemple vaut de longues explications, Hadoop fut souvent adopté au travers de la lentille des bases de données sans pour autant l'être. Aujourd'hui, au travers de projets annexes, Hadoop cherche à se doter de couches de sécurité telles que Sentry ou de temps de réponse au travers d'Impala ou Drill. Comme avec les deux axes précédents Hadoop est une technologie jeune et impose de se doter des outils annexes pour finaliser un produit. La couche est d'autant plus critique qu'elle est proche du client et il faut souvent choisir des compléments à Hadoop pour remplir ce rôle.

Quel est le modèle d'API/Framework ?

Avec la sortie de Hadoop 2 et sa couche de ressources Yarn, l'ensemble du framework prend en maturité. Après une course aux technologies et aux initiatives plus ou moins individuelles, Hadoop prend la place du centre de la donnée et offre les moyens de s'y connecter. En termes de processus, la nouvelle couche Yarn se pose en gestionnaire des ressources de l'ensemble des machines, proposant à d'autres programmes d'en définir l'utilisation. Hier Hadoop était restreint à un paradigme MapReduce qui pouvait être étendu ou répété au sens de Hive ou Pig. Aujourd'hui Yarn permet de brancher des paradigmes différents tels que l'approche graph au travers du projet Giraph, d'une approche plus en mémoire grâce au projet Spark ou encore d'un modèle BSP au travers d'Hama.

Parallèlement à cette ouverture, les différents distributeurs ont fait des efforts pour packager les meilleures interactions possibles et simplifier l'accès au cluster Hadoop. Cloudera

propose son CDK (Cloudera Development Kit) un ensemble d'API java pour créer des datasets organisés, interagir avec le cluster, ordonner des tâches.

Notons aussi des initiatives comme Pail ou toutes les interfaces Thrift ou Rest que propose chacune des briques de Hadoop pour permettre de les piloter à distance.

Point sur les modules à mettre en œuvre également

Comme expliqué Hadoop est un changement de paradigme. On passe d'une solution unique que sont les bases relationnelles à un système distribué de gestion et traitement de l'information. La composante distribuée le rend plus puissant et évolutif tout en rendant son écosystème plus complexe. Pour tirer toute la souplesse et la richesse de Hadoop, il faut lui adjoindre les abstractions et outils nécessaires pour le besoin de l'entreprise. De la même façon que nous avions des bases de données relationnelles ou documentaires ou de compte utilisateur avec Hadoop, nous allons avoir des programmes d'analyse de l'information qui diffèrent selon le besoin, mais tous hébergés par Hadoop.

Hadoop au sens le plus strict, c'est :

- Hdfs : le système de fichier réparti
- Yarn : le système de gestion des ressources
- MapReduce : le paradigme de traitement historique

Zookeeper : un arbitre pour l'ensemble des systèmes.

L'écosystème standard

- Hive : une couche SQL sur MapReduce
- Pig : un langage procédural sur MapReduce
- Oozie : un ordonnanceur de tâche Hadoop
- Hue : une interface graphique pour tous ces outils
- Hbase | cassandra : une base de données

L'écosystème étendu demain

- Impala : SQL temps réel sur Hadoop
- Spark/Shark : calcul en mémoire
- Storm : couche de calcul temps réel

L'écosystème Analyse

- R /Python (scikit learn) pour les analyses machine learning

L'écosystème Service

- Solr : moteur de recherche temps réel

- Elasticsearch : moteur de recherche et analyse temps réel

- Greenplum : base de données et d'analyse temps réel
- Druid.io : base de données temporelle

Distribution	Manager	Communauté	Offre gratuite	Projet Phare	Offre gratuite	Entreprise	Écosystème
Cloudera	1	1	oui	Impala	oui	2	1
Horton	2	2	oui	Hive	oui	3	2
MapR	3	3	non	Nfs Hdfs	limite	1	3

Développer un job Map/Reduce pour Hadoop

Hadoop Map/Reduce est un framework de calcul distribué basé sur le paradigme fonctionnel. Les données vont être manipulées par un ensemble de fonctions ne partageant (en théorie) aucun état mutable.

Ainsi, avant de commencer par les spécificités d'Hadoop, il est utile de comprendre ce que signifient les concepts de map et de reduce en général. Ce sont des abstractions que vous utilisez probablement tous les jours, même si vous n'avez pas nécessairement l'habitude de les identifier ni de les nommer.

Map

Soit deux types A et B ; si on possède une fonction capable de transformer A en B, alors, avec map, on est capable de transformer une collection de A en une collection de B. Plus concrètement, si on possède une classe *Person* avec une fonction *getFirstname()* retournant son prénom, * en java8, on pourrait écrire :

```
persons.stream().map(person -> person.getFirstName())
```

* et dans le langage de tous les jours :

```
Collection<String> firstnames = new ArrayList<String>(persons.size());
for (Person person : persons) {
    firstnames.add(person.getFirstname());
}
```

L'opération map permet d'abstraire l'implémentation réelle d'un ensemble de transformations sur une collection qui ne partagent pas d'état mutable. Dans les cas simples, sur un faible volume de données, une boucle séquentielle est sûrement la meilleure approche. Dès que le volume va grossir, paralléliser en local les transformations peuvent améliorer le temps de réponse mais il faut alors gérer la synchronisation. Distribuer le traitement sur plusieurs machines permet de traiter des volumes encore plus gros mais il faut alors gérer les cas de pannes. C'est ce que fait Hadoop map/reduce. Étant donné que l'implémentation n'est pas à la charge du client, il n'a plus à se soucier du nombre de fois ni de l'endroit où les transformations ont eu lieu. Seul le résultat compte.

Reduce

Reduce est une opération se réalisant aussi sur une collection, mais permettant le partage d'un état local au détriment de la parallélisation des opérations. Faire une somme est sûrement l'exemple le plus concret. Supposons que nos précédentes *Person* soient des employés, on peut vouloir calculer le salaire total en se basant sur les rémunérations personnelles *getSalary()*. Le partage d'état local est nécessaire pour ce besoin, on ne peut donc pas uniquement se baser sur map. L'implémentation interne de reduce est, dans le cadre d'Hadoop, peu intéressante, et entièrement à la charge du développeur. Mais il est important de voir qu'un reduce sur une collection peut être envisagé comme une opération de transformation sur un élément (la collection). Lorsque plusieurs collections doivent être transformées en parallèle, il est possible de faire un map de reduces. Cela permet de profiter une fois de plus de la gestion automatique de la parallélisation et de la distribution des tâches par le framework.

Hadoop Map/Reduce gère des couples clef/valeur

Hadoop Map/Reduce est une implémentation spécifique de map et reduce pour faire du traitement distribué en se basant sur un système de fichier distribué (HDFS). Conceptuellement, une contrainte a été ajoutée :

le framework va travailler sur des collections de clefs/valeurs.

Soit 6 types : A, B, C, D, E, F. Map va transformer une collection de (A,B) en une collection de (C,D). Toutes les différentes valeurs D associées à une même clef C vont être regroupées. Et reduce va transformer une collection de (C, collection<D>) en une collection de (E,F).

Chaque reduce garantit de traiter en même temps toutes les valeurs associées à une même clef C ; de recevoir les clefs C dans leur ordre naturel, mais pas nécessairement de recevoir toutes les clefs puisque le travail est partagé. Les types sont arbitraires, car c'est au développeur de spécifier ce qu'il souhaite. (A,B) est lié à la manière dont le fichier est lu. (C,D) est un format d'échange intermédiaire entre map et reduce. Et (E,F) est le format final, pouvant être relu par un autre job.

Le nombre de map est déterminé par le framework en fonction des fichiers lus.

Par défaut, il y a au minimum une tâche de map par fichier et au maximum une tâche de map par block composant chaque fichier. Étant donné que pour chaque tâche correspondant à un split des données une jvm est lancée, il est pertinent de faire attention à la fois aux nombres de fichiers et à la taille des blocs afin de minimiser le surcoût lié à la parallélisation, la distribution et la gestion des erreurs.

C'est une problématique qui peut nécessiter un travail commun de la part du développeur et de l'administrateur. Le nombre de reduce est quant à lui explicité manuellement et vaut par défaut 1.

Le nombre de reduce est lié au volume de (C,D) mais également à leur distribution. Par défaut, le reducer est choisi en fonction de la valeur du hash de la clef. Si la fonction de hashage est non uniforme, ou si une clef possède beaucoup plus de valeurs qu'une autre clef, la charge entre les reducers peut être mal équilibrée.

C'est encore une fois une problématique devops où chaque partie a un rôle critique à jouer. Un job se compose nécessairement d'une étape de map (qui peut être une identité). Le reducer est cependant optionnel.

Certaines problématiques peuvent être traitées exclusivement en parallèle mais pas toutes.

Qu'est-ce qu'un job ?

Un job est essentiellement constitué de deux fichiers. Le premier est un jar (archive java) qui contient le code devant être exécuté, propre à Hadoop. Et le second est un fichier xml contenant sous forme de clef/valeur la configuration du job. Cela dit, ce fichier est très rarement renseigné tel quel. Les apis vont vous permettre de le configurer de façon transparente. Mais il est important de comprendre que c'est par son biais que la configuration (statique) peut être passée aux différentes tâches map et reduce fonctionnant sur des machines différentes.

Enfin, des API

Hadoop Map/Reduce a un historique. En ce qui concerne la version 1, la version stable actuelle, il existe deux API permettant de faire sensiblement la même chose.

org.apache.hadoop.mapred est l'ancienne API, qui fut un moment dépréciée. Étant donné que beaucoup de documentations mentionnent celle-ci, elle sera utilisée durant cet article. Il faut cependant savoir que *org.apache.hadoop.mapreduce* est une alternative, plus récente. Dans la mesure où les concepts sont assimilés, la migration d'une API vers une autre n'est pas une difficulté en soi.

Définir un job

Traditionnellement, le premier programme affiche "Hello World". Dans le monde d'Hadoop, il s'agit en fait d'un wordcount : calculer le nombre d'occurrences de chaque mot dans un texte.

La partie map va lire les fichiers ligne à ligne et émettre des couples (mot,1). La partie reduce recevra ensuite l'ensemble des occurrences associées à chaque mot (celui-ci étant la clef) et il suffira de faire la somme. L'utilisation du combiner sera expliquée en fin d'article.

Pour la définition du job, étendre la classe *Configured* et implémenter l'interface *Tool* est une bonne pratique à adopter. Cela permet de réutiliser le parsing d'options génériques à Hadoop, et il est ainsi possible de préciser au lancement du job n'importe quelle clef/valeur devant se retrouver au final dans le fichier xml de configuration. Par exemple, le nombre de reducer peut être augmenté à 4 en ajoutant simplement "-D mapred.reduce.tasks=4". Il faut bien noter l'espace entre le D et la propriété. Il ne s'agit pas de propriétés système propres à la jvm, ces dernières n'ont aucun intérêt ici, car le traitement est effectué sur d'autres jvms. *ToolRunner* va parser ces arguments génériques, renseigner la *Configuration* fournie puis appeler la fonction run avec les arguments non interprétés. La configuration récupérée par *getConf()* possédera bien la configuration additionnelle présente sur la ligne de commande.

Le reste de la configuration est essentiellement sans surprise.

```
public class WordCount extends Configured implements Tool {
    public static final String REMOVE_KEY = «wordcount.remove.
    regex»;

    public static void main(String[] args) throws Exception {
        int res = ToolRunner.run(new Configuration(), new Word
        Count(), args);
        System.exit(res);
    }

    public int run(String[] args) throws Exception {
        // préciser une classe permet au client de savoir
        // quel est le jar qui doit être envoyé au cluster
        JobConf conf = new JobConf(getConf(), WordCount.class);
        conf.setJobName(«wordcount»);

        // on précise les types des données
        // sauf pour l'entrée qui est à la charge de l'input format
        conf.setMapOutputKeyClass(Text.class);
        conf.setMapOutputValueClass(IntWritable.class);
        conf.setOutputKeyClass(Text.class);
        conf.setOutputValueClass(IntWritable.class);

        // on définit le mapper et le reducer
        conf.setMapperClass(Map.class);
        conf.setCombinerClass(Reduce.class);
        conf.setReducerClass(Reduce.class);

        // et comment les données sont lues et écrites
        conf.setInputFormat(TextInputFormat.class);
        conf.setOutputFormat(TextOutputFormat.class);

        // enfin on parse les arguments pour un argument custom
        List<String> other_args = new ArrayList<String>();
        for (int i = 0; i < args.length; ++i) {
            if («-remove».equals(args[i])) {
```

```
                conf.set(REMOVE_KEY, args[++i]);
            } else {
                other_args.add(args[i]);
            }
        }
        // et les fichiers à lire et la destination des résultats
        TextInputFormat.setInputPaths(conf, new Path(other_args
        .get(0)));
        FileOutputFormat.setOutputPath(conf, new Path(other_args
        .get(1)));
        // puis on lance le job
        JobClient.runJob(conf);
        return 0;
    }
}
```

Le format d'échange : Writable

Les données échangées doivent utiliser la sérialisation d'Hadoop et être des *Writable*. Les types primitifs et les String sont gérés par défaut, comme le montre *IntWritable* et *Text* dans cet exemple. S'il est nécessaire d'avoir des types plus complexes, il suffit alors de créer une classe implémentant *Writable* et encapsulant d'autres *Writable* déjà disponibles. Les clefs devront implémenter *WritableComparable*.

Le mapper et le reducer

L'API pour le mapper et le reducer sont très proches. Dans les deux cas, il faut implémenter une interface Mapper ou Reducer. Etendre la classe abstraite *MapReduceBase* permet d'éviter de devoir implémenter à vide les deux fonctions du cycle de vie : *configure* et *close*.

Le mapper reçoit en value chaque ligne des fichiers un par un. Elles sont découpées et chaque mot est émis un par un. Sa méthode *configure* permet de récupérer le paramètre précédemment renseigné, afin de ne pas compter certains mots qui correspondent à une expression régulière. La réutilisation des instances *Writable* permet de soulager la gestion de la mémoire. Et enfin un système de compteurs permet en fin de job d'obtenir un aperçu de son fonctionnement.

```
public class Map extends MapReduceBase implements Mapper
<LongWritable, Text, Text, IntWritable> {

    static enum Counters {
        REMOVED
    }

    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    private Pattern pattern;

    @Override
    public void configure(JobConf job) {
        String remove = job.get(WordCount.REMOVE_KEY);
        if (remove != null) {
            pattern = Pattern.compile(remove);
        }
    }

    @Override
    public void map(LongWritable key, Text value, OutputCollector
<Text, IntWritable> output, Reporter reporter) throws IOException {
```

```
String line = value.toString().toLowerCase();

 StringTokenizer tokenizer = new StringTokenizer(line);
 while (tokenizer.hasMoreTokens()) {
    word.set(tokenizer.nextToken());
    if (pattern != null && pattern.matcher(line).matches()) {
        reporter.incrCounter(Counters.REMOVED, 1);
    } else {
        output.collect(word, one);
    }
}
}
```

Le reducer s'explique de lui-même.

```
public class Reduce extends MapReduceBase implements Reducer<Text, IntWritable, Text, IntWritable> {
    @Override
    public void reduce(Text key, Iterator<IntWritable> values,
                      OutputCollector<Text, IntWritable> output, Reporter reporter)
        throws IOException {
        int sum = 0;
        while (values.hasNext()) {
            sum += values.next().get();
        }
        output.collect(key, new IntWritable(sum));
    }
}
```

Et le combiner?

Le combiner est une optimisation de performance. Il utilise la même API que le reducer, consomme les données en sortie du mapper, et doit produire en sortie le même format. Le framework est responsable de son appel éventuel et potentiellement multiple. Le reducer n'est pas conscient de son existence.

Ici, il permet de regrouper des instances d'un même mot. Transformant

$\{(mot,1),(mot,1),(mot,1)\}$ en $\{(mot,3)\}$, il peut ainsi réduire considérablement le volume de données à transférer depuis la tâche de map vers la tâche de reduce.

Il ne reste plus qu'à lancer

Le lancement ressemble à l'exécution standard d'un jar.

```
hadoop jar wordcount.jar fr.programmez.WordCount /a/lire  
/a/ecrire -remove le
```

La fin du début

Hadoop est un sujet vaste, et cet article ne prétend pas pouvoir tout présenter. Il serait possible d'écrire des livres entiers sur ce sujet et cela a été fait. "Hadoop: The Definitive Guide" par Tom White est la référence à consulter.

Il est cependant possible de lister quelques sujets à creuser pour ceux souhaitant aller plus loin. Il faut mentionner l'existence d'un cache distribué pour les fichiers de références couramment utilisés. Le pattern "secondary sort" est intéressant pour mieux comprendre le fonctionnement interne de la plateforme. Maven est un outil permettant de simplifier la gestion de dépendances et MRunit une librairie pour écrire plus facilement vos tests.

Tout cela dit, même si le potentiel de l'API présentée est énorme, il faut aussi se poser la question du niveau d'abstraction. D'autres outils comme Pig, Hive ou Cascading permettent de simplifier considérablement le développement. Mais la présentation de chaque outil séparément pourrait faire l'objet d'un article en soi.

Pour conclure, tout projet, même informatique, est avant tout une aventure humaine. Hadoop possède des mailings lists et un channel irc. Des groupes utilisateurs existent aussi tout autour de la planète, la France n'étant pas une exception (hugfrance.net).



HDInsight : Hadoop en tant que service dans Windows Azure

Hadoop fonctionnant sur Java, le portage sur Windows devait être possible; c'est le défi qu'ont relevé les équipes d'Hortonworks et de Microsoft. Hadoop fonctionne sur Windows Azure sous la forme du service HDInsight.

Hadoop est né en Java sur Unix. Windows Azure peut faire tourner des machines virtuelles Linux, et on peut installer un cluster Hadoop sur Linux (on trouve sur Internet des articles qui montrent comment faire). Ne serait-il pas plus simple d'avoir juste à créer un cluster en indiquant uniquement quelques paramètres tels que le nombre de nœuds que l'on souhaite ? Comme Hadoop tourne en Java, le portage sur Windows devait être possible et c'est le défi qu'ont relevé les équipes d'Hortonworks et de Microsoft. Il existe donc maintenant une distribution HDP (Hortonworks Data Platform) sur Linux et sur Windows. HDP sur Windows est intégrée dans Azure sous la forme du service HDInsight dont il est question dans cet article.

Architecture

Un cluster HDInsight est composé d'un nœud principal, de taille XL (8 coeurs, 14 Go de RAM), et de n nœuds de travail de taille L (4 coeurs, 7 Go de RAM). Les nœuds tournent en Windows Server.

Deux systèmes de fichiers sont proposés : HDFS et WASB qu'on atteint par hdfs://(...) ou wasb://(...). Les deux implémentent les interfaces HDFS et s'utilisent de la même façon. Par exemple, hadoop fs -ls hdfs://(...) ou hadoop fs -ls wasb://(...).

Le système de fichiers HDFS (Hadoop Distributed File System) est l'implémentation classique du système de fichiers distribué Hadoop. Les blocs de fichiers sont stockés sur les disques locaux des nœuds de travail. Le système de fichiers WASB (Windows Azure Storage Blob) est le système de stockage de blobs Windows Azure avec l'interface HDFS, de la sorte d'un job Map/Reduce, les commandes classiques en ligne de commande, Pig, Hive etc. peuvent travailler directement sur les blobs.

Pour accéder aux blobs en SSL, on peut utiliser wasbs:// au lieu de wasb://. Le système de fichiers HDFS présente l'avantage de mettre en œuvre le principe de distribution du code près des blocs de données.

Le système de fichiers WASB présente l'avantage de survivre à la



disparition du cluster, ce qui est particulièrement utile dans le Cloud où l'on veut bénéficier d'un paiement à l'usage. Cela permet en effet d'alimenter les données alors que le cluster n'est pas encore démarré, de récupérer les résultats de traitement après l'arrêt du cluster, et de conserver les données à un moindre coût. Enfin, sur un même compte de stockage, on peut créer plusieurs clusters HDInsight si nécessaire (cela peut permettre par exemple de gérer la priorité des différents travaux sur des données communes). En terme de performances, en fonction du type de traitement que l'on exécute (ex : jointures), WASB peut se révéler plus rapide qu'HDFS. Tout cela fait que WASB est le système de fichiers par défaut sur HDInsight.

Création d'un cluster

La façon la plus simple de créer un cluster est d'aller dans le portail Windows Azure (<https://manage.windowsazure.com>) et de choisir Nouveau, Services de données, HDInsight, Création rapide Fig.1.

Le cluster s'appuie sur un compte de stockage Windows Azure.

Si vous n'avez pas de compte de stockage dans la région où vous créez le cluster (ex : Europe du Nord) en dehors d'un groupe d'affinité, vous pouvez préalablement en créer un depuis le portail de la façon suivante : Nouveau, Services de données, Stockage, Création rapide. Là, vous saisissez un nom (ex : « monstockageazure »), un emplacement (ex : « Europe du Nord »), et un mode de réplication (ex : « Redondant en local »).

Connexion en interactif

Pour se connecter sur le nœud principal du cluster avec le bureau à distance, il suffit d'aller dans le portail de gestion de Windows Azure, dans la rubrique HDInsight ; dans la configuration du portail, il faut utiliser

le bouton « Activer Distant » et donner un compte et un mot de passe. Le bouton « Activer Distant » sera remplacé par deux boutons « Connecter » et « Désactiver Distant ». Le premier sert à se connecter en ouvrant le fichier .rdp fourni. En ouvrant le raccourci du bureau « Hadoop Command Line », on peut regarder dans les systèmes de fichiers. Dans la copie d'écran suivante, la première commande montre que par défaut on voit ce qu'il y a dans le compte de stockage Windows Azure. La seconde montre qu'on peut aussi accéder à HDFS Fig.2.

Exécution de jobs en Java, .NET, Hive, PIG

On peut exécuter des jobs sous forme de jars, ou avec les langages de script de plus haut niveau PIG ou HIVE. Les nœuds ont également C Python 2.7 installé ce qui permet d'exécuter par exemple des jobs HIVE qui déléguent une partie de leur traitement difficile à exprimer de façon ensembliste à un langage procédural. Il y a également un SDK pour développer des jobs Map/Reduce en

Fig.2

C#. Ce dernier est en incubation à <http://hadoop sdk.codeplex.com>. Prenons ici pour exemple un code Hive + Python qui s'appuie sur la table exemple fournie par défaut dans HDInsight. On met le code Hive dans simple_sample.txt :

```
add file c:\simple_sample.py;

SELECT TRANSFORM (clientid, devicemode, devicemode)
  USING 'D:\Python27\python.exe simple_sample.py' AS
  (clientid string, phoneLabel string, phoneHash string)
FROM hivesampletable
ORDER BY clientid LIMIT 50;
```

On met le code Python dans simple_sample.py :

```
import sys
import string
import hashlib

while True:
    line = sys.stdin.readline()
    if not line:
        break

    line = string.strip(line, "\n ")
    clientid, devicemode, devicemode = string.split(line, "\t")
    phone_label = devicemode + ' ' + devicemode
    print "\t".join([clientid, phone_label, hashlib.md5(phone_
label).hexdigest()])
```

On exécute de la façon suivante depuis l'Hadoop Command Line :

```
cd ..\hive< touche TAB>\bin
c:\apps\dist\hive-0.11.0.0.1.3.2.0-05\bin>hive -v -f c:\simple_
_sample.txt
```

ce qui donne : Fig.3.

Soumission de jobs en C# et PowerShell

La connexion en RDP est pratique en développement, mais en production, il est préférable d'automatiser la création, le lancement du job et la suppression du cluster. Cela peut se faire en .NET ou en PowerShell. Toutes les informations sont disponibles à <http://hadoop sdk.codeplex.com>. A titre d'exemple, voici le même job que précédemment lancé en PowerShell :

```
Import-Module azure
Add-AzureAccount
Use-AzureHDInsightCluster "monclusterhadoop"
Select-AzureSubscription -Default "Azdem169A44055X"
$clusterAdmin = 'admin'
$clusterPassword = 'PutYours_Here'
$passwd = ConvertTo-SecureString $clusterPassword -AsPlain
Text -Force
$clusterCredentials = New-Object System.Management.Automation.
PSCredential ($clusterAdmin, $passwd)

$hiveJobVT = New-AzureHDInsightHiveJobDefinition -File "wasb://messscripts@monstockageazure.blob.core.windows.net/simple_
sample.txt"
$hiveJobVT.Files.Add("wasb://messscripts@monstockageazure.blob.
core.windows.net/simple_sample.py")
$startedHiveJobVT = $hiveJobVT | Start-AzureHDInsightJob -
```

```
Credential $clusterCredentials -Cluster "monclusterhadoop"
$startedHiveJobVT | Wait-AzureHDInsightJob -Credential $cluster
Credentials Get-AzureHDInsightJobOutput -TaskSummary -JobId
$startedHiveJobVT.JobId -Cluster "monclusterhadoop" Get-Azure
HDInsightJobOutput -StandardError -JobId $startedHiveJobVT.
JobId -Cluster "monclusterhadoop" Get-AzureHDInsightJobOut
put -StandardOutput -JobId $startedHiveJobVT.JobId -Cluster
"monclusterhadoop"
```

NB : Il faut remplacer la ligne « add file c:\simple_sample.py; » du fichier simple_sample.txt par « add file simple_sample.py; » et mettre les deux scripts dans le compte de stockage Windows Azure.

Combien ça coûte ?

Le coût peut être calculé en se rendant à <http://windowsazure.com>, rubrique Tarification. Mais prenons plutôt un exemple qui montre qu'avec un cluster, on peut n'allumer que le temps d'un batch, Hadoop se révèle peu cher. Le scénario consiste à écouter les réseaux sociaux, garder 1 To dans le Cloud et les analyser une fois par semaine sur 80 cœurs. Avec un machine virtuelle de taille A1 qui tourne 24h/24 et ses 130 Go de stockage associés, on peut écouter les réseaux sociaux. Par exemple, écouter Twitter correspond à l'écoute d'un GET HTTP dont la réponse ne se termine jamais. Cette même machine peut aller effectuer sur d'autres threads des requêtes sur les API Facebook et YouTube par exemple. Coût estimé : 40,04 € / mois. Les données récupérées sont stockées sur le stockage en blobs où l'on estime un volume de 1 To. Coût estimé 53,45 € / mois. On peut ensuite exécuter des jobs PIG ou HIVE sur un cluster Hadoop (service HDInsight) de 18 nœuds, soit (1 nœud de 8 cœurs + 18 nœuds de 4 cœurs chacun = 8 + 72 cœurs soit 80 cœurs). Ce cluster peut être utilisé 2 heures / semaine soit 8 heures par mois. Coût estimé : 14,30 € / mois. Le tout peut générer quelques 90 millions de transactions de stockage. Coût estimé : 6,75 € / mois. Enfin, les résultats nécessitent de faire sortir du datacenter Windows Azure 10 Go / mois, soit 5 Go au-delà des 5 Go inclus gratuitement. Coût estimé : 0,45 € / mois. On voit donc que sur un budget de moins de 115 € par mois, Hadoop représente 15 €.

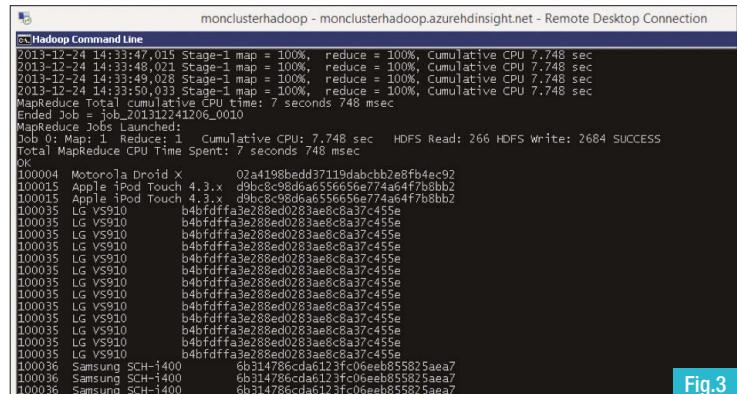


Fig.3

Conclusion

HDInsight est donc un moyen simple de traiter beaucoup de données stockées dans des blobs Windows Azure, avec les outils standard d'Hadoop, et pour un coût réduit. Ressources complémentaires <http://www.windowsazure.com/en-us/services/hdinsight/>, <http://hadoop sdk.codeplex.com/>, <http://aka.ms/bigdatafrance>

● Benjamin Guinebertière

évangélise technique à Microsoft France sur Windows Azure, en particulier sur la plateforme de données. Cela consiste à écrire, parler et aider. Vous pouvez le suivre et lui poser des questions sur <http://twitter.com/@benjuin>.

ESIEA : des écoles à la pointe de l'informatique, de la sécurité et de la technologie

Aux portes de Paris, le campus du Groupe ESIEA attend ses étudiants. Articulé autour d'une grande cour, on y trouve l'école d'ingénieur l'ESIEA (école supérieure d'informatique électronique automatique) et In'Tech Info, orienté technologie et informatique. Les origines de l'ESIEA remontent à 1957 à Chamonix. Aujourd'hui, l'école se partage entre Ivry sur Seine et Laval, et tisse des partenariats en France et à l'international. In'Tech Info fait partie du groupe. Focus.

In'Tech Info est ce que l'on pourrait appeler une petite école : 200 étudiants. Elle est récente (fondée en 2002). Sa mission est claire : former des experts en ingénierie informatique capables d'être immédiatement opérationnels. « L'encaissement est très important. Les cours sont souvent donnés par des personnes venant de l'entreprise. Nous voulons que l'étudiant soit proche de l'entreprise et de sa réalité. » précise Eric Lalitte, directeur de l'école.

La pédagogie par projet est mise en avant; grand classique de ce type de formation. Les projets durent en général 3 à 4 mois. Mais au-delà du projet, il s'agit aussi de savoir travailler en équipe, de gérer le relationnel. Les équipes se forment et le rôle de chacun est défini. Chaque semaine, l'équipe, et le professeur qui



D.R.

suit le projet, se réunissent pour faire le point. Le professeur est tout autant coach que manager et chef de projet. Ces projets sont souvent des projets venant d'entreprises et sont donc très réels. Les étudiants doivent choisir les outils et les technologies.

Comment se déroule l'admission ?

Tout débute sur le site web. « On remplit le dossier. Puis il y a une journée de tests à l'admission. Un entretien en Anglais est réalisé. Les équivalences ne sont pas automatiques. Un étudiant avec le bac peut très bien entrer au 2e semestre s'il a toutes les compétences informatiques requises. » explique Eric Lalitte.

Ensuite, le cursus se découpe en semestres.

Les premiers semestres permettent d'apprendre, d'approfondir les bases, puis de choisir son orientation et sa spécialisation. Un semestre entier est dédié au stage de fin de cycle. Ce parcours est le cursus des 3 premières années. Les deux années suivantes s'articulent sur le travail en entreprise :

- 4 jours en entreprise, 1 journée à l'école
- 3 semaines à l'école par semestre

Les frais de scolarité sont de 26 100 euros pour les trois premières années (= 4 350 €/semestre). Pour les deux dernières années, obligatoirement en apprentissage, l'entreprise paie l'intégralité des frais de scolarité ainsi que le salaire de l'étudiant.

Motiver l'entrepreneuriat

Même si tous les étudiants n'ont pas vocation à créer une société, l'école possède une pépinière pour donner un cadre, des conseils. Aux étudiants de proposer un business plan et de définir les ambitions de la start-up. Plusieurs startups existent toujours, c'est le cas de dress-things.fr.

L'école annonce que 98 % des étudiants trouvent un emploi après leur cursus. « Pour nos étudiants, nous ne voyons pas de chômage. Le CV est déjà bien rempli à la sortie. » recadre Eric Lalitte. Comme pour les autres écoles d'informatique, les filles sont très peu présentes. La direction annonce 5-15 % selon les promotions.

Journées portes ouvertes :

Sites web : <http://www.intechinfo.fr/>
<http://www.esiea.fr>

ESIEA : le vaisseau mère

Le vaisseau amiral est l'école ESIEA. Elle accueille plus de 1000 étudiants sur les différents campus. Elle délivre un diplôme d'ingénieur (habilité par la CTI), en cursus classique et en alternance. « Les 3 premières années, les étudiants reçoivent un socle informatique/électronique. C'est à la fin de la 3e année qu'ils vont approfondir et choisir les systèmes d'informations ou le monde de l'em-

barqué. Nous faisons beaucoup de mathématiques et de physiques. » évoque Philippe Volle (directeur général du groupe).

L'école possède 4 laboratoires de recherches. Le plus connu est sans aucun doute celui dirigé par Eric Filiol, une des figures de la sécurité en France. Le laboratoire de cryptologie et de virologie a participé à la conception du démonstrateur de l'antivirus

français, DAVFI. C'est incontestablement une vitrine. Là aussi, le mode projet est très important. Les étudiants choisissent dans une liste de projets, mais ils peuvent aussi proposer (par exemple, la section Systèmes d'information anime 26 projets). Des stages à l'étranger sont possibles et durent au minimum 2 à 3 mois. Les frais de scolarité s'élèvent à 7850 €/année.

Lycée Jean Perrin : l'électronique, l'embarqué, le duo de choc

Le concours de robotique, coupe NXT, organisé par la société National Instruments est l'occasion pour des étudiants de différents lycées français de rivaliser d'ingéniosité pour créer un robot répondant à un cahier des charges précis.

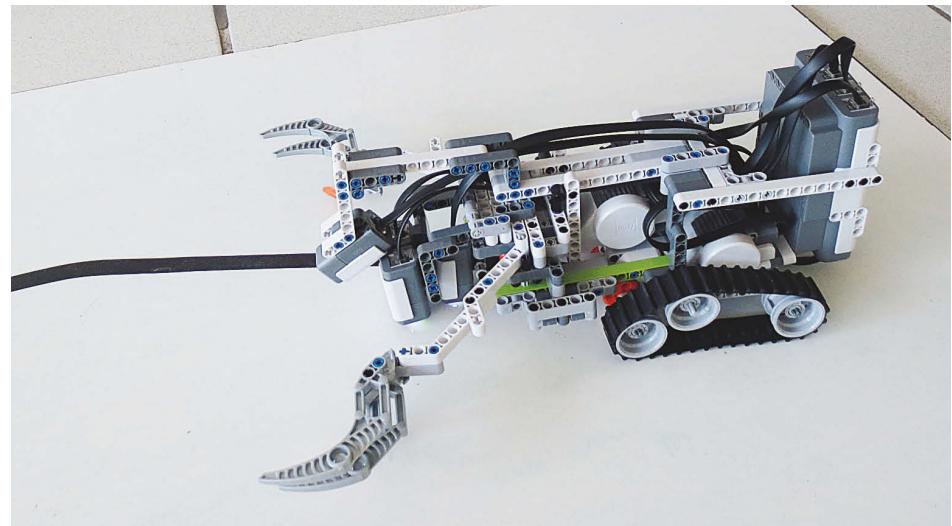
Nous sommes allés voir une classe du lycée Jean Perrin (Saint Ouen l'Aumône, 95).

Le lycée possède un BTS très spécialisé : le BTS Systèmes électroniques qui changera de nom en 2014 pour s'appeler BTS Systèmes Numériques. Les étudiants de cette section acquièrent une solide connaissance en électronique/informatique; il faut une bonne dose de mathématiques et de physiques pour en comprendre les subtilités, notamment dans la domotique, l'électronique embarquée ou la robotique. C'est la 3e année que les étudiants participent à ce concours. Comme dans d'autres filières scientifiques et technologiques, les filles sont peu présentes. Le Lycée accueille environ 940 élèves et dont seulement 76 filles.

Trois équipes ont été formées mais des éliminatoires permettent de choisir celle qui va représenter l'école. La Coupe NXT utilise les Lego MindStorm; il faut utiliser LabVIEW pour programmer le comportement du robot.

LabView, C et Android

Les étudiants doivent acquérir les pré-requis du projet de BTS avant janvier en 2nde année de formation. A leur programme, notions de cahier des charges, langages comme LabVIEW, ou le C, ou encore l'Android, utilisation et mise en œuvre des capteurs et actionneurs, et, enfin, intégration des cartes électroniques et réalisation des circuits imprimés pour les conditionnements. L'intérêt du concours est directement lié



© Roselyne Griffon

aux études comme nous l'a précisé Roselyne Henri, professeur. Avec LabVIEW, les étudiants apprennent la robotique, ils programment des systèmes GPS et GSM, réalisent des interfaces d'oscilloscope, etc. « *On travaille avec des systèmes informatiques diversifiés, on doit maîtriser toute la chaîne d'information et d'énergie* » poursuit Mme Henri. Les étudiants travailleront sur des projets industriels après janvier autour de l'embarqué, de l'instrumentation et de la programmation. Ces projets comptent pour la délivrance du BTS.

Ce concours a deux missions : respecter le cahier des charges, concevoir et coder une stratégie. Ce travail permet de développer la collaboration et le savoir en équipe, ce n'est pas simple. Les équipes doivent composer avec les missions imposées, gérer l'environnement para-

site, gérer les capteurs parfois trop sensibles ou ne réagissant pas comme ils devraient. La vitesse des robots est aussi prise en compte mais le robot doit avoir le bon équilibre sinon, il reste l'accident. La conception du programme en LabVIEW se base sur un diagramme d'état qui intègre la logique fonctionnelle. Il sera téléchargé dans l'électronique embarquée du robot.

Des partenariats entre des entreprises et le lycée font que les étudiants participent aussi à la Recherche & Développement. Ils voient comment l'innovation et des projets industriels dans le monde de l'embarqué, de la mécatronique, de l'instrumentation, se gèrent de l'idée à la conception réelle. Ces projets R&D concernent aussi bien l'instrumentation que la mécatronique.

Sur la partie programmation, si la Coupe NXT utilise LabView, en dehors de ce concours, les

étudiants utilisent aussi bien LabVIEW que le langage C, ce qui est très important dans l'univers de l'embarqué. Le système Android est aussi utilisé parfois comme système d'exploitation. Une des difficultés pour ces cursus est de suivre les évolutions scientifiques et technologiques qui progressent fortement. Ainsi, le Raspberry Pi a fait récemment son apparition dans les BTS. Les cartes ChipKit de Digilent et leur Shields sont aussi utilisées.

Dans le cursus, le projet représente 120 heures/étudiant pour encoder les fonctions et le système embarqué, intégrer les cartes et produire des circuits pour le conditionnement, seul le travail d'équipe permet d'arriver au bout.

Merci à l'équipe pédagogique du lycée.



© Roselyne Griffon

Les réalisations.

François Tonic

A la découverte des Google Glass



Les Google Glass sont des lunettes dont la particularité réside dans la présence d'un écran situé au-dessus de l'œil droit. Elles contiennent une branche tactile sur le côté droit de la tête, dans laquelle est intégré tout un système d'exploitation fonctionnant sous Android 4.x.

Au niveau du hardware, l'écran a une résolution de 640x360 pixels. Les Glass ont également une caméra qui capture ce que l'œil droit voit. Elle permet de filmer en 720p ou prendre des photos en 5 Mégapixels.

Niveau connectivité, on dispose des connexions Wi-Fi 802.11b/g et Bluetooth. Les Glass proposent 16 Go de stockage, dont 12 Go réellement utilisables, le reste étant dédié au système. Toutes les données sont synchronisées sur le Google Cloud Storage, les Glass tournent sous **Android Ice Cream Sandwich** (4.0) avec un launcher spécialement conçu. Le processeur est un Texas Instruments 1.2 Ghz Dual Core (ARM v7). La mémoire est de 682MB de RAM (1Gb pour les plus récentes). L'interaction se fait avec les doigts, grâce à la branche tactile, ou bien par la voix, grâce au micro intégré. Les Glass disposent aussi d'un gyroscope, d'un accéléromètre et d'une boussole, utilisés par exemple dans l'application Compass qui affiche la direction regardée par l'utilisateur.

Ces lunettes technologiques qui n'en sont pas moins esthétiques sont disponibles en cinq coloris : anthracite, gris, orange, rouge, blanc et bleu **Fig.1**.

Des verres solaires peuvent se rajouter sur les Glass et sont également fournis dans le but d'améliorer le confort d'utilisation **Fig.2**.

A quoi ça sert ?

Les Google Glass donnent des informations utiles dans la vie de tous les jours sans avoir à sortir son téléphone portable. Avec l'écran, les Glass permettent de superposer le réel au virtuel. C'est justement là que réside le principal intérêt de ces lunettes.

Il y a deux manières d'interagir avec les Glass : soit par la reconnaissance vocale (disponible

uniquement en anglais actuellement), soit par le biais d'un menu à l'aide de la branche droite des Glass, entièrement tactile. Avec cette dernière, on navigue via un menu horizontal en faisant glisser son doigt d'avant en arrière et inversement. Afin de sélectionner un élément, il nous suffit d'appuyer une fois sur le pavé tactile. Enfin, un glissement vers le bas nous fera très simplement revenir en arrière. Pour activer la reconnaissance vocale, il suffit de dire "ok glass" puis d'énoncer l'action souhaitée. Voici la liste non exhaustive des principales actions disponibles :

► Prise de photos et de vidéos via les commandes "take a picture" et "record a video". Il est également possible de prendre une photo en clignant de l'œil bien que les Glass soient en veille (à noter que les glass font bien entendu la distinction entre un clignement volontaire d'un clignement naturel, cela évite de prendre des photos par inadvertance). Les photos et vidéos pourront être partagées avec vos contacts sur les réseaux sociaux (Google+, Facebook, Twitter ou Instagram).

► Navigation GPS : il suffit de dire "get directions to (votre destination)" pour afficher Google Maps qui vous guidera jusqu'au lieu de votre choix. Vous devrez néanmoins connecter les Glass à votre téléphone en Bluetooth.

► Envoi de messages et mails : la commande "send a message to (votre destinataire)" enverra un message.

► Appels téléphoniques et Hangouts vidéos : "make a video call with (votre interlocuteur)" démarrera une vidéo conférence avec Google Hangouts.

► Assistant vocal : les Glass sont conçues pour que nous puissions poser des questions

oralement. Comme par exemple : "Google, what is the height of the Eiffel Tower?". La réponse s'affichera alors sur l'écran des Glass.

► Traduction visuelle d'un texte : si vous avez installé l'application Word Lens, vous pourrez dire "translate this" qui lancera l'application. Celle-ci permettra de prendre en photo un panneau ou un affichage contenant du texte afin de traduire son contenu et d'afficher le résultat sur l'écran des Glass.

La traduction se superpose sur l'image prise en photo de telle sorte que la traduction puisse être lue sur le même décor.

En revanche cette application ne permet pas la traduction de textes avec une police fantaisiste ou celle d'une écriture manuscrite.

Les Google Glass étant essentiellement destinées aux développeurs, pour le moment, il est bien sûr possible de créer des applications sur Glass. Ces applications peuvent être installées via le Glassware (<https://glass.google.com/myglass>). Il existe déjà quelques applications, en voici quelques exemples :

► **GolfSight** permet d'afficher de nombreuses informations lors d'une partie de golf comme la distance du trou ou le score.

► **Allthecooks** est une application de cuisine qui vous guide dans la confection de vos repas.

► **Evernote for Glass** vous permet d'ajouter des notes vocalement sur la célèbre plate-forme Evernote.

► **Glass Stocks** affiche quant à lui des graphiques et autres informations sur la bourse.

Voyons maintenant comment développer vos propres applications sur les Google Glass !

Comment développer des applications sur Glass ?

Nous allons voir dans cette section les deux moyens possibles pour développer sur les Glass.

► La Mirror API : cette technologie permet de développer des applications pour Google Glass dans le Cloud.

Cette approche facilite le développement et ne nécessite aucun code sur les Glass. Les interactions étant gérées automatiquement par le système et les serveurs de Google, celui-ci met à disposition son backend Google App Engine. Il est possible de développer son application en Go, Java, .NET, PHP, Python ou Ruby.

Il est bien sûr autorisé d'héberger son application sur son propre serveur et d'utiliser les langages de programmation de son choix.

► Le GDK (Glass Development Kit) : il permet de réaliser des applications se trouvant sur les Glass, ne nécessitant pas une connexion Internet. Le développement est similaire à celui d'Android et se base sur l'API 15 (Android 4.0.3). L'intérêt du GDK est de pouvoir développer des applications nécessitant de tourner en temps réel (navigation, réalité augmentée, etc.) avec un accès au hardware des Glass (caméra, gyroscope, accéléromètre, microphone, boussole).

Nous allons voir dans cette section comment créer une application avec la Mirror API et une avec le GDK.

Explorons la Mirror API

Il est possible d'interagir avec les Google Glass via la Mirror API dans le langage de son choix. Nous allons ici réaliser une application simple qui vous permettra d'obtenir les derniers articles parus sur le site Internet de Programmez!. Google propose sur son site des packs "QuickStart" pour PHP, Python, Java, Ruby et .NET.

I. Activation Mirror API

Tout d'abord il suffit de se rendre sur <https://cloud.google.com/console/project> pour créer un projet. L'onglet API permettra d'activer l'accès à "Google Mirror API".

Ensuite, il faut se rendre dans "Credentials" pour créer un "Client id for Web Applications" : vous obtiendrez un "Client ID" et un "Client Secret" à conserver pour votre application.

Il faut ensuite indiquer "<http://localhost/glass/oauth2callback>" dans "Redirect URIs" pour autoriser votre poste local. Pour finir, l'onglet "Consent screen" permet de définir le nom de notre service avec "Product Name".

Pour proposer un développement différent, nous allons programmer notre application en utilisant NodeJS. Les consignes pour installer NodeJS sont disponibles sur le site <http://nodejs.org/>

II. package.json

Commençons notre projet NodeJS en créant un fichier package.json.

[Code Javascript : NodeJS/package.json]

Ce fichier contient la description de notre application au format JSON (JavaScript Object Notation). On peut remarquer le champ "dependencies" qui sert à indiquer les modules que nous voulons utiliser.

Il vous suffit de lancer "npm install" pour que le fichier soit interprété et que les modules soient téléchargés automatiquement.

III. app.js (Code sur www.programmez.com)

1. Les Bases

Nous pouvons maintenant créer le fichier qui va contenir notre application : app.js.

Commençons par le code de base de notre application.

[Code Javascript : NodeJS/part1.js]

Le premier bloc de texte permet d'inclure toutes les dépendances de notre projet. Le second permet d'utiliser express pour gérer les

points d'accès de notre application, par exemple '/glass/' , '/glass/timeline' et de mettre en place les cookies et les sessions.

Le dernier bloc permet de lancer notre application en écoute sur le port 8084.

Pour lancer l'application, il suffit de taper la commande "nodejs app.js" ou bien "node app.js" selon votre système d'exploitation.

2. Connexions aux APIs Google

Ajoutons maintenant la connexion aux APIs Google.

[Code Javascript : NodeJS/part2.js]

Les deux "app.get" permettent à notre application d'être en écoute sur les url '/glass' et '/glass/oauth2callback'.

La fonction checkLogged() permet, comme son nom l'indique, de vérifier que l'utilisateur est connecté aux APIs Google. Dans le cas contraire, l'utilisateur est redirigé vers une page de connexion.

Les fonctions grabToken() et restoreClientSession() servent à gérer le retour de connexion aux APIs Google et à conserver la session utilisateur.

Maintenant, lorsque l'utilisateur se rend sur <http://localhost:8084/glass>, celui-ci se connecte aux APIs Google. Lorsque le message "{ok:ok}" s'affiche, le client est donc correctement connecté à l'application.

3. Récupérer sa timeline

Nous rajoutons dans notre code un point d'entrée sur '/glass/timeline'. On vérifie alors que l'utilisateur est connecté.

La fonction getToken() qui vérifie la validité du token appelle la fonction listTimeline() pour afficher en Json la liste des cartes de la timeline.

Cette liste est pour le moment vide !

En effet nous n'avons pas encore ajouté de carte à la timeline.

[Code Javascript : NodeJS/part3.js]



Fig.1

Fig.2



4. Ajouter une carte à sa timeline

Nous ajoutons un point d'entrée pour demander la création d'une carte, '/glass/card'. La fonction `insertCard()` télécharge le flux RSS du magazine Programmez!, change l'encodage de celui-ci en UTF-8 pour interagir au mieux avec NodeJS et transcrit le format XML en objet javascript. On sélectionne ensuite un article au hasard avec `Math.random()` et nous l'envoyons dans la timeline avec la fonction `insert()`. On permet à l'utilisateur d'ouvrir le site, de partager la carte ou de la supprimer via le "menulitems". En allant sur <http://localhost:8084/glass/card>, une carte est ajoutée : il n'y a donc plus qu'à se rendre sur <http://localhost:8084/glass/timeline> pour observer l'ajout de la carte à la timeline.
[Code Javascript : NodeJS/part4.js]

5. Pour aller plus loin

Une archive contenant d'autres fonctions est disponible sur le site de Programmez ! : elle contient notamment les fonctions de suppression de cartes et un système d'envoi automatique de notification de nouveaux articles. Vous pouvez la retrouver sur leur site.
[Code Javascript : NodeJS/nodejs.zip]

Explorons le GDK

Le GDK permet de développer une application sur Glass à la manière d'une application Android classique, que ce soit sur téléphone ou tablette. La seule différence est la gestion du menu Android (ActionBar) : il est en plein écran et l'Activity est en background de ce menu. Ce qui donne une approche sensiblement différente : on ne peut pas porter directement

une application fonctionnant sur téléphone vers Glass.

Il existe aussi une autre différence : sur téléphone et tablette, la première action lancée lors de l'exécution d'une application est la launch Activity, tandis que sur Glass c'est un Service; c'est alors lui qui devra se charger de lancer l'Activity que vous voulez.

Nous verrons ici uniquement comment gérer les interactions avec l'utilisateur : reconnaître les mouvements de **swipe** ou de **tap** ainsi que les mouvements de têtes via l'accéléromètre.

Commençons tout d'abord par le **GestureDetector** des Glass. Cette classe vous permettra d'interagir avec les gestes que l'utilisateur peut effectuer sur la branche tactile. On crée donc une Activity qui va instancier un GestureDetector.

[Code JAVA : GDK/gesturedetector1.java]

Ensuite, on crée la méthode `createGestureDetector()` qui attribue les listeners au GestureDetector puis retourne la classe instanciée.

[Code JAVA : GDK/gesturedetector2.java]

NB : Il existe beaucoup d'autres gestes comme **TWO_LONG_PRESS** par exemple. Vous pouvez tous les trouver sur cette page :

<https://developers.google.com/glass/develop/gdk/reference/com/google/android/glass/touchpad/Gesture>.

Enfin, il faut créer la méthode qui sera automatiquement appelée lorsqu'un geste sera effectué.

[Code JAVA : GDK/gesturedetector3.java]

Maintenant, nous allons voir comment utiliser l'accéléromètre des Glass afin de détecter les mouvements de la tête. On crée donc une Activity qui implémente l'interface

`SensorEventListener`.

[Code JAVA : GDK/sensor1.java]

Ensuite, on ajoute la méthode `onSensorChanged()` qui sera appelée lorsqu'une accélération sera enregistrée : c'est-à-dire lorsque l'utilisateur bougera sa tête.
[Code JAVA : sensor2.java]

LIMITES DES GLASS

Encore en développement, les Google Glass ont plusieurs limites. Il en est ainsi de son prix : il vous faudra d'abord être sélectionné par Google pour faire partie du programme Glass Explorer afin, ensuite, de pouvoir acquérir les fameuses lunettes de Google pour 1500\$. De plus, vous devrez vous déplacer aux États-Unis pour les récupérer : Google ne les livrant pas à l'étranger. Vous pouvez remplir le formulaire à cette adresse :

<http://www.google.com/glass/start/how-to-get-one/>.

Autre point négatif, cette fois au niveau de l'utilisation des lunettes : elles chauffent beaucoup lorsqu'on les utilise pour des applications assez lourdes (lecture vidéo, jeux...) et la batterie a une durée de vie assez courte (environ 4 heures en utilisation normale). Aussi, il est parfois impossible de se connecter à certains réseaux WiFi, car certains protocoles comme le WPA Entreprise ne sont pas encore compatibles.

On peut facilement imaginer que ces problèmes soient corrigés par Google lors de la sortie de la version grand public.

 Victor Leduc, Vincent Néel, Denis Mondon, Michaël Ohayon, Jonathan Picques.
Epitech

Oculus Rift : la réalité virtuelle à portée de code ?

Une des grandes nouveautés de l'année 2013 fut l'arrivée du casque de réalité virtuelle Oculus Rift dans les mains des développeurs. Ceux qui ont financé le projet sur Kickstarter ou qui ont acheté le kit de développement sur le site ont pu essayer ce casque, et démarrer leurs projets les plus fous.



HDMI

LE RIFT VU DE L'INTERIEUR

A l'intérieur du casque, un écran de 7 pouces est placé au fond de l'appareil, sur lequel 2 lentilles sont superposées. Une fois le casque mis devant les yeux, les lentilles permettent de remplir le champ de vision jusqu'à 110 degrés horizontal, pour une vision la plus large possible. L'écran est divisé en deux, une moitié pour chaque œil. Le jeu vidéo utilisé devra calculer ces deux images avec un léger décalage afin de créer l'effet 3D stéréoscopique. A l'intérieur du casque, 3 capteurs vont envoyer les positions du casque avec une très faible latence. Ainsi, chaque mouvement de l'utilisateur sera retranscrit à l'écran immédiatement, et la sensation d'être dans le jeu n'en sera que plus prenante. Le casque dans sa version actuelle est plutôt léger. Une fois mis sur la tête, il sait se faire oublier. Le prix actuel est de 300\$ hors frais d'envoi. Attention : Il s'agit d'une version destinée aux développeurs Fig.2. Une fois le kit reçu, il suffit de brancher les câbles USB et HDMI, télécharger la dernière version du SDK sur le site www.developer.oculusvr.com. La mise à jour du micro-logiciel du casque est conseillée. Le casque est maintenant prêt à être utilisé !

CONTENU DU SDK

Le SDK contient plusieurs dossiers :

- ▷ « 3rdParty » contient des bibliothèques externes nécessaires au projet,
- ▷ « Doc » contient un PDF expliquant le fonctionnement du SDK,
- ▷ « Firmware » contient la dernière version du logiciel. Pour mettre le casque à jour, voir le dossier « Tools »
- ▷ « LibOVR » contient les sources du SDK pour le compiler sur la plateforme associée,
- ▷ « Samples » contient plusieurs démos techniques pour tester le casque.
- ▷ « Tools » Contient un logiciel permettant la calibration du casque. Il permet aussi de mettre à jour la version du logiciel interne au casque, étape largement recommandée au premier lancement.

ET NIVEAU LOGICIEL ?

L'ordinateur voit l'Oculus Rift comme un simple écran secondaire, avec un périphérique USB renvoyant les positions du casque.

Le casque étant compatible PC Mac et Linux, 3 versions du SDK sont disponibles sur le site et chacune installera les pilotes nécessaires pour



Fig.2

la reconnaissance du périphérique USB. Afin de récupérer les positions du casque, le jeu doit intégrer une bibliothèque C++, fournie dans le SDK. Cette bibliothèque renverra toutes les informations liées au casque, ce qui permettra au jeu de modifier correctement son rendu. Ainsi, n'importe quel logiciel pouvant intégrer cette bibliothèque est dès aujourd'hui capable d'utiliser l'Oculus Rift. Coté Web pour l'instant, la détection ne se fait pas, faute d'interaction entre les pilotes renvoyant les positions et une page web.

COMMENT L'INTEGRER

Le SDK fournit la bibliothèque C++ permettant d'interagir avec le casque. Elle est compatible Windows/Mac/Linux, et très facile à prendre en main. Voici comment l'intégrer dans un code C++ :

```
#include <OVR.h>
using namespace OVR;
System::Init(Log::ConfigureDefaultLog(LogMask_All));
```

Ici la bibliothèque est initialisée. Elle est prête à renvoyer les caractéristiques du casque connecté. Afin de pouvoir garder son code compatible avec les prochaines versions du casque, il faut se baser sur les caractéristiques renvoyées par la bibliothèque :

```
Ptr<DeviceManager> pManager;
Ptr<HMDDevice> pHMD;
HMDInfo hmd;

pManager = *DeviceManager::Create();
pHMD = *pManager->EnumerateDevices<HMDDevice>().CreateDevice();
if (pHMD->GetDeviceInfo(&hmd))
{
    MonitorName = hmd.DisplayDeviceName;
    EyeDistance = hmd.InterpupillaryDistance;
    DistortionK[0] = hmd.DistortionK[0];
    DistortionK[1] = hmd.DistortionK[1];
    DistortionK[2] = hmd.DistortionK[2];
    DistortionK[3] = hmd.DistortionK[3];
}
```

Fig.3

Avec les caractéristiques du Rift en mains, la prochaine étape est de récupérer les positions de la tête. Les 3 capteurs de position envoient leurs informations à travers la classe SensorFusion. La fonction SensorFusion::GetOrientation() renvoie la rotation en y, l'angle d'inclinaison x ainsi que la rotation en z.

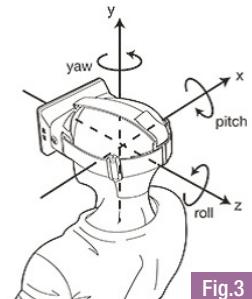


Fig.3

```

Ptr<SensorDevice> pSensor;
pSensor = *pHMD->GetSensor();
SensorFusion SFusion;
if (pSensor)
    SFusion.AttachToSensor(pSensor);
Quatf hmdOrient = SFusion.GetOrientation();
float yaw = 0.0f;
float EyePitch = 0.0f;
float EyeRoll = 0.0f;
float EyeYaw = 0.0f;
hmdOrient.GetEulerABC<Axis_Y, Axis_X, Axis_Z>(&yaw,
&EyePitch, &EyeRoll);
EyeYaw += (yaw - LastSensorYaw);
LastSensorYaw = yaw;
// Il est aussi possible de récupérer les positions
// dans une matrice
Matrix4f hmdMat(hmdOrient);

```

A chaque tour de boucle du jeu, la position de la tête doit être récupérée afin de calculer le mouvement de caméra adéquat en temps réel. Il est également possible de récupérer les valeurs brutes du casque telles que l'accélération, la rotation, la force du champ magnétique ainsi que la température en degrés Celsius à travers la classe MessageBodyFrame :

```

Vector3f Acceleration;
Vector3f RotationRate;
Vector3f MagneticField;
float Temperature;
float TimeDelta;

```

Occupons-nous maintenant du rendu stéréoscopique. On se souvient que l'Oculus Rift dispose d'un écran séparé en deux verticalement, avec une partie pour chaque œil. Il faut donc appliquer 2 rendus. Puisqu'il existe de nombreux moteurs de rendu avec des noms de méthodes différents, le code ci-dessous est donné à titre indicatif afin de comprendre comment l'intégrer dans votre moteur de rendu :

```

// HResolution est la résolution horizontale de
// l'écran, VResolution la résolution verticale
// Rendu de l'œil gauche
SetViewport(0, 0, HResolution/2, VResolution);
SetProjection(LeftEyeProjectionMatrix);
RenderScene();
// Rendu de l'œil droit
SetViewport(HResolution/2, 0, HResolution, VResolution);
SetProjection(RightEyeProjectionMatrix);
RenderScene();

```

Les lentilles placées devant les yeux sont utilisées pour étendre le champ de vision. Elles causent une déformation de l'image, qu'il est nécessaire de corriger. Une distorsion en bâillet doit être appliquée. Après un léger temps d'utilisation, un décalage peut apparaître entre la position droite du corps et la rotation en y de l'image de l'écran. Afin de pouvoir corriger cet écart, le jeu doit pouvoir proposer au joueur de réinitialiser la position de l'image.

Les jeux à la première personne ne présentent pas ce défaut car le joueur utilise la souris pour s'orienter et garder une position face au clavier. Les développeurs de l'Oculus s'appliquent à corriger cette dérivation pour la version finale du casque.

MOTEUR DE JEU INTEGRANT LE SDK

En effet, tout le monde n'est pas enclin à vouloir se mettre au C++ pour créer ses premiers jeux adaptés à la réalité virtuelle. Il existe heureusement des moteurs de jeu intégrant un support complet de l'Oculus Rift. L'une des tendances actuelles en matière de création de jeu vidéo est le moteur Unity. Unity dans sa version Pro intègre à travers un plugin les bibliothèques nécessaires. Il est donc tout à fait envisageable de rendre compatible un jeu précédemment créé avec Unity avec l'Oculus Rift. De plus, Oculus VR fournit un mois gratuit à Unity Pro avec le kit de développement. Un autre moteur est également compatible avec l'Oculus Rift, il s'agit de l'Unreal Engine 3 et 4. Ces solutions sont les meilleures dans le cas où le développeur souhaite démarrer son projet utilisant l'Oculus Rift le plus rapidement. De nombreuses démos jouables sont disponibles sur le site d'Oculus VR, et nombre d'entre-elles utilisent Unity comme moteur de jeu : <https://share.oculusvr.com/>

JEU INTEGRANT L'OCULUS RIFT

Il existe à l'heure actuelle peu de jeux utilisant cette technologie. Le produit n'étant pas encore sorti pour le grand public, il faudra attendre encore quelques temps avant que les grands groupes tels qu'Ubisoft ou EA commencent à intégrer cette technologie dans leurs jeux. En revanche, certaines entreprises comme Valve supportent entièrement Oculus VR. Après avoir intégré le SDK Oculus à Half-life 2, Team Fortress 2 et Portal, Valve a ajouté un mode à Steam pour l'utiliser à travers le casque de réalité virtuelle. Il est donc possible de naviguer dans sa bibliothèque de jeu avec le casque.

COMPATIBILITE AVEC LES ANCIENS JEUX

Après avoir entendu parler de l'Oculus Rift, beaucoup se posent la question de savoir s'il est possible de rendre compatible d'anciens jeux afin de pouvoir s'y replonger d'une manière inédite. Même si une vraie intégration nécessite d'accéder au code source du jeu, plusieurs équipes de développement ont réfléchi à une autre manière d'intégrer le casque. Ces équipes ont développé des drivers qui vont s'ajouter à la couche DirectX sous Windows et appliquer le rendu nécessaire à travers ces interfaces. Ils vont ensuite simuler d'ajouter les mouvements du casque en les traitant comme ceux d'une souris classique.

Même si l'effet d'immersion est présent, le rendu n'est parfois pas à la hauteur de ce qui peut se voir avec un jeu intégrant directement le SDK. Cette différence est comparable à celle d'un film tourné avec une caméra 3D et un film tourné en 2D et mis en 3D durant la post production. Pourtant, il est tout à fait plaisant de pouvoir utiliser l'Oculus Rift avec des jeux tels que Skyrim, Battlefield ou GTA. Plusieurs projets tels que VorpX ou VIREio utilisent ce procédé. En revanche la liste des jeux compatibles n'est pas très grande, mais on retrouve les principaux jeux de ces dernières années.

AVENIR DU CASQUE

Durant le CES un nouveau prototype a été présenté, intégrant un écran avec une latence très faible (-1 milliseconde), ainsi qu'un système enregistrant les mouvements de la tête de façon encore plus précise. Malgré une énorme avancée technique, les dirigeants d'Oculus VR ne veulent pas s'aventurer à donner une date de sortie. Pourtant, ils évoquent le fait qu'une version autonome fonctionnant sous Android serait envisageable.

 Vincent Munoz | Développeur
vmunoz@infinitesquare.com
Infinite Square



Communiquer avec un GPS Bluetooth sous Windows Phone 8

A l'époque de Windows Mobile, communiquer avec un périphérique en utilisant une liaison série était quelque chose de très simple à mettre en œuvre. Le système d'exploitation intégrait tout ce qu'il fallait pour ouvrir un port série et permettre l'envoi et la réception de données par ce canal. Simple, ouvert, pratique.

Dans les premières versions de l'OS mobile de Microsoft, du temps de Pocket PC 2000 sous Windows CE 3 par exemple, la liaison série était physiquement réelle. On branchait un câble qui reliait le PDA avec le périphérique, comme on peut le voir dans cet exemple de [borne interactive animée par un Pocket PC](#) que j'ai mis en ligne en 2001. Le programme était réalisé en eVB (ça ne me rajeunit pas cette histoire).

Ensuite, avec la démocratisation du Bluetooth, les machines ont été en mesure de créer des liaisons séries virtuelles, le câble ayant été remplacé par les ondes radio. Classiquement, à cette époque, on se connectait avec des GPS Bluetooth car les machines sous Windows Mobile n'intégraient pas encore de puce GPS, il fallait donc passer par des appareils externes pour être en mesure de proposer une géolocalisation et de la cartographie.

A l'époque j'avais d'ailleurs écrit [un article sur la manière de se connecter à un GPS](#) et être en mesure de décoder les trames NMEA qu'il fournissait périodiquement et pour les TechDays 2008 j'avais présenté une [application permettant de prendre des photos géo-localisées](#) avec la possibilité de les afficher sur une carte. A l'époque ça en a impressionné plus d'un, maintenant c'est d'une grande banalité je vous l'accorde.

Pour moi la sortie de Windows Phone fût un grand choc. Belle plateforme, belles et puissantes machines, fluidité, design moderne, en gros que du bonheur par rapport à la plateforme Windows Mobile. Mais en creusant un peu, quel retour en arrière sur certains points ! Certes les machines intégraient nativement une puce GPS et le SDK permettait d'en tirer parti, mais impossible par exemple d'ouvrir une liaison série (réelle ou virtuelle), impossible de créer un socket serveur, impossible de faire plein de choses qui étaient devenues si naturelles avec l'ancienne plateforme. Heureusement avec l'arrivée de Windows Phone 7.5, puis Windows Phone 8, une grande partie du retard a été comblée (ouf). Dans certains cas il est indispensable d'avoir plusieurs GPS en même temps. Par exemple, si vous faites du GPS différentiel, vous ne pouvez vous contenter d'utiliser celui inclus dans le device; vous devez collecter les données de plusieurs GPS et effectuer des traitements en temps réel pour arriver à une précision de positionnement par exemple de l'ordre du cm. De plus vous ne pouvez-vous contenter de récupérer simplement la position (latitude, longitude), vous devez aussi récupérer des données très importantes sur le niveau de précision, le nombre de satellites en vue et j'en passe. Je ne vais pas entrer dans le détail du GPS différentiel; ce n'est pas le but de cet article. Donc avec Windows Phone, si vous voulez faire du GPS différentiel vous êtes cuit. Sauf depuis la sortie de Windows Phone 8, qui a ouvert encore un peu plus le périmètre des libertés anciennes (Windows Mobile) retrouvées.

Avec Windows Phone 8 vous avez la possibilité d'ouvrir un canal de communication entre votre application et un périphérique Bluetooth. Donc pour faire simple, on peut communiquer très facilement en Bluetooth avec

par exemple... un GPS, et même plusieurs GPS en même temps !

Je vais donc, dans cet article, vous montrer comment dialoguer avec un GPS en Bluetooth et vous allez voir que c'est assez simple à faire.

Le petit programme « GpsBT » que j'ai réalisé pour l'occasion permet de faire 3 choses :

- ▷ Lister les appareils liés en Bluetooth
- ▷ Sélectionner un appareil et s'y connecter
- ▷ Afficher les trames NMEA envoyées par le GPS

1/ Les droits

La 1ère chose à faire, c'est de donner à votre application les droits pour utiliser les fonctions Bluetooth :

- ▷ ID_CAP_PROXIMITY
- ▷ ID_CAP_NETWORKING

2/ Lister les appareils liés

Il faut partir à la découverte des appareils liés. Ces appareils sont ceux qui sont listés dans l'écran des settings du Bluetooth et qui ont été liés à votre téléphone par une opération d'appairage. On ne peut se connecter qu'avec un appareil qui a été au préalable lié et impossible dans l'état actuel du SDK de provoquer par programme ce lien.

```
private async void BouPeerDiscovery_Click(object sender,
RoutedEventArgs e)
{
    BouPeerDiscovery.IsEnabled = false;
    PeerFinder.AlternateIdentities[«Bluetooth:Paired»] = «»;
    try
    {
        var wPairedDevices = await PeerFinder.FindAllPeersAsync();
        this.ViewModel =
            new MainPageViewModel(wPairedDevices.ToList<PeerInformation>());
        BouPeerDiscovery.Visibility = System.Windows.Visibility.Collapsed;
    }
    catch (Exception ex)
    {
        PeerFinder.Stop();
        MessageBox.Show(ex.Message, «Peer Discovery», MessageBoxButton.OK);
        BouPeerDiscovery.IsEnabled = true;
    }
}
```

La liste des appareils liés est retornée par la fonction `PeerFinder.FindAllPeerAsync`. Dans mon exemple, la liste est fournie au ViewModel de la page pour un affichage dans une liste comme dans l'écran suivant : **Fig.1**.

Le GPS que j'utilise est le Holux GR-230, c'est donc à lui que je vais essayer de me connecter. Il ne faut pas oublier au préalable de le mettre en route, bien sûr.

3/ Connexion à un appareil

La fonction permettant de lister les appareils disponibles retourne pour chacun d'entre eux un objet de type `PeerInformation`. Parmi les informations disponibles, on a le `DisplayName` qui est le nom en clair (celui que j'utilise dans la liste), et le `HostName` qui sera utilisé pour effectuer la connexion.

La connexion est réalisée via un objet de type `StreamSocket` en utilisant sa méthode asynchrone `ConnectAsync` avec en premier paramètre le `HostName` provenant du `PeerInformation` de l'appareil auquel on veut se connecter et « 1 » en second paramètre (pour le moment, je ne sais pas vraiment à quoi correspond ce second paramètre, dans tous les exemples de la documentation la valeur utilisée est la chaîne « 1 » alors va pour ça).

```
public async Task<Exception> ConnectAsync(PeerInformation wPeer)
{
    // Si un précédent device était déjà connecté, on débranche

    if (_StreamSocket != null)
    {
        _StreamSocket.Dispose();
        _StreamSocket = null;
    }

    // On essaye de se connecter au device

    _StreamSocket = new StreamSocket();

    try
    {
        await _StreamSocket.ConnectAsync(wPeer.HostName, <<1>>);

        // Pas d'erreur ?
        // Alors c'est qu'on est connecté
        // On peut mettre en place le scan des données qui arrivent
        // du device

        _CurrentGpsInput = <<>;
        this.DeviceName = wPeer.DisplayName;
        ReadGps();
    }
    catch (Exception ex)
    {
        // Erreur pendant la connexion

        _StreamSocket = null;
        return ex;
    }

    return null;
}
```

La fonction de connexion, placée dans le ViewModel de la page, retourne l'erreur rencontrée en cas de problème, ou `null` si tout s'est bien passé. Si la connexion se passe bien, on passe en mode pooling dans lequel on va lire le contenu du buffer d'entrée du `StreamSocket` **Fig.2**.

Pour lire les données en provenance du `StreamSocket` il faut passer par un `DataReader` à qui on donne l'`InputStream` utilisé par le `StreamSocket` pour collecter les données entrantes. Il faut aussi le paramétriser pour qu'il effectue une lecture partielle des données. Ainsi, si on demande de lire 100 octets et qu'il n'y en a que 50 dans le buffer d'entrée, le `DataReader` va lire les 50 octets et rendre la main au lieu d'attendre la fin des rats que les 100 octets soient effectivement disponibles.

La lecture en elle-même est très simple. Il suffit de dire combien on veut lire d'octets maximum et on reprend la main quand on a lu au maximum ce nombre d'octets. Simple et rapide. La fonction de lecture étant asynchrone à la base, on n'oublie pas pour l'attendre de placer le petit `await` qui va bien devant. La fonction retourne le nombre d'octets effectivement lus.

Si elle retourne zéro c'est que rien n'était disponible dans le buffer d'entrée. Suivant les matériels connectés, ça peut vouloir dire qu'il ne cause plus pour le moment (il boude par exemple) ou alors qu'il n'est plus connecté. Dans le cas d'un GPS, qui est je vous le rappelle un appareil très bavard, le fait de ne plus recevoir de données signifie souvent qu'il n'est plus là.

Idéalement, il faudrait gérer un timeout. En effet, comme expliqué un peu plus haut, quand le périphérique est déconnecté, il cesse d'envoyer des données, mais aucune erreur n'est levée au moment où l'on essaye de notre côté de lire de nouvelles données; on n'a seulement comme indice le fait que la fonction de lecture du `DataReader` retourne zéro.

Au bout d'un certain temps, il sera donc nécessaire d'en conclure que la liaison est rompue. La durée de ce timeout dépendra du comportement « normal » du périphérique auquel vous êtes connecté. Par exemple, pour le GPS que j'ai utilisé, une absence de données pendant plus d'une seconde n'est pas normale, le timeout sera donc compris en 1000 et 1500 millisecondes. Dans les sources de « GpsBT » fournies avec cet article, le

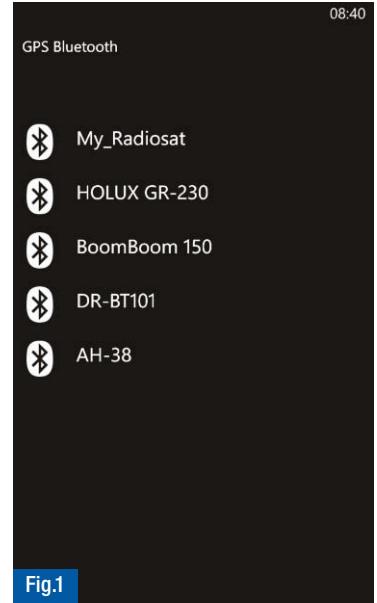


Fig.1

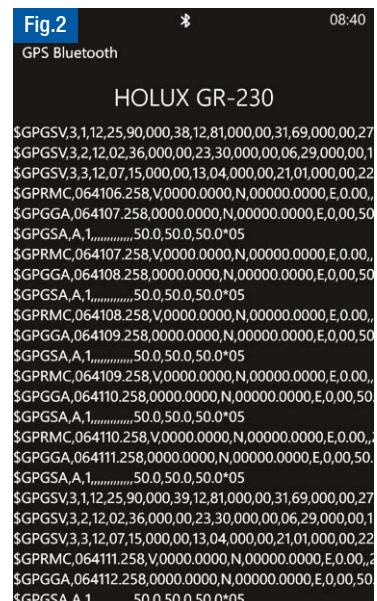


Fig.2

The screenshot shows raw GPS data from the HOLUX GR-230 device. The data starts with SGPGSV3,1,12,25,90,00,38,12,81,000,00,31,69,000,00,27 and continues with several SGPGRMC, SGPGLL, SGPDTG, and SGPVIA messages. The title bar shows 'GPS Bluetooth' and the time '08:40'.

timeout est géré, vous pourrez regarder dans le code comment il est implémenté.

Une fois qu'on a des données disponibles dans le buffer d'entrée, il suffit d'aller les récupérer de la manière qui convient le mieux en utilisant le **DataReader** et ses fonctions de récupération (**ReadByte**, **ReadBytes**, **ReadDouble**, etc...). C'est lui qui s'occupe d'aller puiser le bon nombre d'octets dans le buffer d'entrée pour récupérer les informations dans le format que vous souhaitez.

Dans le cas du GPS nous devons récupérer les données en ASCII. Malheureusement la méthode **ReadString** ne convient pas car elle lit la chaîne en UTF8 ou en Unicode, mais ne sait pas le faire en ASCII. En fait le format ASCII a été retiré des formats proposés dans **System.Text.Encoding**. Quelle idée ?

Donc nous allons récupérer les octets comme si c'étaient de simples caractères ASCII et construire une chaîne ASCII, à la mano, comme le faisait mon grand-père. L'opération est renouvelée tant que le **StreamSocket** n'est pas déconnecté.

```
private async void ReadGps()
{
    DataReader wReader = new DataReader(_StreamSocket.InputStream);
    wReader.InputStreamOptions = InputStreamOptions.Partial;

    while (_StreamSocket != null)
    {
        // Lecture des données

        uint r = await wReader.LoadAsync(300);

        if (r > 0)
        {
            // On a des données

            // On prépare le buffer de réception

            byte[] wBuffer = new byte[r];

            // On récupère les données

            wReader.ReadBytes(wBuffer);

            // On transforme ces données en string

            string wString = <>;

            for (int i = 0; i < r; i++)
            {
                wString += (char)wBuffer[i];
            }

            // On ajoute ces données aux trames générales reçues du GPS

            _CurrentGpsInput += wString;

            // On essaye de découper la trame générale en lignes NMEA
        }
    }
}
```

```
do
{
    // On recherche les caractères CRLF

    p = _CurrentGpsInput.IndexOf(<<\r\n>>);

    if (p > -1)
    {
        // On récupère le début (avant les caractères CRLF)

        string wDebut = _CurrentGpsInput.Substring(0, p);

        // On enlève cette partie de la trame générale

        _CurrentGpsInput = _CurrentGpsInput.Substring(p + 2);

        // Généralement une trame NMEA valide commence par $GP

        if (wDebut.StartsWith(<<$GP>>))
        {
            // On ajoute la partie à la liste des trames NMEA

            this.GpsNmea.Add(wDebut);

            // On ne garde que les 25 dernières trames NMEA
            // dans la liste

            while (GpsNmea.Count > 25) { GpsNmea.RemoveAt(0); }

        }
    }
}
```

Les nouvelles données lues sont ajoutées à un « buffer » général qui est ensuite découpé en trames se terminant par les caractères CRLF. Une trame valide commence par les caractères « \$GP ». Si c'est le cas, la trame est extraite du buffer général et ajoutée à la liste des trames qui s'affichent à l'écran.

Pour utiliser vraiment le GPS, il faudrait traiter les trames reçues (\$GPRMC, \$GPGGA, etc...) pour en extraire les informations telles que la latitude et la longitude, l'altitude, la précision, etc. Ça, je vous laisse faire, ce n'est que du traitement basique de chaînes de caractères.

Ensuite, on reprend la lecture des données suivantes, et la boucle est bouclée. Nous avons réalisé toutes les opérations pour se connecter et communiquer avec un GPS Bluetooth depuis un device Windows Phone 8.

Comme vous pouvez le constater, c'est très simple à mettre en œuvre. Je vous laisse décortiquer les sources de « GpsBT » pour bien comprendre comment tout cela fonctionne.

 Stéphane Sibue
CTO chez SOFTELITE
stephane.sibue@softelite.fr
Développeur iOS, Android et Windows Phone
Blog : www.coding4phone.com

API : une mise en œuvre

L'objectif de cet article est de présenter la démarche à mettre en œuvre pour la mise en place d'une Web API. Nous nous placerons selon deux points de vue; dans un premier temps la partie serveur : exposition d'une Web API, ensuite nous verrons la partie cliente : consommation de la Web API.

Nous allons traiter un problème assez courant dans les systèmes d'information (SI) actuels : la rationalisation des référentiels de données.

Il n'est pas rare de voir une information stockée à plusieurs endroits dans le SI. Cette information est interprétée différemment selon des règles métiers qui peuvent différer selon l'endroit où la donnée est traitée.

Le fait de disperser l'information oblige, dans certains cas, à devoir effectuer des synchronisations entre les "référentiels". Le problème étant qu'il n'est pas possible d'identifier une source qui fera office de référence officielle pour l'ensemble du SI. Pour notre exemple, nous allons prendre le cas de la société **ContactMania** qui, dans le cadre de son activité, a un besoin de gérer des contacts. Cette société n'échappe pas à la règle et sa gestion des contacts est disséminée dans tout son SI. Comme nous l'évoquions un peu plus tôt, la notion de contact est un élément central dans le métier de cette entreprise. Fort de cette prise de conscience, le service informatique décide de mettre en place un référentiel unique de gestion des contacts. Les contraintes posées sont les suivantes :

- Ce référentiel doit être consommable par tous les systèmes du SI,
- La consommation du service doit être indépendante d'un langage spécifique,
- Le point d'entrée doit respecter une interface standardisée,
- Les systèmes seront forcément distants.

Dans cette démarche, un composant métier de gestion des contacts a été développé par l'équipe informatique. Dans le cadre de cet article, nous traiterons ce composant comme une boîte noire. Il a la charge de fournir des fonctionnalités de manipulation des contacts jusqu'à la persistance de ceux-ci. Nous nous concentrerons sur notre objectif, c'est-à-dire sur la mise en place d'une Web API. L'API que nous allons mettre en place exposera les fonctionnalités suivantes :

- Lister l'ensemble des contacts,
- Créer un contact,
- Éditer un contact,
- Supprimer un contact,
- Recherche de(s) contact(s) selon des critères.

Selon les contraintes imposées, l'api sera exposée via une architecture de type REST.

CHOIX DES OUTILS

Pour mettre en place cette API, les choix se sont arrêtés sur les outils suivants :

PlayFramework

Play framework est un framework '*nouvelle génération*'. Je vous propose de l'utiliser par rapport aux concepts mis en avant par l'outil. En effet ce framework a comme leitmotiv de respecter à la lettre les principes architecturaux du Web (architecture stateless, ressource unique, verbe HTTP, etc.). De plus, le serveur propose, entre autres, une gestion des I/O non bloquante ce qui, pour des problématiques de scalabilité, est un vrai atout.

Json

JSON pour JavaScript Object Notation. Json est une notation couramment utilisée au sein de JavaScript. Il permet de représenter de l'information structurée comme le permet XML. Une spécification le décrit précisément : RFC 4627. L'utilisation de Json est intéressante dans notre contexte, car il est devenu un standard. Il existe des librairies dans de

nombreux langages et sur de nombreuses plateformes permettant de parser et d'encoder au format Json. Sa syntaxe est assez simple à écrire et à lire. On trouve également pas mal d'outils de validation de syntaxe d'un flux Json sur le Web.

Heroku

Heroku est une plateforme hébergée dans le Cloud qui permet de déployer automatiquement une application respectant un certain format. Il existe plusieurs licences d'utilisation, dont une gratuite.

La plateforme supporte plusieurs langages ou frameworks comme Ruby, Java, node.js, Python, Scala et PlayFramework (vous trouverez une liste exhaustive en suivant ce lien : <https://devcenter.heroku.com/categories/language-support>). Pour notre besoin, cette plateforme nous permet de rapidement voir le résultat sans se soucier de tous les problèmes de configuration et d'infrastructure.

Git

Nous n'allons pas dans notre étude créer un repo pour gérer différentes versions de notre application de gestion de contact. Nous avons besoin de Git pour faire les déploiements de l'application sur la plateforme Heroku. Pour ceux qui n'auraient pas encore franchi le pas sur l'utilisation de Git comme seconde langue maternelle, je vous rassure, les commandes à utiliser sont relativement simples et ne nécessitent pas une connaissance poussée de l'outil.

INSTALLATION

Passons aux choses sérieuses. Dans un premier temps, nous allons nous assurer que l'environnement technique est correctement installé et configuré. Pour commencer, il faut avoir une version 1.6 de Java installée. Vous trouverez une procédure d'installation pour votre environnement en suivant le lien suivant :

<http://www.oracle.com/technetwork/java/javase/index-137561.html>

Ensuite, installons le framework Play ! Pour faire l'installation, je vous renvoie à la procédure d'installation décrite sur le site de l'outil :

<http://www.playframework.com/documentation/2.2.x/Home>

Enfin nous allons installer Git pour pouvoir effectuer le déploiement à terme. Pour effectuer l'installation, je vous renvoie à la procédure d'installation décrite ici :

<http://git-scm.com/book/fr/D%C3%A9marrage-rapide-Installation-de-Git>

Une fois terminée l'installation de Git, votre environnement est prêt.

CRÉATION DU PROJET

Une ultime étape reste à faire avant de passer à la partie "coding". Il faut créer le projet. La création d'un projet Play se fait en ligne de commande. Nous allons créer un projet nommé : **contactManagerAPI**.

Ouvrez une console et placez-vous dans le répertoire qui accueillera le répertoire du futur projet. Ensuite taper la commande suivante :

```
play new contactManagerAPI
```

En validant cette commande, Play va vous poser 2 questions :

- What is the application name? [contactManagerAPI] -> appuyer sur [entrer]

► Which template do you want to use for this new application? -> saisissez 2, pour travailler à partir d'un template en Java

Après validation de la dernière étape, vous devriez obtenir le message suivant :

```
OK, application contactManagerAPI is created.
```

```
Have fun!
```

Ensuite placez-vous dans le répertoire qui a été créé :

```
cd contactManagerAPI
```

Voilà vous êtes maintenant prêt à lancer votre serveur Play; pour cela, tapez la commande suivante :

```
play run
```

Le serveur se lancera, vous pourrez voir quelques lignes indiquant que SBT résout des dépendances et enfin le message suivant d'affichera :

— (Running the application from SBT, auto-reloading is enabled) —

[info] play - Listening for HTTP on /0:0:0:0:0:0:0:0:9000

(Server started, use Ctrl+D to stop and go back to the console...)

votre serveur est démarré, ouvrez votre navigateur et allez à l'adresse suivante : <http://localhost:9000>

Vous devriez obtenir une page aux couleurs de playframework.

La dernière étape concerne la configuration du projet afin que celui-ci soit compatible avec votre IDE favori. Je vous renvoie à la documentation de playframework pour configurer votre IDE :

<http://www.playframework.com/documentation/2.2.x/IDE>

Avant de terminer avec la configuration de l'environnement, il faut ajouter un fichier de config destiné à l'outil de build SBT (sur lequel Play se base pour gérer, entre autres, les dépendances du projet). Dans le cadre du projet, nous utiliserons la librairie Guava.

```
name := «ContactManager»  
  
version := «1.0-SNAPSHOT»  
  
libraryDependencies ++= Seq(  
  «com.google.guava» % «guava» % «15.0»  
)
```

Et collez-le dans un fichier nommé **build.sbt**, que vous aurez créé préalablement à la racine du projet. Une fois créé, taper la séquence de commandes suivantes :

```
play  
reload  
update  
idea [ou eclipse] selon votre IDE.
```

Vous devrez redémarrer votre IDE pour que les modifications soient prises en compte.

IMPLÉMENTATION

Nous allons commencer par implémenter la partie serveur. Je vous rappelle brièvement le contexte, la société **ContactMania** souhaite exposer une API pour centraliser la gestion de ses contacts. Actuellement, elle possède un composant qui est capable de gérer des contacts. Nous utiliserons ce composant comme une boîte noire. Au travers de l'API, nous exposerons ses services. Voici l'interface de ce composant :

```
public interface ContactManager {  
  
  Contact loadContact(final Long id);
```

```
Collection<Contact> loadAllContacts();  
  
void saveContact(final Contact contact);  
  
void deleteContact(final Long id);  
  
void updateContact(final Contact contact);  
  
Collection<Contact> findContactByName(final String name);  
  
Collection<Contact> findContactPhoneNumber(final String phone  
Number);  
  
}  
  
Ainsi que l'implémentation :  
  
public class ContactManagerImpl implements ContactManager {  
  
  final Map<Long, Contact> bd = new HashMap();  
  
  public ContactManagerImpl() {  
    bd.put(42L, new Contact(«Fabrice», «Sznajderman», «6 rue  
des maisons», «0102030405»));  
    bd.put(53L, new Contact(«Géraldine», «Dupond», «4 rue  
des fleurs», «0503020104»));  
    bd.put(12L, new Contact(«Francois», «Duprin», «Bâtiment  
5 résidence chambrey», «»));  
    bd.put(70L, new Contact(«Loic», «Martin», «Dijon centre»,  
«1002304030»));  
  }  
  
  @Override  
  public Contact loadContact(Long id) {  
  
    saveContact(null);  
    return bd.get(id);  
  }  
  
  @Override  
  public Collection<Contact> loadAllContacts() {  
    return bd.values();  
  }  
  
  @Override  
  public void saveContact(Contact contact) {  
    final Collection<Long> c = Sets.newTreeSet(bd.keySet()).  
descendingSet();  
    bd.put((c.iterator().next()) + 1, contact);  
  }  
  
  @Override  
  public void deleteContact(Long id) {  
  
    bd.remove(id);  
  }
```

```

@Override
public void updateContact(final Long id, Contact contact) {
    bd.put(id, contact);
}

@Override
public Collection<Contact> findContactByName(final String name) {

    return Collections2.filter(bd.values(), new Predicate<Contact>() {
        @Override
        public boolean apply(@Nullable Contact contact) {

            return name.equalsIgnoreCase(contact.getFirstName());
        }
    });
}

@Override
public Collection<Contact> findContactPhoneNumber(final String
phoneNumber) {
    return Collections2.filter(bd.values(), new Predicate<Contact>() {
        @Override
        public boolean apply(@Nullable Contact contact) {
            return phoneNumber.equalsIgnoreCase(contact.getNumTel());
        }
    });
}

```

L'implémentation a pour objectif de fournir un composant capable de répondre à des demandes précises sans aucune dépendance externe (bdd ou autres). *Veiller à récupérer cette implémentation avec son interface au sein de votre projet.*

PARTIE SERVEUR

L'API que nous allons exposer se basera sur une Architecture de type REST. Nous allons devoir mettre en place plusieurs choses au sein de notre application. Voici la liste des tâches que nous allons effectuer :

- ▷ Définition des URIs exposées par l'API,
- ▷ Mapping des URI sur le composant contactManagerAPI,
- ▷ Déploiement de l'application sur la plateforme Heroku,

Construction des routes

Nous allons mettre en place une liste d'URIs représentant les différentes fonctionnalités mises à disposition par l'API. Avant d'écrire concrètement les URIs à mettre en place, nous allons voir comment elles se composent :

<nom du schéma> : <partie hiérarchique> [? <requête>]

- ▷ **Nom du schéma** : correspond au protocole de communication, c'est-à-dire HTTP. Dans la définition générale des URIs le scope du schéma est beaucoup plus large.
- ▷ **Partie hiérarchique** : correspond à la désignation précise de la ressource visée. Cette partie se décompose en 2 parties : le domaine et le chemin

- **le domaine** : correspond au nom de l'hôte où est hébergée la ressource
- **le chemin** : littéralement correspond à l'endroit où se trouve la ressource dans le domaine.

- ▷ **Requête** : cette partie correspond à un ensemble de paramètres complémentaires sous forme de clef=valeur. La requête n'a aucune influence sur la partie précédente.

Voici un exemple : <http://fr.wikipedia.org/wiki/REST>

Après ce rappel, nous allons créer les URIs nécessaires à notre API. Dans le titre du paragraphe, j'ai employé le terme de "route". Il correspond au terme couramment employé dans le cadre de playFramework pour désigner les différentes URIs exposées.

Pour saisir les routes au sein de Play, il faut éditer le fichier

```
./contactManagerAPI/conf/routes
```

Nous allons avoir besoin d'une URI pour récupérer un contact à partir de son identifiant (id). Avant de décrire le domaine et la partie hiérarchique, il faut indiquer le méthode HTTP avec laquelle la requête sera envoyée à l'API. Pour ce premier cas, nous faisons une action de lecture à partir de la ressource, donc nous utiliserons le verbe GET.

À partir d'une nouvelle ligne, commencer par indiquer le verbe GET ensuite nous allons renseigner la partie hiérarchique, c'est-à-dire le chemin précis permettant d'accéder à la ressource (en l'occurrence un contact).

Le chemin doit exprimer clairement les intentions :

```
GET /load/contact/:id
```

Voilà notre première route. Comment doit-on l'interpréter? Lorsqu'une requête arrive sur le serveur de type GET et la partie hiérarchique matche avec celle décrite dans la route, alors on mappera cet appel sur le traitement correspondant. Dans la route que nous venons de décrire, en dernière position, on trouve la chaîne suivante :

```
... :id
```

Cette chaîne est une partie dynamique. L'id prendra une valeur différente (définie par l'émetteur de la requête).

Par exemple, si je veux récupérer des informations sur le contact ayant l'id 42, alors l'URI ressemblera à ceci : <http://localhost:8080/load/contact/42>

Dans cet exemple on voit bien que l'URI pointe sur une ressource de manière unique.

Voyons maintenant les autres fonctionnalités :

- ▷ Lister l'ensemble des contacts :

- GET /load/contact/all

- ▷ Éditer un contact :

- GET /load/contact/:id

- ▷ Supprimer un contact :

- DELETE /del/contact/:id

- ▷ Créer un contact :

- POST /add/new/contact/

- ▷ Rechercher un contact par son nom :

- GET /find/contact/name/:name

- ▷ Rechercher un contact par son numéro de téléphone :

- GET /find/contact/phone/:phoneNumber

Ci-dessus l'ensemble des URIs constituant notre API. Chaque route correspond à un point d'entrée sur l'API.

Point à noter : Il faut utiliser avec précaution le verbe DELETE, celui-ci n'est pas supporté par certains clients de type navigateur Web.

Aussi, le passage des paramètres dans le cadre de la création d'un nouveau contact se fait via le header de la requête HTTP de type POST.

Pour le moment la définition du fichier de configuration de routes est incomplète. En effet, il faut associer à chaque URI un composant qui effectuera le traitement attendu sur la ressource.

Mapping des contrôleurs

Nous allons créer maintenant les contrôleurs qui seront mappés aux URIs précédentes. Pour cela nous allons nous placer dans le répertoire dédié aux contrôleurs

```
./contactManagerAPI/controllers
```

A cet endroit on trouvera une classe, nommée Application, qui fait office de controller par défaut. Nous allons utiliser ce contrôleur en ajoutant des méthodes, chaque méthode sera mappée sur une URI.

Si l'on regarde d'un peu plus près la classe Application, on peut voir que celle-ci hérite de la classe contrôleur. Cette classe mère fournit un contexte technique pour gérer le flux des requêtes.

Nous avons créé une instance de ContactManager :

```
private final static ContactManager cm = new ContactManagerImpl();
```

Son scope est global à la classe. Nous allons maintenant créer un contrôleur pour le premier cas, le chargement d'un contact à partir de son identifiant. Chaque contrôleur prendra la forme d'une méthode dans la classe Application.

```
public static Result loadContact(final Long id) {

    final Contact c = cm.loadContact(id); //1

    if (c != null) {
        return ok(Json.toJson(c)); // 2
    }
    return notFound("No contact found"); // 3
}
```

Bien que le code soit relativement clair, voilà une description du comportement de celui-ci :

- **Ligne 1** : à partir de l'id récupérée depuis la requête (souvenez-vous du :id), on interroge le composant métier ContactManager,
- **Ligne 2** : si un contact a été trouvé pour l'identifiant donné, alors on renvoie, au format Json, le contact qui a été trouvé,
- **Ligne 3** : sinon on indique au client de l'API (par une erreur 404) que la ressource demandée n'a pas été trouvée.

Maintenant que le contrôleur a été créé pour le premier service, nous allons le mapper à l'URI correspondant, pour cela nous reprenons le fichier **routes**. et nous complétons la ligne comme suit :

```
GET /load/contact/:id controllers.Application.loadContact(id:Long)
```

L'élément que nous avons ajouté désigne précisément la méthode ainsi que le type du paramètre attendu par ce contrôleur. Une fois ajouté votre fichier de configuration routes devrait ressembler à ceci :

```
# Routes
# This file defines all application routes (Higher priority
routes first)
# ~~~~

# Home page
GET / controllers.Application.index()

# Map static resources from the /public folder to the
/assets URL path
```

```
GET /assets/*file
controllers.Assets.at(path=>/public», file)
```

```
GET /load/contact/:id
controllers.Application.loadContact(id:Long)
```

Nous allons pouvoir tester le résultat immédiatement. Pour cela, il faut démarrer le serveur Play. Si votre serveur est toujours démarré suite au test que nous avons fait tout à l'heure, redémarrez-le (ctrl + c puis play run). Une fois redémarré, ouvrez votre navigateur et tapez l'url suivante : <http://localhost:9000/load/contact/42>

Vous devriez obtenir le résultat suivant :

```
{«firstName»:>»Fabrice»,»lastName»:>»Sznajderman»,»adresse»:>»6 rue des maisons»,»numTel»:>»0102030405»}
```

Voilà le premier service de notre API est accessible via une architecture REST. Ci-dessous vous trouverez l'implémentation complète de tous les contrôleurs :

```
public class Application extends Controller {

    private final static ContactManager cm = new ContactManager
    Impl();

    /*
    Default
    */
    public static Result index() {
        return ok(index.render("Your new application is ready."));
    }

    public static Result loadContact(final Long id) {

        final Contact c = cm.loadContact(id);

        if (c != null) {
            return ok(Json.toJson(c));
        }
        return notFound("No contact found");
    }

    public static Result all() {
        return ok(Json.toJson(cm.loadAllContacts()));
    }

    public static Result deleteContact(final Long id) {
        cm.deleteContact(id);
        return ok("Contact has been deleted!");
    }

    public static Result saveContact() {

        final DynamicForm rdata = Form.form().bindFromRequest();
        final String firstName = rdata.get("firstName");
        final String lastName = rdata.get("lastName");
        final String adresse = rdata.get("adresse");
        final String phone = rdata.get("phone");
        final Contact newContact = new Contact(firstName, lastName,
        adresse, phone);
    }
}
```

```

        cm.saveContact(newContact);
        return ok(<>Contact saved!>);
    }

    public static Result findByName(final String name) {
        return ok(Json.toJson(cm.findContactByName(name)));
    }

    public static Result findByPhone(final String phone) {
        return ok(Json.toJson(cm.findContactByPhoneNumber(phone)));
    }
}

```

Maintenant que nous avons tous les contrôleurs, il faut les mapper sur les routes que nous avions définies précédemment. Le fichier routes devrait ressembler à cela : voir le code sur le site www.programmez.com. On peut remarquer que pour le service de création d'un contact aucun paramètre n'est déclaré. En effet, le framework Play ira chercher les informations directement dans les headers de la requête (cf contrôleur `saveContact()`). Vous pouvez dès à présent lancer votre serveur et tester les différents services. Actuellement les tests que vous pourrez faire seront en local.

Déploiement de l'API (Heroku)

Pour exposer notre gestion de contact sur Internet (oui la société est ambitieuse ;)) nous allons déployer notre code sur service Heroku. L'avantage de cette plateforme est la prise en charge de manière automatique de certains outils comme le framework Play.

C'est-à-dire qu'il suffit juste d'envoyer notre code sur le compte créé et Heroku se charge de tout pour déployer l'application et la rendre accessible.

La communication avec le service se fait exclusivement via Git. C'est-à-dire que pour déployer une nouvelle version, il faut *commiter* son code localement et faire un push vers Heroku. Ensuite la magie opère!

Au préalable, il faudra naturellement créer un compte sur la plateforme :

<https://www.heroku.com/>

Une fois le compte créé, il faut créer une application. Lorsque l'application est créée, Heroku nous fournit deux Urls :

► Url d'accès à l'application

► Url Git d'accès au repo sur lequel il faudra pousser notre code pour le déploiement.

La première chose que nous ayons à faire ensuite est de mettre notre projet sous Git. Pour cela il faut se placer dans le répertoire racine du projet et taper la commande suivante :

```
git init
```

Ensuite d'effectuer un premier commit, pour que tous les fichiers soient reconnus et versionnés :

```
git add .
git commit -m "initiale commit"
```

(nous utilisons la commande `add` avec comme paramètre un '.', ce n'est pas forcément une bonne pratique de faire comme ça. Dans notre cas cela simplifie la tâche pour les personnes ne maîtrisant pas Git).

Maintenant notre projet est versionné avec Git, il est donc prêt à être déployé. Avant de pouvoir pousser nos sources sur la plateforme Heroku, il faut ajouter la référence (cf l'url Git fournie au moment de la création de l'application). Pour ajouter la référence, il faut taper la commande suivante :

```
git remote add heroku git@heroku.com:contactmanagerapi.git
```

Vous pouvez vérifier que la référence a été bien ajoutée en utilisant la commande suivante :

```
git remote -v
```

Ce qui donnera le résultat suivant :

```
heroku git@heroku.com:contactmanagerapi.git (fetch)
heroku git@heroku.com:contactmanagerapi.git (push)
```

Maintenant, nous sommes fin prêts pour déployer. Il faut saisir la commande suivante :

```
git push heroku master
```

Une fois la commande validée, on peut suivre dans la console toutes les étapes de déploiement en live (le temps de déploiement peut prendre quelques minutes). Une fois le déploiement terminé, vous pouvez accéder à l'API en utilisant l'URL que vous fournit Heroku au moment de la création du compte. Vous pouvez tester celle que j'ai déployée dans le cadre de cet article : <http://contactmanagerapi.herokuapp.com/load/contact/all> <http://contactmanagerapi.herokuapp.com/load/contact/42> etc.

Toutes les fonctionnalités sont opérationnelles.

PARTIE CLIENTE

Dans cette partie nous allons mettre en place 3 clients différents qui viendront consommer notre API. Le premier client sera écrit en Java, le deuxième sera écrit en JavaScript (dans une page Web) et, enfin, le dernier sera consommé par une requête initiée avec l'outil CURL.

L'objectif de cette partie est de montrer différentes façons de consommer une API basée sur une architecture de type REST. On notera qu'avec ce type d'architecture aucune contrainte n'est posée concernant la technologie utilisée côté client. J'aurais pu présenter un exemple développé en .NET ou encore en Haskell ou Scala, pourvu que le langage utilisé propose des composants permettant de faire des requêtes HTTP.

Client Java

Vous trouverez ci-dessous une version du client écrit en Java. Pour cette implémentation nous avons recours aux classes natives du JDK 1.6. Aucune dépendance n'est requise. Par conséquent, vous pouvez prendre le code ci-dessous et le copier dans un fichier (en respectant naturellement les règles du langage Java) et l'exécuter pour obtenir le résultat.

Dans cet exemple, le programme va directement interroger la version de l'API déployée sur Heroku.

```

public class ContactManagerJavaClient {

    public static void main(String[] args) {
        System.out.println(<>Find : << +
        getResult(<http://contactmanagerapi.herokuapp.com/find/contact/name/Fabrice>);

        System.out.println(<>All : << +
        getResult(<http://contactmanagerapi.herokuapp.com/load/contact/all>));
    }

    public static String getResult(String urlToRead) {
        URL url;
        HttpURLConnection conn;
        BufferedReader rd;
        String line;
        String result = <<>;

```

```

try {
    url = new URL(urlToRead);
    conn = (HttpURLConnection) url.openConnection();
    conn.setRequestMethod("GET");
    rd = new BufferedReader(new InputStreamReader(conn.getInputStream()));
    while ((line = rd.readLine()) != null) {
        result += line;
    }
    rd.close();
} catch (IOException e) {
    e.printStackTrace();
} catch (Exception e) {
    e.printStackTrace();
}
return result;
}
}

```

Client JavaScript

Je vous propose maintenant une version du client implémentée en JavaScript. Cette implémentation se trouve au sein d'une page Web.

Ci-dessous le code JavaScript permettant de faire la requête :

```

$(function() {

    $('#query').click(function() {
        var uri = $('#uri').val();
        $.get(uri, function(data) {
            $('#result').html( data.firstName + ' ' + data.lastName );
        });
    });
});

```

Ce code fait référence à des éléments d'une page au sein de laquelle il est exécuté. La page HTML contenant ce code se trouve dans l'application déployée sur Heroku dans le répertoire `view`, dont voici le code source :

```

<HTML>
<head>
    <title>JavaScript Client</title>
    <link rel=>stylesheet media=>screen href=>@routes.Assets.at(<stylesheets/main.css>)>
    <link rel=>shortcut icon type=>image/png href=>@routes.Assets.at(<images/favicon.png>)>
    <script src=>@routes.Assets.at(<javascripts/jquery-1.9.0.min.js>)>
<type=>text/JavaScript></script>
</head>
<body>
    <div id=>query style=>width: 100%;height: 40px;>
        <input type=>text id=>uri value=>/load/contact/42>
    <input type=>button value=>Run id=>query>
</div>
    <div id=>result style=>width: 100%;border: solid 1px #A3A3A3;></div>
    <script>
        $(function() {
            $('#query').click(function() {
                var uri = $('#uri').val();

```

```

                $.get(uri, function(data) {
                    $('#result').html( data.firstName + ' ' + data.lastName );
                });
            });
        });
    </script>
</body>
</HTML>

```

Voici le rendu une fois l'action "run" exécutée :

Client Javascript

Fabrice Sznajderman

En cliquant sur le bouton "run", une requête HTTP GET ajax est déclenchée vers l'api. A cette requête, l'API a répondu le contact correspondant à l'identifiant passé en paramètre dans l'URI.

Client Curl

Curl est un outil principalement disponible sous les systèmes Unix/Linux. Il permet d'effectuer des requêtes HTTP à partir de la ligne de commande. Nous allons voir dans cette section, comment consommer notre API. Dans un premier temps, nous allons charger le fameux contact numéro 42

`curl http://contactmanagerapi.herokuapp.com/load/contact/42`

résultat attendu :

```
{<firstName>:»Fabrice»,<lastName>:»Sznajderman»,<adresse>:>6 rue des maisons»,<numTel>:»0102030405»}
```

Nous allons maintenant créer un nouveau contact :

```
curl -X POST -d <firstName>Michel<&><lastName>Martin<&><adresse>:>adr1<&>numTel:1234567890<&>
http://contactmanagerapi.herokuapp.com/add/new/contact
```

ensuite vous pourrez chercher ce contact par son nom :

`curl http://contactmanagerapi.herokuapp.com/find/contact/name/Michel`

Résultat attendu :

```
[{<firstName>:»Michel»,<lastName>:»Martin»,<adresse>:>adr1,<numTel>:»1234567890»}]
```

CONCLUSION

Nous arrivons au terme de cet article. J'espère qu'il vous aura donné une vision un peu plus claire sur la démarche à suivre pour créer une Web API basée sur une architecture REST. Les outils utilisés sont un choix personnel. Il existe beaucoup d'autres outils (comme Node.js, SpringBoot, HttpServer du JDK, etc...) permettant de réaliser avec autant d'efficacité l'exposition d'une API Web. Le code présenté dans cet article est disponible à partir d'un repository hébergé sur GitHub : <http://goo.gl/KGOZA1>

N'hésitez pas à faire un fork du projet et jouer avec l'implémentation côté serveur ainsi que les différents clients proposés ici. Vous trouverez également l'API déployée et à disposition sur la plateforme Heroku : <http://goo.gl/KT0cGL>

 Fabrice Sznajderman

Senior Java & Scala developer at SFEIR

Maîtriser l'asynchronisme de C# 5.0

Implémentation asynchrone de C# 5.0

3^e partie



Une des nouveautés majeures de C# 5.0 est sans aucun doute l'intégration de l'asynchronisme. Avec cette offre, il devient difficile de produire des applications peu disponibles aux yeux de l'utilisateur. Cependant, l'expérience montre que cette grande nouveauté n'est pas toujours bien comprise et que des erreurs sont parfois difficiles à résoudre lorsque le développeur n'a pas conscience des mécanismes sous-jacents.

L'objectif est de répondre à la question : comment est implémentée cette nouvelle technologie asynchrone ? Cette partie est la suite directe de la 1^{re} partie publiée dans Programmez ! n°168.

ASYNCHRONISME EN C# 5.0

Présentation

L'ensemble des points négatifs évoqués avec les modèles APM, EAP, TAP, évoqués dans l'article précédent, ont poussé les équipes Microsoft à développer une solution asynchrone intégrée au niveau des langages C# 5.0 et VB 11. La motivation était de produire du code 100 % asynchrone, facilement maintenable dans un contexte fortement impératif.

Dans l'exemple suivant, nous réutilisons la classe FileStream qui offre une nouvelle méthode orientée TAP, ReadAsync qui retourne un type Task<int>. Une des conséquences de l'apparition de l'asynchronisme dans les langages C# et VB est qu'une très grande partie des classes du Framework .NET 4.5 déclinent de nouvelles méthodes orientées TAP prenant en charge l'asynchronisme. Par convention, le nom des méthodes doit être post-fixé avec le mot Async.

```
async static public void ReadAsync(string filename)
{
    var buffer = new byte[100];
    using (var fs = new FileStream(filename, FileMode.Open, FileAccess.Read,
        FileShare.Read, 1024, FileOptions.Asynchronous))
    {
        int bytesRead = await fs.ReadAsync(buffer, 0, buffer.Length);
        Console.WriteLine("Number of bytes read={0}", bytesRead);
        Console.WriteLine(BitConverter.ToString(buffer, 0, bytesRead));
    }
}
```

Le code est bien plus simple que la version APM, EAP et TAP décrite dans l'article précédent. Le code semble à la fois séquentiel et facile à maintenir. Il est temps de détailler, comment ce code fonctionne, afin de mieux appréhender ce nouveau modèle. À l'instar des modèles APM, EAP et TAP, une demande de lecture au driver NTFS sous-jacent est réclamée et retourne immédiatement. Une structure IRP interne permet d'édifier un lien avec le code de continuation qui permettra de récupérer le résultat (pour plus explication, relire l'article précédent). Pour l'instant, je ne vous ai pas expliqué d'où vient cette continuation.

```
int bytesRead = await fs.ReadAsync(buffer, 0, buffer.Length);
Console.WriteLine("Number of bytes read={0}", bytesRead);
Console.WriteLine(BitConverter.ToString(buffer, 0, bytesRead));
```

Tout le code en jaune, comprenant la déclaration int bytesRead, jusqu'à la dernière ligne Console.WriteLine(BitConverter.ToString(Buffer, 0, bytesRead)), est capturée par le compilateur afin de devenir une continuation. Cette capture est réalisée par le compilateur à votre insu : lorsque la combinaison des deux identifiants async et await est utilisée dans le contexte décrit précédemment, alors le compilateur va prendre en charge l'implémentation de l'asynchronisme pour vous.

Nous pouvons aussi reprendre l'exemple EAP de l'article précédent, pour le transformer avec l'offre asynchrone de C# 5.0.

```
string html = await new WebClient().DownloadStringTaskAsync("http://blogs.msdn.com/b/nativeconcurrency/");
listView.Items.Add(new PageLength(uri, uri.Length));
txtResult.Text = "LENGTH: " + html.Length.ToString("N0");
```

Dans cet exemple, tout le code en jaune est capturé par le compilateur pour devenir une continuation. Sur le plan de l'exécution, le code capturé s'exécute au-delà de l'appel de la méthode. En effet, l'appel à la méthode DownloadString retourne immédiatement vers l'appelant. Ce n'est qu'une fois la page téléchargée, que le code capturé par le compilateur est exécuté. Par défaut, le code de la continuation s'exécute dans le thread de l'appelant. Généralement dans un contexte graphique, ce comportement vous conviendra certainement. Cependant, si vous développez une librairie sans affinité graphique ou plus simplement du code serveur, vous pouvez configurer l'appel d'un code await, via la méthode Task.ConfigureAwait. Cette méthode permet à travers un booléen de préciser, si le code de la continuation doit se synchroniser ou pas avec le contexte de l'appelant. Par exemple dans le code suivant nous signalons que nous ne souhaitons pas de synchronisation en passant la valeur false à la méthode ConfigureAwait.

```
int bytesRead = await fs.ReadAsync(buffer, 0, buffer.Length)
    .ConfigureAwait(false);
```

L'exemple suivant reprend l'exemple de la méthode PopulateAsync, mais dans un contexte un peu plus complexe. C'est un fragment tiré d'une application WPF, qui cette fois boucle sur liste d'URLs.

```
async void PopulateAsync()
{
    listView.Items.Clear();
    btnGo.IsEnabled = false;
    btnCancel.IsEnabled = true;
    txtResult.Text = String.Empty;
    try
    {
        int totalLength = 0;
```

```

foreach (string uri in _uris)
{
    string html = await new WebClient().DownloadString
TaskAsync(uri);
    totalLength += html.Length;
    listView.Items.Add(new PageLength(uri, uri.Length));
}
txtResult.Text = «TOTAL LENGTH: <> + totalLength.ToString(<>N0»);
}
catch (Exception ex)
{
    txtResult.Text = «ERROR: <> + ex.Message;
}
finally
{
    btnGo.IsEnabled = true;
    btnCancel.IsEnabled = false;
}
}

```

Ce dernier exemple démontre toute la puissance de l'offre asynchrone C# 5.0 (difficile de faire moins invasif). Toutes les offres précédentes, APM, EAP et TAP sont parfaitement incapables de proposer l'équivalent de ce code, sans devenir un plat de spaghetti.

Quelques contraintes sur le passage de paramètre

Sur le plan de la signature d'une méthode déclarée asynchrone, il existe quelques contraintes que vous devez connaître. Ces contraintes portent sur des restrictions sur le passage de paramètres concernant les identifiants `out` et `ref` qui sont interdits.

```
public static async void UseFooWithAwaitAsync(ref string
value, out string res) ...
```

À la compilation, on obtient des erreurs de type: *Async methods cannot have ref or out parameter*. Ces contraintes s'expliquent parfaitement au regard du fonctionnement interne des expressions contenant le mot clef `await` qui peuvent être mises en pause afin de rendre la main au code appelant, puis être relancées en fonction de la durée de l'appel asynchrone afin de poursuivre le fil d'exécution avec le code de la continuation. On comprend alors pourquoi ces contraintes existent, car maintenir le contexte de ces variables entre différents appels rend l'implémentation interne compliquée.

Communication asynchrone

Dans les offres asynchrones précédentes, APM, EAP et TAP (Programmez ! 169) nous avons commenté les « véhicules » de communications asynchrones permettant de transmettre « le résultat » une fois, le traitement asynchrone terminé. Dans l'offre TAP, une instance du type `Task` ou `Task<TResult>` est utilisée. Le support des identifiants `async` et `await` assure parfaitement cette transparence. La motivation de l'offre asynchrone de C# 5.0 est d'être la moins invasive possible vis-à-vis de votre code original synchrone.

Comprendre l'implémentation

Introduction

Dans les modèles précédents, nous avons détaillé le fonctionnement sous-jacent de chaque solution asynchrone. Dans le cadre de l'offre `async/await`, le sujet est un peu plus vaste. En effet, nous pouvons distinguer deux sujets :

- Le pattern `await`
- La capture des variables.

Pattern `await`

Voici quelques explications qui vous permettront de mieux comprendre l'implémentation interne du mot clef `await`. Le mot clef `await` est placé au sein d'une méthode préfixée par l'identifiant `async`, afin de permettre une simplification de la construction de la continuation.

```
var result = await expression ;
statement(s);
```

Cette fonctionnalité est traduite par le compilateur

```
var awaiter = await expression.GetAwaiter();
await.OnCompleted(() =>
{
    var result = awaiter.GetResult();
    statement(s);
})
```

Si le traitement est déjà terminé avant que la continuation ne soit lancée, le compilateur engendre un raccourci afin d'exécuter immédiatement en mode synchrone, le code de la continuation, comme l'illustre le code ci-dessous.

```
var awaiter = DownloadPageAsync(uri).GetAwaiter();

if (awaiter.IsCompleted)
    Console.WriteLine(awaiter.GetResult());
else
    awaiter.OnCompleted(() => Console.WriteLine(awaiter.
GetResult()));
```

L'essentiel des appels des API asynchrones qui utilisent le mot clef `await` retourneront une tâche TPL. Cependant, le compilateur C# n'impose pas ce type de résultat. Il est parfaitement légitime d'attendre autre chose qu'une tâche. La seule contrainte est que l'objet retourné supporte quelques méthodes bien spécifiques. Le compilateur impose que l'objet retourné contienne une méthode nommée `GetAwaiter` (cependant une méthode d'extension du même nom serait compatible aussi). Cette méthode doit retourner un objet ou une valeur, qui supporte les trois points suivants :

- L'objet doit supporter une propriété `IsCompleted` de type `bool` qui permet au code généré par le compilateur d'engendrer un raccourci: si l'exécution est déjà terminée, éviter de perdre du temps à enregistrer une continuation pour rien. Dans le code situé juste après, le mot clef `await` est alors exécuté dans la foulée.
- L'objet doit aussi supporter une méthode nommée `GetResult`. Le type de retour est sélectionné en fonction du type l'expression `await`. Dans notre exemple c'est un type `string` qui doit être retourné.
- Enfin, l'objet doit fournir une méthode de type `callback` dans le cas où la propriété `IsCompleted` retourne `false`, le code généré pour l'expression `await` créera un délégué qui exécutera le reste de la méthode. Pour ce support, le compilateur exige le support de l'implémentation de l'interface `INotifyCompletion` ou l'implémentation de l'interface `ICriticalNotificationCompletion`. Ces deux interfaces sont similaires et définissent une seule méthode ; respectivement `OnCompleted` et `UnsafeOnCompleted` qui est passé à une instance d'un délégué `Action`, que l'objet retourné appellera une fois que l'opération asynchrone sera terminée. La sélection de l'implémentation d'une des deux interfaces est à la charge du compilateur en fonction du contexte. Par défaut c'est l'implémentation de l'interface `ICriticalNotificationCompletion` qui est utilisée si l'exécution du code est considérée Full Trust. Notons que cette interface est

marquée par un attribut de sécurité `SecurityCriticalAttribut`. Enfin, c'est l'interface `INotifyCompletion` qui est utilisée dans le cadre d'un code marqué `Partial Trust`.

Pour synthétiser tous ces éléments, je vous propose un exemple simplifié qui illustre le code simplifié d'une classe contenant un appel avec `await`, qu'on appelle `FooWithAwait`, généré par le compilateur.

```
using System.Runtime.CompilerServices;

public class FooWithAwait
{
    public SimpleAwaiter GetAwaiter()
    {
        return new SimpleAwaiter();
    }

    public class SimpleAwaiter : INotifyCompletion
    {
        public bool IsCompleted { get { return true; } }

        public string GetResult()
        {
            return «Hello world Async/Await»;
        }

        public void OnCompleted(Action continuation)
        {
            throw new NotImplementedException();
        }
    }
}
```

Ce type contient naturellement une méthode `GetAwaiter` qui retourne une classe interne `SimpleAwaiter`. La classe `SimpleAwaiter` implémente naïvement l'interface `INotifyCompletion`. Cette interface est définie au sein de l'espace de nom `System.Runtime.CompilerServices`. La propriété `IsCompleted` retourne systématiquement `true`, ce qui signifie que la méthode `OnCompleted` ne sera jamais appelée. C'est pour cette raison qu'elle n'est pas implémentée.

Pour illustrer le fonctionnement de cette implémentation utilisée en interne par le compilateur, je vous propose le code suivant reposant sur un simple projet de type console (Code complet sur www.programmez.com).

La méthode `UseFooWithAwaitAsync` est une méthode asynchrone qui respecte la syntaxe `async/await`: la signature de la méthode est préfixée par le mot clef `async`, le nom de méthode est post fixé par le mot `Async`. Le corps de la méthode utilise le mot clef `await` sur l'appel d'une méthode appelée `UseCompilerAsync`. Cette méthode est une fabrique, elle instancie `FooWithAwait` qui implémente le pattern `await` du compilateur C# 5.0.

Si vous lancez ce petit programme, vous obtiendrez dans votre console le fameux message « Hello world ».

Capture des variables

Pour l'implémentation d'`async/await`, les ingénieurs Microsoft ont appliqué le même principe utilisé dans le cadre de l'implémentation simplifiée du pattern Énumérateur avec des mots «`yield return`» et «`yield break`»: une machine à états générés par le compilateur. Cette machine à états est engendrée afin de prendre en compte un déroulement correct d'appels d'une ou plusieurs méthodes asynchrones, pouvant lever des exceptions au sein de boucles, par exemple.

Cependant, la vraie puissance du couple `async/await` réside dans la cap-

ture des variables. En effet, vous pouvez placer des appels `await`, n'importe où dans votre code (même dans des boucles, des événements et même des expressions lambda), excepté au sein de blocs `catch` et `finally`, des blocs de code verrouillés par le mot clef `lock`, au sein d'un contexte `unsafe` ou dans le point d'entrée d'un programme (`Main`).

Pour étudier la transformation d'un code décoré avec le couple `async/await`, je vous propose la méthode ci-dessous qui reprend en partie un exemple précédent.

```
static async Task<List<string>> DownloadPagesAsync(IEnumerable<string> uris)
{
    var result = new List<string>();

    using (var wc = new WebClient())
    {
        foreach (var uri in uris)
        {
            result.Add(await wc.DownloadStringTaskAsync(uri));
        }
    }
    return result;
}
```

Lorsque la méthode `DownloadPagesAsync` va exécuter la méthode `DownloadStringTaskAsync` pour la première fois, le retour de cette méthode va retourner immédiatement vers l'appelant en vertu du fonctionnement du couple `async/await`. Lorsque la continuation sera terminée, le programme va reprendre son exécution là où il l'avait laissée avec toutes les variables correctement renseignées au sein de la boucle. Le compilateur génère un système de capture des variables respectant le contexte d'exécution courant. La phase où l'expression `await` retourne vers l'appelant est complètement prise en charge par le compilateur. Le compilateur doit aussi gérer un fil d'exécution identique à la version synchrone, ce qui signifie que dans notre exemple les résultats de toutes les itérations de notre boucle doivent être ordonnés à l'identique à l'exécution synchrone. Sans les mots clefs `async/await`, il serait extrêmement difficile de coder à la main une solution équivalente et encore plus dur de la maintenir, d'où l'importance de la prise en charge du compilateur.

Pour vous sensibiliser au travail réalisé par le compilateur, je vous propose d'étudier une implémentation personnalisée d'une machine à états, similaire (mais simplifié) à ce que génère le compilateur C# 5.0. Cette nouvelle implémentation (ci-dessous) peut sembler intimidante. La difficulté provient de la traduction de la boucle qui itère la liste des uris pour appeler successivement la méthode asynchrone `DownloadStringTaskAsync`. Mais pas d'inquiétude, car si vous avez compris tous les éléments précédents, cette méthode ne devrait pas vous poser de problèmes de compréhension.

```
static Task<List<string>> DownloadPagesAsync(IEnumerable<string> uris)
{
    var tcs = new TaskCompletionSource<List<string>>();
    List<string> result = null;
    WebClient wc = null;
    Ienumerator e = null;
    int state = 0;
    Action moveNext = null;
    Task<string> downloadTask = null;
```

```

moveNext = () =>
{
    switch (state)
    {
        case 0:
        {
            result = new List<string>();
            wc = new WebClient();
            goto case 2;
        }
        case 2:
        {
            bool shouldDispose = true;
            try
            {
                switch (state)
                {
                    case 0:
                    {
                        e = uris.GetEnumerator();
                        goto case 1;
                    }
                    case 1:
                    {
                        if (e.MoveNext())
                        {
                            var uri = (string) e.Current;
                            downloadTask = wc.DownloadStringTaskAsync(
                                uri);

                            state = 2;
                            shouldDispose = false;
                            downloadTask.ContinueWith(_ => moveNext());
                        }
                    }
                    else
                    {
                        tcs.TrySetResult(result);
                    }
                }
                break;
            case 2:
            {
                if (downloadTask.IsFaulted)
                {
                    tcs.TrySetException(downloadTask.Exception);
                    return;
                }
                else
                {
                    result.Add(downloadTask.Result);
                    goto case 1;
                }
            }
        }
        finally
        {
            if (shouldDispose)
                wc.Dispose();
        }
    }
}

```

```

        break;
    }
};

moveNext();

return tcs.Task;
}

```

La signature de la méthode ne change pas, mais comme vous pouvez le constater l'implémentation est totalement bouleversée.

Pour simuler le travail du compilateur, la première chose à faire est de produire une méthode TAP.

Pour cela nous créons une nouvelle instance d'un TaskCompletionSource <T>, telle que nous l'avons présentée précédemment.

```
var tcs = new TaskCompletionSource<List<string>>();
```

Puis nous déclarons une liste de variables qui sont utilisées dans l'implémentation de la boucle via une machine à états :

```
List<string> result = null;
```

La variable results sera utilisée pour entretenir les résultats des appels asynchrones de la méthode wc.DownloadStringTaskAsync.

```
WebClient wc = null;
```

La variable wc représente une instance du type WebClient utilisée pour télécharger les pages Web.

```
IEnumerator e = null;
```

La variable e représente l'énumérateur de la liste des URLs des pages à télécharger.

```
int state = 0;
```

La variable state représente l'état courant de la machine à états.

```
Action moveNext = null;
```

La variable moveNext représente l'action courante de la machine à états. L'action est initialisée par une expression lambda qui, par nature, capture toutes les variables utilisées.

```
Task<string> downloadTask = null;
```

La variable downloadTask représente un retour de l'appel à la méthode DownloadStringTaskAsync.

La première étape consiste à initialiser l'action moveNext par expression lambda contenant tout le code de la machine à états. Cette machine à états va être animée par la variable state au cours des appels successifs à la variable moveNext.

L'action est appelée à la sortie de la méthode DownloadPagesAsync avant de retourner la propriété Task de l'instance TaskCompletionSource<List<string>>. Cependant, vous pouvez remarquer dans le corps de la machine à états, la ligne suivante :

```
downloadTask.ContinueWith(_ => moveNext());
```

Pour conserver les variables de chaque itération interrompue par le retour à l'appelant de la méthode DownloadPagesAsync, la continuation de la tâche de retour de la méthode DownloadStringTaskAsync est renseignée par l'action moveNext, créant ainsi une nouvelle tâche contenant une capture des variables associées à l'état courant.

La première fois où la machine à états est appelée, la variable state est fixée à 0. De ce fait le code suivant est exécuté :

```

case 0:
{
    result = new List<string>();
    wc = new WebClient();
    goto case 2;
}

```

La liste des résultats est allouée et une instance du type WebClient est instantiée. Le code se poursuit à la case 2 :

```

case 2:
{
    bool shouldDispose = true;
    try
    {
        switch (state)
        {
            case 0:
            {
                e = uris.GetEnumerator();
                goto case 1;
            }
        }
    }

```

À la case 2, une variable est positionnée pour mémoriser que l'instance du WebClient doit être nettoyée. À ce stade, la variable state contient toujours la valeur 0, ce qui permet de récupérer un énumérateur sur la liste des URIs. Puis le code se poursuit à la case 1 :

```

case 1:
{
    if (e.MoveNext())
    {
        var uri = (string) e.Current;
        downloadTask = wc.DownloadStringTaskAsync(uri);
        state = 2;
        shouldDispose = false;
        downloadTask.ContinueWith(_ => moveNext());
    }
    else
    {
        tcs.TrySetResult(results);
    }
}

```

La case 1 permet d'itérer les éléments de la liste des URIs via l'interface `IEnumerable` et sa méthode `MoveNext`. Si la méthode retourne `false`, il n'y a plus d'élément à itérer, le code peut donc stocker la liste des résultats, `results`, dans l'instance du type `TaskCompletionSource<List<string>>` via la méthode `TrySetResult`.

```

else
{
    tcs.TrySetResult(results);
}

```

Si la méthode `MoveNext` retourne `true`, alors la valeur courante de l'item est récupérée via l'appel à la méthode `DownloadStringTaskAsync`. Le résultat de l'appel est stocké dans la variable `downloadTask`. La variable `state` est modifiée par pas de 0 à 2. La variable `shouldDispose` repasse à `false` car nous sommes en cours de traitement et nous ne devons pas

détruire prématièrement l'instance du `WebClient`. Puis nous passons au code de continuation où l'expression lambda relance l'action `moveNext`. Mais cette fois la variable `state` représentant l'état courant de la machine à états est fixée à 2 et le code de l'action `moveNext` est plongé dans un thread du Pool de Threads. Le code de l'action `moveNext` reprend une nouvelle exécution et aboutit dans la portion de code suivante :

```

case 2:
{
    if (downloadTask.IsFaulted)
    {
        tcs.TrySetException(downloadTask.Exception);
        return;
    }
    else
    {
        result.Add(downloadTask.Result);
        goto case 1;
    }
}

```

Si la tâche retournée par la méthode `DownloadStringTaskAsync` a échoué, cela signifie qu'une exception a été levée. Cette exception enregistre l'instance du type `TaskCompletionSource<List<string>>` via la méthode `TrySetException`, puis le code retourne vers l'appelant. Si la tâche n'a pas échoué, le résultat est inséré dans la liste des résultats via la variable `result`. Puis le code retourne à la case 1 que nous avons déjà commentée. À ce stade, nous comprenons que la machine à états va se poursuivre de la même manière tant que la liste des URIs n'a pas totalement été visitée et qu'il n'y a pas d'erreur.

Avec cette étude, vous devriez apprécier que le codage à la main d'une machine à états, afin de retrouver le comportement asynchrone similaire à la version `async/await`, est particulièrement compliqué. Il est évident que le code devient difficile à lire, et très difficile à faire évoluer. Finalement, le principe utilisé par Microsoft est simple, puisque le code engendré pour obtenir une solution pertinente est complexe, alors, autant laisser le compilateur générer le code afin que le développeur gagne en efficacité et en productivité. Pour des raisons de compréhension, la machine à états présentée est bien plus simple que celle générée par le compilateur. L'utilisation du pattern `await` a été volontairement oubliée afin d'obtenir un code plus facile à comprendre. L'objectif était de vous expliquer comment on peut gérer la capture des variables dans le cadre d'une boucle faisant appel à une méthode asynchrone. Vous pouvez retenir que le compilateur C# 5.0, lorsqu'il rencontre une méthode utilisant `async/await`, engendre sous le capot un code utilisant à la fois le pattern `await` et une machine à états dédiée au contexte courant.

CONCLUSION

Dans cette troisième partie consacrée à l'implémentation de l'offre asynchrone avec C# 5.0, nous avons détaillé comment cette nouvelle fonctionnalité est modélisée en interne, afin de maîtriser son implémentation. Cette nouvelle fonctionnalité est sans doute la fonctionnalité majeure de C# 5.0. Elle apporte un service considérable aux développeurs utilisant Visual Studio 2012 pour produire des applications .NET 4.5 ou pour proposer sur le Store Microsoft des nouvelles applications « *Fast & Fluid* ».

 Bruno Boucard

Technical Fellow | MVP C++ chez Cellenza

Software Development Done Right

Son Blog : <http://blogs.msdn.com/b/devpara>

Dart : la plateforme orientée Web de Google

Google ne cesse de proposer des solutions visant à rendre le Web plus rapide et plus riche. Toujours prompt à faire bouger les lignes, le géant de Mountain View est à la pointe de l'innovation et nous le prouve une fois de plus avec sa plateforme Dart sur le point de fêter sa version 1.0. Indissociable de Javascript dont elle assure la compatibilité tout en comblant ses lacunes, Dart offre un environnement moderne et efficace pour réaliser des applications web riches. Plongée au coeur d'une technologie prometteuse combinant performance et productivité.

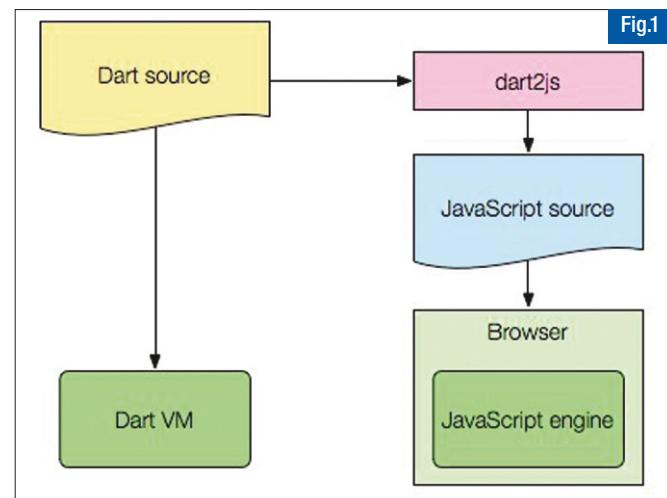
Révélée au grand public en Octobre 2011, la plateforme Dart se base sur le langage éponyme. Bien souvent d'ailleurs, beaucoup pensent que Dart se limite à un langage, mais il constitue en réalité un véritable écosystème applicatif ayant pour but de fournir une alternative aux limites de Javascript. Plutôt que de chercher à faire évoluer le langage Javascript pour combler ses lacunes, les ingénieurs de chez Google ont opté pour une solution nouvelle avec Dart, tout en assurant la pleine compatibilité avec Javascript puisque la plateforme propose un compilateur de code vers ce dernier. A l'origine de la création de Dart : un constat assez simple. Véritable assemblleur du Web, le langage Javascript pose un certain nombre de problèmes tant au niveau de sa syntaxe (typage dynamique, langage prototypé, ...) que de son implémentation différente au sein des navigateurs. En outre, les performances ne sont pas toujours au rendez-vous, du fait d'une mauvaise utilisation du langage. En effet, de par ses spécificités, le Javascript nécessite une forte maîtrise de la part des développeurs pour construire des applications web à la fois rapides, sûres et riches. Trouver de tels développeurs est souvent ardu, ce qui amène à la réalisation d'applications web de piètre qualité. Enfin, les outils de développement autour de Javascript ne sont pas de qualité suffisante, ce qui ralentit la productivité des développements. Parmi les autres acteurs du Web ayant fait ces constats, Microsoft a pris une voie différente avec TypeScript qui s'appuie sur la syntaxe Javascript en proposant un "sur-ensemble" au langage. A contrario, Google a donc préféré partir sur un nouveau langage pour corriger au mieux les lacunes de Javascript. C'est dans ce contexte qu'est né Dart qui se veut être, selon ses concepteurs, tout ce que Javascript aurait du être s'il avait été inventé aujourd'hui. En bref, il tente de garder un côté dynamique dans un cadre plus strict pour offrir des outils de qualité permettant de réaliser des applications web complexes et pérennes.

Architecture

Un programme Dart est exécuté au sein d'une machine virtuelle dédiée, utilisable en standalone, ou bien embarquée dans un navigateur exécutant directement le code Dart. Le SDK s'amène d'ailleurs avec Dartium, une implémentation spécifique de Chromium intégrant la machine virtuelle Dart. Excepté Chrome qui prévoit d'intégrer cette machine virtuelle dans un avenir proche, la plupart des navigateurs ne supportent pas Dart. Pour contourner ce potentiel problème, le SDK propose un compilateur Dart vers Javascript compilant un programme Dart en un Javascript équivalent, compatible avec les normes ECMAScript 5 et HTML 5. Le code produit fonctionne sur l'ensemble des navigateurs récents : Chrome, Safari, Firefox, IE 9 et IE 10 ainsi que les versions mobiles de Safari et de Chrome. Ceci offre l'avantage d'abstraire le développeur du travail de compatibilité inter navigateurs.

L'approche retenue pour l'exécution d'un programme Dart (figure 1) consiste à proposer le code Dart et son équivalent Javascript tout en testant à l'exécution la présence ou non d'une machine virtuelle Dart afin de savoir quel code exécuter. Ce test se fait via le script dart.js mis à disposition par la plateforme.

Le compilateur dart2js compile toutes les sources d'une application, y



Possibilités d'exécution d'un code Dart

compris les bibliothèques utilisées, en seul fichier Javascript. Juste avant cette compilation, dart2js optimise le code source, afin de réduire sa taille, en éliminant le code mort via le Tree Shaking à l'instar de ce que fait le framework GWT par exemple. Le modèle d'exécution de Dart n'engage qu'un seul thread mais autorise néanmoins un mécanisme de concurrence simple. En outre, la gestion des dépendances n'est pas oubliée avec l'outil pub permettant de découvrir de nouvelles bibliothèques, de les installer et de les intégrer tout en gérant leur versionning. Enfin, la machine virtuelle possède 2 modes d'exécution. Plus rapide, le mode production est activé par défaut et ne lève pas d'avertissements sur le code comme peut le faire le mode checked.

SDK et Outils

Récupérable ici : <http://www.dartlang.org/tools/sdk/>, le SDK Dart met à disposition les outils nécessaires pour la création et le développement d'applications web de qualité (figure 2) proposant ainsi :

- ▶ La machine virtuelle Dart
- ▶ Le compilateur dart2js
- ▶ Le gestionnaire de dépendances pub
- ▶ Le navigateur Dartium embarquant une machine virtuelle Dart
- ▶ Des outils pour la qualité des projets Dart (le générateur de documentation dartdoc et l'analyseur statique dartanalyzer)

A côté du SDK, Google met à disposition l'IDE Dart Editor. Basé sur Eclipse, il offre les outils de base pour développer des applications web avec Dart. On retrouve ainsi la complétion du code, l'analyse de code, la navigation au sein d'un projet, le débogage ainsi que l'exécution du code depuis l'IDE. En sus, des plugins pour Eclipse et IntelliJ sont proposés.

Bibliothèques

Le SDK Dart comporte près d'une vingtaine de bibliothèques standards couvrant les besoins de base dans la réalisation d'applications web mais égale-

ment ceux pour le développement côté serveur parmi lesquelles on retrouve :

- ▷ dart:core inclut par défaut contient l'ensemble des types et des structures de données de base
- ▷ dart:html qui est la porte d'entrée pour accéder au DOM HTML 5
- ▷ dart:io réservée au développement d'applications serveur avec le support des entrées/sorties et un serveur HTTP
- ▷ dart:math pour les fonctions mathématiques basiques
- ▷ dart:json pour travailler avec le format JSON
- ▷ dart:utf pour la gestion de l'Unicode
- ▷ dart:mirrors propose des fonctionnalités de réflexion
- ▷ dart:isolate supportant une concurrence simple et sûre
- ▷ dart:async pour des opérations asynchrones avec notamment le concept des Futures

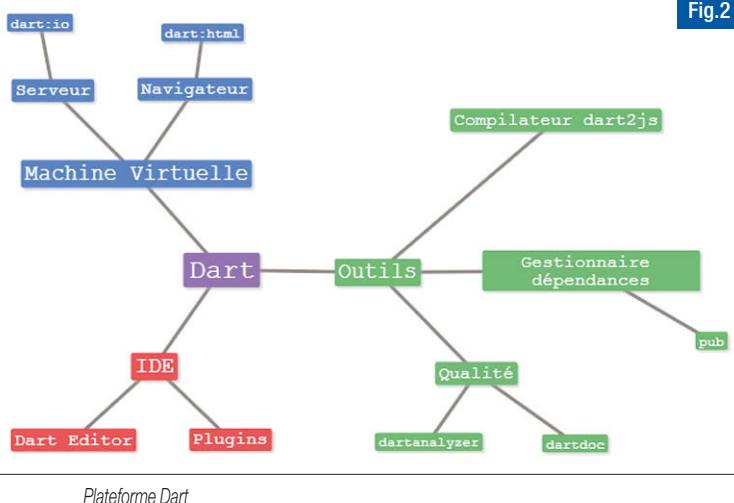
En plus de ces composants standards, de nombreuses autres bibliothèques sont accessibles via le gestionnaire de dépendances pub. On peut ainsi citer Javascript Interop pour utiliser du Javascript directement au sein de Dart, Web UI pour construire des applications web basées sur les composants web et Unit Test pour l'écriture de tests unitaires.

Langage

Conçu pour palier aux défauts de Javascript, Dart est un langage de script web dynamique pouvant être compilé en Javascript ou exécuté au sein d'une machine virtuelle dédiée. Pleinement orienté objet, il repose sur des classes concrètes et abstraites avec un héritage simple. Dérivant du C, sa syntaxe se rapproche d'autres de ses descendants tels que Java et C#. Supportant également les interfaces et les génériques, il propose un système de typage optionnel ignoré à l'exécution et qui sert à déclencher des avertissements en phase de développement. D'autre part, Dart supporte les fonctions de haut niveau considérées comme des objets à part entière et donc utilisables en tant qu'arguments en entrée de fonctions, ou méthodes au sein d'une classe. En sus, le concept de mixins autorise des constructions puissantes. Un modèle de concurrence simple est supporté via les Isolates implémentés sous forme de Web Workers lors du passage en Javascript. Au niveau des différences avec ce dernier, on peut citer l'absence des prototypes et l'impossibilité d'exécuter du code à la volée. Enfin, un programme Dart requiert la présence d'une fonction de haut niveau main appelée à son lancement.

De par sa conception, le langage Dart et plus encore la plateforme, autorisent la construction d'applications web allant du simple script au projet complexe, avec une grande efficacité. Ses concepts issus de langages répandus le rendent facile à prendre en main, et le fait qu'il puisse être employé côté client et côté serveur lui confère des atouts considérables dans la perspective de réaliser des applications web via un seul langage.

Fig.2



Types

Tous les types Dart sont des objets dérivant de la classe Object. Le mot clé var sert à déclarer une variable sans définir de type. Comme en Java, la méthode `toString()` d'un objet est appelée par défaut lorsque celui-ci est converti en String comme avec la fonction `print` :

```
var myObj = 2013;
// équivalent
print(myObj);
print(myObject.toString());
```

Dans l'idéal, l'absence de typage doit se limiter à des portions de codes internes. Pour représenter des numériques, Dart propose le type général num et les implémentations spécialisées int et double. Bien qu'il s'agisse d'objets, il n'est pas nécessaire de les instancier via new :

```
num myNum = 12.4;
num myRounded = myNum.round();
int myInt = 12;
// possible
var myInt = 123;
var intAsHex = 0xD;
double myDouble = 12.4;
var doubleWithExp = 1.24e3;
```

Les objets numériques proposent les opérations de calculs +/* qu'il est possible de surcharger, ainsi que des fonctions classiques comme abs, ceil, floor et round. Contrairement à Javascript, Dart considère uniquement le littéral booléen true comme valeur vraie. Toutes les autres valeurs sont donc fausses. En mode checked, les opérations booléennes sont autorisées uniquement sur des variables de type bool :

```
bool myBool = true;
var myBool = true;
bool myFalse = false;
print(myBool != myFalse); // true
print(myBool && myFalse); // false
print(myBool || myFalse); // true
print("Cond is : " is String); // true
```

On remarque l'opérateur is pour tester si une instance est d'un type ou non. Élément indispensable, les chaînes de caractères sont implémentées au sein du type String. Leur principale spécificité concerne la concaténation qui s'appuie sur l'interpolation et non l'opérateur + :

```
String hello = "Hello";
var name = "Sylvain";
var concat = "$hello $name";
print(concat); // Hello Sylvain
```

Les variables définies au sein d'une interpolation sont évaluées, ce qui autorise des opérations en se servant de \${...} :

```
var test = "${1+2.5.round()}";
print(test); // 3
```

Structures de données

Au niveau structures de données, Dart n'a pas de type tableau mais propose le type List instanciable à partir de la syntaxe raccourcie ou via son constructeur :

```
List myList = []; // vide
var myList = [];
var myList = [1, 2, 3];
var myList = [1, "val 2", 3];
```

```
// constructeur
List myList = new List();
var myList = new List(10); // 10 elts null
var myList = new List.filled(10, "SS"); // 10 "SS"
```

L'ajout d'éléments se fait via la méthode add() alors que l'accès à un élément de la liste passe par l'utilisation des crochets []. En outre, il est bon de souligner que les listes sont indexées sur le zéro :

```
myList.add("Elt1");
myList.add("2");
print(myList[1]); // 2
myList[1] = "Elt2";
```

Le langage supportant les génériques, il est possible de typer une liste pour préciser le type d'élément qu'elle peut contenir. L'ajout d'un objet de type différent lèvera ensuite un avertissement de type :

```
List<String> myList = new List<String>();
myList.add(123); // avertissement 123 non String
myList = <String>["a", "b"];
```

Le parcours des éléments d'une liste peut se faire de manière classique en accédant aux éléments via leur index, soit en utilisant une boucle for-in, ou bien en appelant la méthode forEach de l'objet List :

```
for(var i = 0; i < myList.length; i++) {
  var val = myList[i];
}

for(var item in myList) {
  print(item);
}
myList.forEach((item) {
  print(item);
});
```

La bibliothèque standard propose également des implémentations spécialisées pour les collections HashSet, Queue et Set. D'autre part, le langage supporte les maps qui sont des listes de couples clé / valeur. La création d'une map se faisant via la syntaxe raccourcie ou via son constructeur. Les génériques restent applicables et l'accésion aux éléments d'une map se fait via les crochets []. Enfin, le parcours d'une map se réalise via une boucle for ou via la méthode forEach :

```
Map m = {};
Map<String, String> map = {"A":"1", "B":"2", "C":"3"};
// ajout
map["D"] = "4";
print(map.containsKey("D")); // true

for (var keys in map.keys){
  print(keys);
}

for (var val in map.values){
  print(val);
}

map.forEach((k,v) {
  print("$k => $v");
});
```

Fonctions

Les fonctions peuvent être de haut niveau ou bien définies au sein d'une classe; auquel cas on parle de méthodes. En sus, des facilités syntaxiques permettent de raccourcir l'écriture de fonctions :

```
sum(num val1, num val2) => val1 + val2;
printMsg(msg) => print("Msg : $msg"); // renvoie null
// équivalent
num sum(num val1, num val2) {
  return val1 + val2;
}
```

Le typage d'une fonction est optionnel tant au niveau des entrées que de sa sortie. Définir les types attendus reste néanmoins une bonne pratique permettant aux outils Dart de travailler efficacement. Les paramètres d'entrée peuvent être obligatoires ou optionnels. Dans tous les cas, les paramètres obligatoires sont définis en premier. Alors que ces derniers doivent être passés en respectant l'ordre de définition, les paramètres optionnels peuvent être également définis comme associés à leur nom :

```
void optParams(String name, [int version = 1, url]) { ... }
void optNamed(String name, {int version:1, url}) { ... }
optParams("name");
optParams("name", 1);
optNamed("name", url:"www.programmez.com");
```

Comme pour les autres types, une fonction peut être affectée à une variable, ce qui implique que l'on peut passer une fonction en entrée ou en sortie d'une autre fonction :

```
num getResult(num val1, num val2, Function calcFunc) {
  return calcFunc(val1, val2);
}
var add = (a,b) => a+b;
var sub = (a,b) => a-b;

var res1 = getResult(1,2,add); // 3
var res2 = getResult(10,5,sub); // 5
```

Afin de définir une fonction d'un type donné et d'éviter d'utiliser le type générique Function, il faut recourir au mot clé typedef :

```
typedef CalcFunction(num value1, num value2);
```

Cette définition permet de lever des avertissements à l'exécution si le type de fonction passé en entrée ne correspond pas au type attendu.

Classes / Interfaces

Le modèle de classes retenu par Dart se rapproche grandement de ce qui existe en Java ou en C#. Ceci permet notamment aux développeurs venant de ces univers de ne pas être trop dépayrés. Une classe peut ainsi définir un constructeur, des méthodes, des propriétés ainsi que des getters et setters. Ces derniers étant définis par défaut mais pouvant être redéfinis le cas échéant :

```
class Person {
  // private
  String _firstName;
  String _lastName;
  int _age;
  // public
```

```

String gender = MALE;
static String MALE = "MALE";
static String FEMALE = "FEMALE";

Person(fname, lname, age, g) {
    _firstName = fname;
    _lastName = lname;
    _age = age;
    gender = g;
}

Person.fromJson(Map data) {
    // ...
}

int get age => _age;

bool isMale() => this.gender == MALE;

String toString() {
    return "${_firstName} ${_lastName} - ${_age}";
}

```

Le langage propose un héritage simple via le classique mot clé extends. L'accès à l'objet parent se faisant via le mot clé super. Les constructeurs ne sont pas hérités et le constructeur parent doit être appelé dans le bloc d'initialisation du constructeur :

```

class Employee extends Person {
    int _id;
    Employee(fname, lname, age, g) : super(fname, lname, age, g) { }
    Employee.fromJson(Map data) : super.fromJson(data) {
        // ...
    }
}

```

Pour définir une classe sans implémentation ou avec implémentation partielle, le langage propose les classes abstraites. Les classes qui en héritent doivent définir les méthodes sans implémentation :

```

abstract class Person {
    // ...
    void fromJson(Map data);
}

class Employee extends Person {
    // ...
    void fromJson(Map data) {
    }
}

Person p = new Employee("Sylvain", "Saurel", 29, Person.MALE);

```

Le modèle de classes de Dart ne propose pas de mot clé interface. Si le langage ne propose pas de définition explicite, c'est que chaque classe est en fait une interface. En effet, l'interface d'une classe est la représentation des membres publics de la classe. Quand une classe implémente une interface, elle doit fournir une implémentation pour tous les membres :

```

class Employee implements Person {
    // implémentation des membres publics
}

```

```

void fromJson(Map data);
}
```

Pour terminer, un mot sur les mixins que le langage propose depuis sa milestone 3. Ceux-ci permettent d'injecter des comportements au sein d'une classe sans recourir à l'héritage. Le rajout d'un comportement Persistence à la classe Employee s'effectuant de la sorte :

```

abstract class Persistence {
    void save(String filename) { }
    void load(String filename) { }
}

class Employee extends Person with Persistence {
    // ...
}

```

Premier programme

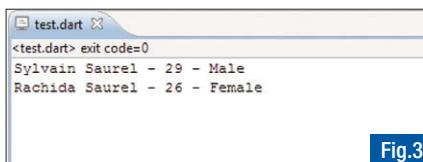
Un programme Dart contient obligatoirement une fonction de haut niveau main servant de point d'entrée lors de l'exécution du programme. Notre premier programme Dart met en oeuvre les classes définies précédemment :

```

void main() {
    List<Person> persons = <Person>[];
    Person p = new Employee("Sylvain", "Saurel", 29, Person.MALE);
    persons.add(p);
    p = new Employee("Rachida", "Saurel", 26, Person.FEMALE);
    persons.add(p);

    persons.forEach((e) {
        print("${e} - ${e.isMale() ? 'Male' : 'Female'}");
    });
}

```



L'exécution de ce code au sein de l'IDE Dart Editor produit la sortie présentée à la figure 3.

Fig.3

Premier programme Dart

Conclusion

Bien plus qu'un simple langage, Dart constitue une véritable plateforme de développement orientée Web. Proche du Java et du C#, sa syntaxe associée à un outillage de qualité favorise une prise en main rapide. Fonctionnant au sein d'une machine virtuelle dédiée, le code Dart est également compilable en Javascript, condition sine qua non à ses chances d'adoption par le plus grand nombre. Les nombreuses bibliothèques standard proposées ainsi que le gestionnaire de dépendances intégré attestent de la volonté de Google de faciliter le développement de projets complexes grâce à Dart. Enfin, son fonctionnement côté client et côté serveur offre de réelles perspectives pour réaliser des applications web avec un seul langage. C'est justement ce qui constituera l'objet de la seconde partie de cet article.

Les Nouveautés d'ASP.NET MVC 5

Les technologies destinées au développement d'applications Web sont en constante évolution. Afin de suivre cette dynamique, Microsoft continue de mettre à jour et d'améliorer ses technologies liées au développement d'applications Web, notamment le framework MVC, qui avec sa version 5 nous amène un lot de nouveautés des plus intéressantes. Nous allons commencer par faire un bref rappel de ce qu'est ASP.NET ainsi que le pattern MVC puis nous explorerons les nouveautés apportées par cette nouvelle version d'ASP.NET MVC 5.

ASP.NET est le standard pour le développement d'applications Web avec les technologies Microsoft. Son utilisation permet aux développeurs de réaliser des sites Web dynamiques, des applications Web et des Web services tout en ayant accès aux fonctionnalités que peuvent offrir le Framework .NET (support du XML, les interactions avec la base de données, email, les expressions régulières, etc...).

Il existe différentes technologies Web sous le nom ASP.NET :

- ▷ **WebForm** : cette technologie se rapproche du monde Winform, le développeur crée des écrans Web en utilisant des composants prêts à l'emploi. C'est le moteur ASP.NET qui se charge ensuite de générer le code HTML/Javascript qui sera utilisé par le navigateur.

- ▷ **ASP.NET MVC** : application du patron de conception MVC, utilisé depuis longtemps dans le monde Java ou Ruby, à ASP.NET.

A chaque nouvelle version d'ASP.NET, Microsoft a su améliorer sa plate-forme et y intégrer des nouveautés intéressantes comme de nouveaux contrôles ou fonctionnalités :

- ▷ Les contrôles de navigation comme le SiteMapPath ou encore le TreeView qui sont apparus dans la version 2.0 d'ASP.NET, ou encore les contrôles de données ListView et DataPager dans la version 3.0,
- ▷ Les fonctionnalités de mise en page comme les Master Pages et les thèmes,
- ▷ L'introduction d'AJAX dans ASP.NET,
- ▷ Le support du HTML 5,
- ▷ ASP.NET MVC et ASP.NET Web API,
- ▷ ASP.NET Web Pages,
- ▷ ...

Qu'est-ce que le MVC ?

Le MVC (Modèle Vue Contrôleur) est un pattern qui permet de séparer différentes couches d'une application. Il permet de séparer les données, le traitement et l'affichage. Le contrôleur interagit avec le modèle et la vue car il manipule les données avant de les transmettre à la vue.

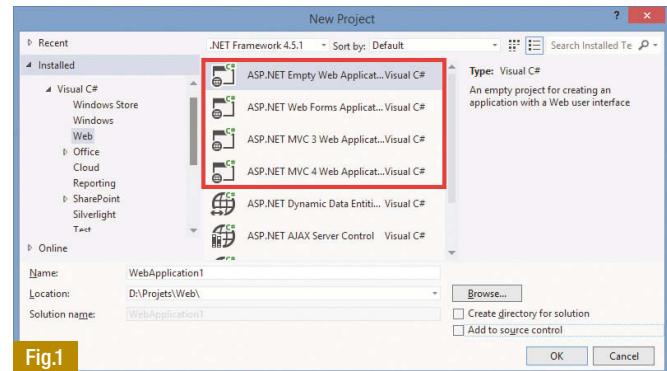
2013 voit l'arrivée de la version d'ASP.NET MVC 5 et de Visual Studio 2013. La plateforme a mûri pour devenir aujourd'hui une des technologies les plus utilisées pour le développement Web et apporte son lot de nouveautés, dont en voici quelques-unes : One ASP.NET, Bootstrap, ASP.NET identity, Attribute routing, Authentication filters, Filter overrides.

Nouveautés d'ASP.NET MVC 5

One ASP.NET

Dans les précédentes versions du Framework, lors de la création d'un nouveau projet web, il nous était demandé de choisir entre différents types de projets. (Fig. 1). Il était donc nécessaire de connaître la technologie que l'on souhaitait utiliser dès le début du projet avec l'impossibilité de modifier son choix en cours de route. Microsoft a modifié la création d'un projet Web afin de pallier ces limitations. Désormais, lors de la création d'un projet Web, nous devons faire un seul choix.

Après avoir sélectionné le type de projet, une nouvelle fenêtre s'affiche, permettant de sélectionner les fonctionnalités que l'on souhaite intégrer à



notre projet. On remarque alors les cases à cocher sous la liste des templates. Il est donc possible de créer un projet MVC avec une référence vers les répertoires Webform afin de lier les deux technologies.

Routing par attribut

Dans les précédentes versions du Framework ASP.NET, le routage des URL était centralisé dans le fichier RouteConfig.cs; au démarrage de l'application le moteur ASP.NET chargeait les routes définies dans la méthode RegisterRoutes. Microsoft a maintenant ajouté la possibilité de définir les routages des URL sur le contrôleur ou sur une action du contrôleur via le routing par attribut. C'est une fonctionnalité très intéressante notamment pour le debugage de l'application, en effet cela permet de voir clairement les URL qui permettent d'accéder à un contrôleur ou une action. Il est bien sûr toujours possible d'effectuer le routage via la méthode RegisterRoutes.

```
[RoutePrefix(<>Accueil>)]
[Route(<>{action=Index}<>)]
public class HomeController : Controller
{
    public ActionResult Index()
    {
        return View();
    }

    [Route(<>aPropos>)]
    public ActionResult About()
    {
        ViewBag.Message = <>Your application description page.<>;
        return View();
    }

    public ActionResult Contact()
    {
        ViewBag.Message = <>Your contact page.<>;
        return View();
    }
}
```

ASP.NET Identity

La nouvelle version du Framework embarque aussi un nouveau système d'authentification : ASP.NET Identity.

ASP.NET Identity comporte de nombreux avantages :

- Il peut être utilisé dans toutes les déclinaisons du Framework ASP.NET : MVC, WebForm, Web API, etc.
- Il est facile de personnaliser les attributs d'un utilisateur. C'est à dire, quand vous créez de nouveaux utilisateurs dans votre application, il est maintenant simple d'ajouter des informations supplémentaires concernant l'utilisateur. Par exemple, si vous voulez ajouter la date de naissance dans le formulaire d'inscription de votre application Web.
- Il utilise Entity Framework Code First afin de stocker les informations en base de données. Mais le mécanisme d'enregistrement peut être modifié afin de répondre aux besoins de votre application. Si vous avez besoin que les informations soient stockées ailleurs comme sur SharePoint, Azure Table Service ou encore dans une base de données NoSQL, il est maintenant possible de le faire.
- Gestion de rôle intégrée, simple à mettre en place et à utiliser.
- Possibilité d'ajouter facilement l'authentification externe (Compte Microsoft, Facebook, etc...) à votre application.
- ASP.NET Identity rend votre application encore plus testable au travers des tests unitaires. Vous pouvez en effet écrire des tests pour les parties de votre application utilisant ASP.NET Identity.

Bootstrap

Dans Visual Studio 2013, le nouveau template d'ASP.NET MVC comprend un grand nombre de bibliothèques très intéressantes pour vous aider à améliorer votre site Web et accélérer votre développement. L'une de ces bibliothèques, incluses par défaut par Microsoft est le « Twitter Bootstrap ». La version du framework de ce dernier est à jour (3.0.0).

Bootstrap est un framework Web très puissant et intuitif qui permet de concevoir des interfaces utilisateur réactives à base de HTML et CSS mais aussi compatibles entre les différents navigateurs.

Filtres d'authentification

Pour ceux qui ont déjà fait du développement ASP.NET MVC, vous avez peut-être déjà dû utiliser le filtre « Authorization » qui permet de renforcer la sécurité sur les rôles au sein de votre application Web. Avec MVC 5, il est maintenant possible d'appliquer un filtre d'authentification à une action ou à un contrôleur, ou encore à l'ensemble des contrôleurs. Il s'agit d'un nouveau type de filtre qui permet aux utilisateurs de s'authentifier au site via un custom provider ou bien un provider provenant d'applications tierces. Ces filtres permettent également de gérer les requêtes non autorisées.

Filter overrides

Dans les versions précédentes de MVC, il n'y avait aucun moyen d'« overrider » un filtre pour une seule action ou controller. Il fallait appliquer le filtre pour chaque action et contrôleur un par un.

Dans l'exemple ci-dessous, on va déclarer la classe HomeController avec le filtre « Authorization » en y indiquant l'utilisateur « Admin ». Ainsi, ce filtre va s'appliquer à toutes les actions à l'intérieur de notre controller.

```
[Authorize(Users = "Admin")]
public class HomeController : Controller
{
    public ActionResult Index()
    {
        return View();
    }
}
```

```
public ActionResult About()
{
    ViewBag.Message = "Your application description page.";
    return View();
}

public ActionResult Contact()
{
    ViewBag.Message = "Your contact page.";
    return View();
}
```

Supposons que vous souhaitez contourner ce filtre seulement pour l'action 'Contact'. Vous souhaitez donner accès à cette page uniquement à l'utilisateur 'David'. Avec MVC 5, vous pouvez le faire en ajoutant deux lignes de code, comme illustré ci-dessous :

```
[Authorize(Users = "Admin")]
public class HomeController : Controller
{
    public ActionResult Index()
    {
        return View();
    }

    public ActionResult About()
    {
        ViewBag.Message = "Your application description page.";
        return View();
    }

    [OverrideAuthorization]
    [Authorize(Users = "David")]
    public ActionResult Contact()
    {
        ViewBag.Message = "Your contact page.";
        return View();
    }
}
```

L'utilisateur «Admin» a donc accès à toutes les actions du controller à l'exception de l'action «Contact» qui n'est accessible que par l'utilisateur «David». Il existe 5 types de «Filter override» correspondant à chaque type de filtres : OverrideActionFilters, OverrideAuthentication, OverrideAuthorization, OverrideExceptionFilters, OverrideResultFilters.

Conclusion

Cet article a pour but de faire un rapide état des lieux des nouveautés d'ASP.NET MVC 5. Nous sommes convaincus que cette migration vaut le coup d'être effectuée. Microsoft enrichit encore son framework et met à disposition des développeurs des nouveautés qui leur facilitent toujours plus la vie.



 David Tran
Consultant chez Cellenza
Software Development Done Right
Son Blog : <http://blog.cellenza.com>



 Eric Galiano
Consultant chez Cellenza
Software Development Done Right
Son Blog : <http://blog.cellenza.com>

Initiation à IntuiFace

La révolution tactile, amorcée avec l'essor des smartphones et des tablettes, s'est accompagnée de la démultiplication d'écrans tactiles dans notre quotidien. Les applications de consultation de catalogues, de présentation produit ou de wayfinding fleurissent dans les lieux publics et les salons; les réaliser exige des compétences de développement et de design spécifiques.

Afin de rendre la réalisation d'expériences interactives accessible aux développeurs, aux designers, voire aux non-initiés, IntuiLab a développé une plateforme logicielle nommée IntuiFace.

IntuiFace est une plateforme associant un éditeur intuitif, le Composer, deux Players autonomes (Windows et iPad), et une interface serveur de gestion des déploiements.

Elle vise à faciliter la réalisation de front-end complexes, réagissant aux interactions naturelles (écrans ou tables tactiles, Kinect, Leap Motion, ...). Elle permet d'y ajouter, par des mécanismes de programmation visuelle des comportements évolutifs. Enfin, les applications réalisées sont aisément interconnectables avec un écosystème complexe. IntuiFace est particu-

lièrement adapté au développement rapide d'applications interactives, riches et connectées, mais aussi de prototypes.

IntuiFace se décline en trois versions. Une version « Free », téléchargeable gratuitement sur www.intuilib.com, permet d'utiliser immédiatement et sans limites le Composer. Pour des usages plus professionnels, des versions « Pro » et « Enterprise » sont aussi disponibles.

Cet article va vous présenter les grands concepts d'IntuiFace au travers de la réalisation d'une application ludo-éducative sur l'Egypte. Elle proposera un écran d'accueil reprenant un slide show d'images et un écran représentant les grandes dates de son histoire (Fig.1).

Présentation des concepts

IntuiFace permet de définir des écrans (**Space**), agrégeant des composants (**Asset**), organisés dans des groupes (**Collections**).

Les assets sont principalement de deux types : Les médias (texte, images, vidéo, objet 3D, document, ...) et les contrôles (boutons, text inputs..)

De même, si certaines collections sont uniquement dédiées à l'organisation visuelle d'éléments (*Group*, *Carousel*...), d'autres se destinent à des usages plus spécifiques (*Timeline*, *Map*,...).

Les assets et les collections gèrent nativement leurs comportements et les interactions associées : un carrousel ajouté dans une application sera directement manipulable par l'utilisateur en multitouch. Mais bien entendu, chacun de ces

éléments est configurable par le biais d'un ensemble complet de propriétés. De plus, chaque composant peut émettre des événements et exécuter des actions paramétrées.

L'interface du Composer ambitionne ainsi de présenter ces concepts, bien connus des développeurs, de manière simple pour faciliter la réalisation de front-end interactifs et communicants.

Interface du Composer

Découvrons le Composer à travers notre exemple. Notre zone de travail est initialement vide. Commençons à construire simplement notre écran d'accueil. La Toolbar (Fig.3) nous permet d'accéder aux différents composants. Choisissons le *Slideshow*, une collection réalisant un ... slideshow entre ses items ! Nous pouvons le positionner sur la scène, de manière à ce qu'il prenne tout l'écran.

Ses propriétés communes sont modifiables au travers du panneau dédié, et l'ensemble des propriétés est accessible par le biais de la fenêtre que fait apparaître le bouton « Show all properties ».

Pour ajouter des images à notre Slideshow, nous pouvons utiliser le bouton '+' de la Content Library, à droite. Ensuite, un simple drag'n'drop suffit à les ajouter sur le Space ou comme contenu de notre slideshow (Fig.4).

Ajoutons aussi le texte d'accueil, et, enfin, un

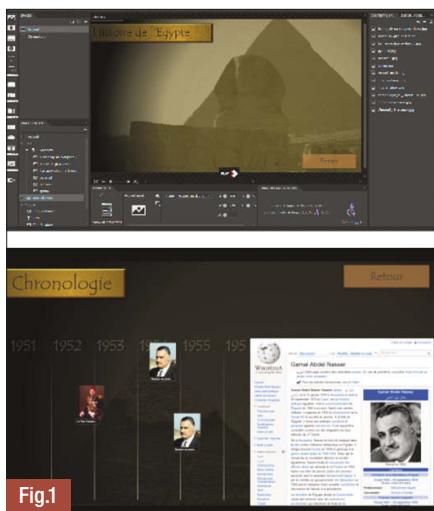
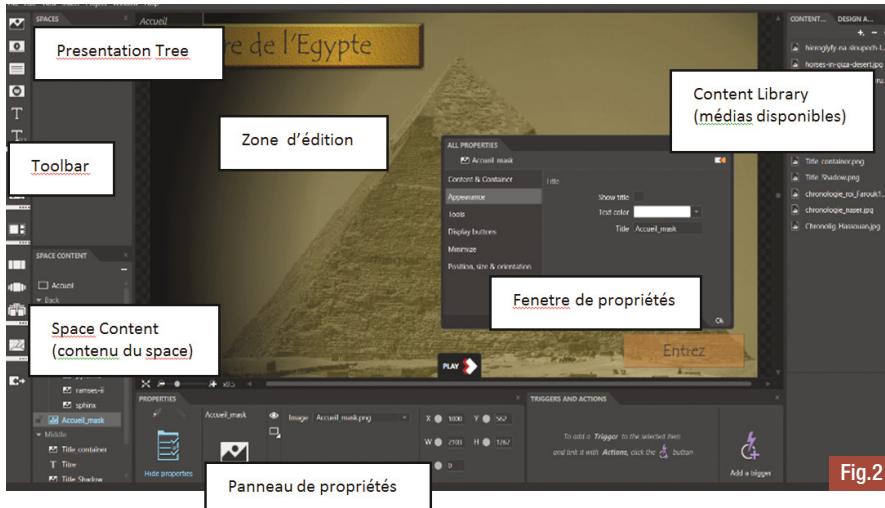


Fig.1
Interface générale du Composer



Ecrans finaux de l'application

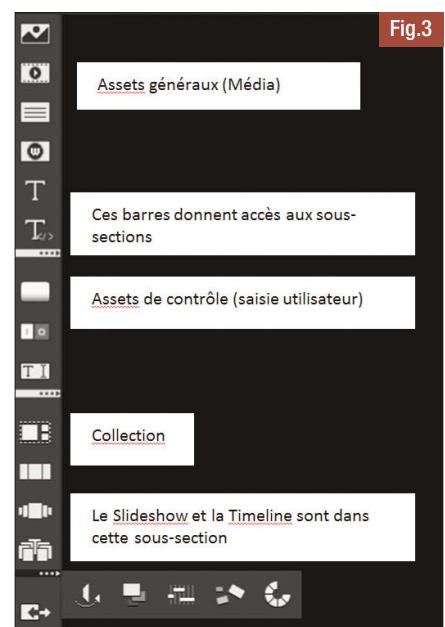


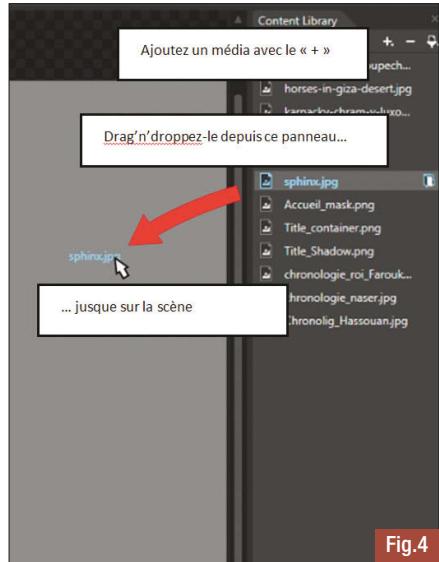
Fig.3

bouton pour accéder à la chronologie. Vous pouvez éventuellement ajouter d'autres médias, pour enrichir le design de votre page d'accueil. Passons maintenant à l'écran dédié à la chronologie. Pour cela, vous devrez passer par le Presentation Tree, qui permet (entre autres) d'ajouter et de sélectionner les Spaces.

L'idée de cet écran est d'afficher une échelle de temps (timeline). Au clic sur un élément, un navigateur web s'affiche avec la présentation Wikipedia associée à la date.

Ajoutons notre collection *Timeline* depuis la toolbox, et insérons-y, comme précédemment, des éléments (images mais aussi des vidéos, des objets 3D,...).

Déplacez vos éléments à leurs dates respectives, ou fixez directement leur propriété *Start Date* dans le panneau de propriétés. Notez que vous avez, à votre disposition le **Property Painter**, proche cousin du pinceau de Word, pour copier la mise en forme d'un élément sur un autre.



Ajout d'un média

Comme toutes les collections, la Timeline possède des propriétés permettant d'affiner les interactions sur les éléments contenus. Ainsi, cocher la propriété « *Tap item to open* » permettra aux utilisateurs de sortir un élément de la Timeline en appuyant dessus.

Pour finir, ajoutons le composant Navigateur Web qui affichera les détails d'un événement historique. Changez sa visibilité par défaut pour le cacher : il ne devra apparaître qu'au tap sur un élément de la timeline.

Les éléments sont en scène, il nous reste désormais à donner vie à notre application !

Trigger et Actions

Les mécanismes généraux d'interactions entre assets reposent sur l'invocation d'**Action** en réponse à un **Trigger** (« When Trigger then Action »).

Un **Trigger** est un événement levé par un asset ou résultant de la reconnaissance d'un geste ou d'un tag, de l'expiration d'un timer,...

Une **Action** permet d'agir sur un élément de la présentation (pour le cacher, changer sa source...). L'édition des triggers / actions s'effectue par la fenêtre dédiée (**Fig.5**) : lorsqu'un asset est sélectionné, ses triggers apparaissent dans le panneau en question. Bien entendu, un trigger peut déclencher plusieurs actions.

Commençons simplement par définir la logique de navigation de l'écran d'accueil :

Selectionnez le bouton « Entrer » de la page d'accueil : le panneau d'action indique qu'un Trigger par défaut (*Released*) est défini. Cliquez-le pour afficher la fenêtre d'édition : celle-ci rappelle, dans sa partie haute, le trigger (modifiable, au besoin), et permet de lui associer des actions. La partie inférieure permet de choisir une action et d'éditer ses paramètres... Les actions générales (navigation, quitter, appeler une URL,...) sont accessibles aux travers des deux catégories générales **Space-to-Space**

Fig.5 Edition des Trigger/Actions

Navigation et Overall Experience, et une représentation du contenu du Space courant permet d'accéder aux actions de chaque asset (il est aussi possible d'accéder aux éléments d'un autre Space). Pour notre bouton, le Trigger *Released* nous convient. Assignez-lui l'action de navigation « *Go To Space* », et choisissez « Chronologie » comme cible.

Sur le second écran, nous allons définir la dynamique des éléments de la Timeline : pour chaque élément, ajoutez-lui, sur le trigger *Tapped* (section *Touch*), une action *Show* ciblant le Web Browser pour l'afficher et une action *Open URL* menant sur l'adresse de la page Wikipedia souhaitée.

Et voilà ! Il vous suffit d'appuyer sur F5 pour voir fonctionner votre première application !

Interface Assets et Bindings

Les **Interface Assets** (IA) définissent des points d'extensions à IntuiFace. Ils permettent d'intégrer des composants externes, définis en Excel ou par des Web Services REST et des DLL, comme sources de données, de services ou pour définir une logique métier à l'application.

Cette intégration passe par la définition d'un fichier JSON de contrat (suivant la syntaxe Google Discovery), permettant de définir le mapping entre le composant externe et le Composer. La syntaxe de ce fichier pourrait faire l'objet d'un tutoriel propre, nous vous orientons donc vers les ressources présentes sur le site d'IntuiFace. Une fois ce fichier défini, l'élément sera disponible dans le Composer. Il pourra être instancié et ses propriétés, événements et actions seront automatiquement accessibles partout dans l'application.

Pour commencer, ajoutons un IA gérant l'heure courante (il fait partie des IA prédéfinis et fournis avec IntuiFace), et affichons-la sur l'écran d'accueil : Ouvrez la toolbar de choix des IA (**Fig.6**), et sélectionnez « *Clock* » dans la liste. L'IA ins-

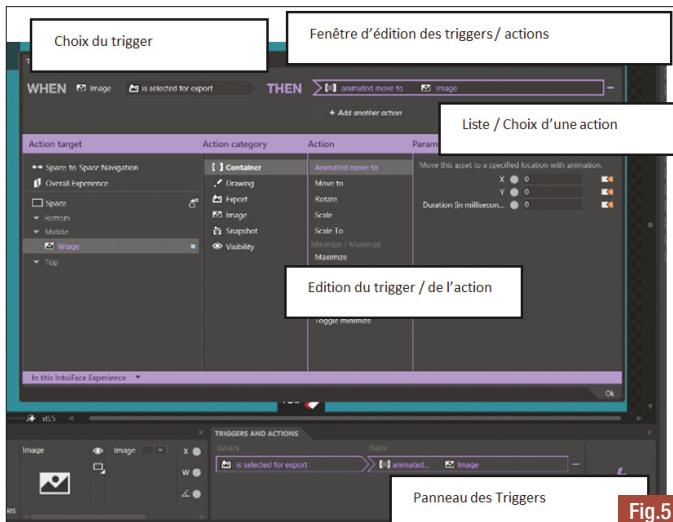


Fig.5

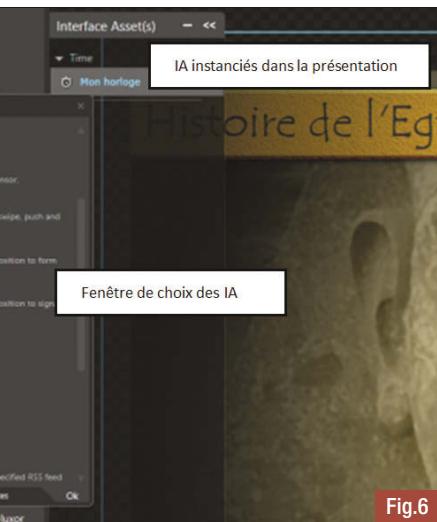


Fig.6

tancié apparaît désormais dans le tiroir dédié. Nous pouvons le configurer, nous abonner à ses événements (par ex. *Reached Time*) ou invoquer ses services. Il nous reste désormais à lui définir une représentation. Nous exploiterons pour cela le mécanisme de liaison de données d'IntuiFace : le **Binding**.

Connus des développeurs IHM, ce principe permet de lier une propriété cible à une propriété source de n'importe quel élément. La cible reflète automatiquement tout changement de la source. Ainsi, IntuiFace permet de lier entre elles des propriétés d'Asset, de Collections ou d'Interface Asset. Il est ainsi facile de synchroniser l'affichage de composants IHM ou de les lier à des données externes.

Revenons à notre projet et créons un Asset **Text**. Passez la fenêtre de propriétés en mode **Binding** en cliquant sur le bouton dédié. La fenêtre s'agrandit (Fig.7) : Dans la partie droite apparaissent les propriétés sources auxquelles se connecter Choisissez la *Clock* puis la propriété *Hour* et reliez-la à la propriété *Text* de l'Asset **Text** : votre texte affiche maintenant l'heure courante.

Collection de données, templates

IntuiFace permet d'aller encore plus loin en permettant par exemple l'affichage de larges collections de données structurées.

Dans notre cas, nous prendrons l'exemple de données issues d'un IA très pratique : L'IA

Excel. Il expose directement sous forme de propriétés les données d'un tel fichier. Il nous servira ici à peupler notre Timeline de données.

Créez un fichier Excel reprenant la structure illustrée (Fig.8). Choisissez d'ajouter un IA Excel dans le Composer et sélectionnez ce fichier. L'IA importé proposera entre autres propriétés une collection d'éléments reprenant nos colonnes.

Drag-and-dropper votre Interface Asset sur le Space courant : il apparaîtra comme sur l'illustration suivante. Cette collection définit un **Data Template** qui sera utilisé pour représenter chaque élément de la collection (dans notre cas, les lignes du fichier Excel) (Fig.9).

Changeons le style de la collection principale en **TimeLine**.

Plutôt que d'utiliser le **Data Template** par défaut, définissons-en un nouveau. Pour cela, créons dans le Space un groupe qui nous servira de modèle, en reproduisant l'illustration suivante. Bindez vos champs sur les propriétés du premier élément de la collection. Après quelques manipulations on obtient le visuel suivant : Fig.10.

Replacez votre groupe dans la collection et, automatiquement, tous les éléments de notre collection vont appliquer le même modèle.

Pour positionner nos éléments à leurs dates respectives, ouvrons la fenêtre de Binding sur la racine de notre Data Template : Nous y voyons apparaître le nom de notre IA, reprenant la donnée présentée par le template. Il nous suffit de

lier la propriété « *Start Date* » de la racine du Data Template à la propriété « *Date* » de cette donnée. Enfin ajoutons sur le Trigger « *Tapped* » de la racine du template, une action *Open URL* sur le Web browser : Plutôt que rentrer l'Url en dur, nous allons lier la propriété « *Url Wikipedia* » de la donnée au paramètre de l'action *Open URL* (Fig.11) et le tour est joué ! (Fig.12)

C'est fini ! Lancer l'application par F5 pour observer le résultat final.

Déploiement

Une fois votre projet terminé, vous pouvez en quelques clics depuis l'interface Web de gestion le déployer sur des Players installés sur votre flotte de machines, sans intervention de votre part sur les postes clients.

Enfin, vous pourrez, sur l'App Store d'Apple, télécharger une version iPad du Player IntuiFace. Cette nouvelle version du client peut exécuter les mêmes applications que le client Windows. Elle vous permet notamment d'exécuter votre application depuis le composer, lors du lancement par F5, si votre iPad et l'ordinateur exécutant le Composer partagent le même réseau Wifi. Seuls quelques Assets et IA ne sont pas encore supportés, et le gap est amené à se réduire avec le temps.

Conclusion

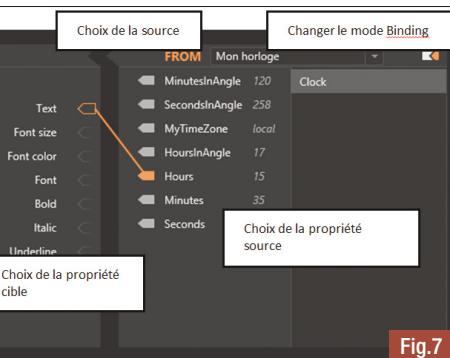
Ce court tutoriel vous a présenté les grands principes d'IntuiFace, et n'a qu'effleuré ses possibilités. Les Interfaces Assets vous permettront de vous intégrer à un environnement complet, accéder ou contrôler des sources de données complexes, voire de contrôler, sans code, des objets connectés (vous trouverez des exemples sur le site Web www.intuilab.com).

Que vous soyez développeur, designer, ou simple hobbyiste, IntuiFace pourra vous rendre bien des services ! N'hésitez pas à le télécharger, et à nous faire des retours sur notre site www.intuilab.com, sur notre compte Facebook « IntuiFace » ou via Twitter, en suivant @IntuiLab.

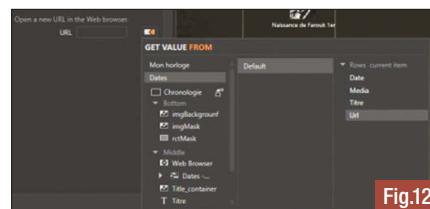
Roland Tomczak
tomczak@intuilab.com

A	B	C	D
3	Date	Media	Titre
2	11/02/1920	http://upload.wikimedia.org/wikipedia/commons/Naissance_de_Farouk_1er.jpg	Naissance de Farouk 1er
3	15/01/1918	http://upload.wikimedia.org/wikipedia/commons/Naissance_de_Gamal_Abdel_Nasser.jpg	Naissance de Gamal Abdel Nasser
4	16/01/1952	http://upload.wikimedia.org/wikipedia/commons/Naissance_de_Farouk_II.jpg	Naissance de Farouk II

Fichier Excel utilisé comme source de données



Création d'un Binding



Utiliser les données comme argument d'action



Notre Data Template édité



Assignation d'un Template

A	B	C	D
11/02/1	http://	Naissance	http://
920	upload.wikimedia.org/wikipedia/commons/Naissance_de_Farouk_1er.jpg	Media	fr.wikipedia.org/wiki/Farouk_1er
15/01/1	http://	Samal	http://
918	upload.wikimedia.org/wikipedia/commons/Naissance_de_Gamal_Abdel_Nasser.jpg	Media	fr.wikipedia.org/wiki/Gamal_Abdel_Nasser
16/01/1	http://	Naissance	http://
952	upload.wikimedia.org/wikipedia/commons/Naissance_de_Farouk_II.jpg	Media	fr.wikipedia.org/wiki/Farouk_II

Représentation générée de l'IA Excel

eMobilité et responsive design : l'art de s'adapter



La révolution mobile est plus que jamais en marche, plus de 23 millions de français ont déjà accédé à Internet via leur mobile et plus de 3 millions de foyers possèdent déjà une tablette tactile en France. Résultat, toutes ces différentes tailles d'écran et de façons de naviguer sur le web poussent les entreprises à revoir leur présence multi-écrans et à proposer de nouvelles expériences utilisateur plus confortables et ergonomiques.

Le responsive design répond à ce défi en permettant d'optimiser ou d'adapter automatiquement l'affichage d'un site web en fonction du terminal utilisé. Désormais, les internautes profitent d'une expérience utilisateur unique et séduisante ; le site web responsive peut changer de forme, s'adapter automatiquement selon la taille d'écran et l'orientation.

Concrètement, comment ça marche ? Le responsive design est rendu possible grâce à l'évolution des divers langages de programmation utilisés tels que le HTML5 et les nouvelles propriétés CSS3.

Dans la pratique, mettre en œuvre un site caméléon qui saura s'adapter automatiquement à chaque terminal en fonction de ses caractéristiques spécifiques est délicat. Il faut reconstruire tous les cas de figure possibles, les agencements de contenus, les redimensionnements d'images et optimiser l'utilisabilité du site, peu importe le support.

QUELQUES AVANTAGES

Avant d'aller plus loin dans le développement en responsive design, voici quatre avantages très intéressants de cette pratique. D'abord, le temps de maintenance est fortement réduit car l'adaptation du site web se fait côté client avec un script serveur commun. Deuxième avantage, l'adaptabilité du produit. En effet, peu importe le terminal utilisé, le contenu, le design et les fonctionnalités du site web seront adaptés automatiquement. Résultat, le responsive design garantit une expérience utilisateur homogène sur l'ensemble des supports et permet ainsi de conserver une certaine cohérence et plus-value pour l'image de marque de l'entreprise.



Troisième avantage et pas des moindres, le responsive design est la technique recommandée pour optimiser le référencement naturel d'un site. Favorisée par Google, le poids des pages (Google juice) est concentré sur une URL unique. D'ailleurs, la mise en place d'une URL unique permet aussi d'éviter la duplication de contenus et favorise le netlinking.

Dernier avantage, gardez à l'esprit que les coûts de production pour un site web en responsive design sont moins élevés que pour le développement de versions séparées, ou encore d'applications mobiles.

A moyen et long terme, le responsive design offre à ses utilisateurs une meilleure réactivité du site web, et surtout s'adapte pleinement à leurs besoins d'accéder en permanence à l'information.

Cependant, rien n'est parfait. Le responsive design impose des contraintes techniques lourdes sur la conception et implique de revoir en profondeur les rôles et expertises de l'équipe projet.

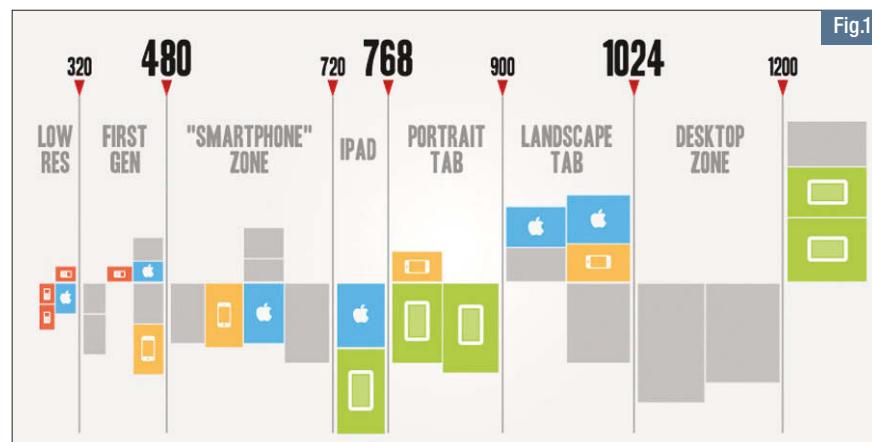
CONTRAINTE ET SOLUTIONS

Si vous vous lancez dans la conception de sites en responsive design, vous risquez de vous retrouver face à certains obstacles.

Tout d'abord, la phase de conception est plus lourde. En effet, concevoir un site responsive implique de repenser l'interface sous forme de grilles et de blocs, de réfléchir à l'architecture des contenus, et encore de décider des points de rupture du design du site.

Un autre inconvénient majeur, le temps de chargement est plus long en raison des ajustements d'images et de contenus. Lorsque l'on sait qu'une seconde de délai de réponse sur un site web peut entraîner une perte de 7% du taux de transformation selon les études de Gomez et Akamai, il devient crucial de maîtriser le temps d'affichage des pages. Une gestion fine des images flexibles permet de pallier cette limite notamment avec l'utilisation de javascript.

La compatibilité avec les différents navigateurs est un point sensible en raison des codes HTML et CSS plus travaillés que pour un site classique. En effet, tous les navigateurs ne gèrent pas les media queries notamment IE7 et IE8. Il existe toutefois plusieurs librairies javascript comme par exemple le plugin jQuery qui permettent de contourner cette limite.



Troisième inconvénient, l'architecture de l'information des écrans nécessite un temps de développement supérieur à un site classique.

Ainsi, certes le coût d'un site en responsive design reste inférieur aux coûts cumulés d'un site en version mobile et d'une application native, mais il vous faut malgré tout compter sur des coûts de développement plus élevés puisque la création d'un site responsive nécessite des temps de développement plus longs qu'un site web classique.

ALLER A L'ESSENTIEL

La réalisation d'un site en responsive design implique de réaliser au préalable une interface sous forme de grille et de blocs. Les blocs doivent être distincts les uns des autres et il faut éviter d'inclure une information d'un bloc dans un autre.

Une fois la grille de blocs réalisée, il suffit de déplacer les blocs en fonction des familles de résolutions **Fig.1**.

Une bonne solution est de commencer la conception d'interfaces responsive du plus petit format (approche appelée *Mobile first*) au plus grand. Cette approche permet de se focaliser sur l'affichage des contenus, les fonctionnalités clés et l'expérience utilisateur enrichie.

Cependant, en fonction du projet et des cibles, il peut être préférable de produire un layout dense pour une version bureau qui sera à priori plus utilisée par les internautes, puis décliner chaque gabarit de page pour des écrans moins larges. En réalité, à chaque projet ses spécificités. D'ailleurs, pourquoi ne pas opter pour le développement cumulé d'un site classique et d'un site en version mobile ?

RESPONSIVE DESIGN VS SITES MOBILES : QUE CHOISIR ?

Les paris sont ouverts, entre responsive web design et développement de sites en version mobile, quelle technologie est la plus efficace ?

Sans prendre en compte les considérations techniques et budgétaires, la meilleure solution serait bien sûr de proposer une déclinaison sur mesure pour chaque support utilisé par le consommateur. Une telle proposition, non seulement bien utopique, serait rapidement obsolète au vu de la recrudescence des terminaux aux formats différents.

L'ORGANISATION DU CONTENU

Point clé du responsive design, l'assurance d'un contenu unique et donc d'une expérience utilisateur homogène quel que soit le terminal utilisé pour la consultation du site s'avère pratique mais nécessite malgré tout d'avoir peu de textes, d'images et de photos afin de ne pas alourdir la consultation sur les écrans plus petits.

A contrario, il est vrai qu'un site mobile permettra de sélectionner le contenu que l'on souhaite proposer et surtout de les organiser à votre manière pour améliorer la lisibilité du site.

Inconvénient majeur pour les sites en version mobile, vous risquez des pénalités de Google car ce dernier assimilera la version mobile de votre site à du duplicate content. Au final, vous encourrez une baisse de votre positionnement sur les pages de résultats du moteur de recherche.

Le choix d'un développement en responsive design permet donc d'éviter ces risques de duplication de contenu, et aura l'avantage de ne pas perdre l'utilisateur; il lui sera offert une expérience unique et cohérente sur l'ensemble de ses supports.

Autre élément important, les backlinks, c'est-à-dire les liens d'autres sites web pointant sur le vôtre, sont importants en SEO.

Avec une URL unique, vous évitez l'éparpillement de l'information et concentrez tous les backlinks.

Si toutefois vous faites le choix de développer une version mobile, optez pour une solution CMS fiable qui facilite le partage de contenu et évite ainsi le copier-coller.

UN ENJEU DE TAILLE : L'EXPERIENCE UTILISATEUR

Construire une version mobile dédiée, ou encore une application mobile, permet de proposer aux utilisateurs une expérience utilisateur souvent plus riche. En effet, en proposant un site totalement différent du site web classique, il sera possible d'ajouter des contenus, des pages, des fonctionnalités spécifiques.

Cependant, de tels développements impliquent bien sûr des coûts et des délais de production importants.

Le responsive design, loin d'être une solution miracle en définitive, s'impose plutôt comme un excellent compromis pour fournir une expérience utilisateur cohérente sur l'ensemble des supports.

En fonction du projet et des cibles, il peut être pertinent de développer un site responsive et de le coupler à une application mobile native qui permettra d'utiliser au mieux les fonctionnalités du terminal, les rendus graphiques et optimisera l'expérience utilisateur.

DÉMYSTIFIER LE DÉVELOPPEMENT EN RESPONSIVE DESIGN

Les media queries (CSS3), les grilles fluides et les images flexibles sont les composants fondamentaux du responsive design. Examinons en détails comment maîtriser ces outils pour répondre aux attentes des internautes. Critères en CSS3, les media queries permettent d'adapter dynamiquement le design en fonction des caractéristiques de chaque support utilisé et de la taille d'écran correspondante. Autre élément clé, les grilles fluides permettent de créer des interfaces composées de blocs en unités relatives. Cette conception modulaire permet ainsi de jouer sur la composition de la page en fonction de la largeur de l'écran.

Enfin, il est aussi essentiel de dompter les images, et ainsi de proposer des images dont la taille et la disposition s'ajustent en fonction de l'écran. Examinons tous ces aspects.

JOUER AVEC LES MEDIA QUERIES

Les media queries sont de nouvelles règles CSS spécialement destinées pour les media (écrans). Ils permettent donc l'application de feuilles de styles différentes en fonction des terminaux cibles.

Avec CSS2 et HTML4, il était déjà possible de spécifier une feuille de style en fonction du media par exemple pour l'impression. L'attribut *media* précisant le contexte dans lequel les styles doivent s'appliquer était limité : *print* pour l'impression, *handheld* pour les terminaux mobiles et de petite taille, *speech* pour les lecteurs vocaux, *embossed* pour les imprimantes braille...

```
Media "print"
<link rel= "stylesheet" media= "print"
      href = "print.css" type= "text/css"/>
```

L'objectif des media queries est d'élargir les possibilités en combinant un ensemble de critères à partir des capacités des terminaux et non à partir du type des terminaux. Les différentes déclarations possibles sont associées avec un opérateur logique pour définir l'ensemble des conditions à réunir pour l'application des styles compris dans le bloc.

Les combinaisons sont multiples.

L'exemple suivant vise les écrans de largeur inférieure à 480 pixels.

```
Media "screen" et terminal de 480px de largeur maximale
@media only screen and (max-device-width: 480px) {
    div#wrapper {
        width: 400px;
    }
}
```

L'exemple ci-dessous cible les écrans de largeur supérieure à 800 pixels grâce à la règle **max-width** associée à la valeur 800px.

```
Media "screen" et une largeur maximale d'écran
supérieure à 800px

<link rel="stylesheet" type="text/css"
media="screen and (max-width > 800px)"
href="css/desktop.css"/>
```

```
Media queries utilisées dans Twitter bootstrap

/* Landscape phones and down */
@media (max-width:480px) { ... }
/* Landscape phones to portrait tablet */
@media (max-width:767px) { ... }
/* Portrait tablet to landscape and desktop */
@media (min-width:768px) and (max-width:979px) { ... }
/* Large desktop */
@media (min-width:1200px) { ... }
```



ADOPTER UNE MISE EN PAGE MODULAIRE

Utiliser une conception modulaire est le meilleur moyen d'être prêt à toute taille d'écran/orientation. Les grilles flexibles utilisent des colonnes pour organiser le contenu, et un *width relatif* plutôt que fixe afin de s'adapter au viewport.

Ainsi, les éléments fluides occupent le même pourcentage d'espace sur les différents écrans. Par exemple un élément prendra 50% de la largeur, quel que soit le terminal.

```
Tailles de blocs en %

#nav
{
    float: left;
    width: 25%;
    height: 150px;
    background-color: #999;
    margin-bottom: 10px;
}

#content
{
    float: left;
    margin-left: 5%;
    width: 65%;
    height: 150px;
    background-color: #999;
    margin-bottom: 10px;
}

#footer
{
    clear: both;
    height: 50px;
    background-color: #666;
    margin-bottom: 10px;
}
```

DOMPTER LES IMAGES

Plusieurs solutions sont envisageables pour adapter automatiquement la taille et la disposition des images. Attention cependant à prendre en compte la taille en pixels et le poids de l'image.

> Définir les images en unités relatives

Un cas fréquent est de définir les images en unités relatives, le redimensionnement de l'image est alors laissé au navigateur.

Si cette solution a l'avantage de la simplicité, le résultat est malheureusement loin d'être optimal en terme de qualité, et le poids de l'image reste inchangé, ce qui peut poser problème lors de connexion à faible débit (GPRS, Edge, ...).

> Opter pour les media queries

Une autre solution est d'afficher certaines images (par exemple celles liées au design) à l'aide des media queries.

Par exemple, en définissant une feuille de style dédiée aux écrans de moins de 480px, les images utilisées dans cette feuille de styles peuvent être spécifiquement adaptées à cette résolution : le redimensionnement est fait à l'avance par un logiciel spécialisé et le poids de l'image est réduit. Pour les images intégrées dans le corps de la page, il suffit de déclarer chacune des images et d'afficher/cacher les images à l'aide des media queries.

Tous les navigateurs ne gèrent pas les media queries notamment IE7 et IE8. Toutefois, il existe plusieurs librairies javascript qui permettent de contourner cette limite : *adapt.js* ou *encore css3-mediaqueries.js*

Attention toutefois, le navigateur télécharge toutes les feuilles de styles ! Le principal défaut de cette méthode est que les images cachées seront aussi téléchargées.

> Utiliser javascript et les media queries

L'utilisation de javascript est une troisième piste intéressante puisqu'elle permet d'optimiser le chargement des images, et d'éviter que le navigateur ne charge des images inutiles ; les media queries étant utilisées pour le reste.

En utilisant le plugin jQuery, il est possible de tester la taille de la fenêtre `$(window).width()` au chargement de la page `$(document).ready(function())` et d'appliquer une expression régulière pour modifier la source de toutes les images afin d'y ajouter une information de taille. Il faut bien sûr disposer sur le serveur d'un outil de redimensionnement des images ou, simplement, d'images déjà redimensionnées.

Si cette solution prend en considération la taille de la fenêtre au chargement, elle laisse cependant de côté l'ajustement d'affichage pour les périphériques pouvant être orientés (passage de portrait à paysage).

D'autres solutions sont envisageables notamment en jouant sur l'image de fond.

> Adopter HTML5 et le plugin jQueryPicture

Le plugin `jQueryPicture` permet d'ajouter des fonctionnalités pour gérer les images en responsive. Il propose l'utilisation de deux balises : `<figure>` ou `<picture>`. Ces balises ont été proposées à la standardisation, mais pas encore validées.

Nous pouvons espérer qu'elles soient implémentées en natif dans les navigateurs dans un futur proche.

Il est possible d'utiliser le tag `<figure>`, ou le tag `<picture>` et le tag `<source>`. La balise `<noscript>` est utilisée pour la compatibilité avec les autres navigateurs.

Pour utiliser la balise `<figure>`, il suffit d'initialiser le plugin dans le header.

```
<script type="text/javascript">
    $(function() {
        $('figure.responsive').picture();
    });
</script>
```

Puis dans le code de déclarer la `<figure>` en utilisant `data-mediaXXX`. XXX est remplacé par la taille souhaitée.

```
<figure class="responsive" data-media="image-400px.png"
data-media600="image-600px.png"
data-media800="image-800px.png"
title="Test picture">
    <noscript>
        
    </noscript>
</figure>
```

Pour insérer un lien sur l'image, il suffit d'entourer la balise `<noscript>` par le lien `<a href>`

```
<figure class="responsive" data-media="image-400px.png"
data-media600="image-600px.png"
data-media800="image-800px.png"
title="Test picture">
  <a href="http://www.ametys.org">
    <noscript>
      
    </noscript>
  </a>
</figure>
```

Pour utiliser la balise `<picture>`, il faut de même initialiser le plugin dans le header.

```
<script type="text/javascript">
  $(function() {
    $('picture').picture();
  });
</script>
```

Puis dans le code de déclarer la `<picture>` en utilisant des `<source>` et des attributs *media* sous forme de media queries.

```
<picture alt="Mon image">
  <source src="image-400px.png">
  <source src="image-600px.png" media="(min-width:640px)">
  <source src="image-800px.png" media="(min-width:840px)">
  <noscript>
    
  </noscript>
</picture>
```

Concernant la gestion des images, le CMS tient un rôle très important. Veillez donc à choisir une solution efficace.

L'image téléchargée par l'internaute doit être optimisée en termes de taille, de qualité et de poids.

15 OUTILS POUR BIEN DÉMARRER

Nous allons dans cette dernière partie parcourir rapidement quelques outils gratuits disponibles sur la toile pour faciliter la vie des concepteurs et intégrateurs de sites internet.

Liste non exhaustive, voici une petite sélection d'outils efficaces.

3 OUTILS POUR COMMENCER !

> Simple Grid pour une interface souple et extensible

Outil très complet, qui permet de simplifier la mise en page d'une grille par l'utilisation d'une classe par élément. Le point fort de l'outil : une utilisation particulièrement simple !

> Foundation pour créer facilement des prototypes

Foundation est un framework très complet se basant sur Sass et sur Compass plutôt que sur du simple CSS.

Il utilise jQuery pour certains éléments graphiques.

L'outil, plus léger et plus simple que Twitter Bootstrap, permet de créer rapidement un prototype, et dispose de nombreux éléments graphiques. Un prototype facilite la compréhension et la simulation de l'ergonomie.

> Adapt.js

Petit javascript très efficace, *Adapt.js* permet de définir le bon CSS en fonction de la résolution de l'internaute.

Il peut détecter plusieurs dimensions de plusieurs navigateurs et servir en fonction les feuilles de style nécessaires.



FitText
A jQuery plugin for inflating web type [Download on Github](#)
FitText makes font-sizes flexible. Use this plugin on your fluid or responsive layout to achieve scalable headlines that fill the width of a parent element.

2 OUTILS POUR DES IMAGES RESPONSIVE

> Adaptive Images : une solution efficace

Adaptive Images est une solution très pratique pour avoir des images responsive, l'outil génère automatiquement des images flexibles optimisées en fonction de la résolution d'écran du visiteur.

> FitText pour ajuster la taille des textes et titres

FitText est un plugin jQuery très pratique qui permet d'ajuster automatiquement la taille des textes et titres en fonction de la résolution d'écran.

2 OUTILS PRÉTS A L'EMPLOI OU «BOILERPLATES»

> Skeleton : un des meilleurs framework responsive

Skeleton utilise seulement des CSS. L'outil est particulièrement adapté à de petits sites et sa prise en main est rapide.



> Bootstrap : un framework simple et complet

Twitter bootstrap utilise jQuery. Il inclut de nombreux éléments graphiques et permet de mettre en place rapidement un site multi-terminal.

3 OUTILS POUR TESTER LE RESPONSIVE DESIGN D'UN SITE

> Responsive Web Design Testing Tool

Outil bien utile pour tester le responsive de votre site. Il suffit d'indiquer l'URL de votre site dans la barre d'adresse prévue à cet effet pour tester une page spécifique dans différentes résolutions. Cependant, il est pour l'instant impossible de naviguer dans votre site directement au sein du service.

> Screenfly

Outil très simple d'utilisation et agréable à prendre en main, Screenfly permet de tester un site sur les terminaux les plus populaires. Il suffit d'entrer l'URL du site web et de se laisser guider !

> Responsinator

Le site Responsinator permet de visualiser rapidement l'affichage d'un site web sur les mobiles et tablettes les plus connus. Le service ne propose pas un rendu extrêmement fidèle, cependant il sera suffisant pour se faire une opinion rapide des axes d'amélioration à travailler pour vos sites responsive.



Frédéric Ravetier

Directeur Projet chez Anyware Services, éditeur du CMS Open Source Ametys - <http://www.ametys.org>

Timeline : 1987

Objet : Amiga 500 contre Atari 520 ST

À la fin des années 80, l'informatique personnelle passe à la génération 16 bits. Il n'y a plus qu'Atari et Commodore pour séduire les passionnés, lesquels partiront dans une guerre de clochers.

Fallait-il passer à l'Atari 520 ST ou à l'Amiga 500 de Commodore ? À la fin des années 80, le débat fait rage pour savoir quel modèle de dernière génération 16 bits allait supplanter dans les foyers les premiers micro-ordinateurs 8 bits. La polémique est d'autant plus passionnée qu'il n'y a alors plus que deux constructeurs dans la course. Les autres, soit, ont disparu, soit, comme Amstrad, se sont reconvertis dans la fabrication de compatibles PC. À cette époque où Internet n'existe pas encore, le PC reste une machine essentiellement vendue hors de prix aux professionnels (entre 10000 et 15000 francs de l'époque) pour faire de la bureautique en mode texte. Les caractéristiques techniques du PC sont nulles et le grand public s'en désintéressera tant que rien de ludique ne sera publié dessus. Apple, tout de même, pousse mollement un Apple IIGS 16 bits sans logiciel pour soi-disant reprendre le flambeau de ses 8 bits à succès. Mais l'essentiel de l'énergie du constructeur est désormais consacrée au Mac, lancé depuis 84, qui caracole en couverture des magazines comme la machine la plus belle du monde avec sa souris et son interface graphique. Mais à 25000 Frs sans disque dur, peu nombreux sont ceux qui peuvent se l'offrir. Justement, l'Amiga 500 et l'Atari 520 ST débarquent alors pour apporter au grand public le fantasme de l'expérience du Mac (souris, processeur 68000, disquettes 3,5 pouces), avec en plus des capacités graphiques inédites.

Deux mêmes machines

De prime abord, Amiga 500 ou Atari 520 ST, c'est bonnet blanc et blanc bonnet. Les noms se ressemblent, l'apparence - un clavier gris clair/blanc cassé fortement incliné et allongé à l'arrière pour accueillir la carte mère et le lecteur de disquettes - est identique, le processeur et la mémoire (512 Ko) sont les mêmes. Et puis leur histoire s'entremêle : élaboré par le concepteur des Atari 8 bits, le projet Amiga a été mis au point au sein d'une startup, Hi-Toro, dès 1982, et devait à terme retourner chez Atari. Mais en 1984, le patron démissionnaire de Commodore, Jack Tramiel; il rachète Atari



L'Amiga 500 était plus beige, l'Atari 520 ST plus gris, mais il s'agissait globalement du même type de machine pour adresser le même marché.



Les premières versions des machines, sorties durant l'été 1985, étaient l'Amiga 1000 dans un boîtier desktop et l'Atari 520ST avec lecteur de disquettes et alimentations externes.

avec la volonté de remplacer tout le monde par son ancienne équipe. Hi-Toro préfère alors se vendre avec son projet Amiga à Commodore. Rageant de voir l'affaire lui passer sous le nez, Jack Tramiel presse ses ingénieurs de sortir un clone Atari de l'Amiga avant Commodore. L'Atari ST et l'Amiga arriveront en définitive tous les deux sur le marché durant l'été 1985. Les premières versions des machines (l'Amiga 1000 qui ressemblait plutôt à un PC de bureau et l'Atari 520 ST dont le lecteur de disquettes était séparé) n'ont pas de BIOS (on dit « ROM », à l'époque) et doivent charger leur système sur disquette à chaque redémarrage, ne laissant plus que 256 Ko de mémoire disponible. Ce qui est inutilisable. Il faudra attendre 1986 pour que l'Atari ST intègre le lecteur de disquettes ainsi que le BIOS et devienne plus exactement un Atari 520 « STf ». L'Amiga 500, qui copie le boîtier du 520 STf et intègre aussi le BIOS, arrive, lui, en 1987.

mêmes graphismes sur les deux plateformes. À la rigueur, seule la qualité d'écoute permet à l'Amiga de faire la différence. Les puces audio sont en effet incompatibles entre les deux machines, ce qui oblige les éditeurs à produire des bandes sonores spécifiques. En définitive, seul le phénomène des démos - des animations plus ou moins oniriques calculées en temps réel par l'ordinateur et qui résultent d'un effort de programmation en assembleur pour tirer le meilleur parti des capacités de la machine - mettra l'Amiga 500 sur un piédestal. Au point que la rivalité entre l'Amiga 500 et l'Atari 520 ST sera en fait entretenue par la compétition que se livreront durant des années les groupes de démomakers constitués de part et d'autre.

Propreté contre bidouille

Ce que les défenseurs de chaque plateforme se jettent à la figure, ce sont les détails des caractéristiques techniques. Celles de l'Atari 520ST sentent le savon propre et celles de l'Amiga 500 la bidouille poussée à bloc. En France, l'Atari 520 ST c'est simple : c'est comme un Amstrad CPC 6128 qui fait tout en mieux. Il a la même puce audio nasillarde, la même grosse bande de pixels inutiles tout autour de sa zone d'affichage et le même système de fichiers CP/M (avec ses noms en 8+3 caractères, comme sur DOS). Mais la vitesse (8 MHz contre 4 MHz), l'affichage (320x200 contre 160x200 en 16 couleurs), le stockage (360 Ko contre 180 Ko sur chaque face de disquette) sont deux fois meilleurs. A noter que son interface graphique à la Macintosh, GEM, est aussi fournie sur le

L'Amiga, l'Atari des riches

À ce stade, en 1987, la vraie différence entre les deux appareils est le prix : l'Amiga 500 coûte 4700 Frs, alors que l'Atari 520 ST s'arrache à seulement 3000 Frs (il faut débourser 2500 Frs de plus pour un moniteur couleur dans les deux cas). L'écart de prix se justifie par des capacités graphiques et sonores théoriquement supérieures sur l'Amiga. Théoriquement, car c'est ce que disent les documentations techniques. En pratique, et par soucis d'économie, les éditeurs développeront de nombreux jeux avec le même code assembleur 68000 générique et les

compatible PC 1512 d'Amstrad, puisque Windows n'est pas encore disponible. Enfin, chose inédite sur une machine personnelle, l'Atari ST dispose d'un contrôleur de disque dur. Hélas, il ne servira pas à grand-chose sur le 520 ST, du fait du prix prohibitif des disques durs à la fin des années 80, et aussi parce qu'il faudra attendre la sortie d'adaptateurs pour utiliser des modèles SCSI standards ; le contrôleur du ST était propriétaire. Du côté de l'Amiga 500, en revanche, il faut avoir fait polytechnique pour comprendre comment il marche. Son processeur est cadencé à la fréquence bizarre de 7,14 MHz quand il est branché sur un écran PAL européen et à 7,09 MHz avec un moniteur NTSC. La puce audio a tellement de ressources qu'elle pilote aussi le lecteur de disquettes et lui offre des fonctions incongrues, comme le fait de faire vrombir la tête de lecture, en particulier pendant le jeu de course Out Run. Pour l'anecdote, l'Amiga 500 était à ce point capable d'écrire des données de n'importe quelle manière sur ses disquettes qu'il était le seul appareil disponible dans le commerce à pouvoir copier les jeux protégés de l'Atari 520 ST. Ce qui donnera lieu à des trêves régulières entre les deux clans. La puce graphique affiche de base 320x256 pixels en 32 couleurs, mais grimpe à 704x576 pixels en 16 couleurs pour couvrir toute la surface de l'écran, même celle qui n'est pas affichable, et offrir un mode haute résolution qui clignote tellement qu'il écorche les yeux. On peut aussi monter à 64 couleurs, dont la moitié est forcément sombre, et même à 4096 en codant chaque pixel d'après celui qui le précède. Ce qui limite l'usage de ce mode à l'affichage de photos fixes qui bavent. Au centre de l'Amiga 500, un chipset offre de l'accélération 2D, pour bouger des blocs de pixels



En haut de gamme, l'Amiga 2000 était censé séduire les professionnels de la vidéo et l'Atari MegaST (ici avec disque dur) ceux de la PAO.

sans que cela coûte du temps machine au processeur. Enfin, l'Amiga est le premier micro-ordinateur à fonctionner avec un système d'exploitation multitâche. Mais, avec seulement 512 Ko de mémoire, cela permet juste de jouer avec la calculatrice pendant que l'on copie des fichiers entre deux disquettes.

En quête désespérée de marchés pros

L'idée géniale de l'Atari 520 ST est d'être le seul micro-ordinateur à intégrer un connecteur MIDI. Avec celui-ci, il devient possible de faire jouer sur des synthétiseurs professionnels la musique que l'on a composée et stockée sur l'ordinateur (à l'aide du logiciel Cubase, notamment). Jean-Michel Jarre, Madonna ou encore Eurythmics utiliseront ainsi l'Atari ST en concert. Le 520 ST sera décliné en version 1040 ST avec 1 Mo de mémoire (au même prix que l'Amiga 500 !), plus adaptée pour les musiciens professionnels. Une version Mega ST desktop (clavier à part), avec 2 à 4 Mo de mémoire et une imprimante laser, existera aussi pour concurrencer le Mac sur le marché de la PAO (avec le logiciel Calamus), mais avec un succès mitigé. L'Amiga 500 s'accompagnera d'un Amiga 2000 au format desktop avec, comme sur PC, des connecteurs d'extension.



Cela ouvrira à la machine un vrai créneau professionnel : l'incrustation d'effets vidéo sur les fonds verts pour la télé et le cinéma. Grâce au chipset qui manipule la mémoire et une carte dite Genlock, on mixe des cartes de France ou des vaisseaux spatiaux dessinés sur Amiga avec le flux vidéo d'un présentateur météo ou d'une série de science-fiction. Du coup, des logiciels de création graphique s'imposent sur la machine : Deluxe Paint pour dessiner en 2D et Lightwave pour produire des scènes en 3D.

Des constructeurs qui n'ont pas su évoluer

En 1989, Atari intègre dans ses machines devenues « STe » les mêmes capacités graphiques et sonores que celles de l'Amiga. Mais les éditeurs ne s'en serviront pas. En 90, Commodore et Atari lancent des versions 32 bits destinées aux marchés professionnels : l'Amiga 3000 pour la 3D et l'Atari TT030 pour la PAO. Il s'agissait de proposer des alternatives bon marché aux stations de travail de Silicon Graphics et aux Mac d'Apple. Les professionnels n'ont pas été séduits. Il faudra attendre 1992 pour que les constructeurs lancent enfin des versions 32 bits pour le grand public : l'Amiga 1200 et l'Atari Falcon. Mais il est trop tard : les jeux Wolfenstein 3D puis Doom sortent sur PC et prouvent que cette austère machine est désormais capable de proposer des univers 3D calculés en temps réel, un concept jamais vu jusqu'ici. Et dont sont totalement incapables les Atari-Amiga optimisés pour la 2D. Ringardisés, ruinés, Atari et Commodore ne cesseront depuis lors de disparaître à petit feu en se faisant racheter régulièrement par des personnes plus ou moins fantasques.

● Yann Serra



En 1992, les Amiga 500 et Ataris 520 ST sont enfin améliorés en version 32 bits : ce sont respectivement les Amiga 1200 et Atari Falcon. Mais le marché n'en veut plus.

Abonnement : Programmez, 17, Route des Boulangers, 78926 Yvelines Cedex 9 - Tél. : 01 55 56 70 55 - abonnements.programmez@groupe-gli.com - Fax : 01 55 56 70 91 - du lundi au jeudi de 9h30 à 12h30 et de 13h30 à 17h00, le vendredi de 9h00 à 12h00 et de 14h00 à 16h30.
Tarifs abonnement (magazine seul) : 1 an - 11 numéros France métropolitaine : 49 € - Etudiant : 39 € - CEE et Suisse : 55,82 € - Algérie, Maroc, Tunisie : 59,89 € Canada : 68,36 € - Tom : 83,65 € - Dom : 66,82 € - Autres pays : nous consulter.
PDF : 30 € (Monde Entier) souscription exclusivement sur www.programmez.com



Directeur de la publication

& **rédacteur en chef** : François Tonic

Ont collaboré à ce numéro : Yves Grandmontagne, Sylvain Saurel, Yann Serra

Secrétaire de rédaction : Olivier Pavie

Experts : G. Renard, J.D., J. Paumard, Nicolas Maillard, B. Dechoux, B. Guinebertière, J-P Sanchez, S. Challa, T. Joubert, V. Leduc, V. Néel, D. Mondon, M. Ohayon, J. Picques, V. Munoz, S. Sibué, F. Szajnberman, B. Boucard, D. Tran, E. Galiano, R. Tomczak, F. Ravetier

Une publication Nefer-IT
7 avenue Roger Chambonnet
91220 Brétigny sur Orge
redaction@programmez.com
Tél. : 01 60 85 39 96

Crédits couverture : © iStock - Dolgachov

Maquette : Pierre Sandré

Publicité : Régie publicitaire, K-Now sarl. Pour la publicité :
Tél. : 01 41 77 16 03 - diff@programmez.com.

Imprimeur : S.A. Corelio Nevada Printing, 30 allée de la recherche, 1070 Bruxelles, Belgique.

Marketing et promotion des ventes :
Agence BOCONSEIL - Analyse Media Etude

Directeur : Otto BORSCHA oborscha@boconseilame.fr

Responsable titre : Terry MATTARD
Téléphone : 0967320934

Contacts

Rédacteur en chef : ftonic@programmez.com

Rédaction : redaction@programmez.com

Webmaster : webmaster@programmez.com

Publicité : diff@programmez.com

Événements / agenda :

redaction@programmez.com

Dépôt légal : à parution - Commission paritaire : 1215 K 78366 - ISSN : 1627-0908

© NEFER-IT / Programmez, février 2014

Toute reproduction intégrale ou partielle est interdite sans accord des auteurs et du directeur de la publication.

DÈS MAINTENANT

Le nouveau site web

www.programmez.com

**Nouvelle présentation
Nouveaux contenus
Nouvelle boutique
Nouvelle newsletter**

Quelques chiffres :
+ 16 semaines de développement
100 % Drupal
3 développeurs
1 designer

PROGRAMMEZ!
le magazine du développeur

Newsletter Hebdomadaire
Mercredi 29 Janvier 2014

Microsoft
tech-days
2014

11,12,13 février
au Palais des Congrès de Paris

A LA UNE CETTE SEMAINE

Java 8 arrivera le 18 mars, même bogué !
Oracle semble en avoir fini avec les retards pour la sortie de Java 8 en raison de problèmes de sécurité des versions précédentes.

Sécurité
Pirater un site web devient légal
Depuis le 18 décembre 2013 (article de [l'ONPC](#) du 2013-11-6 : art. 25) "fouiller" dans un site ou une application web à la recherche de failles de sécurité devient...

iOS
Windows 9 dès 2015
D'après Paul Thurrott, journaliste extrêmement bien informé de ce qui se passe chez Microsoft, Redmond travaille déjà à...

IDE
Plutôt monter chez Target... et alliées
A la fin de l'année 2013, le groupe de distribution américain Target a été victime d'un piratage monstrueux, sans doute le plus gros piratage...

AVIS D'EXPERTS

Les botnets, ces acteurs majeurs de l'ombre
Internet est d'une complexité insoupçonnée notamment pour sa part obscure. Au cœur de cet écosystème, les botnets, l'ennemi de tous les utilisateurs, sont l'origine d'un très grand nombre

Comment conserver le contrôle de la sécurité mobile aujourd'hui

Comment piéger les logiciels malveillants avec un bac à sable

Sécurité des applications : les logs sont une mine d'informations pour les pirates informatiques

Lire la suite des Avis d'Experts

LA SUITE DE L'ACTUALITÉ SUR PROGRAMMEZ!

Ne tardez plus pour votre migration depuis Windows XP

Accenture au secours du site de l'Obamacare

Android 4.4 KitKat disponible pour Samsung Galaxy S4 !

Windows 8.1 : une mise à jour au printemps ?

Gfi Informatique recrute une trentaine d'experts PLM

Les données sont plus précieuses que les machines

Des sites du Ministère de la Santé massivement hackés !

Comment conserver le contrôle de la sécurité mobile aujourd'hui

Lire le reste de l'actualité sur [Programmez.com](#)

11,12,13 février
au Palais des Congrès de Paris

Je m'inscris

PROGRAMMEZ!
Le magazine du développeur

CLOUD COMPUTING
La révolution pour les développeurs

DIPLOME
TIZEN
CODEUR
GAMMA TEST
HADOOP
IDE
JAVA
NETBEANS
NODEJS
PHP
SQL
TEST
UI/UX
VB.NET

Programmez! N°170
Janvier 2014

Cloud Computing : La révolution pour les développeurs

Tizen : La plateforme mobile Linux

Consulter le sommaire complet

SABONNER

ACHETER LE PDF

LIVRES BLANCS

Le groupe Volkswagen teste son système d'infotainment avec ABYSS FineReader Engine
Paru le 02-12-2012

Consulter les Livres Blancs

NIDays
GRAPHICAL SYSTEM DESIGN CONFERENCES
11 février 2014 | Paris
1200 VISITEURS
CONCEPTION - TEST - MESURE - CONTRÔLE
Inscription gratuite

Suivez nous [Facebook](#) [Twitter](#) [Google+](#) [RSS](#)

AGENDA

Evenements à venir

TechDays 2014
11/02

ITIX : Google Glass et Réalité Augmentée
20/02

Consulter l'agenda complet

Solutions Logiciels
N°43 - Novembre

Découvrez le sommaire

Microsoft
tech-days
2014

11,12,13 février
au Palais des Congrès de Paris

Je m'inscris

PROGRAMMEZ!
le magazine du développeur

Actualités Magazine Livres Livres Blancs Tutorials Téléchargement Forum Agenda Avis d'experts Emploi

Microsoft
tech-days
2014

11.12.13 février
au Palais des Congrès de Paris

Je m'inscris

Dans Programmez N° 170

Cloud Computing : La révolution pour les développeurs

- Diplôme : Comment choisir son école d'informatique
- CODING4FUN : Construire son robot Curiosity
- Tizen : La plateforme mobile Linux

S'ABONNER ACHETER LE PDF

PROGRAMMEZ
CLOUD COMPUTING
L'volution pour les dveloppeurs

DATA SCIENCE
DATA SCIENCE
DATA SCIENCE
DATA SCIENCE

APPLICATIONS MOBILES
APPLICATIONS MOBILES
APPLICATIONS MOBILES
APPLICATIONS MOBILES

CODES SOURCE
CODES SOURCE
CODES SOURCE
CODES SOURCE

INFORMATIQUE
INFORMATIQUE
INFORMATIQUE
INFORMATIQUE

INTERNET DES objETS
INTERNET DES objETS
INTERNET DES objETS
INTERNET DES objETS

NETWORKING
NETWORKING
NETWORKING
NETWORKING

SECURITÉ
SECURITÉ
SECURITÉ
SECURITÉ

SOFTWARE ENGINEERING
SOFTWARE ENGINEERING
SOFTWARE ENGINEERING
SOFTWARE ENGINEERING

TECHNOLOGIES
TECHNOLOGIES
TECHNOLOGIES
TECHNOLOGIES

TOOLBOX
TOOLBOX
TOOLBOX
TOOLBOX

UNIVERSITÉS
UNIVERSITÉS
UNIVERSITÉS
UNIVERSITÉS

VERSIONS
VERSIONS
VERSIONS
VERSIONS

WIKIS
WIKIS
WIKIS
WIKIS

Abonnez vous à la newsletter

NIDays
GRAPHICAL SYSTEM DESIGN CONFERENCES
11 février 2014 | Paris
1200 VISITEURS
CONCEPTION - TEST - MESURE - CONTRÔLE
Inscription gratuite

Suivez nous

ACTUALITÉS

Java 8 arrivera le 18 mars, même bogué !

Oracle semble en avoir fini avec les retards pour la sortie de Java 8 en raison de problèmes de sécurité des versions précédentes.

Pirater un site web devient légal

Alphamen : un OS open source pour robots

Windows 9 dès 2015 ?

Piratage monstre chez Target... et ailleurs

Facebook : des fraudes au bouton 'J'aime' ?

Lire toute l'actualité >

AVIS D'EXPERTS

Les bonnets, ces acteurs majeurs de l'ombre

Comment conserver le contrôle de la sécurité mobile aujourd'hui

Comment piéger les logiciels malveillants avec un bac à salade

Sécurité des applications : les logs sont une mine d'informations pour les pirates informatiques

Hisser les voiles vers les API : La Coupe de l'América navigue dans un océan

Les cyber-attaques avancées en 2014

EN CE MOMENT SUR LE FORUM

Quand est ce que ma session a-t-elle expiré ?
Avec Luc... PHP

33

Mon compte

Retrouvez-nous sur Facebook

PRC Programmez.com

490 651 personnes aiment Programmez.com

Facebook social Facebook

Livres Programming with MS Visual C++ Web Analytics Drupal 7 CMF 1.3 Web Analytics Ajax, jQuery et PHP Cuda par l'exemple

tech-days
2014

11.12.13 février
au Palais des Congrès de Paris

Je m'inscris

NIDays
GRAPHICAL SYSTEM DESIGN CONFERENCES
11 février 2014 | Paris
1200 VISITEURS
CONCEPTION - TEST - MESURE - CONTRÔLE
Inscription gratuite

Suivez nous

Editeurs, Chefs de projets logiciels donnez votre avis sur le Cloud

Participez à l'enquête. Vous pourrez gagner une soirée de rêve : tour en ballon, dîner et nuit au château pour deux personnes

Participer à l'enquête

Mon compte

Retrouvez-nous sur Facebook

PRC Programmez.com

490 651 personnes aiment Programmez.com

Facebook social Facebook

Livres Programming with MS Visual C++ Web Analytics Drupal 7 CMF 1.3 Web Analytics Ajax, jQuery et PHP Cuda par l'exemple

ir le logiciel libre à l'arrivée du printemps

ts

lative, l'initiative Libre en Fête est relancée par l'April, en partenariat avec la

la réseau Cyber-base de la Caisse des dépôts et l'Agenda du Libre.

La NSA vous surveille quand même !

raut-elles une fin ? Nous savons déjà que la NSA, parmi autres facéties,

tre Borland sur le développement logiciel

ultats de son second Baromètre du Développement Logiciel.

ux États-Unis pour 2013 Big Blue

age

mis à jour Windows RT 8.1, et s'est vue contrainte de...

même bogué !

retardera pour la sorte de Java 8 en raison de problèmes de sécurité des

appel à communications

Linux, Libres et Open Source se déroulera les 20 et 21 mai 2014...

technologies : Net, Java/JEE et méthodes Agile, ouvre 80 nouveaux postes

légal

de loi n°2013-1168 - art. 25) "fouiller" dans un site ou une application web

gration depuis Windows XP

osoft prendra fin le 8 avril prochain. Sans mises à jour techniques...

e de l'Obamacare

Etats-Unis, ils ont le site de l'Obamacare. Ce site, healthcare.gov, portait

1 2 3 4 5 ... Suivant

Tout change sauf le code

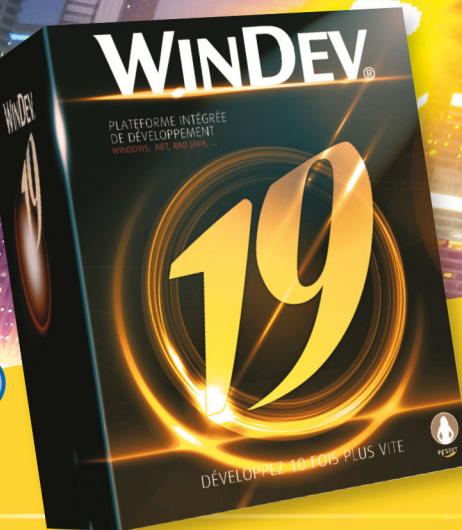
Programmez ! expert du code depuis 1998

Le saviez-vous ?



De Janvier à Avril 2014,

WINDEV est parrain de l'émission CAPITAL sur M6



Fournisseur Officiel de la
Préparation Olympique

Siège 3 rue de Puech Villa - 34197 MONTPELLIER Cedex 05 Tel. 04 67 032 032 Fax 04 67 03 07 87 Tél. Support Technique: 04 67 03 17 17

Agence 142 Avenue des Champs Elysées 75008 PARIS Tél. 01 48 01 48 88

PC SOFT Informatique - SAS au capital de 2 297 548 Euros - RCS 330 318 270 - Code APE 5829B

www.pcsoft.fr