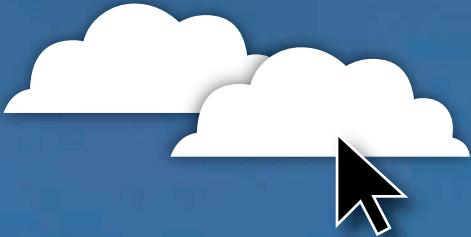


Adapter, optimiser votre site web sur smartphone



Cloud Computing



Maintenant ou jamais ?

Comment migrer une application ?

Quelle architecture adopter ?

Déployer facilement PHP et MySQL sur Amazon EC2 et Windows Azure

Refactoring

Simplifiez-vous le code

Réseau

Comprendre et utiliser IPv6



FreeBSD

Système

Osez BSD !

Emploi

+ 70% en janvier !

Langage

Falcon : une nouvelle approche de l'objet

Outils

Maven 3 : tout ce qu'il faut savoir

Tactile

Gérer les gestes Surface

NuGet

Les dépendances binaires en .Net

M 04319 - 139 - F: 5,95 €



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

N°1 EN FRANCE

- Crée des applications Windows, Linux, Mac, ... compatibles Internet, Intranet, Android, Windows Phone 7
- Gestion complète du cycle de vie
- Éditeur d'analyses (UML, Merise...)
- RAD (avec patterns)
- Lien avec toutes les bases de données (ODBC, OLE DB, Lien natif), Oracle, SQL Server, AS/400, Informix, DB2, MySQL, PostreSQL, SQLite, ...
- HyperFileSQL: puissante base de données Client/Serveur gratuite incluse
- Génération d'états PDF, Codesbarres
- Accès natif SAP R/3, Lotus Notes, ...
- Gestion de planning, des Exigences
- Audit automatique de vos applications
- Langage de 5^e génération (L5G)
- SNMP, Bluetooth, TAPI, OPC, FTP, HTTP, Socket, Twain, API, DLL, XML...
- Domotique • Interfaces série et USB
- Débogage à distance, Profiler
- Multilingue (64 langues)
- Installateur 1-clic & Push
- Gestionnaire de versions
- Gestion de l'infrastructure
- Tout en français
- etc...

support technique gratuit

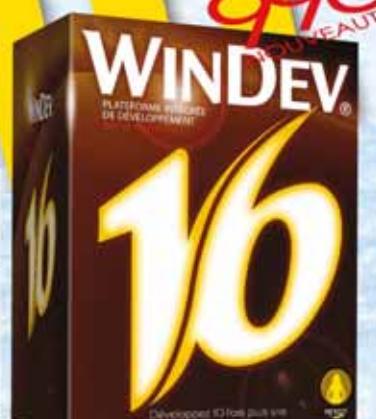
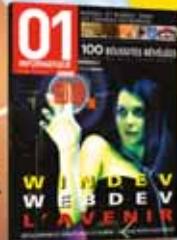
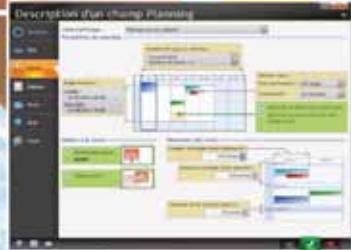
DÉVELOPPEZ
CONCENTREZ-VOUS SUR LES BESOINS MÉTIERS

Élu «Langage le plus productif du marché»

VERSION EXPRESS GRATUITE

Téléchargez-la !

100 TÉMOIGNAGES SUR SIMPLE DEMANDE



16 CRÉEZ VOS APPLICATIONS POUR PC, MAC, LINUX, INTERNET, SMARTPHONES, TABLETTES, ...

VOTRE CODE EST MULTI-PLATEFORMES

Windows, .Net, Java, PHP, Linux, Mac, J2EE, XML, Internet, SaaS, Pocket PC, Windows Phone 7, Android, ...

WINDEV GÈRE LE CYCLE DE VIE COMPLET DE VOS APPLICATIONS.

► **DEMANDEZ LE DOSSIER GRATUIT**

Dossier gratuit 260 pages sur simple demande. Tél: 04.67.032.032 info@pcsoft.fr

Fournisseur Officiel de la Préparation Olympique



www.pcsoft.fr

॥ actus

En bref	6
Agenda	8
Hardware	10

**॥ webmaster**

Votre site web sur smartphone de A à Z

Le navigateur mobile : le nerf de la guerre	15
Comment adapter son site web aux smartphones	16
Web, mensonges et vidéos	18
S'adapter aux interactions utilisateurs	20
Adapter, développer, optimiser : les modèles de développement	22

**॥ carrière**

Les offres d'emploi informatique décollent en janvier	25
---	----

**॥ dossier**

Cloud Computing : le guide de survie du développeur

Qu'est ce qu'une application cloudisée, quel cloud choisir ?	27
Windows Azure : les nouveautés 2011	30
PHP dans le cloud : les offres	32
Windows Azure VM Role : mise en œuvre	34
Adapter une application PHP pour Azure	38
Déployer une application PHP/MySQL dans le cloud avec Amazon EC2	40

**॥ technique**

Refactoring : les meilleures pratiques	44
Les 10 commandements du refactoring	46

॥ recherche

Swelen : le défi de l'information	50
---	----

**॥ code**

Falcon, un langage de programmation procédural, OO, fonctionnel et OM	53
La gestuelle avec MVVM sur Microsoft Surface	57
Maven-épisode III, la revanche du (plugin) site	59
Concevoir une application PHP avec ZendFramework & Doctrine 2	62
La dépendance binaire selon Nuget	65
Windows Identity Foundation	69
A la découverte de FreeBSD	73
Petite prise de contact avec IPv6	78

॥ temps libre

Les livres du mois	82
--------------------------	----

L'info continue sur www.programmez.com

CODE

Les sources des articles

NOUVEAU

Livres blancs : langages, outils...

TELECHARGEMENT

Les dernières versions de vos outils préférés + les mises à jour

QUOTIDIEN

Actualité, Forum Tutoriels, etc.



Spread for Windows Forms

à partir de € 739

GrapeCity PowerTools

Feuille de calcul pour les applications Windows Forms, compatible avec Microsoft Excel.

- Accélérez le développement avec les concepteurs de feuilles de calcul, l'Assistant de prise en main et les concepteurs de graphiques
- Renseignement automatique : anticipation de la frappe dans la cellule
- Nouveau - outil intégré de création de diagrammes avec 85 styles
- Nouveau - préserve les .XLS et restaure les fonctions non supportées
- Inclut des apparences prédéfinies ainsi que la possibilité de créer des apparences personnalisées



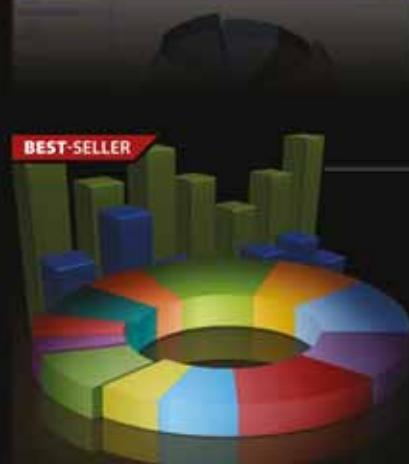
Codejock Xtreme Chart Pro ActiveX

à partir de € 175

CODEJOCK

Incluez des diagrammes dans vos applications ActiveX en quelques lignes de code.

- Affiche un ensemble riche de classes de personnalisation et d'amélioration
- Histogrammes, barres de dispersion, barres empilées, 100% barres empilées et barres horizontales
- Graphiques en secteurs 2D/3D, anneaux 2D/3D et tores 3D, ainsi que les secteurs éclatés
- Inclut les lignes standard, à dispersion, à traçage rapide, en escalier et spline
- Inclut aussi les diagrammes de points, de zones, entonnoir, financiers et de Gantt



FusionCharts

à partir de € 147

InfoSoft Global

Graphiques Flash & JavaScript (HTML5) interactifs pour les applications Web.

- Animez vos applications Web avec des graphiques interactifs et pilotés par les données
- Créez des graphiques AJAX avec des possibilités d'exploration en quelques minutes
- Exportez les graphiques au PDF et les données en CSV directement depuis les graphiques
- Créez des jauge, des tableaux de bord, des graphique financiers et plus de 550 types de carte
- Adopté par plus de 17 000 clients et 330 000 utilisateurs dans 110 pays



Google Data Provider for ADO.NET from RSSBus

à partir de € 221

Créez des applications liées à Google avec les Assistants Visual Studio.

- Connectez facilement les applications .NET à Google Services en incluant GMail, Google Docs, Google Calendar, Google Adwords, Google Talk, Recherche...
- Offrez une prise en charge CRUD (créer, lire, mettre à jour et supprimer)
- En ligne et hors ligne d'accès aux données grâce à la mise en cache configurable
- Caractéristiques de sécurité de classe entreprise telles que les données cryptage SSL/TLS



Elle arrive quand, l'interface naturelle ?

Mine de rien, depuis 27 ans (lancement du Macintosh), l'interface utilisateur de nos ordinateurs n'a pas fondamentalement changé : fenêtres, boutons, cadres, métaphore du bureau, icônes, menus déroulants. Les innovations se limitent souvent à des aspects 'cosmétiques' : effets « 3D », variations des thèmes, nouvelles fonctions. Les technologies multitouch /tactile permettent aujourd'hui de modifier notre approche de l'interface, et plus précisément, de la manière dont on interagit avec elle. Cependant, des technologies comme Surface modifient des notions ergonomiques et fonctionnelles qui sont là depuis l'apparition de l'Apple Lisa ! C'est dire. Les plus jeunes lecteurs n'ont sans doute pas connu ce fantastique ordinateur, ni les idées du PARC.

Ce que l'on voit changer depuis des années, dans les centres de recherche ou les démonstrations (comme durant le 3e jour des Microsoft TechDays), ce n'est pas tant l'interface de l'ordinateur, que l'interaction entre les objets du quotidien et l'humain, (nous). Et là, la liberté paraît immense. Les usages à la « Minority Report », l'interface virtuelle, intégrée directement à son miroir, sa table, son frigo, ne sont plus une absurdité. Et cet avenir n'est pas aussi lointain qu'il y paraît. Même le corps humain pourra devenir une interface: par exemple, le bras servant de clavier au téléphone et la paume de la main d'écran. [*]

Déjà, des lunettes 3D, totalement intégrées permettent une interface immersive, ou plus précisément la réalité augmentée. C'est-à-dire apporter des informations, une interaction virtuelle, complémentaire à la réalité.

La console Kinect est une nouvelle approche de l'interface ou plus exactement de l'interaction, de l'usage d'objet, d'élément informatique. Nous ne pouvons plus appeler cela une interface même si le système Kinect conserve des composants d'interface.

Sur l'ordinateur, il paraît difficile à court terme d'avoir des « interfaces » de ce genre. Au quotidien, est-ce productif ? Pas certain. Même le tactile, tel qu'il existe aujourd'hui, a un intérêt limité pour travailler intensivement tous les jours. Les nouveaux usages restent encore à découvrir au-delà de la nouveauté, du côté fun. Toutes ces questions, il faut les poser, comprendre les enjeux.

Steve Jobs n'a jamais été partisan d'offrir du multitouch sur les Mac. Cela changera s'il y voit un intérêt. Cette attitude se comprend si on considère que pour des terminaux mobiles, embarqués, le tactile remplace des périphériques de saisie (souris, clavier). Dans les années 1990, de nombreux éditeurs parlaient d'interface 3D sur nos PC, de nouvelles manières de visualiser le web ou encore d'une informatique entièrement vocale. 15-20 ans plus tard, constatons que nous en sommes encore loin. Plusieurs raisons : des technologies limitées et perfectibles à l'équipe, une puissance machine incapable de fournir les ressources nécessaires et une utilisabilité discutable. La prospective, notamment dans l'interface utilisateur, est cruciale. Les démonstrations font rêver. Mais entre le laboratoire et l'utilisateur, le fossé est parfois un abysse.

■ François Tonic

[*] Nous vous conseillons vivement de regarder, pour l'exemple :

- le projet Skinput :

<http://research.microsoft.com/en-us/um/redmond/groups/cue/skinput/>

- Lightspace Project : <http://research.microsoft.com/en-us/projects/lightspace/>

Editeur : Go-02 sarl, 21 rue de Fécamp 75012 Paris - diff@programmez.com.

Rédaction : redaction@programmez.com

Directeur de la Rédaction : Jean Kaminsky.

Rédacteur en Chef : François Tonic - ftonic@programmez.com. **Ont collaboré à ce numéro :** F. Mazué, S. Saurel, G. Delamarre, A. Bariault **Experts :** Q. Sun, J.L. Boucho, A. Dauba, C. Villeneuve, G. Seguy, G. Daniel, P. Couzy, D. Negrier, V. Fontaine, V. Borrel, R. Sabin Mompelat, N. Calvi, N. De Loof, D. Camus, V. Labatut, A. Zanchetta

Illustrations couverture : ©istockphoto/Cinek20

Publicité : Régie publicitaire, K-Now sarl. Pour la publicité uniquement : Tél. : 01 41 77 16 03 - diff@programmez.com.

Dépôt légal : à parution - Commission paritaire : 0712K78366 ISSN : 1627-0908. Imprimeur : S.A. Corelio Nevada Printing, 30 allée de la recherche, 1070 Bruxelles Belgique. **Directeur de la publication :** J-C Vaudecrane

Abonnement : Programmez 22, rue René Boulanger, 75472 Paris Cedex 10

Tél. : 01 55 56 70 55

abonnements.programmez@groupe-gli.com

Fax : 01 40 03 97 79 - 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

L'INFO PERMANENTE WWW.PROGRAMMEZ.COM



PROCHAIN NUMÉRO

N°140 avril 2011
parution **31 mars**

✓ **Créer un code universel :**
le multiplateforme a-t-il
encore un sens ?

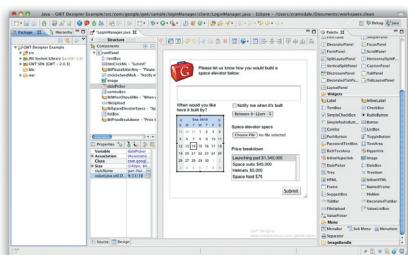
✓ **Maître en C++**
Pourquoi le C++ est redevenu
LE langage incontournable ?

✓ **Webmaster**
Comment créer un site web sécurisé ?

■ **GTK** arrive en version 3. Très attendue, cette v3 apporte de nombreuses nouveautés : partie graphique refaite et harmonisée (centrée sur Cairo), nouveaux supports d'interfaces d'entrée, nouvelles API pour créer plus rapidement des applications, apparition d'une API proche de la syntaxe CSS pour la configuration, les transitions. Site : <http://www.gtk.org/>

■ **WebODF** est une nouvelle bibliothèque javascript pour supporter rapidement le format ODF sur un site web, une application mobile ou desktop. Elle utilise HTML et CSS pour l'affichage. Une excellente idée pour diffuser l'utilisation de ce format plus largement. Site : <http://webodf.org/>

■ **GNU Hurd** existe toujours. On en parle peu et pour cause, le projet avance lentement. Hurd vise à remplacer le noyau Unix en utilisant une approche multiserveur fonctionnant sur un micronoyau de type Mach et implémentant le système de fichier, réseau... 2010 fut une année active avec le support de Xen domU, le travail autour d'un nouveau framework de pilotes, l'utilisation d'un nouvel installateur.



■ **Google** propose GWT 2.2, plus exactement le plug-in pour Eclipse GPE. Cette version permet de supporter dans l'IDE GWT 2.2, avec notamment le GWT Design inclus directement dans le plug-in. Le SDK supporte aussi HTML (canvas, balise vidéo).

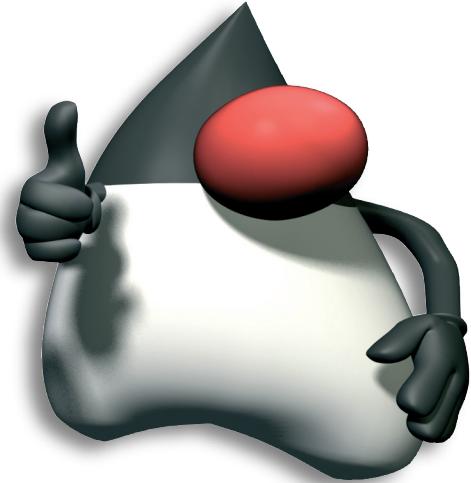
Gouvernance

Oracle redessine la gouvernance du monde Java

Bonne ou mauvaise nouvelle ? Difficile de le dire, mais après les errements autour de Java 7 et 8, des critiques de la communauté, du retrait de la fondation Apache ou encore des plaintes contre Android, la communauté Java regarde cette nouvelle initiative avec prudence. Il faut dire que le document dévoilé début février dernier concerne la manière dont Java est géré et le processus de prise de décision. Deux éditeurs sont clairement mis en avant (les auteurs) : Oracle et IBM, avec un soupçon d'Eclipse. Il s'agit d'améliorer le processus de gouvernance et de décision du JCP : Java Community Process. Le JCP se compose de différents groupes, de projets, d'un « chef », le OpenJDK Lead et d'un comité de gouvernance. Le Lead a la charge des questions techniques et de l'activité des groupes.

Le document définit outre ce qui a été dit plus haut, différents niveaux de participations :

- un participant : personne individuelle pouvant être inscrite à une ou plusieurs listes d'OpenJDK, il peut aussi proposer des (simples) patchs.
- Un contributeur : c'est un participant ayant signé un document spécifique (Oracle Contributor Agreement) ou qui travaille dans une



organisation ayant signé un tel document. Seuls les contributeurs peuvent soumettre des éléments autres que des patchs simples.

- OpenJDK Member : contributeur qui a montré la valeur de ses contributions, de son travail. A cela s'ajoute une définition des processus de vote. Ainsi, le OpenJDK Lead peut, dans certains cas, revenir sur un vote. Pour raccourcir les délais, si dans le cadre d'une décision, le vote est unanime chez les participants d'un groupe, le vote peut être clos avant la date imposée. Et il est question d'avoir des votes à simple majorité, au 2/3 et consensus unanime. L'un des points concerne le comité (governing board) qui sera composé de 5 membres, dans lequel, Oracle et IBM nommeront trois sièges... Les deux autres se feront par vote des membres OpenJDK. Tous les points soulevés par le document nécessitent encore des commentaires et ajustements.

Site : <http://cr.openjdk.java.net/~mr/bylaws/draft-openjdk-bylaws-07.html>

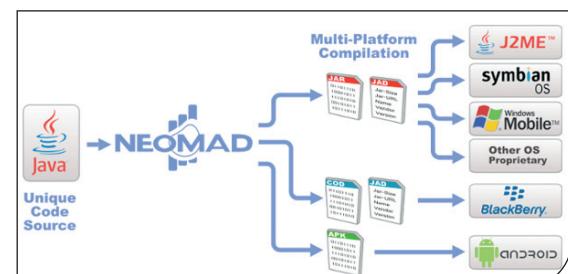
Mobile

Un code pour tous les mobiles ? Oui selon Neomad

Neomad, petite société française (!), propose une approche unique et globale pour développer une fois et déployer partout, sur plus de 1000 modèles de téléphones. L'offre NeoMAD 2.0 repose sur le langage et un environnement 100 % Java (Java ME). La structure du projet s'appuie sur une description XML.

La v2 supporte pleinement désormais Android et s'intègre à Eclipse. Cette version propose une nouvelle API Bluetooth, la gestion des contacts ou encore une meilleure gestion du multimédia.

Les cibles annoncées par l'éditeur sont Java ME, Symbian, Windows Mobile, Blackberry, Android, étonnamment iOS



n'est pas explicitement indiqué. Le développement se fait uniquement sous Windows et un JDK Java 1.5 ou supérieur. La licence dépend du nombre d'applications et de modèles supportés.

Pour les développements de tests, la licence est gratuite. Site : <http://www.neomades.com>

DESIGN DEVELOPPEMENT EXPERIENCE



NetAdvantage® **ULTIMATE**

for ASP.NET, Windows Forms, WPF, Silverlight,
WPF Data Visualization, Silverlight Data Visualization

De la conception au développement , nous avons les outils pour créer une excellente expérience utilisateur.



DECOUVREZ COMMENT UTILISER NOS CONTROLES
POUR CREER CETTE "KILLER APP" SUR
INFRAGISTICS.COM/IMPRESS

Infragistics

Infragistics Ventes France 0800 667 307 • Infragistics Europe Ventes +44 (0) 800 298 9055
Infragistics India +91 80 41518042 • @infragistics

■ Le directeur du développement de Firefox (Mozilla), Mike Beltzner a annoncé son départ du projet mi-février. Cette annonce est une surprise au moment où Firefox 4 peine à se terminer suite aux multiples retards du projet. Mike dit juste qu'il souhaite poursuivre sa carrière ailleurs mais sans préciser les motivations et les raisons de son départ...

Site : <http://beltzner.ca/mike/2011/02/14/as-the-french-say-until-we-meet-again/>

■ Sur le salon Mobile World Congress, **Adobe** a annoncé que plus de 20 millions de smartphones ont été équipés (directement ou après mise à jour) de Adobe Flash Player 10.1 dans les six mois qui ont suivi son lancement et que le logiciel est à présent installé sur plus de 35 équipements certifiés. D'après les prévisions d'Adobe, en 2011, Flash Player devrait conquérir plus de 132 millions d'unités à travers le monde.

■ **Jetbrains**, l'éditeur d'IDE très connu pour IntelliJ IDEA, son outil de développement pour Java, a annoncé la sortie de PHPStorm 2.0. Dans l'esprit d'IntelliJ IDEA, cet IDE se veut un outil intelligent pour le développement PHP, en fournissant des outils essentiels tels que la refactorisation (refactoring) automatisée, l'analyse de code en profondeur, le contrôle d'erreur à la volée, et la résolution rapide des bugs.

■ **Parasoft** dévoile un nouvel outil : Parasoft Virtualize. Il fait partie de l'approche ALM de l'éditeur. Cet outil doit permettre une flexibilité dans l'environnement de tests et de qualité des équipes de développement, en utilisant la virtualisation. Il doit permettre de déployer plus rapidement des cas de tests, des batteries de tests. Site : www.parasoft.com/virtualize

TechDays 2011

cloud attack

L'édition 2011 des TechDays de Microsoft, grand'messe de 3 jours à Paris, a un étrange nuageux. Eh oui, le cloud computing était la vedette incontestée. Et l'éditeur veut le faire savoir en mettant en avant très fortement Windows Azure, Online Services.

Toute la session inaugurale a été centrée autour de ces offres : un peu de code, un peu d'administration et d'architecture. Aucune annonce n'a ponctué l'ouverture de l'événement...

Les sessions autour de Windows Azure étaient pleines et l'affluence importante partout. Nous avons particulièrement aimé la présentation de DyadLinq. C'est un projet actuellement en pré-version visant à massivement distribuer les données très volumineuses pour les calculs intensifs de type HPC. Et sa combinaison



avec Linq permet des requêtes souples mais surpuissantes. Deux annonces autour de cette technologie : une version finale courant de l'été et un support de Windows Azure, dans les tuyaux. Occasion aussi de découvrir la nouvelle offre SaaS, Office 365, actuellement en pré-version. Côté développement fun, une session à revoir absolument : .Net microframework, petits montages électroniques à la portée des softueux (sic) ! Voilà une heure bien geek, de quoi nous réjouir !

Nous avons été particulièrement excité par la session robot et Windows Azure, où comment un projet robotique peut utiliser un service Azure ! Le prototype clorofeel de Nicolas Clerc s'appuie sur un service AppFabric pour faire une passerelle entre le robot et

la partie cliente (application de remote contrôle ou via une page web). Il est possible d'exposer des services robotiques sur un service AppFabric puis d'utiliser ce point d'entrée dans une application remote pour contrôler le robot ou récupérer des données. Cela prouve que le cloud peut servir à de nombreux utilisateurs... La dernière journée proposa un peu de prospective avec les notions d'interface naturelle et comment l'informatique allait évoluer dans les 5 à 10 ans à venir (voir notre édito sur le sujet).

Cette édition 2011 a montré que le cloud est devenu une réalité même si les questions demeurent pour les développeurs et utilisateurs. Et le plus intéressant n'est pas dans les sessions d'ouverture mais dans les sessions !

agenda \

MARS

• Du 02 au 05 mars 2011, Cité Universitaire Internationale, 17 bd Jourdan 75014 Paris,

Conférences Symfony Live 2011

Sensio Labs annonce ses conférences Symfony pour 'évangéliser autour du framework, et plus largement autour de la professionnalisation de PHP et la dissémination des bonnes pratiques de

développement." <http://www.symfony-live.com/paris/schedule>

• Du 08 au 10 mars, Porte de Versailles - Paris. **In the Business**. L'événement regroupe : In the Cloud, In the Datacenter, Security et SaaS, www.inthe-cloud.com

• Du 08 au 10 mars 2011, Porte de Versailles - Pavillon 4 - Paris. **Serious Games 2011**, dans le cadre d'elearning expo.

<http://www.solutions-ressources-humanaines.com>

• Du 23 au 24 mars 2011, CNIT Paris La Défense, 17e édition du salon **Documentation 2011** sur la gestion de l'information, du contenu et du document. www.documentation.fr

• Le 28 mars 2011, 14h30, **Webinar : Zend Studio 8** : Tout pour que vos développements PHP atteignent un autre niveau ! <http://www zend.com>

AVRIL

Le 05 avril 2011, Paris, Quai de Grenelle, **Journée française du test logiciel 2011**. <http://www.cftl.net>

ETRANGER

Du 31 mars au 1er avril 2011, Hambourg, Allemagne, **quatrième colloque européen sur le langage Lisp**

<http://www.european-lisp-symposium.org>

Nokia se recentre sur le matériel

Le paysage du mobile se redessine peu à peu. Nokia a du se rendre à l'évidence. On a beau vendre des millions de téléphones mobiles et être leader du marché, on ne s'invente pas fabricant de smartphones comme cela, malgré Symbian, Qt, Meego. Et l'offensive d'Apple et Google a montré que Nokia avait du mal à réagir et à proposer une alternative crédible. Malgré quelques nouveautés depuis un an, le constructeur n'a pas su trouver la bonne formule. Et le 11 février dernier, le couperet est tombé après plusieurs mois de rumeurs, de mise en garde. Nokia

facto le système mobile de référence chez Nokia, au détriment de Symbian et de Meego. Nokia précise tout de même :

- **Symbian** devient une plate-forme en franchise, Nokia continuera à le supporter mais sera surtout utilisé par des constructeurs tiers
- **Meego** : le système open source pour la mobilité continue son chemin avec une disponibilité attendue vers la fin 2011. En revanche, si Nokia annonce qu'il sera un axe de long terme, on attend un positionnement clair du constructeur. Meego n'est pas totalement oublié mais quel est



va fabriquer des smartphones basés sur Windows Phone 7 de Microsoft ! Pour l'éditeur, c'est une bonne nouvelle pour contrer la vive concurrence et un départ en retard de 18 mois, mais il manquait un acteur de poids. C'est chose faite avec Nokia. Pour le constructeur finlandais, c'est un revirement technologique et stratégique sans précédent. Jusqu'à présent, Nokia maîtrisait matériel et logiciel. Avec l'arrivée de Windows Phone, le constructeur perd une partie de sa maîtrise même si Microsoft autorise des modifications, ajouts dans son système comme sur la cartographie, le mail. Par contre, le Marketplace de Windows Phone s'imposera. Ovi Store avait beau présenter des millions de téléchargement, il ne faisait pas le poids contre iOS et Android. De nombreuses questions se posent, en attendant la sortie des premiers smartphones. Windows Phone 7 devient de

son avenir exact chez Nokia ? La question se pose à l'heure où nous écrivons. Idem pour Symbian qui n'a plus beaucoup d'intérêt hormis dans les téléphones mobiles « simples ». Et dès l'annonce de l'accord, plusieurs milliers d'emploi risquent d'être supprimés.

Quel avenir pour Qt ?

Racheté par Nokia, Qt était une des technologies phare du constructeur dans la mobilité pour bâtir des interfaces universelles en C++. L'équipe Qt (<http://blog.qt.nokia.com>) a voulu rassurer, un peu, mais l'avenir demeure incertain malgré de nouvelles versions en préparation et un lien toujours fort avec Meego. Finalement, Qt a-t-il encore un avenir chez Nokia ? Surtout qu'un support dans Windows Phone 7 n'est pas d'actualité. La réponse dans quelques mois...

Passionné(e) par le développement et les nouvelles technologies Microsoft ?

Connaissez-vous un des sujets XML, .NET Framework, C#, Web Services, WCF, WF, SQL Server, BizTalk Server, SharePoint, SOA, ASP.NET ?

Spécialisez-vous chez Codit, leader en intégration, et devenez expert en Microsoft BizTalk, AppFabric et Azure !

En savoir plus ? Veuillez visiter la page «jobs» sur www.codit.eu et contactez-nous.

www.codit.eu
jobs@codit.eu

codit
integrating your business

Des processeurs « software defined radio » pour améliorer les réseaux

Freescale, fabricant de semi-conducteurs, annonçait une nouvelle gamme de processeurs hybrides multistandard destinés à accompagner à moindre coût et à moindre consommation la croissance du trafic des données dans les réseaux mobiles. Ce sont en fait 3 processeurs, conçus pour équiper les cellules de grande taille (plusieurs centaines d'utilisateurs), les femto cellules (quelques dizaines d'utilisateurs) ou les cellules à la maison (quelques utilisateurs), qui seront disponibles progressivement à partir du 4ème trimestre 2011. Ils offrent une grande flexibilité de migration aux opérateurs car ils peuvent prendre en charge une grande variété de radios (3G et LTE) grâce à des technologies dites « software defined radio » dans des situations très variées, sans conduire à des adjonctions coûteuses de cartes spécifiques à chaque radio. Freescale est aujourd'hui le seul fabricant à intégrer sur une seule puce l'ensemble des capacités de traitement et d'accélération du signal radio en multistandard. Il utilise des technologies de microprocesseurs à 45 nanomètres et passera au 28 nano l'année prochaine.

■ Alain Baritault

Envie d'un nouvel écran ?



Nec propose un écran de 30 pouces, SpectraView Reference 301, affichage panoramique. Cette dalle est idéale pour les développeurs ayant besoin de grands espaces. Il s'utilise parfaitement en environnement de production photo et vidéo grâce à ses fonctions de calibration. Le moniteur affiche une résolution de 2560 x 1600 pixels au format 16:10, un confort de travail parfait pour une utilisation intensive. Le prix est à la hauteur des prestations : plus de 3800 euros...

HP lance (enfin) une vraie tablette !



Après le rachat de Palm et des systèmes de l'éditeur, HP avait beaucoup hésité sur la stratégie notamment sur le marché des tablettes tactiles. Désormais, le constructeur veut réellement concurrencer Apple et les autres fournisseurs avec le TouchPad. Il supportera diverses technologies : Exchange, codecs vidéo divers et variés, réseaux sans fil, le clavier virtuel, pour un poids de 740 grammes. Et certains échanges avec un téléphone mobile seront possibles, un peu comme avec la tablette Surface. Pour le système, webOS est utilisé. C'était une des valeurs technologiques de Palm. Un nouveau SDK est disponible pour les développeurs. L'OS a été visiblement retaillé, optimisé et complété de nombreuses fonctions pour arriver à rivaliser avec un iPad ou un Android 3.0. Flash sera présent par défaut. Ce nouveau webOS sera implémenté sur plusieurs autres terminaux HP courant 2011.

Tabbee 2 : stop ou encore ?

Orange a présenté la nouvelle évolution de son terminal tactile Tabbee, l'opérateur rêve de concurrencer Apple avec sa propre solution mais le succès et l'intérêt du produit restent à démontrer. Elle utilise un processeur ARM, possède une sortie HDMI, une mémoire Flash limitée à 2 Go, la possibilité de faire de la vidéo-conférence. Le tout utilise un système Windows CE mais avec une personnalité propre à Orange. Disponibilité : printemps 2011.



Intel propose un affichage sans-fil

Widi, une technologie d'affichage sans fil (via un simple réseau WiFi) sur un téléviseur, ou tout autre écran compatible. Il faut que le netbook supporte la fonction et que le téléviseur soit connecté à un boîtier externe pour recevoir le signal. Cela ressemble à la fonction Air Play d'Apple iOS. Il faut que le processeur soit tout de même



très récent pour en bénéficier. A terme, Intel souhaite intégrer Widi aux téléviseurs sans passer par des boîtiers de type Netgear Push2TV. Pour le moment la technologie fonctionne uniquement sur Windows. Et un petit logo Widi indique si le matériel est compatible...

La table tactile pour DJ : trop fort !

Une des attractions de la dernière journée des Microsoft Techdays 2011 était la présence de la superbe table tactile pour DJ : Emulator de Smithson Martin. Système entièrement multitouch, Emulator



est impressionnant par son ergonomie et la souplesse d'utilisation. Actuellement disponible sous Windows 7, une version MacOS X est en développement. L'outil montre parfaitement les possibilités du tactile dans une utilisation ludique. Le système complet comprend un « écran », son support et un projecteur. Le prix est à la hauteur : plus de 3 800 dollars (prix de départ).

HTML 5 : disponible en... 2014 !

Les consortiums et autres alliances avancent parfois très lentement sur les spécifications de nouveaux standards. Ces critiques sont parfois justifiées. Le W3C va sans doute subir quelques critiques car il a déclaré mi-février que HTML 5 pourrait être disponible d'ici à 2014. Une phase de ce langage sera lancée en mai prochain avec la notion de dernier appel. Le W3C précise sur cet appel qu'il s'agit d'une « étape importante marquant la satisfaction des exigences techniques. C'est aussi l'étape où les communautés de développeurs sont invitées à envoyer leurs commentaires sur la spécification ». Puis, le groupe en charge de HTML 5 se concentrera sur les retours des implémentateurs et les tests. Le but étant d'obtenir une interopérabilité maximale d'ici à 2014 (dixit le W3C). « Même si l'innovation suit

son cours, l'évolution de HTML5 vers la recommandation fournit à l'ensemble de l'écosystème Web une spécification interopérable, testée et stable » déclare Jeff Jaffe, CEO du W3C. « La décision de planifier le dernier appel de HTML5 en mai 2011 était un pas important pour définir les attentes de l'industrie. Aujourd'hui, nous franchissons un autre cap en annonçant 2014 comme date cible pour un standard. »

HTML5 constitue une étape majeure pour le Web et l'éditeur. Depuis plus d'un an, les annonces se multiplient, parfois à tort car les spécifications ne sont pas sèches pour implémenter des balises non définitives. Microsoft a voulu limiter le support de HTML5 dans IE9 pour ne pas être obligé de revenir en arrière, causant ainsi des problèmes de compatibilité pour les développeurs. Quand vous utilisez

une fonction HTML5, il faut donc être prudent. La notion de compatibilité est vitale dans ce contexte et le W3C se veut rassurant : « Le Web est devenu une ressource mondiale décisive qui transforme tous les secteurs, dont l'industrie mobile, la télévision, les jeux, l'édition et la santé. Les industries tournées vers l'avenir adoptent le Web comme plate-forme privilégiée pour intégrer différents périphériques, services et modèles économiques. Les audiences se diversifient, le besoin d'interopérabilité parmi les nombreuses technologies de la plate-forme Web ouverte du W3C, comme HTML5, CSS, SVG, WOFF, et les différentes API, devient donc crucial. La plupart de ces technologies sont déjà utilisées, à des degrés divers de maturité et d'implémentation. Etant donné que HTML5 est au cœur même de la plate-forme Web



ouverte, le W3C a commencé de travailler sur une suite de tests complète pour assurer un niveau élevé d'interopérabilité tel que le demandent les différentes industries. Le W3C invite les membres de la communauté à contribuer à la suite de tests qui permettra aux implémentateurs logiciels de satisfaire les critères de mise en œuvre du W3C et facilitera la création de contenu et d'applications. Ces tests joueront un rôle important dans l'achèvement du standard dans les temps. »

Cependant, il vaut mieux attendre une version totalement figée et les premières implémentations complètes pour juger de la qualité de HTML 5 et l'intérêt de migrer.



WHAT'S NEXT ?

LA conférence technique sur l'avenir de Java
26-27 Mai 2011, Le Grand Rex, Paris

Un événement exceptionnel à Paris !

Les meilleurs experts rassemblés pour vous

Découvrez les technologies & acteurs de demain...

www.whatsnextparis.com



J'ai choisi 1&1 car ils bénéficient de serveurs d'une grande stabilité, de coûts abordables, d'une gamme de services très complète et d'une bonne réactivité en hotline.

Sylvain Jacquet
www.lemagasinbio.com



VOTRE SITE TOUT COMPRIS GRATUIT PENDANT UN AN !*



*Le Pack Initial est gratuit pendant 1 an sous réserve d'un engagement de 24 mois. Frais de mise en service de 5,97 € TTC. A l'issue de la première année, le Pack Initial est au prix habituel de 2,38 € TTC/mois. Offre domaine applicable uniquement la première année au lieu du prix habituel de 6,99 € HT/an (8,36 € TTC). Conditions détaillées sur www.1and1.fr. Offres sans engagement également disponibles.



1&1 PACK INITIAL 1 AN GRATUIT* OFFRE PROLONGÉE !

1&1, des packs hébergement tout compris au meilleur prix !

.com
.fr
.org
.net

Nom de domaine inclus

Profitez d'un nom de domaine inclus pendant toute la durée de votre pack, sans coût supplémentaire.



Traffic illimité

Restez à l'abri des mauvaises surprises sur votre facture grâce au trafic illimité.



Logiciels avancés de web design

Concevez votre site comme un pro et optimisez-le pour la consultation sur terminaux mobiles.



Fonctions pour développeurs

Disposez de toutes les fonctions nécessaires pour développer votre site et vos applications dans votre environnement favori.



Outils de marketing en ligne

Propulsez votre site en tête des moteurs de recherche à l'aide des outils de référencement 1&1.



Centres de données verts

Réduisez votre impact sur l'environnement en optant pour un hébergement 1&1.

1&1 PACK INITIAL :

- Nom de domaine inclus
- 2,5 Go d'espace disque
- Trafic ILLIMITÉ
- 10 comptes email (POP3 et IMAP)
- Accès FTP
- Base de données MySQL
- e-Boutique Start
- Outils de référencement
- Assistance technique par email et téléphone
- Et bien plus encore...

0 €* ~~1.00€~~
HT/mois (3,36 € TTC/mois)
pendant un an

Retrouvez toutes nos offres du moment sur 1and1.fr, comme par exemple le nom de domaine en **.fr à seulement 3,99 € HT/an (4,77 € TTC/an)*** !



Appelez le **0970 808 911** (appel non surtaxé) ou consultez notre site Web

www.1and1.fr

Votre site web sur smartphone de A à Z



Avec l'explosion de l'internet mobile, grâce au succès croissant des smartphones, la question n'est plus de savoir s'il faut un site web mobile mais comment le réaliser rapidement tout en prenant soin de l'optimiser et de l'adapter au mieux aux terminaux mobiles.

Il existe plusieurs plateformes mobiles (iOS, Android, Windows Phone 7, Bada, Blackberry, etc.), mais finalement les fondamentaux restent identiques : un navigateur, un moteur de rendu. On affichera du javascript, du HTML, et différentes technologies selon le support proposé par les terminaux. Cependant, le développeur, le webmaster devra tout de même se méfier de la compatibilité, des technologies



disponibles, des contraintes techniques comme la taille des écrans, l'ergonomie. Mais au-delà du pur aspect développement, il faut se poser la question : quel contenu je veux y mettre ?

Car, un site mobile n'est pas un site desktop. On ne peut pas tout y mettre et ce n'est pas l'objectif. Il faut donc choisir le contenu à exposer sur mobile et selon les contraintes techniques du terminal. D'autre part, ne jamais oublier que l'on est sur du mobile donc avec des connexions pas toujours optimales même en 3G, plus un contenu, un design sera complexe, plus il sera long à charger. Il faudra donc trouver un juste équilibre.

Le sujet est très vaste et nous allons vous proposer un dossier complet dessus afin de bien comprendre les enjeux, les difficultés, les avantages d'une telle adaptation.

■ François Tonic

Le navigateur mobile : le nerf de la guerre



Les versions ne sont pas toujours simples à trouver sur un téléphone, ni les technologies réellement supportées et si chaque plateforme arrive avec son navigateur intégré, il est souvent possible d'en installer un autre. Il existe de très nombreux navigateurs en dehors de ceux des fournisseurs de systèmes mobiles. Le web mobile représente une opportunité considérable pour les développeurs et le contenu. Il ne faut donc pas le délaisser.

Par développement web mobile, nous entendons le développement, l'adaptation de sites web s'affichant dans des navigateurs internet, contrairement aux applications natives s'exécutant directement sur le système du terminal. Bien entendu, il faudra tester un minimum votre site sur différentes plateformes et divers navigateurs car comme sur le desktop, l'affichage, le fonctionnement peuvent varier. Cibler au moins les principaux navigateurs.

WebKit : le géant sous le capot

Plusieurs navigateurs mobiles (iOS, QtWebkit, Android, Blackberry...) s'appuient sur le même socle technique, le moteur de rendu WebKit. La version embarquée sur smartphone

Qui dit internet mobile, dit obligatoirement navigateur mobile sur les terminaux. Aujourd'hui, il existe deux principaux types de navigateurs pour smartphones : ceux basés sur le moteur Webkit et les autres (Opera, IE, etc.). WebKit est sans doute le moteur de rendu le plus performant et le plus en avance sur les plateformes mobiles grâce à Android et iOS. Par contre, Mozilla travaille toujours à la finalisation de son navigateur mobile. Et Microsoft peaufine aussi les prochaines mises à jour.

est identique à la version desktop, hormis qu'elle nécessite moins de mémoire et affiche des performances très satisfaisantes. Sa modularité l'a rendu populaire. Chaque éditeur peut l'adapter à ses propres besoins et faire évoluer le navigateur à sa guise car le moteur évolue chaque jour...

Pour en savoir plus : <http://developer.apple.com/devcenter/safari/index.action>

Mozilla Fennec

Pour la mobilité, la fondation Mozilla développe le projet Fennec qui se présente comme un Firefox taillé pour les smartphones. Il fonctionne sur Android et Maemo. Pas d'autres versions de prévues pour le moment (iOS, Windows Phone, etc.). Fennec se base sur XUL, supporte les écrans tactiles et partage de nombreux éléments de Firefox. Ainsi les deux versions possèdent le même moteur de rendu, le support des extensions, la gestion des téléchargements, le marque-page, le javascript avec support JIT. Selon les développeurs, Fennec prend en compte les contraintes des terminaux mobiles : taille de l'écran, batterie, CPU, mémoire, clavier et multitouch. Il propose un système d'onglets avec différentes ergonomies selon l'orientation du terminal.

Fennec 2 devrait être disponible fin mars. Actuellement, le projet est en bêta 4. Cette version se focalise beaucoup sur l'interface, l'approche multiprocess (projet Electrolysis). La partie layer sera largement améliorée avec de nouvelles API. Et l'intégration avec Android est une des priorités.

Pour en savoir plus : <https://wiki.mozilla.org/Mobile/Fennec/Architecture>
et
<https://wiki.mozilla.org/Mobile/Planning/2.0>

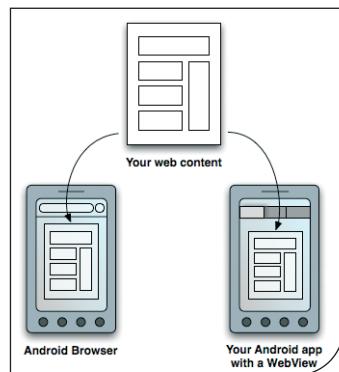
Internet Explorer sur Windows Phone 7

Comme toute plateforme mobile, le nouveau Windows Phone 7 intègre un navigateur mobile. Celui-ci est par défaut compatible HTML 4, xHTML, XML, CSS 2, JavaScript, ECMAScript 3, Ajax, etc. Il utilise un socle IE 7. Sur Silverlight, la technologie n'était pas disponible dans le navigateur et elle est prévue dans une future mise à jour. Pour en savoir plus : [http://msdn.microsoft.com/enus/library/ff462082\(v=vs.92\).aspx](http://msdn.microsoft.com/enus/library/ff462082(v=vs.92).aspx)

Bada embarque son propre navigateur

Samsung utilise pour son système Bada, Dolphin. Il s'agit d'un mini navigateur supportant HTML 5, HTML,

Flash. Une version est disponible sur Android. Il se base sur Webkit. Et il passe plutôt bien les tests de performances et de compatibilité, par contre, sur le Javascript il est plutôt en retrait. Mais le marché continue d'évoluer car chaque fournisseur étoffe son offre web et les nouveaux formats HTML5, WebM, la 3D, vont rapidement devenir des standards du smartphone. Certains proposent une autre manière de naviguer comme SurfCube...



■ F. T.

Comment adapter son site web aux Smartphones ?

Accéder à internet à partir de son smartphone est devenu courant. En France, entre janvier et août 2010, la vente de ces appareils a connu une croissance en volume de plus de 138% selon l'institut GfK. Désormais, un quart des téléphones vendus est un smartphone.

Malgré tout, il y a encore de nombreuses contraintes qui rendent la navigation difficile : résolution réduite, temps de chargements importants etc. Autant de raisons de vouloir offrir à vos visiteurs une version de votre site optimisée pour les mobiles.

Impensable de faire une version pour tous les mobiles

Vous trouvez fastidieux d'assurer une compatibilité de votre site pour Firefox, Chrome et Internet Explorer 6, 7 et 8 ? Bienvenue dans le monde des mobiles ! Ici, vous aurez bien plus de navigateurs à supporter, chaque fabricant ayant son propre navigateur avec ses spécificités. Vous n'avez probablement pas le temps ni les moyens de tester tous les modèles de téléphones existants. Nous pouvons cependant séparer les smartphones en 2 catégories :

- Les smartphone récents dans lesquels nous mettrons iPhone, Android, les dernières versions de Blackberry, WP7
- Les smartphones low-cost ou plus anciens (Nokia E71, anciens Blackberry...)

Les stratégies de support seront très différentes selon la catégorie visée. Pour la première catégorie, on peut adapter un site existant, avec quelques CSS et Javascript personnalisés. Pour les plus anciens, il faudra penser à fournir des templates spécifiques.

Si vous développez un site web d'entreprise, vous optimiserez votre site pour les smartphones de l'entreprise. Si vous développez un site grand public, vous pouvez vous décider en fonction des parts de marché des différents appareils. Attention, celles-ci sont très différentes suivant les pays! Alors qu'au niveau mondial Opera mobile est le navigateur le plus utilisé (principalement à cause de l'Asie), aux Etats-Unis, le marché est réparti équitablement entre iOS, Android et Blackberry. Si votre site web est destiné à un public exclusivement français, le choix sera facile. En effet, la France fait figure d'exception, avec une écrasante domination d'Apple, les mobiles sous iOS effectuant 64% des requêtes, très loin devant les autres navigateurs mobiles.

Quelles sont les techniques pour détecter un mobile ?

Pour les téléphones les plus récents : Media Queries

La grande majorité des Smartphones récents (iPhone, Android et Blackberry ...) utilise un navigateur basé sur le moteur open-source Webkit, qui supporte bien les feuilles de style CSS3. Il est donc possible d'utiliser les « media queries ». Ceux-ci permettent d'inclure un contenu CSS de manière conditionnelle, donc en fonction des capacités d'affichage :



```
<link rel="stylesheet" media="screen and (max-width: 480px)" href="480px.css" />
```

L'exemple ci-dessus inclura dynamiquement la feuille CSS « 480px.css » si la largeur de l'écran est inférieure à 480 pixels. On peut ainsi créer différents layouts en fonction de la taille de l'écran. Attention, l'iPhone croit disposer d'une largeur d'écran plus grande que sa résolution native puisqu'il est possible de zoomer et de zoomer grâce au « pinch ». Aussi, pour que les media queries fonctionnent correctement, il faudra indiquer :

```
<meta name="viewport" content="width=device-width" />
```

Pour les téléphones anciens : Détection par User-Agent

Chaque navigateur, même sur un mobile, envoie son nom et sa version dans chaque requête HTTP, dans le header User-Agent. Il est donc possible de savoir si on accède à votre site à partir d'un mobile ou non.

De manière générale, se baser sur l'User-Agent est cependant une mauvaise pratique. En effet, il existe de très nombreux User-Agent et de nouveaux apparaissent pour chaque génération de téléphone. Il est donc très complexe de garder une liste à jour.

Aussi, on préfère généralement se baser sur la détection des capacités du téléphone en Javascript. Cependant, si vous voulez supporter des téléphones basiques, qui n'ont pas de support Javascript, vous n'aurez pas d'autre choix que d'utiliser le User-Agent.

Une détection en PHP peut ressembler à ceci :

```
<?php
$iphone = strpos($_SERVER['HTTP_USER_AGENT'], "iPhone");
$android = strpos($_SERVER['HTTP_USER_AGENT'], "Android");
$palmpre = strpos($_SERVER['HTTP_USER_AGENT'], "webOS");
$ipod = strpos($_SERVER['HTTP_USER_AGENT'], "iPod");
if($iphone || $android || $palmpre || $ipod) {
    // le code à executer pour les smartphones
}
?>
```

Si vous désirez avoir plus d'informations concernant les User-Agent, les sites Internet suivants peuvent vous aider : Handset Detection.com, WURFL ou Zytrax.

Si vous avez un CMS, un module existe peut-être

Si vous avez de la chance et que votre site web est construit autour d'un CMS, il est possible qu'un module adapte directement votre site

internet. Par exemple, pour Drupal, le module « Mobile Plugin » fait presque tout à votre place : il détecte le mobile puis change automatiquement le template selon la plateforme du mobile. Il permet également de configurer des blocs différents pour le site mobile. Pour Joomla, il s'agit du module « Mobilebot ».

Adapter son site Internet : les éléments à éviter Flash

L'annonce du non support de Flash dans l'iPhone a provoqué un buzz qui a même été repris par la presse généraliste. Inutile donc de vous dire que Flash n'est pas une option dans le monde mobile et ne le sera pas tant qu'Apple sera dominant / important sur ce marché.

Javascript

Le support Javascript sur un site web mobile est un sujet complexe. Javascript n'est pas supporté ou activé par défaut sur tous les mobiles. Si vous vissez le support le plus large, mieux vaut renoncer au Javascript ou ne l'utiliser que de façon cosmétique. Si vous vissez principalement les smartphones récents, le support Javascript peut alors se révéler important.

Vous pourrez par exemple tirer parti de l'API de géolocalisation, qui a été récemment standardisée par le W3C, ou encore exploiter les API permettant d'effectuer du multi-touch. Attention cependant, plus vous souhaiterez élargir le nombre de téléphones supportés, plus l'effort d'adaptation sera important. Le site Quirksmode référence les fonctionnalités supportées par chaque navigateur mobile : <http://www.quirksmode.org/m/table.html>

Adapter son site Internet : les éléments à modifier

Penser au confort de lecture

Chaque navigateur mobile offre une police avec une taille adaptée. Il vaut donc mieux éviter de préciser une taille en pixels pour les polices de caractères. Utilisez plutôt les valeurs relatives comme «medium, large, small, x-large, etc.» qui seront interprétées selon la configuration du navigateur et les préférences de l'utilisateur.

Alléger vos pages - quelques conseils

Du fait de la taille de l'écran, réduisez au maximum l'encombrement du site sur les côtés. Verticalisez le plus possible votre mise en page. Un site constitué par une seule colonne est beaucoup plus lisible. Il faut être attentif aux hyperliens. Un lien un peu long sur un écran d'ordinateur peut s'afficher sur 3 à 4 lignes sur un écran mobile. C'est moche !

L'apparition de l'écran tactile procure un vrai plaisir de navigation sur mobile, il faut cependant tenir compte que la précision des doigts est beaucoup moins importante que celle du stylet. Il est parfois préférable de simplifier la navigation. N'oubliez pas non plus qu'en l'absence de souris, les aides contextuelles de type « hover » ne s'afficheront pas. N'hésitez donc pas à revoir votre ergonomie !

Adapter l'affichage

Si on exclut les tablettes, les écrans de smartphones ont des résolutions allant de 240x240px jusqu'à 640x960px pour l'iPhone 4. Donc, il vaut mieux éviter les layouts « fixes » et préférer les mises en pages de type « fluid », dans lesquelles les dimensions sont indiquées en pourcentage de la largeur de l'écran.

Autre problème : les feuilles de style se comportent différemment selon les navigateurs. Encore une fois, le site Quirksmode peut vous

aider avec un tableau très détaillé des comportements CSS en fonction des navigateurs mobile : <http://www.quirksmode.org/m/css.html>

Améliorer les performances

Comparés à une ligne ADSL, les réseaux mobiles sont bien plus lents. Le temps de latence peut dépasser 300ms. Il est donc très important de respecter les bonnes pratiques de performance sur un site web mobile :

- Limiter la taille des fichiers envoyés : compresser fortement les images, minimiser les fichiers CSS et Javascript
- Limiter le nombre de fichiers par page : autant que possible, agréger les styles dans un fichier CSS unique, agréger les fichiers Javascript également.
- Concaténer les images du thème en une image unique, et les afficher grâce à la technique du « sprite CSS ».

Des outils comme YSlow de Yahoo! ou PageSpeed de Google se chargeront de détecter pour vous les points à améliorer pour optimiser vos performances.

Il est temps de tester !

Votre site web est prêt ? Il est temps de le tester ! Heureusement, pas besoin de disposer de tous les mobiles pour tester le rendu !

Les fabricants mettent à disposition des développeurs des outils de simulation. Rendez-vous sur le site de [Programmez](http://Programmez.com) pour la liste des liens vers les outils de tests !



■ **Qing Sun**

Responsable de l'offre mobilité chez The Coding Machine, société spécialisée dans le développement web

En complément

La meilleure façon de savoir si votre site web mobile est adapté aux Smartphones, est de le tester. Au lieu d'acheter tous les types de Smartphone pour tester votre site web. Il existe aussi les outils qui sont gratuits et simples à utiliser. Les principaux constructeurs de Smartphones proposent des simulateurs (ou émulateurs) de leur Smartphone. Cela permet aux développeurs de tester leur site web mobile sur leur ordinateur.

Opera mobile simulator : <http://www.opera.com/developer/tools/>

Blackberry : <http://www.blackberry.com/developers/downloads/simulators/index.shtml>

Android : <http://developer.android.com/sdk/index.html>

Nokia : http://www.forum.nokia.com/info/sw.nokia.com/id/db2c69a2-4066-46ff-81c4-caac8872a7c5/NMB40_install.zip.html

Windows Phone 7 : <http://www.redmondpie.com/standalone-windows-phone-7-series-emulator-9140536/>

Palm Pre : <http://developer.palm.com/>

Site web de test

Un simulateur en ligne : <http://emulator.mtd.mobi/emulator.php>

iPhone Simulator en ligne <http://www.testiphone.com>

Pour aller plus loin, nous vous conseillons de lire les bonnes pratiques du Web mobile en Flipcards édité par le W3C. C'est un excellent résumé du guide complet des Pratiques exemplaires du Web mobile 1.0.

Web, mensonges et vidéos

Et tout d'abord, il nous semble utile de faire le point sur l'intégration de la vidéo dans les navigateurs classiques.



Le règne des plug-ins
Avant HTML 5, la vidéo sur le Web a été l'apanage d'implémentations propriétaires dans les navigateurs, et surtout de plug-ins comme Apple Quicktime ou le lecteur Adobe Flash. Dans le contenu HTML, ces plug-ins sont activés via la balise `<object>`.

Pour la plupart des utilisateurs, peu importent les détails techniques

tant que la vidéo fonctionne, et la nécessité d'installer un plug-in au préalable est rapidement oubliée lorsqu'on se promène sur YouTube. Sur le Web classique d'aujourd'hui, les vidéos Flash ont donc une bonne longueur d'avance. Cette approche pose néanmoins quelques problèmes liés au fait que, du point de vue du navigateur, une balise `<object>` est une « boîte noire » contrôlée par le plug-in. Impossible de dessiner du contenu au-dessus de cette boîte noire, si on veut par exemple ajouter du texte sur une vidéo en HTML ou appliquer des styles CSS, voire un filtre SVG. De même, il n'existe pas de moyen standard pour contrôler la boîte noire depuis le reste de la page. En particulier, l'interface graphique permettant la manipulation de la vidéo (pause, lecture, avance rapide, volume sonore...) est imposée par le plug-in.

Enfin, le navigateur et le plug-in restent deux entités séparées, qui



suivent des agendas différents, et le plug-in n'est donc pas forcément disponible ou adapté à un environnement donné. C'est – surprise ! – notamment le cas du monde mobile où ces plug-ins commencent tout juste à faire une timide apparition.

La balise `<video>`

HTML 5 définit la balise `<video>` pour résoudre ces problèmes. C'est une balise comme une autre et, au même titre qu'une balise `<p>` ou `<div>`, parfaitement intégrée au reste de la page. Du contenu HTML peut être affiché par-dessus la vidéo, dont la présentation à l'écran peut être précisée en CSS.

Aucun plug-in n'est nécessaire. HTML 5 définit une interface de programmation standard pour manipuler la vidéo depuis JavaScript.

HTML 5 sur les téléphones

HTML 5 (et son jumeau XHTML 5) est la prochaine version de HTML en cours de développement au sein du W3C. Bien qu'il soit encore loin d'être finalisé, les navigateurs s'empressent d'en implémenter les fonctionnalités les plus emblématiques, comme les balises `<video>` et `<canvas>` ou le fonctionnement hors-ligne via l'attribut `manifest`. Au-delà des ajouts de HTML 5 proprement dits, les navigateurs se concurrencent aussi sur les API JavaScript additionnelles qui gravitent autour de HTML 5 : citons par exemple l'API de parallélisme (Web Workers), l'API de base de données locale IndexedDB, ainsi que toute une panoplie d'intégration du navigateur aux différents capteurs embarqués, comme l'API de géolocalisation, que nous évoquerons au dernier chapitre. Les navigateurs mobiles sont loin d'être en reste

dans cette saine compétition et les versions les plus récentes sur les téléphones haut de gamme (smartphones de type iPhone, Android, BlackBerry, séries N de Nokia) commencent à intégrer ces nouvelles fonctionnalités.

Néanmoins, si HTML 5 est sans nul doute d'une importance extrême pour le Web mobile, il convient de réaliser que son déploiement complet effectif est à ce jour très marginal et que de nombreux utilisateurs n'y auront pas accès avant un certain temps, soit parce que leurs téléphones ne disposent pas de navigateurs compatibles, soit parce qu'ils n'auront pas effectué les mises à jour le leur permettant. Il conviendra donc d'utiliser ces fonctionnalités avancées avec parcimonie et, autant que possible, dans une perspective d'amélioration progressive,

c'est-à-dire en s'assurant que le contenu HTML reste compatible avec les navigateurs moins avancés.

Le Document Type de HTML 5 peut cependant d'ores et déjà être utilisé sans conséquence néfaste, si vous remplissez tous les critères suivants :

- Vous avez effectivement l'intention d'utiliser des fonctionnalités introduites par HTML 5 – typiquement l'attribut `manifest`.
- Vous savez distinguer les fonctionnalités prises en charge par les téléphones mobiles des fonctionnalités avancées qui ne sont pas encore déployées.
- Vous êtes prêt à vous astreindre à la discipline et à ne pas utiliser les fonctionnalités avancées qui pourraient rendre le contenu incompatible avec des navigateurs moins avancés.
- Vous savez utiliser les outils de validation d'un document HTML pour contrôler le bon respect de cette discipline.

Fini les contrôles imposés, l'interface graphique peut être ainsi complètement repensée.

La balise <video> est déjà présente dans les dernières versions des navigateurs de bureau, y compris Internet Explorer 9 (à venir à l'heure où nous écrivons ces lignes).

Quels formats pour le Web mobile ?

Tout n'est pas rose pour autant. La prise en charge de la vidéo achoppe malheureusement sur la définition d'un format de vidéo commun à tous.

Comme pour les images, plusieurs formats sont possibles et l'existence de nombreux brevets dans le domaine rend très difficile la définition d'un format acceptable par tous. En outre, les techniques de diffusion de la vidéo (streaming) sont en pleine effervescence et auront sans doute, au final, un impact sur les formats.

Trois formats – on parle plutôt de codec pour les vidéos – semblent se dégager du lot pour la vidéo sur le Web (H.264, Theora et VP8), sans qu'aucun d'eux ne soit pris en charge par tous les navigateurs. H.264 est un codec mûr qui dispose d'un large choix d'implémentations faisant appel à l'accélération matérielle pour réduire la puissance nécessaire à la lecture de la vidéo ; les brevets sous-jacents sont clairement identifiés – de même que les royalties à verser pour ceux qui l'implémentent. À l'inverse, Theora et VP8 sont plus récents, pas encore complètement optimisés et, s'ils sont bien supposés être libres de droits, personne ne semble être en mesure de le prouver réellement. Difficile de prédire l'avenir dans ce domaine...

La vidéo dans les mobiles

Les téléphones mobiles autorisent-ils la vidéo ? La réponse était négative il y a un an encore. Les plug-ins nécessaires à la lecture de

Code, code, codec

Un fichier vidéo se divise en réalité, d'une part, en des flux audio et vidéo et, d'autre part, en un conteneur pour ces flux. Il existe donc des formats pour les conteneurs et d'autres pour la vidéo et pour l'audio : les codecs (abréviation de codage-décodage). Il existe également des formats pour les sous-titres et les annotations. Enfin, certains codecs vidéo (comme le H.264) se déclinent en une bonne dizaine de profils.

Pour être un peu plus exacts, nous devrions donc plutôt parler du profil principal du codec H.264 dans un conteneur MP4, du codec Theora dans un conteneur Ogg et du codec VP8 dans un conteneur WebM.



vidéo n'existaient pas et la balise <video> n'était pas non plus gérée par les navigateurs. Certains permettaient néanmoins d'ouvrir certaines vidéos à l'extérieur du navigateur, via une application dédiée du système d'exploitation (sur Android ou iPhone, par exemple). La situation a commencé à évoluer. Les navigateurs mobiles récents traitent bien la balise <video>, le plus souvent avec des fichiers de type MP4 qui utilisent le profil principal du codec H.264. C'est notamment le cas pour l'iPhone et l'iPad, pour les téléphones Android (à partir de la version 2.0), du Palm Pre et de quelques Nokia haut de gamme.

Certains navigateurs légers intègrent également la gestion des vidéos Flash, mais ils sont encore peu connus du grand public.

Extraits de « **Relever le défi du Web mobile Bonnes pratiques de conception et développement 1re édition** »

Editions Eyrolles

François Daoust et Dominique Hazaël-Massieux
29,90 € - 300 pages



L'information permanente

- L'actu de Programmez.com : le fil d'info quotidien
- La newsletter hebdo : la synthèse des informations indispensables.
Abonnez-vous, c'est gratuit !

www.programmez.com

S'adapter aux interactions utilisateur



“ Optimiser le contenu pour l'utilisateur mobile est nécessaire, mais pas suffisant pour s'assurer que celui-ci se sente à même de réaliser ses objectifs. Il est en effet capital de prendre aussi en compte les interactions qu'il peut avoir avec la page web. Sur des terminaux où les moyens usuels (souris et clavier) sont inexistant ou difficiles d'utilisation, les techniques web classiques, appliquées sans prendre garde, peuvent en effet ruiner tous vos efforts. Tout comme le contenu, les interactions doivent être claires et précises, à commencer par les liens.

Liens : cliquer n'est pas jouer

L'interaction avec l'interface graphique d'un navigateur mobile se fait principalement de trois manières :

- avec un curseur séquentiel ne permettant que de se déplacer d'élément en élément ;
- en déplaçant un curseur à l'écran via le clavier ou un joystick ;
- via un écran tactile.

Ces trois méthodes sont très différentes. Dans le premier cas, si votre contenu est correctement agencé de manière linéaire, le déplacement de lien en lien est aisément et vous n'avez pas besoin de prêter une attention particulière aux liens de vos pages... sous réserve de ne pas en abuser dans une même page et de réussir à les ordonner du plus utile au moins utile. L'utilisateur doit en effet survoler tous les liens avant d'atteindre le dernier. Une vingtaine de liens par page constitue donc un grand maximum.

Dans le deuxième cas, le déplacement du curseur est rapidement fastidieux, surtout s'il s'agit de viser un lien contenu dans du texte et qui n'apparaît pas nécessairement à gauche de l'écran. Les liens insérés dans un bloc de texte sont donc à éviter et la page doit être d'une taille raisonnable.

Dans le dernier cas – qui est certainement le plus en vue à l'heure actuelle – la précision de sélection des liens est, limitée par la taille du doigt de l'utilisateur. En guise d'ordre de grandeur, la surface d'interaction d'un doigt représente environ 40 pixels de hauteur sur un téléphone de résolution moyenne. L'utilisation de liens dans un bloc de texte peut alors se faire en s'assurant que ces liens sont suffisamment espacés dans la page. Quant aux menus de navigation, ils doivent bien souvent être « aérés » en ajoutant des marges (via CSS, avec margin, padding, line-height ou height,).

Du bon usage des formulaires

Les liens constituent une brique élémentaire des interactions possibles sur le Web. Les formulaires proposent une brique supplémentaire, souvent nécessaire pour des interactions plus riches : recherche, saisie de critères, sélection de choix, identification, etc. Les formulaires sont bien sûr une composante importante, dont il convient néanmoins de prendre en compte les spécificités d'utilisation dans le contexte mobile.

Limiter les entrées manuelles

En l'absence de clavier, ou d'un clavier suffisamment efficace, la saisie de données est toujours difficile sur un téléphone mobile. Un bon site web mobile limite au maximum le nombre d'entrées manuelles que l'utilisateur doit saisir. En particulier, tâchez de remplacer toutes les zones de saisie qui peuvent l'être par une liste de sélection. De même, sélectionnez par défaut les options les plus probables pour gagner du temps.

Pré-sélection de l'option la plus probable

```
<p>Souhaitez-vous recevoir Spam-Spam magazine ?</p>
<select>
<option value="oui">Oui oui !</option>
<option value="non" selected="selected">Non</option>
</select>
```

Enfin, la saisie de caractères spéciaux est particulièrement difficile sur un terminal mobile : elle requiert souvent d'activer un mode spécifique de fonctionnement du clavier, que nombre d'utilisateurs auront du mal à découvrir ou à employer.

En général, les caractères / et @ sont relativement plus faciles d'accès. L'apostrophe ou les caractères accentués le sont moins. Le mélange de différents types de caractères (un peu de chiffres, de symboles et d'alphanumériques en minuscules et en majuscules) est souvent un exercice acrobatique. Évitez autant que possible de demander à l'utilisateur de taper des données nécessitant ce genre de mélanges, y compris pour les mots de passe.

Guider la saisie

Dans le même ordre d'idées, il est essentiel de guider l'utilisateur dans le remplissage du formulaire. L'envoi de données invalides nécessite un aller-retour coûteux avec votre serveur web et engendre une grande frustration chez l'utilisateur qui doit rectifier la saisie après de longues secondes d'attente.

JavaScript peut être utilisé pour aider à la saisie contextuelle, par exemple dans une zone de recherche. Prenez garde, cependant, à ce que la lenteur d'exécution de Java-Script sur les terminaux mobiles ne déclenche pas l'effet inverse de celui recherché en ralentissant encore plus la saisie de l'utilisateur !

La validation des données d'un formulaire avant envoi est en revanche un excellent usage de JavaScript et permet d'améliorer sensiblement l'expérience utilisateur, en supprimant l'aller-retour

inutile lorsque certaines données sont invalides.

En pratique, il s'agit surtout de spécifier les champs obligatoires, typiquement par un astérisque, et d'expliciter le format attendu : un nombre entre 1 et 42, jj/mm/aaaa, etc.

HTML 5 introduit par ailleurs une flopée de nouveaux types de champs pour spécifier ce à quoi la zone de saisie est supposée servir (email, tel, date ou number, pour n'en citer que quelques-uns), ainsi qu'un nouveau jeu d'attributs pour la balise `<input>`. Peu de navigateurs mobiles supportent ces nouveautés à l'heure actuelle. Vous pouvez néanmoins les utiliser dès aujourd'hui sans aucun risque. Les navigateurs mobiles qui ne les comprennent pas se contenteront en effet d'afficher une zone de texte. À l'inverse, ceux qui les supportent sauront exploiter ces valeurs pour guider la saisie de l'utilisateur, en basculant en mode clavier numérique lors de la saisie d'un nombre, en proposant une liste de choix possibles pour la saisie d'un e-mail, ou encore en affichant un calendrier adapté au terminal lorsque l'utilisateur doit choisir une date.

Quelques exemples de champs en HTML 5

```
<p>Votre email :<br/>
<input type="email" /></p>
<p>Votre téléphone :<br/>
<input type="tel" /></p>
<p>Date souhaitée :<br/>
<input type="date" /></p>
<p>Proposez un mot de 8 lettres :<br/>
<input id="motus" pattern="[A-Z]{8}" />
</p>
```

À chaque champ son étiquette

La sélection des champs dans un formulaire est identique à la sélection de liens vue précédemment. En particulier, sur les écrans tactiles, les champs doivent être suffisamment aérés pour que l'utilisateur puisse les manipuler sans encombre. En outre, un bon site web mobile associe une étiquette à proximité de chaque champ. En plus de fournir un libellé logique pour l'utilisateur, cette étiquette est aussi utilisée par le navigateur pour étendre la zone de sélection du champ en question.

L'attribut `inputmode`, défini dans XHTML Basic 1.1, permet en théorie de spécifier le format attendu dans un champ de saisie, de manière à ce que le navigateur mobile puisse basculer le mode de saisie en conséquence lorsque l'utilisateur se prépare à renseigner ce champ. Il n'est malheureusement presque pas implémenté dans les navigateurs mobiles, et donc peu utile en pratique.

Associer des étiquettes aux champs de saisie

```
<form action=' '>
<p>
<label for='n'>Nom de famille :</label>
<input name='n' id='n' /> (<abbr title='requis'>*</abbr>)
</p>
<p>
<label for='p'>Prénom :</label>
<input name='p' id='p' />
(<abbr title='requis'>*</abbr>)
```

```
</p>
<p>
<input type="submit" value="S'inscrire" id='submit' />
</p>
</form>
<p>Les champs marqués d'une astérisque (*) sont requis.</p>
```

Compact et à suivre

Compte tenu de la faible place disponible à l'écran, le bouton d'envoi du formulaire doit être aisément accessible, directement après la saisie du dernier champ. Certains formulaires de pages web destinées aux ordinateurs de bureau prennent en effet parfois des libertés dans le positionnement de ce bouton (avant, après, à gauche, à droite). Préférez un agencement compact sur un site web mobile.

De même, un utilisateur en situation de mobilité s'essouffle vite. Si le formulaire à remplir est long, nous vous conseillons de le découper en plusieurs formulaires plus simples et d'enregistrer les données de l'utilisateur au fur et à mesure côté serveur. Il s'agit aussi d'éviter à l'utilisateur une situation où il doit tout recommencer de zéro, par exemple si la connexion est interrompue momentanément au moment de l'envoi.



B.A.-BA : quel format d'image pour quel usage ?

Le format JPEG (Joint Photographic Expert Group) est optimisé pour la compression des photographies. GIF (Graphic Interchange Format) et PNG (Portable Network Graphics) sont tous les deux des formats destinés aux images synthétiques, comme les diagrammes, logos et icônes. PNG est en général la meilleure solution technique, puisqu'il permet de fournir des images nettement plus riches :

- GIF limite à 256 couleurs la palette utilisable dans une image, tandis que PNG autorise un nombre pratiquement illimité de couleurs – en tout cas, un nombre supérieur à ce que l'œil humain peut percevoir et que la plupart des écrans peuvent afficher.
- GIF permet de créer des images à fond transparent, mais de manière très basique uniquement : un point de l'image est soit de couleur, soit transparent. À l'inverse, PNG permet d'indiquer des niveaux de translucidité, c'est-à-dire qu'un point peut avoir à la fois une couleur et un niveau de transparence.
- À qualité égale, PNG fournit en général un meilleur niveau de compression que GIF. Le seul avantage du format GIF est qu'il permet de créer des images animées bien reconnues sur la plupart des navigateurs d'ordinateurs.

Extraits de

Relever le défi du Web mobile
Bonnes pratiques de conception et développement 1^{re} édition

Editions Eyrolles

François Daoust et Dominique Hazaël-Massieux
29,90 € - 300 pages

Adapter, développer, optimiser : les modèles de développement

Finalement est-ce si compliqué d'adapter du contenu, un site, pour un téléphone portable de type smartphone ? En soi, concevoir un site web pour mobile ne pose guère plus de problèmes qu'un site « normal », juste avoir quelques réflexes et être conscient des limitations techniques.

Il faut faire une différence entre l'optimisation du contenu d'un site web sur smartphone et le développement d'une application web ressemblant à une application native comme on peut en avoir sur iOS. En général, le développeur pourra utiliser les technologies et langages de base : HTML 4, xHTML 1.0, CSS 2.x et 3, Javascript, Dom niveau 2, Ajax. Eviter d'utiliser des technologies web non standard ou non disponibles partout (selon les terminaux ciblés). On

constate d'un simple coup d'œil que l'affichage d'un contenu sur desktop, tablette ou smartphone varie : le navigateur n'est pas le même, l'interface n'est pas la même et surtout la taille de l'écran modifie cette visualisation. Au-delà de l'aspect programmation, adaptation du code, il faut optimiser l'affichage du contenu, c'est vital.

A retenir :

- l'écran des smartphones est plus petit
- se concentrer sur le contenu essentiel pouvant être visible et utile pour



Adapter le contenu au terminal

Le Web repose sur une architecture client-serveur : le client, c'est-à-dire, dans le cas qui nous occupe, un navigateur web, envoie une requête (contenant un certain nombre de paramètres) au serveur, et celui-ci répond avec le contenu correspondant à ladite requête au mieux de ses capacités. Pour faire en sorte que le contenu envoyé par le serveur corresponde aux contraintes et capacités du client, deux possibilités naturelles se dessinent pour le développeur web : sur le client ou sur le serveur.

Derrière cette apparente simplicité se cachent plusieurs variations importantes :

- adaptation côté client uniquement ;
- adaptation côté serveur uniquement ;
- adaptation côté client et côté serveur simultanément ;
- adaptation sur une simple différenciation téléphone mobile/ordinateur ;
- adaptation par grands groupes de téléphones mobiles (en plus des ordinateurs) ;
- adaptation individualisée pour chacun des terminaux mobiles et de leurs navigateurs.

L'idéal, pour tout développeur web, serait bien sûr de développer une seule version

de son contenu, qui s'adapterait automatiquement à tous les navigateurs et tous les terminaux existants – une adaptation côté client uniquement.

Le W3C essaye autant que possible de développer des technologies qui peuvent s'adapter à une large gamme de terminaux et tente par ailleurs de construire des technologies de manière progressive, de sorte qu'une prise en charge partielle d'une technologie donnée permette tout de même d'interpréter une grande partie du contenu considéré. Toutefois, cette approche du développement d'un contenu unique, qui peut être déployé sur une infinité de terminaux, connaît plusieurs limites importantes.

- Les erreurs des navigateurs, les erreurs de conception de certaines technologies ou, tout simplement, des difficultés techniques réduisent fortement les possibilités concrètes de cette adaptation infinie.
- Plus fondamentalement, les attentes et besoins des utilisateurs varient fortement d'un type de terminal à un autre et pour satisfaire ceux-ci, il conviendra souvent de mettre en place une expérience utilisateur largement différente en fonction du terminal utilisé.

Par exemple, l'adaptation des images d'une page web à la taille de l'écran du téléphone a tout intérêt à être effectuée côté serveur afin d'éviter d'envoyer des images trop grosses au client. Envoyer trop de contenu à un téléphone mobile est très coûteux en temps de transmission et, dans le cas des images, leur redimensionnement par le navigateur est par ailleurs coûteux en termes d'utilisation du processeur.

Les coûts de développement, de déploiement et de maintenance de ces différentes approches varient grandement, de même que la qualité des résultats qu'on peut en espérer. De fait, il n'existe pas une solution unique applicable à tous les cas : en fonction du budget disponible, du taux de fréquentation attendu, des besoins spécifiques des utilisateurs en situation de mobilité, de la fréquence et des outils de modification du contenu mis à disposition, du type d'interactions (information, achat, inscription...) mises en place, la balance pèsera d'un côté ou de l'autre de la gamme des adaptations de contenu.

Extraits de « *Relever le défi du Web mobile. Bonnes pratiques de conception et développement 1re édition. Eyrolles.* »

l'utilisateur (par exemple l'affichage d'un forum n'est pas la meilleure idée à avoir ni retranscrire une home page telle quelle, l'intérêt étant quasi nul).

- Alléger le contenu, l'interface pour éviter des temps de chargement trop long entre les pages
- Eviter les contenus trop lourds (audio, vidéo) sauf à les optimiser.

Quels contenus sur son mobile ?

C'est une question délicate et que nous ne pouvons pas résoudre pour vous. Chaque site nécessite une expertise de contenu. Qui sont vos lecteurs ? Quel est votre contenu primaire ? Ai-je besoin de tout le contenu ? Quelles sont les interactions de l'utilisateur avec mon contenu ? Quelles technologies sont utilisées ? Etc. Pour exemple si vous êtes plutôt un site d'actualités, faut-il passer des flux de type RSS sur une simple page HTML ? Idem pour un blog. Faut-il afficher tous les post ou simplement prendre un flux ? D'autre part, certaines fonctions peuvent ne pas passer, comme la saisie d'un post sur mobile, pas très pratique. Si vous diffusez massivement des images et des vidéos, cela va poser la question de l'optimisation des fichiers, de l'uniformisation des formats. Les contraintes d'affichage et de performances réseaux peuvent aussi vous obliger à des choix. Un site mobile n'a pas à ressembler à un site web consultable sur un desktop.

Ayez les bons réflexes

Pour faciliter le travail, vous pouvez utiliser les CSS3 qui permettent de bien contrôler les tailles selon les terminaux ce qui facilite l'adaptation des pages, et du contenu selon le type de terminal. Il ne faudra pas oublier d'indiquer les versions afin que le code détecte convenablement le navigateur. Les réflexes de base, notamment préconisés par Apple :

- utiliser une déclaration Doctype dans les pages HTML
- séparer strictement HTML, CSS et JavaScript
- structurer votre HTML et son arborescence

CSS : adapter les styles au terminal

“ L'un des avantages fréquemment cités de l'utilisation des feuilles de styles CSS réside dans la séparation du contenu (écrit en HTML) et de sa présentation (décrise en CSS). Pour les navigateurs mobiles, cette séparation constitue la base de l'utilisation de CSS pour l'adaptation côté client : il va s'agir en effet de créer des présentations différentes en fonction du navigateur et du terminal. CSS est de ce point de vue un outil très puissant, puisqu'il permet facilement d'envisager les modifications suivantes, par exemple :

- adaptation des espacements et des marges ;
 - passage d'un affichage en plusieurs colonnes à un affichage purement linéaire ;
 - suppression de contenu inutile ou superflu ;
 - remplacement d'images par des images plus légères ;
 - changement d'ordre de présentation des éléments d'une page.
- La clef de ces techniques repose sur la possibilité d'appliquer un jeu de règles de styles à un navigateur ou à un terminal donné ; il existe pour cela deux principales approches :
- l'utilisation des types de média CSS et des requêtes de média CSS qui ciblent des navigateurs en fonction de certaines propriétés du terminal ;
 - l'utilisation des CSS Hacks, c'est-à-dire de la connaissance d'erreurs ou de limitations de certains navigateurs pour leur faire ignorer tel ou tel lot de règles de styles.

Prenez garde, cependant, dans notre optique d'amélioration progressive, de faire en sorte que vos pages restent utilisables lorsque ces feuilles de styles sont désactivées, plus particulièrement lorsque seules certaines feuilles le sont.

JavaScript

Nous nous contenterons ici de rappeler quelques règles concernant JavaScript afin de s'assurer que son emploi se fasse dans une optique d'amélioration progressive et que, entre autres, celui-ci n'empêche pas les

utilisateurs sans JavaScript d'interagir avec un site web.

- N'utilisez pas de liens javascript : car ces « liens », qui exécutent du code lorsque l'utilisateur les active (pour ouvrir une fenêtre pop-up, effectuer une validation avant de passer à une autre page, etc.), sont tout simplement inopérants lorsque JavaScript n'est pas disponible. Ils sont par ailleurs toujours remplacables par l'utilisation des événements du DOM (type onclick).

- De même, l'envoi d'un formulaire ne doit pas reposer sur l'exécution d'un code JavaScript.

- Idéalement, les mises à jour partielles de page rendues possibles par l'utilisation de XMLHttpRequest sont disponibles aussi par d'autres biais (envoi de formulaires, navigation) gérés par le serveur.

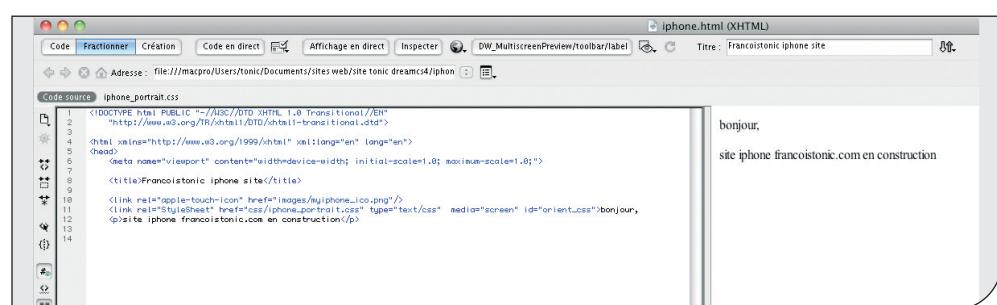
Par ailleurs, plutôt que de se reposer sur des conditions basées sur l'analyse de la chaîne identifiant le navigateur (disponible depuis window.navigator.userAgent), il est en général préférable de tester l'existence des interfaces ou objets visés pour décider de l'exécution d'un code donné. Prenons l'exemple de l'utilisation de l'API de géolocalisation :

Déterminer la disponibilité de l'API de géolocalisation

```
if (navigator.geolocation && navigator.geolocation.getCurrentPosition) {
  navigator.geolocation.getCurrentPosition(affichePlan);
} function affichePlan(position) {...}
```

Au lieu d'analyser le nom du navigateur de manière arbitraire et fragile, il est nettement préférable de simplement vérifier que l'objet implémentant l'interface de géolocalisation (navigator.geolocation) est disponible.

Extraits de « Relever le défi du Web mobile Bonnes pratiques de conception et développement 1re édition », Eyrolles

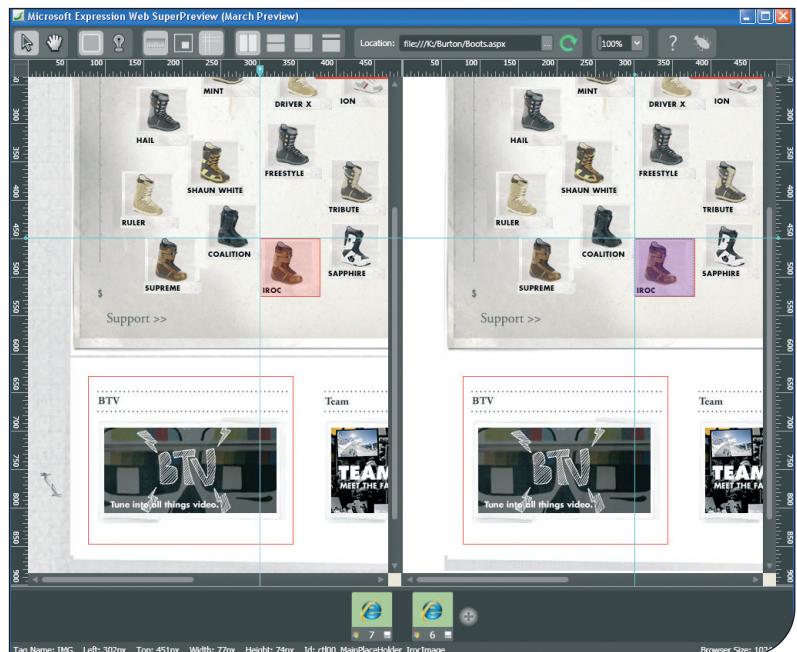


- être indépendant du navigateur mobile
- les frames entre un desktop et un mobile ne s'organisent pas de la même manière, penser à les réorganiser.

Sur les frames, il est fortement conseillé de ne pas les utiliser, à cause des problèmes de place et le fait qu'il faut bannir au maximum les ascenseurs. Comme sur iOS, préférez l'usage des colonnes et des blocs. Le contenu sera mieux présenté et plus lisible. Plus une page mobile sera lisible, simple, plus l'utilisateur sera à l'aise. Pour éviter de proposer une ergonomie disparate, créez des feuilles CSS types qui pourront être reprises d'une page à l'autre. Soyez homogène dans votre ergonomie, votre présentation.

D'autre part n'oubliez pas que les smartphones utilisent une navigation, une interface tactile donc avec les doigts et qu'il faut adapter la taille des boutons en conséquence. D'autre part, les menus, les boutons, les listes déroulantes, etc. n'auront pas forcément la même apparence sur une page mobile. Le designer doit en tenir compte.

N'utilisez pas des technologies, des



La fonction superpreview de MS Expression permet de tester rapidement ses pages sur différents navigateurs.

techniques qui ne sont pas disponibles sur le smartphone. Par exemple, sur iOS, vous n'aurez pas d'événement souris, ni de tooltips, d'appels Java, de Flash, d'installation de plug-ins, d'envoi de fichier depuis une page Safari. Vérifiez bien avant tout développement, les possibilités techniques ! Enfin, astuce toute bête : tenez compte de la présence et de la taille des barres et boutons par défaut du système mobile... Par exemple, sur iOS, la taille de l'écran, en hauteur, est de 480 pixels mais la zone réellement visible est 356... Cela a une influence directe sur la taille des images par exemple. D'autre part, cette taille de zone visible varie si on est en affichage portrait ou paysage !

Donc adapter les images, les vidéos... et au-delà

Vous l'aurez compris, vous devrez retailler les images, les adapter à un

format smartphone. Il faut éviter que l'utilisateur soit obligé de zoomer et dézoomer trop fréquemment. Privilégiez aussi des formats compressés plus légers comme gif ou png.

Mais n'oubliez pas non plus la navigation. Il est important de la simplifier entre les pages, il faut simplifier, simplifier et simplifier. Evitez de proposer trop de liens. Surtout, le principe de navigation doit être le même partout. Ne déroutez pas l'utilisateur.

Exemple de guide de développement web :

http://developer.apple.com/library/safari/#documentation/AppleApplications/Reference/SafariWebContent/Introduction/Introduction.html%23apple_ref/doc/uid/TP40002079-SW1

Surtout, n'oubliez jamais : tester ! Il est impératif de tester votre site mobile sur plusieurs terminaux (des vrais, pas uniquement en émulateur).

■ François Tonic

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

INDICATEUR MENSUEL APEC :

Les offres d'emploi informatique décollent en janvier : +70%

En janvier 2011, plus de 40 700 offres d'emploi ont été confiées à l'Apec, en progression de 72 % par rapport à janvier 2010. Sur les 12 derniers mois, la croissance des offres d'emploi cadre est de 59 % !

	Janvier 2011 Nb offres	D 11/10	en cumul sur 12 mois glissants Nb offres	C 11/10	Structures (en %)
Direction informatique	213		2 510	+ 146%	
Exploitation, maintenance informatique	459		5 601	+ 43%	
Informatique de gestion	4 201		42 515	+ 63%	
Informatique industrielle	1 345		12 821	+ 38%	
Informatique web, sites et portails internet	1 255		11 832	+ 328%	
Maîtrise d'ouvrage et fonctionnel	1 320		10 102	+ 40%	
Système, réseaux, données	1 996		19 948	+ 73%	
INFORMATIQUE	10 789	+ 67%	105 329	+ 70%	29

En 2011, près de 4 500 postes, dont 35% pour jeunes diplômés, seront à pourvoir en France chez Cap Gemini (avec Sogeti).

Cap Gemini : 400 postes à pourvoir pour Toulouse !

Capgemini est la première SSII nationale et figure parmi le top 10 mondial du secteur. Le Groupe compte, en France, plus de 20 000 collaborateurs, dont 7 700 sur Application Services, la branche Intégration et développement d'applications du groupe. Toulouse, qui abrite le siège de la division Aérospatiale & Défense, regroupe les activités aéronautiques, spatiales et de défense de Capgemini en France. Fin 2010, la division a conclu d'importants contrats. Aussi, Capgemini lance une vaste campagne de recrutement pour renforcer les équipes toulousaines et à plus large échelle, de la division Aérospatiale & Défense. 400 nouveaux collaborateurs viendront rejoindre les effectifs, lors du premier trimestre de 2011 (voir encadré)

Les profils recherchés

- 100 jeunes diplômés (0-3 ans d'expérience)
- 150 profils intermédiaires (3-7 ans)
- 150 profils expérimentés (au-delà de 7 ans)

Ils interviendront auprès de clients nationaux et internationaux de l'aéronautique, du spatial, de la défense et de la navigation aérienne mais également du 'travel & transport', de l'industrie, de la santé ou du social, précise la SSII.

Le programme de recrutement lancé à Toulouse s'inscrit dans le cadre d'une campagne de recrutement d'envergure nationale baptisée « Mon Job, Ma Vie ». A travers cette campagne de recrutement nationale, Application Services France envisage, en 2011, de recruter 1500 nouveaux collaborateurs pour planifier, concevoir, diriger et développer des projets d'intégration de systèmes d'information.

LA SEMAINE DU RECRUTEMENT

Afin de favoriser le recrutement de ses 400 nouveaux collaborateurs, Capgemini organise une Semaine spéciale, du 21 au 25 mars, dans les locaux toulousains du Groupe, siège national de la division Aérospatiale

Réseau : +73%

La plus forte hausse concerne la fonction Commercial/Marketing (+ 77 %). L'Informatique, qui représente 29 % des offres affiche une augmentation spectaculaire de son volume d'offres : + 70 %.

L'Apec note que la demande dans le domaine des Systèmes, réseaux, données demeure très élevée (+ 73 %), tandis que des fonctions telles que l'Informatique industrielle et la Maîtrise d'ouvrage et fonctionnel enregistrent une croissance plus mesurée (respectivement + 38 % et + 40 %).

& Défense. Les locaux seront spécialement organisés en mode portes ouvertes à l'heure du déjeuner puis en soirée. Chacun pourra suivre le parcours de son choix et découvrir plus en détails un projet, un métier ou un secteur d'activité.

Une vaste campagne de publicité

Pour atteindre son objectif Cap Gemini déploie une vaste campagne de communication locale et nationale :

affichage sur écrans géants lors du match Toulouse/Biarritz Olympique le 19 février, distribution de flyers aux 19 000 spectateurs, distribution de 50 000 flyers et affichage de 1 000 posters dans les écoles et en ville,

lancement d'un minisite événementiel - www.monjob-mavie.com/recrutementtoulouse etc.

■ Jean Kaminsky

Envoi de CV : un iPad à gagner chaque jour

- Une adresse e-mail dédiée à l'événement, accueille les CV : semainedurecrutement.fr@capgemini.com
- Site internet : www.monjob-mavie.com/recrutementtoulouse

Chaque jour, l'un des CV sera tiré au sort : le candidat sélectionné remportera un iPad.

Cloud Computing

Le guide de survie du développeur

Depuis plus de deux ans, le cloud computing a le vent en poupe. Curiosité, puis objet de test et maintenant, plateforme de production, il fait peu à peu ses preuves auprès des développeurs, des utilisateurs et des entreprises même si le nuage représente encore une faible proportion des développements. Par contre, les logiciels sous forme de service (SaaS) prennent une place non négligeable et ne cesseront de prendre de l'importance, même si le logiciel « traditionnel » a encore un bel avenir devant lui.

Pourquoi le cloud ? Pourquoi maintenant ? Formez-vous, apprenez, testez. Plus vous comprendrez les modèles du cloud, les API, les technologies, mieux vous pourrez adapter le code existant, comprendre les architectures à mettre en place. Le cloud ne s'improvise pas. Il nécessite rigueur, optimisation et surtout des choix qui seront difficiles à modifier lorsque l'application sera déployée.

Le cloud exige une grande rigueur technique car si l'information à la demande offre des avantages indéniables, le développeur aura vite tendance à avoir les réflexes de codage desktop. Au contraire, plus le code sera rigoureux, optimisé, mieux votre application fonctionnera en logique cloud tout en minimisant l'impact budgétaire. Car n'oubliez pas que sur le cloud, toute consommation se paie.

Dans notre dossier du mois, nous allons nous pencher sur plusieurs modèles de développement dans le cloud, faire un point complet sur Windows Azure et un focus sur les architectures à adopter.

■ François Tonic

Qu'est-ce qu'une application «cloudisée»? Quel cloud choisir?

Voilà, vous allez migrer, déployer des applications, des données, sur votre cloud tout neuf. Mais avant d'envisager cette ultime étape du cycle de développement, il y a une phase cruciale à ne pas rater : l'architecture de son application... Et là, les réponses vont vite se compliquer si l'on n'y prend pas garde...

Il y a différentes manières d'aborder une application cloud : vous avez un code existant, vous partez de zéro. Puis, il y a deux types d'applications : l'application desktop à cloudiser en SaaS par exemple ou le site web à déployer en production sur un cloud. La situation pourrait se résumer ainsi :

	Code existant	« From scratch » (de 0)
Application desktop / application web	Installation des SDK Cloud	Utilisation des SDK
	Modification du code	Codage orienté cloud
	Migration du code	Modèle des données cloud, mapping
	Mapping de données	Déploiement sur le cloud
	Gestion du multi-tenant	
	Intégration des règles cloud	
Site web	Installation des SDK Cloud	Choix des langages et technologies
	Modification du code	Utilisation des SDK
	Migration du code et technologies non supportées	Mapping et données cloud
	Mapping de données	Déploiement sur le cloud
	Gestion du multi-tenant	
	Intégration des règles cloud	

Bref, que l'on soit en application ou en site web, les critères et obligations changent assez peu. Car bien entendu, les changements peuvent être très importants pour un code existant mal structuré ou très dépendant du système. En effet, le cloud vous oblige à restructurer l'architecture et le code afin de minimiser les transactions, les besoins en bande passante, les besoins en stockage, etc. Car, une constante doit tout le temps être présente : pas d'optimisation du code, du fonctionnement = coût élevé du cloud, optimisation du code et du fonctionnement = réduction de la facture

4 raisons pour passer au cloud

Marche / Arrêt : j'active une instance cloud pour un travail précis, une application, une donnée, ponctuellement.

Trafic en croissance rapide : mon site, mon application connaît une montée en charge de son utilisation, en croissance forte. Comment absorber ? Comment assurer la disponibilité ?

Pour des pics de charge non prévus, non prédictibles : actualité chaude, promotion sur un produit qui marche mieux que prévu.

Pour des pics de charge prédictibles : par exemple chaque mois pour la paie, ou pour le lancement du nouveau numéro d'un magazine.

cloud. Car en cloud, toute ressource a un coût (également en latence et en traitement).

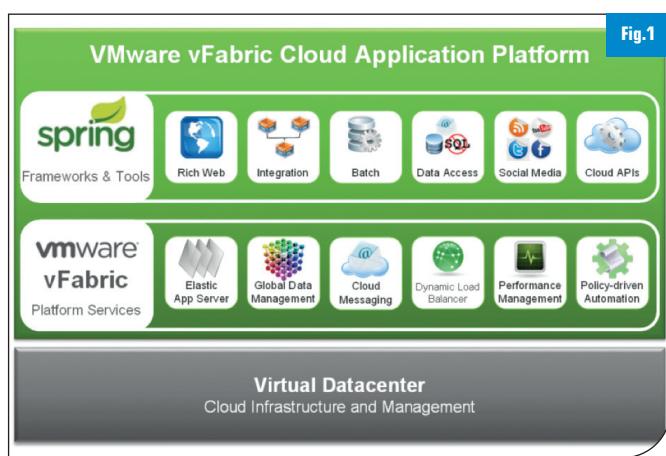
Par exemple, vous avez deux bases identiques l'une sur le serveur interne, l'autre dans le cloud. Vous utilisez un mécanisme de réplication de la base entière. Imaginez que cette base fasse 10 Go et qu'elle doit être répliquée plusieurs fois par semaine. Si votre bande passante entrante et sortante est limitée en Go mensuels, la facture va rapidement exploser. Par contre, le mieux est d'utiliser des fonctions de synchronisation permettant de répliquer uniquement les données nouvelles ou de les modifier, et non la base entière...

PaaS ou IaaS, il faut choisir !

Avant de commencer le développement, il faut trancher dans le vif : plateforme ou infrastructure, bref PaaS ou IaaS ? Pour le développeur et les entreprises voulant passer à des logiciels en mode SaaS / Cloud en général, la plateforme (PaaS) sera le meilleur choix. Nous rappellerons que le PaaS est la couche d'exécution, de développement, de déploiement. Elle masque la complexité de l'infrastructure (IaaS) et permet de faire du SaaS. Sur ce domaine du PaaS, il y a plusieurs offres, par exemple : Google App Engine, Force.com de Salesforce et Microsoft Windows Azure. Mais il en existe d'autres, moins connues : Engineyard, Heroku (racheté par Salesforce.com). [Fig.1]

Ce choix doit être conditionné, au moins partiellement, par les choix techniques :

- qu'est-ce que je veux développer
- quelle est ma plate-forme de développement
- quels langages je veux ou dois utiliser



Vision globale de l'approche vFabric de VMware

- quelles technologies je veux ou dois implémenter dans mon projet
- quels modèles de données (et types)
- quel est le coût financier du cloud, le budget alloué
- me faut-il une formation ?

PaaS	Langage supporté
Google App Engine	Python Java Java EE (SpringSource)
Microsoft Windows Azure	Java PHP Ruby Langages .Net ASP.Net
Force.com	Java (SpringSource) Apex Flex (Flash Builder for force.com)

Un langage est absent du tableau : C++. Oui il est possible, avec certains PaaS, de déployer du C++. On peut invoquer par exemple sur un web role sous Windows Azure, du code natif via un P/Invoke

Deployment scenarios

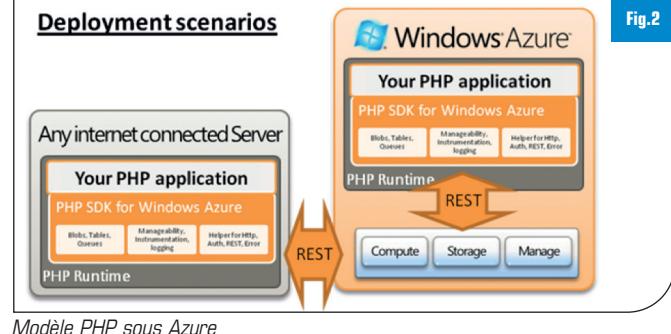


Fig.2

ke. Mais il est aussi possible de déployer du natif directement sous Azure sans passer par du P/Invoke. Dans ce cas, passez en compilation 64-bit et non 32. Windows Azure fonctionne avec des machines virtuelles 64 (Windows Server 2008). Attention, tout PaaS ne supporte pas ce type de déploiement. Méfiance !

Sur le sujet, nous vous invitons à lire : <http://msdn.microsoft.com/en-us/gg271392>

Attention au multi-tenant

L'approche architecturale dite « multi-tenant » est importante pour assurer à son application cloud une montée en charge, de la disponibilité et une utilisation par x utilisateurs. En architecture cloud computing, nous avons une double approche : simple tenant (tenant unique) ou multi-tenant. Basiquement, le multi-tenant permet de partager entre une multitude d'utilisateurs une application tout en faisant des économies d'échelles sur les ressources. Revers de la médaille : complexité de la structure application. Elle doit gérer plusieurs utilisateurs indépendamment les uns des autres. Bref, comment mutualiser une application. Pour un éditeur c'est important car il peut ainsi réduire les ressources nécessaires et donc sa propre facture car une instance applicative n'est pas gratuite.

Basiquement, une application SaaS, PaaS en approche multi-tenant « sécurisée » doit :

- fournir un niveau de sécurité et un contrôle d'accès pour chaque utilisateur selon son profil
- mettre en place des mécanismes d'authentification
- les données utilisateurs sont dans une base à part

Dans l'exemple suivant, dans une approche simple tenant, chaque utilisateur utilise une

instance indépendante des autres. Cela signifie que 10 utilisateurs = 10 instances de l'application. Par contre, dans une approche multi-tenant, 10 utilisateurs = 1 instance applicative. Dans ce cas, les ressources de l'instance sont partagées entre les utilisateurs. Cela implique que l'application est partagée. Il est impératif de séparer les vues, les données, l'identité, le contexte pour chaque utilisateur et de parfaitement isoler chaque « partage ». Si nous prenons l'approche proposée par Windows Azure, le fait que les applications soient (idéalement) formées de multiples composants, chaque composant peut être en mode simple ou multi-tenant. Ainsi, vous pouvez réaliser du multi-tenant sur les modules les plus sensibles à la montée en charge, comme les données. Cependant, outre la

complexité, il ne faut pas oublier que si l'instance Azure tombe, l'ensemble des utilisateurs sera impacté, le temps que le SLA agisse. C'est pour cela que l'on parlera d'architecture multi-tenant et multi-instances. Mais cela nécessite un load balancing entre les instances.

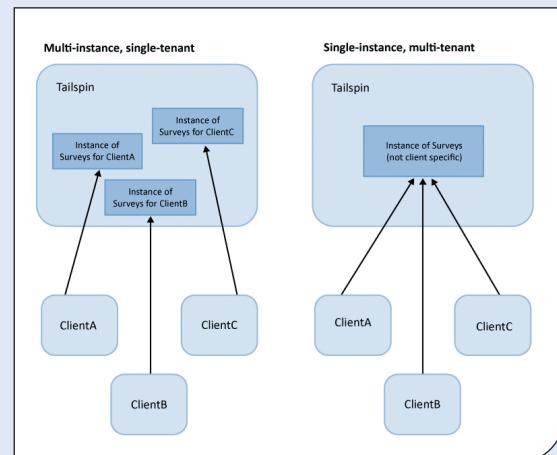
Google propose depuis l'été 2010 des mécanismes multi-tenant. Celui-ci se concentre sur les API des espaces de noms (Namespace API). Ainsi, il est possible d'avoir des données avec un unique namespace servant à chaque utilisateur, une même application pour x utilisateurs. Cette démarche est valable en Java et Python. Force.com.

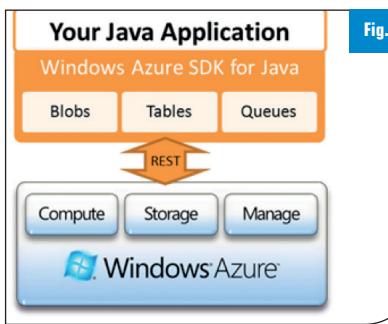
La base de données est un des points sensibles du multi-tenant. Plusieurs solutions sont possibles, soit une base par utilisateur, une base unique avec des segmentations précises pour chaque utilisateur. L'approche multi-base est celle qui est souvent retenue, cela permet une montée en charge plus efficace et surtout une plus grande sécurité et disponibilité des données. Ainsi le crash de la base de données concernera un utilisateur et non l'ensemble.

Pour aller plus loin :

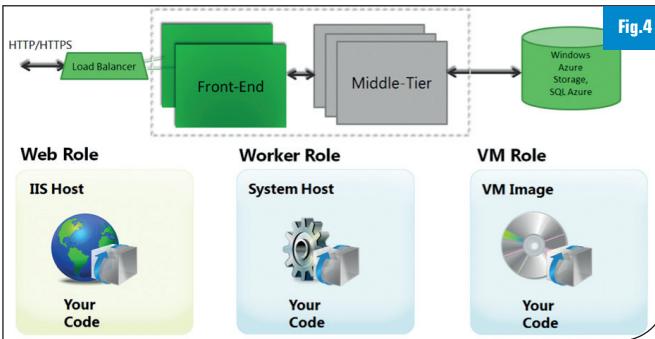
Multi-tenant dans Azure :
<http://msdn.microsoft.com/en-us/library/ff966480.aspx>

Multi-tenant dans App Engine :
<http://code.google.com/intl/fr-FR/appengine/docs/java/multitenancy/overview.html>

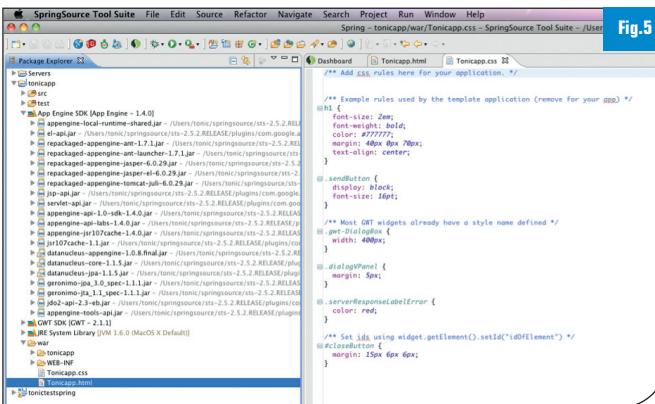




Modèle Java sous Azure



Modèle applicatif classique sous Azure (<http://blogs.msdn.com/b/thomasconte/archive/2010/12/14/windows-azure-vm-role-vue-d-ensemble-et-cas-d-utilisation.aspx>)



Arborescence d'un projet App Engine sous SpringSource Tool Suite

Une autre solution est de passer par un IaaS, de type Amazon EC2, mais en utilisant une solution de développement et de déploiement masquant au mieux qu'il ne s'agit pas d'un PaaS. Cela permet de garder, excepté quelques modifications, votre code existant, mais de le déployer sur du cloud. Par exemple, CloudFoundry (VMware) permet de déployer du Java EE sur une instance EC2.

A quoi ressemble une application sur le cloud ?

En réalité une application cloud n'est guère différente d'une application web, d'un site web. On déploie sur un site accessible depuis un navigateur connecté à Internet. Ce modèle change selon le PaaS [Fig.2,3 et 4]. Si vous prenez une application destinée à Windows Azure, il s'agit d'une application web (appelée web rôle) fonctionnant au-dessus des services IIS. Il existe aussi des services fonctionnant en arrière-plan

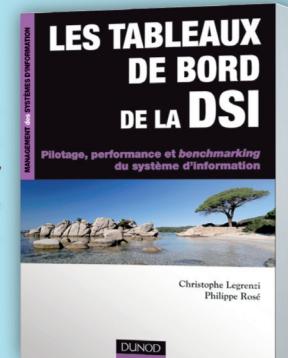
(ou worker rôle), idéaux pour des traitements, l'exécution de code, etc. Par exemple, un web rôle peut être en langage .Net, PHP, Java. Avec App Engine, il s'agit d'une application Java ou Python fonctionnant avec GWT. Guère plus. App Engine fournit le JRE nécessaire, les API. Bref, si vous êtes développeur Java, App Engine ne change guère les habitudes si ce n'est d'utiliser les SDK et de modifier quelque peu l'architecture et le modèle de données [Fig.5].

Une simple conclusion

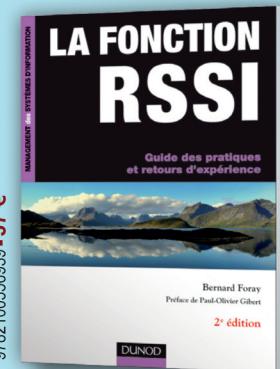
Dites-vous que pour l'utilisateur, une application cloud ne change rien. Ainsi si vous avez monsite.com et que celui-ci est sur App Engine, ce sera : monsite.appspot.com. C'est TOUT ! Oui, il y aura des changements internes, non, c'est transparent pour l'utilisateur.

■ François Tonic

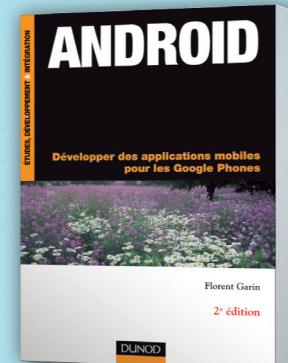
LES OUVRAGES UTILES POUR VOTRE MÉTIER



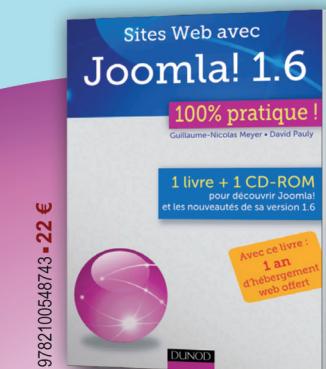
Toutes les réponses pour piloter et réduire les coûts des S.I.



Le guide pratique pour sécuriser les données de l'entreprise



Développer des applications mobiles et tactiles sous Android



Installer et administrer son site web avec Joomla !



Windows Azure : les nouveautés 2011

Vous vous intéressez à Windows Azure ou bien vous avez débuté un développement, et vous souhaitez connaître les nouvelles fonctionnalités : cet article met en évidence les éléments qui sont ou seront disponibles en version commerciale en 2011.

WINDOWS AZURE

Machine virtuelle personnalisée : VM Role

C'était une demande récurrente, Microsoft y a répondu. Les deux rôles Web et Worker sont désormais complétés d'un troisième : le **VM Role**. Il permet d'exécuter dans Windows Azure une machine virtuelle personnalisée et basée sur Windows Server 2008 R2. Pour y parvenir, l'utilisateur crée sa VM en entreprise avec Hyper-V, la personnalise en activant les composants systèmes choisis et en installant ses propres applications, puis enfin télécharge la VM créée dans Windows Azure pour exécution. Noter que la tarification reste identique aux deux autres rôles, et que l'utilisateur pourra bientôt créer ses machines virtuelles personnalisées directement dans Windows Azure d'une part, et en partant des systèmes supplémentaires : Windows Server 2003 et Windows Server 2008 SP2 d'autre part.

Accès complet à IIS 7.0 et 7.5

Le serveur web IIS des machines virtuelles peut maintenant être utilisé et configuré de manière intégrale. Ceci permet notamment de déployer plusieurs sites web et répertoires virtuels, d'installer et de configurer des modules IIS.

Tâches de démarrage et droits élevés

Cette fonctionnalité consiste à indiquer dans le fichier de définition des rôles Web et Worker une ou plusieurs tâches à exécuter au démarrage du rôle et le niveau des droits requis.

En profitant des nouveaux droits d'administration élevés, ceci permet de lancer des scripts d'installation de packages MSI, d'enregistrement de composants COM, ou encore des scripts personnalisés. Si ces tâches de démarrage sont automatisables et pas trop longues, il est préférable d'exploiter ce mécanisme au lieu de configurer entièrement un rôle VM.

Content Distribution Network (CDN)

Ce réseau de vingt-deux nœuds répartis dans le monde entier délivre du contenu mis en cache avec une grande bande passante, ce qui améliore sensiblement les performances des applications. Configuré depuis le portail, il permet de mettre en cache des blobs publics de Windows Azure Storage de tout type : images jpg, css, javascript et autres documents; et supportera bientôt les contenus générés dynamiquement et sécurisés via SSL/TLS.

Virtual Network

Windows Azure Connect est un service qui établit une connectivité sécurisée de niveau IP entre les instances s'exécutant dans Windows Azure et les machines en entreprise. Ainsi, tous les scénarios hybrides sont maintenant ouverts : accéder depuis un rôle Web dans Windows Azure à une base SQL Server en entreprise, à un serveur de fichiers, d'imprimantes, de messagerie, etc. Vous pouvez de plus joindre un rôle Windows Azure au domaine de l'entreprise pour exploiter les méthodes d'authentification existantes, comme Active Directory : les machines virtuelles sont alors vues comme faisant partie intégrante du système d'information de l'entreprise.

Java

Microsoft prévoit une amélioration de l'intégration de Java du point de vue des performances, des outils Eclipse et des librairies clientes. Le choix de l'environnement Java pourra se faire indépendamment de l'exécution dans Windows Azure.

PHP

Le Windows Azure Companion facilite le développement et l'exécution d'applications PHP dans Windows Azure. Il permet notamment d'installer les

composants suivants :

- Runtime PHP
- Windows Azure Tools for Eclipse pour PHP
- Windows Azure Command-line Tools pour PHP
- Windows Azure SDK pour PHP
- Windows Azure Storage plugin pour WordPress

Visual Studio .NET

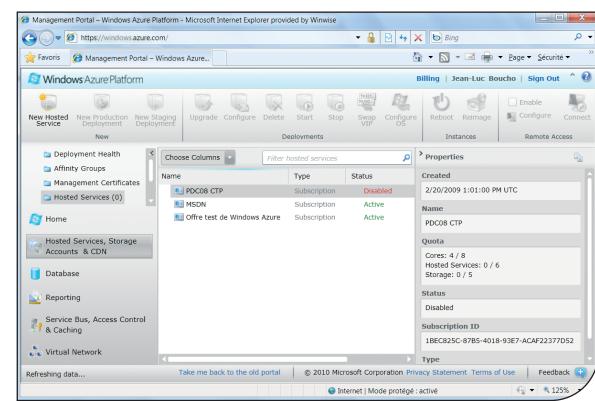
La dernière version 1.3 de Windows Azure Tools for Microsoft Visual Studio supporte les dernières nouveautés de la plateforme, à savoir :

- Configuration du rôle VM
- Activation de l'accès réseau virtuel (Windows Azure Connect) pour un rôle
- Activation de l'accès distant (Remote Desktop) pour un rôle
- Mise à jour de la solution sans faire de redéploiement complet dans Windows Azure (utilise l'accès distant pour déployer directement dans l'instance)
- Le tout en exploitant un nouveau système de plugins pour injecter ces fonctionnalités dans un rôle.

WINDOWS AZURE APPFABRIC

Caching

Ce service est un cache mémoire, distribué et redondant dont les fonctionnalités sont similaires à celles de **Windows Server AppFabric Caching**, mais proposé prêt à l'emploi dans Windows Azure. Il optimise



les performances des applications en réduisant les accès à la base de données, et ne requiert ni administration ni maintenance. Il présente une API permettant de conserver des objets serialisables, mais propose également des fournisseurs de stockage des états de session ASP.NET et de cache de sortie ASP.NET.

Service Bus

Le Service Bus intervient comme un relais de connectivité entre les services et applications dans Windows Azure et/ou en entreprise. Voici les nouveautés :

- Les messages mis en tampon sont désormais **persistants** avec une taille maximale de 256 Ko par message et une file d'attente plafonnée à 100 Mo pour l'ensemble des messages présents
- La communication **AnyCast** autorise plusieurs destinataires à écouter sur le même point de connexion : le Service Bus distribue alors de manière aléatoire chaque message à un seul destinataire, ce qui permet de bénéficier d'une solution simple de load-balancing avec failover
- La gestion est programmable grâce à une API communiquant au format Atom / REST.

Access Control

Ce service met en œuvre la fédération d'identité, et permet de bénéficier du SSO (Single Sign On) dans les applications et services web. Il est compatible avec les principaux fournisseurs publics d'identités : Windows Live ID, Google, Yahoo, Facebook ; mais aussi Active Directory en entreprise. Les protocoles supportés sont variés : OAuth, WRAP, WS-Trust, WS-Federation ; de même que pour les formats de jetons : SAML 1.1, SAML 2.0 et Simple Web Token (SWT).

SQL AZURE

Portail d'administration

Un nouveau portail, le **Database Manager** pour SQL Azure [Fig.3] est accessible depuis le portail général Windows Azure et permet une gestion des bases de données existantes en mode web / Silverlight. Aucun outil à télécharger n'est donc nécessaire. L'outil reprend les grandes fonctions d'un gestionnaire, à savoir la gestion des tables, des vues, des procédures stockées. Il possède de plus un éditeur Transact-SQL permettant d'exécuter des requêtes et consulter les résultats.

Fig.3

SQL Azure Reporting

A l'aide de **Business Intelligence Development Studio** (BIDS), vous avez la capacité de réaliser des rapports liés à des bases de données SQL Azure. Cela permet ensuite de les déployer dans SQL Azure Reporting, de les afficher et de les exporter sous les formats Excel, Word, HTML, PDF, XML, CSV ou ATOM. De plus, une intégration des rapports SQL Azure peut être aussi faite dans les applications web en entreprise ou les rôles Web dans Windows Azure.

SQL Azure Data Sync

Initialement capable de synchroniser plusieurs bases de données SQL Azure, SQL Azure Data Sync synchronise maintenant les bases SQL Azure avec celles de SQL Server en entreprise via l'ajout d'un agent de synchronisation (service Windows) local au serveur. Le portail permet de déclarer plusieurs groupes de bases à synchroniser, les tables à synchroniser, la nature de la synchronisation (mono ou bi-directionnelle) ainsi que la fréquence. Enfin, les opérations de synchronisation se supervisent depuis le portail.

PRÉVISIONS ANNONCÉES MAIS PAS ENCORE DISPONIBLES

Windows Azure AppFabric

Deux nouvelles fonctionnalités ont été annoncées :

- **Integration** : ce service fournira des capacités d'intégration dans Windows Azure des applications métiers et de suivre des processus (BAM) en proposant les patterns : pipeline, transformation et adaptateurs
- **Composite App** : ce service permettra de concevoir graphiquement une application composite complète basée sur les rôles de Windows Azure et les fonctionnalités d'AppFabric (Caching, Service Bus, Access Control) afin d'automatiser le

déploiement, de superviser les instances et de faciliter la montée en charge.

SQL Azure Federation

La fédération est une technique de partitionnement horizontal des données qui répartit les données dans plusieurs bases de données physiques dans l'objectif de paralléliser les requêtes. Néanmoins, les données sont considérées comme appartenant à une seule base de données logique, ce qui ajoute à SQL Azure une capacité native de montée en charge horizontale (scale-out).

Server Application Virtualization

Ce service permettra de déployer dans des rôles Worker des images d'applications virtualisées. Cette solution sera la plus rapide pour migrer des applications existantes sans passer par une configuration dédiée basée sur le rôle VM, tout en bénéficiant automatiquement des mises à jour du système d'exploitation.

CONCLUSION

Pour la plateforme Windows Azure, 2011 se présente donc comme une année riche en nouveautés. Certaines fonctionnalités sont encore en préversion (CTP ou bêta) et seront disponibles en version commerciale dans les mois à venir. Mais surtout, la plateforme est beaucoup plus complète qu'au moment de son lancement, et de nombreux scénarios d'applications d'entreprise sont maintenant accessibles. Tous les composants disponibles forment un ensemble cohérent qui couvre les problématiques classiques du Cloud Computing mais aussi celles de la connectivité et de l'intégration avec les applications existantes de l'entreprise. C'est donc le bon moment pour faire le pas et utiliser la plateforme Windows Azure pour profiter de ses caractéristiques de montée en charge, d'élasticité et de haute disponibilité à un prix compétitif !



■ Jean-Luc Boucho
Solutions Architect & Leader
Windows Azure à Winwise
Blog : <http://blog.jeanlucboucho.com>



■ Arnaud Dauba
Consultant à Winwise

PHP dans le cloud : les offres

Le cloud computing fait référence à une mise en commun des ressources matérielles (IaaS), système et plateforme (PaaS) ou logicielles (SaaS). Le cloud computing rend l'accès à ces ressources aussi simple que l'accès à l'électricité : on se branche sur une prise informatique et l'on dispose des ressources correspondantes à notre besoin. Les infrastructures qui sont à la base des offres de cloud computing reposent sur des plateformes techniques éprouvées comme Linux ou sur des plateformes de développement comme PHP.

Avant de rentrer dans les aspects techniques des offres PaaS PHP, arrêtons-nous quelques instants sur les couches de base du modèle. (voir [Fig.1] : Les différentes couches du Cloud Computing)

La solution IaaS ou Infrastructure as a Service forme le socle de base du cloud computing. Il s'agit de mettre à disposition des utilisateurs ou des couches supérieures une instance de l'OS virtualisée, en masquant tous les aspects physiques (machines, électricité, datacenter, réseau, hyperviseur). Cette couche permet de disposer d'une grande souplesse dans la mise à disposition des ressources, mémoire / puissance de calcul. Le IaaS est « relié » à l'utilisateur final par une connexion internet. La solution PaaS ou Plateforme as a Service est construite au-dessus de la couche IaaS. C'est une plateforme d'exécution d'application entièrement gérée par le fournisseur de service. Elle permet d'exécuter le code applicatif développé par le client de la plateforme. Le prestataire prend en charge les aspects de mise à jour de la plateforme, tant au niveau de l'OS que du framework applicatif. C'est au niveau de cette couche PaaS que le langage PHP prend toute son importance.

L'UTILISATION DU PAAS

Le PaaS, en tant qu'infrastructure logicielle, s'appuie sur le langage PHP pour proposer des applications orientées métiers et peut se décomposer de la façon illustrée par [Fig.2] : Utilisation du PaaS.

Le décryptage de l'image 2, montre un point central. Le « Web Server » se compose d'un environnement AMP (Apache, MySQL, PHP) mais principalement d'un socle commun venant de différents horizons comme par exemple plusieurs frameworks (Symfony, Zend,...) ou CMS, CRM, ERP, etc. Ce socle va alimenter des applications métiers que l'utilisateur peut interroger à tout moment pour obtenir des informations ou compléter son contenu.

Le nuage, quant à lui, ne se trouve pas interrogé directement par l'internaute, mais plutôt par le web server ou sinon en direct des applications. Celui-ci retournera directement les informations lorsqu'il sera sollicité et surtout en suivant les informations qu'il aura en sa possession (voir plus de détails plus loin). Par ailleurs, la gestion des applications peut se baser directement sur « Simple Cloud », qui est un projet open source en PHP pour effectuer les relations. Le cheminement des questions/

réponses, venant de l'utilisateur montre certains avantages et aussi quelques inconvénients. La gestion du socle technique est gérée par le fournisseur, ce qui fait une préoccupation en moins pour les personnes qui doivent s'occuper de la partie applicative des projets PHP. L'autre point avantageux, concerne le déploiement automatisé qui soulage les développeurs. En contre partie, l'application ne fonctionne que par le biais d'internet, provoquant un risque en cas de rupture entre les différentes étapes d'accès vers les données. L'autre inconvénient, touche la limitation des technologies proposées et vous contraint de rester dans celle fournie par le prestataire.

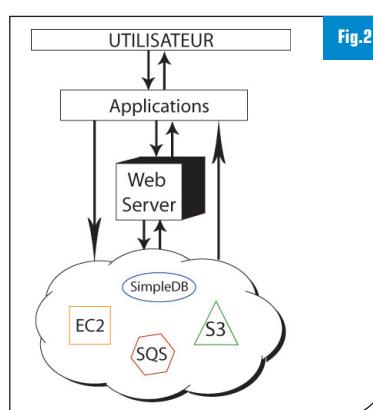
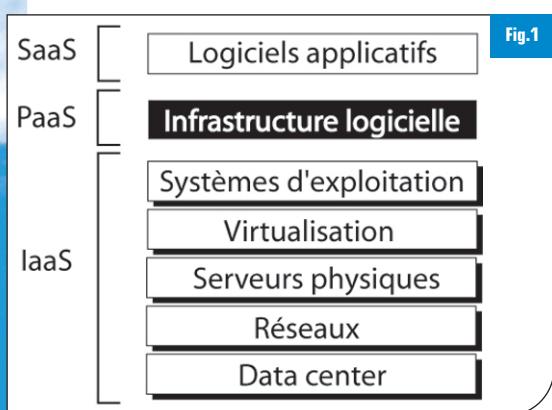
LA DÉCOUPE DU PAAS

Comme expliqué quelques paragraphes avant, le PaaS est géré par le prestataire. Cependant celui-ci laisse la possibilité aux développeurs d'accéder à un espace défini accessible souvent par le biais du FTP.

La décomposition de l'espace proposé permet de réduire les coûts car la charge demandée par les utilisateurs n'est pas tout le temps à son maximum. Le prestataire peut alors décomposer l'espace par application et par client (voir [Fig.3] : décomposition du PaaS). Enfin, un même client peut posséder plusieurs espaces, et par conséquent différentes applications.

LES SERVICES DISPONIBLES POUR PHP

Le langage PHP dans le cloud computing a un rôle important car quel que soit l'environnement, vous pourrez interfaçer avec les différents types de cloud et avec ce langage. L'ensemble de ces services ne possède pas de limite et de nouveaux services



peuvent apparaître. Actuellement, il en existe plusieurs types qui sont :

- S3
- EC2
- SimpleDB
- SQS

Amazon S3

Amazon S3 est un service de stockage. Son but est de vous permettre de placer des fichiers qui ne sont pas systématiquement sollicités à chaque chargement de page. Les fichiers concernés se déclencheront lorsque qu'un clic sera effectué dessus.

Amazon EC2

EC2 est destiné à une utilisation des machines virtuelles, c'est-à-dire la possibilité de configurer et d'exécuter des machines virtuelles dans le cloud. Vous allez pouvoir à tout moment ajouter, supprimer ou modifier la capacité de ces machines, puisque l'ensemble de EC2 est considéré comme modulable.

SimpleDB

SimpleDB est un stockage non relationnel de données à haute disponibilité. Il permet de décharger le travail d'administration de la base de données. La disponibilité des données se trouve accessible à chaque fois que vous en avez besoin.

SQS

SQS est arrivé depuis quelques mois permettant d'obtenir une nouvelle possibilité dans le Cloud. Il correspond aux services de gestion de queues comme une file d'attente. Le SQS n'est pas conçu pour remplacer un système de messagerie que vous trouvez en entreprise mais permet de gérer l'ensemble des demandes par une gestion de file d'attente.

LES OUTILS

Pour réaliser des applications métiers, vous pouvez vous appuyer sur les nom-

breux outils mis à votre disposition. Ces outils peuvent être basés sur des solutions propriétaires ou open source. Actuellement, 2 grandes familles répondent à ce besoin.

Windows Azure SDK pour PHP

L'environnement Windows Azure se voit doté d'un ensemble d'outils pour permettre d'utiliser l'infrastructure logicielle, tout en gardant l'interopérabilité avec l'ensemble des langages du web. Les outils disponibles concernent :

- Une aide à l'utilisation et au paramétrage de la plateforme
- Un outil SDK pour PHP
- Une trousse d'outils en plugins pour Eclipse

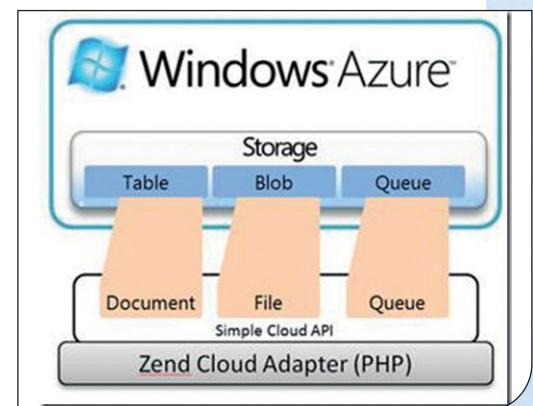
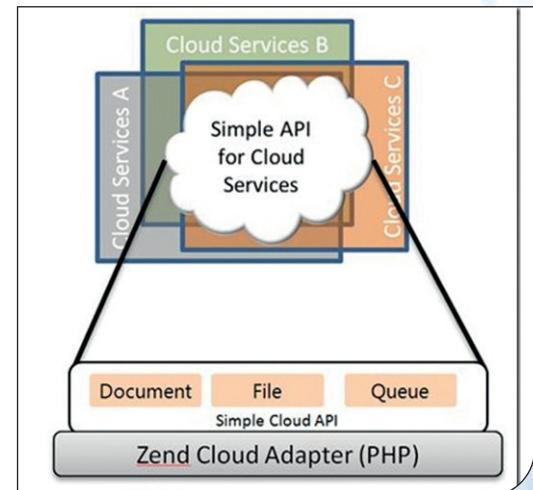
Simple cloud : une API ouverte

Il s'agit de la première API open source, appelée « Simple cloud », qui a été lancée en étroite collaboration avec IBM, Zend, Microsoft et d'autres pour répondre aux différents services et possibilités offertes par le cloud. Cette collaboration a pour but de proposer une norme, et un ensemble d'outils destinés aux développeurs. C'est pourquoi, des extensions et plugins ont été réalisés pour Eclipse permettant de s'interfacer avec l'ensemble des différentes possibilités, tout en gardant la compatibilité avec la plateforme Amazon Web Services (AWS).

Zend a intégré dans son Framework cette API open source et l'ensemble des fonctionnalités sont disponibles depuis la version 1.11 pour répondre aux mieux aux attentes des programmeurs. Vous pourrez concevoir des applications métiers qui seront disponibles sur des web serveurs, tout en gardant la possibilité d'en déporter une partie dans le cloud.

LA SÉCURITÉ

Si vous vous reportez à l'image 2, vous devez ne pas perdre de vue l'importance de la sécurité car on la trouve à tous les niveaux. Un des points importants étant la confidentialité des données. La sécurité du cloud computing sera principalement gérée par le prestataire ou les administrateurs car cette partie est avant tout matérielle (si vous êtes en PaaS, pas en IaaS, ndlr). Celui-ci aura la charge de sécuriser les données et de les garder intactes,



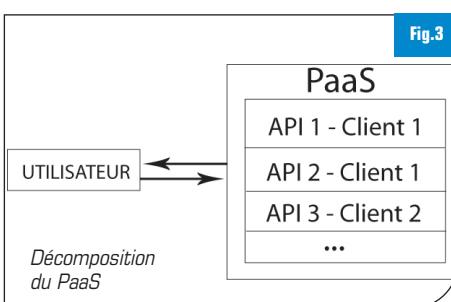
c'est-à-dire proposer une salle climatisée, une gestion de sauvegarde des données. L'autre point important côté sécurité concerne la relation entre les applications métiers et les informations affichées ou gérées. La communication doit s'effectuer en mode sécurisé : HTTPS, sFTP...

CONCLUSION

Le langage PHP tient une place importante dans le cloud comme en témoignent les principaux acteurs de la plateforme. Même si l'ensemble des standards protocoles sont en constante évolution, PHP pourra s'appuyer sur les différents outils disponibles du marché. Bien sûr, l'avenir devrait proposer de nouvelles offres et options pour permettre d'utiliser plus facilement le cloud computing. A vous de tester les migrations !

■ Christophe Villeneuve
consultant pour Alter Way solutions, auteur du livre « PHP & MySQL-MySQLi-PDO, Construisez votre application », aux Éditions ENI. Rédacteur pour nexen.net, Trésorier AFUP et membre du LeMug.fr, PHP-TV, PHPteam...

■ Ghislain Seguy
Directeur Alter Way Hosting



Windows Azure VM Role : mise en œuvre

Windows Azure est la plateforme de Cloud Computing de Microsoft permettant d'héberger des applications et services web. Conçue initialement pour des rôles Web et Worker, respectivement pour cibler les applicatifs avec ou sans interface utilisateur, la plateforme s'enrichit d'un nouveau rôle : VM (Virtual Machine) pour faciliter et accélérer la migration d'applications existantes. Gros plan sur ce nouveau rôle encore en bêta: de son positionnement jusqu'à sa mise en œuvre.

IAAS OR NOT IAAS

Windows Azure est l'offre Microsoft de type **PaaS**, alors qu'en est-il du **VM role** ? En effet, lorsque le terme de VM (machine virtuelle) est évoqué dans le monde du Cloud Computing, il est naturellement associé à l'**IaaS**. Pour bien comprendre comment se positionne le **VM role**, il faut revenir sur certaines caractéristiques de Windows Azure valables pour tous les rôles, VM compris :

- Applications sans état : pour garantir une haute disponibilité et des montées en charge importantes, les applications Windows Azure doivent être sans état : en effet, les répartiteurs de charge vont distribuer les requêtes aux différentes machines disponibles, sans supposer d'affinités particulières.
- Disques non persistants : les machines virtuelles disposent bien d'espaces disques (225Go à 2To) mais ceux-ci ne sont pas persistés : si la machine virtuelle vient à être arrêtée, elle sera remplacée par une nouvelle instance, sans les précédentes modifications disques éventuelles.
- Système d'exploitation : les systèmes d'exploitation supportés sont Windows Server 2008 R2, Windows Server 2008 SP2 (annoncé pour le **VM role**) et Windows Server 2003 (annoncé uniquement pour le **VM role**)
- Azure Fabric Controller : ce contrôleur est le cerveau de Windows Azure, il gère notamment les machines virtuelles : les instancie, les répartit sur les machines phy-

siques, les contrôle et les supervise. Il existe ainsi de nombreuses situations (Détection de mauvaise santé, maintenance...) pour lesquelles une machine virtuelle sera arrêtée et une de remplacement démarrée : ni l'utilisateur, le développeur, ou l'administrateur ne peut intervenir contre ces opérations automatiques.

Les instances du **VM role** sont donc liées à ces contraintes mais aussi aux avantages associés... contrairement à l'**IaaS** où les machines virtuelles peuvent héberger une grande diversité de systèmes d'exploitation, avoir un état (même si c'est déconseillé pour la montée en charge), des disques persistants, et être contrôlées par l'administrateur. Contrairement à ce que pourrait laisser supposer son nom, le **VM role** n'est donc pas une offre de type **IaaS** même si certaines caractéristiques sont voisines.

Mais qu'apporte alors le **VM role ?** Le **VM role** complète les rôles **Web** et **Worker** en autorisant une personnalisation complète de la machine. L'objectif est de faciliter la migration dans Windows Azure d'applications existantes qui comportent des particularités d'installation et/ou de paramétrage difficilement automatisables. Avec le **VM role**, c'est au développeur d'effectuer :

- L'installation du système d'exploitation (parmi les choix disponibles)
- L'installation des composants systèmes et leur configuration
- L'installation des runtimes nécessaires

- L'installation des logiciels serveurs et des applications.

Noter qu'il est également possible de définir des tâches d'installation avec les rôles Web et Worker, mais celles-ci doivent être réalisées par script, sans interaction avec l'utilisateur et s'effectuer rapidement car les tâches sont déclenchées à l'initiative de Windows Azure et dans un délai imparti. Il est donc recommandé de réserver l'usage du **VM role** aux configurations qui ne peuvent pas être déployées avec les rôles Web et Worker.

- Ces rôles ont en effet les avantages :
- De pouvoir être packagés directement depuis Visual Studio
 - D'être plus légers à déployer dans Windows Azure
 - De bénéficier des patchs installés automatiquement par Windows Azure.

Noter enfin que côté tarification, tous les rôles - VM compris - sont facturés de manière identique, selon la taille des instances choisies : de 0,0355 €/heure pour une instance "Extra Small" (1 Core à 1.0 GHz, 768 Mo RAM, 20 Go Disque) à 0,6809 €/heure pour une instance "Extra Large" (8 Core à 1.6 GHz, 14 Go RAM, 2 To Disque).

CRÉATION ET INSTALLATION D'UN VM ROLE

Prérequis

Quelques prérequis sont nécessaires à la conception d'un VM Role :

- Un poste Windows Server 2008 avec le rôle Hyper-V installé afin de créer un disque virtuel au format VHD
- Visual Studio 2010 (Web Edition,

- Standard ou plus)
- .NET Framework 4.0
- Windows Azure SDK 1.3
- Windows Azure Tools for Visual Studio 1.3
- Média d'installation de Windows Server 2008 R2 (ISO ou DVD).

Pour le moment, la plateforme d'hébergement des VM Role restreint le système d'exploitation à Windows Server 2008 R2. Cependant, comme indiqué précédemment, la feuille de route Microsoft prévoit d'élargir ce spectre à Windows Server 2003 et Windows Server 2008 SP2 dans le courant de l'année 2011.

Création du disque virtuel

La première étape consiste à créer un disque virtuel au format VHD (Virtual Hard Disk) contenant toutes les applications et fonctionnalités nécessaires à l'hébergement dans Windows Azure et à l'utilisation souhaitée. Dans cet article, seul le strict nécessaire sera installé pour le bon fonctionnement de la machine virtuelle dans le Cloud.

Vous pouvez créer la machine virtuelle à partir d'un média ou alors à partir d'une machine virtuelle déjà existante, à condition qu'elle ne comporte qu'une seule partition. Par la suite, nous allons considérer que nous la créons totalement.

Pour créer le disque virtuel :

- Lancer Hyper-V Manager
- Créer une nouvelle machine virtuelle [Nom: A définir / RAM: 2048 / Connexion: Local Area Connection - Virtual Network (pour l'exécution de Windows Update durant la création)]
- Création d'un nouveau disque virtuel [Taille: maximum 15 / 35 / 65 Go en fonction de la taille d'instance envisagée : Extra Small / Small / Medium ou plus (Large, Extra Large)]
- Spécifier le média d'installation
- Lancer la machine virtuelle et vous y connecter (double-clic dans Hyper-V Manager pour se connecter)
- Lors de l'installation, choisir l'installation personnalisée (Custom) afin de visualiser l'écran des partitions du disque
- Une fois sur cet écran, appuyer sur

Shift + F10 afin d'ouvrir une invite de commande et taper consécutivement ces commandes dans l'invite, cela forcera le système à n'utiliser qu'une seule partition :

- diskpart
- select disk 0
- create partition primary
- exit
- exit
- Installer Windows Server 2008 R2 (Standard ou Entreprise)
- Ajouter le rôle **Web Server (IIS)**
- Ajouter la fonctionnalité **.NET Framework 3.5.1** et ses prérequis
- Désactiver la mise à jour automatique de Windows (Windows Update) qui pourrait provoquer des redémarrages non souhaités et donc des temps d'indisponibilité
- Effectuer les mises à jour de Windows
- Installer les "Windows Azure VM Role Integration Components" (Pour cela, monter l'ISO présente dans: C:\Program Files\Windows Azure SDK\v1.3\iso\wavmroleic.iso - Ces composants vont permettre à la machine virtuelle de pouvoir être hébergée et pilotée par Windows Azure)
- Une fois cette étape terminée, vous pouvez installer et configurer tout ce que vous souhaitez sur cette machine virtuelle

- **Conseil** : éteindre la machine virtuelle et sauvegarder le VHD
- Relancer la machine virtuelle
- Ouvrir une invite de commande avec les droits d'Administrateur
- Se déplacer dans le répertoire C:\Windows\System32\sysprep, exécuter **sysprep.exe** et renseigner comme suit, l'utilitaire préparera l'image afin qu'elle puisse être utilisée sur n'importe quel type de machine :

- **System Cleanup Action** : Enter System Out-of-Box Experience (OOBE)
- Cocher : Generalize

- **Shutdown Options** : Shutdown

Une fois l'exécution de l'utilitaire sysprep terminée, la machine virtuelle s'éteint et est prête à l'emploi.

Création d'un certificat

Une fois notre disque virtuel terminé, un certificat doit être créé pour que

l'utilisateur soit authentifié de manière certaine auprès de Windows Azure lors de la mise en ligne du disque virtuel et des autres opérations semblables.

Pour créer un certificat :

- Lancer l'invite de commande Visual Studio avec les droits d'Administrateur
- Se déplacer dans le répertoire dans lequel nous voulons générer le certificat (via la commande CD)
- Et enfin taper la commande suivante (remplacer <CertificateName> par le nom désiré) : makecert -sky exchange -r -n "CN=<CertificateName>" -pe -a sha1 -len 2048 -ss My "<CertificateName>.cer"

Ajout du certificat dans Windows Azure

Une fois le certificat créé, il faut l'insérer dans Windows Azure depuis le portail, dans la partie "Management Certificates" afin de pouvoir l'utiliser plus tard lors de nos transactions.

Pour ajouter le certificat :

- Se connecter au portail de gestion Windows Azure : <http://windows.azure.com>
- Sélectionner la partie "Hosted Services, Storage Accounts & CDN"
- Sélectionner "Management Certificates"
- Cliquer sur "Add Certificate", et enfin dans la fenêtre qui s'ouvre, cliquer sur "Browse" pour sélectionner le certificat créé au cours de l'étape précédente

Note : depuis quelque temps, si vous n'avez jamais ajouté de certificat, le bouton "Add Certificate" est inactif. Ce souci devrait être rapidement corrigé mais en attendant, voici un petit coup de pouce pour débloquer la situation. Le nouveau portail de gestion laisse la possibilité de retourner sur l'ancienne version du portail, qui permet l'ajout de certificats en cliquant conséutivement sur Windows Azure > 'Votre nom projet' et ensuite sur "Account" puis "Manage My API Certificates". Après avoir ajouté le certificat, vous pouvez retourner sur le nouveau portail.

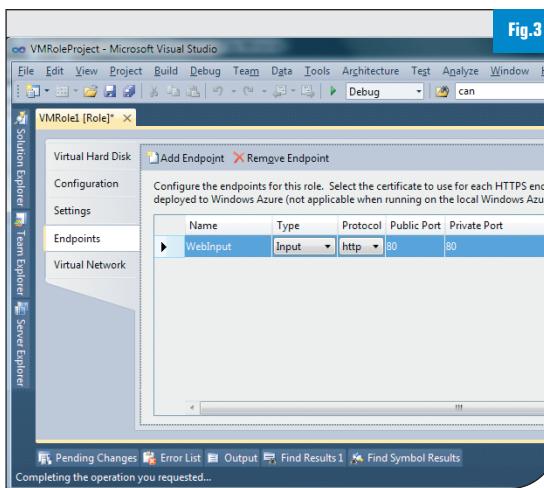
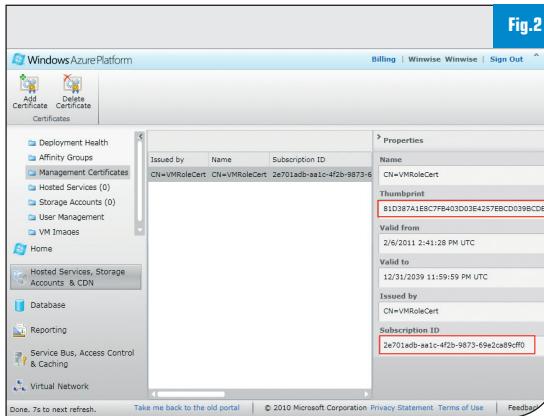
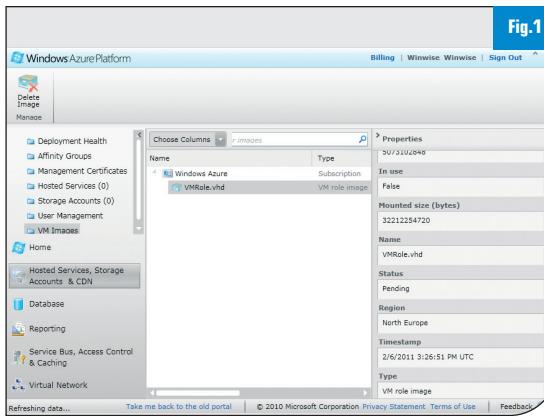
Mise en ligne du disque virtuel

Maintenant que le certificat et le disque virtuel sont prêts, il ne reste plus qu'à déposer le disque virtuel dans Windows Azure. Pour réaliser cette tâche, nous allons exécuter un

utilitaire du SDK Windows Azure du nom de *csupload*. Cet exécutable finit de préparer le disque virtuel en modifiant certains éléments de son contenu puis en le compressant. Il crée ensuite un Blob dans l'espace de stockage Windows Azure pour conserver ce disque virtuel, le transfert vers ce blob et ajoute une image VM Role dans le portail dans la rubrique "VM Images" [Fig.1].

Pour mettre en ligne le disque virtuel:

- Lancer l'invite de commande du SDK Windows Azure avec les droits



d'Administrateur (Démarrer/Tous les programmes/Windows Azure SDK v1.3)

- Taper la ligne de commande suivante : `csupload Add-VMImage -Connection "SubscriptionId=<YOUR-SUBSCRIPTION-ID>;CertificateThumbprint=<YOUR-CERTIFICATE-THUMBPRINT>" -Description "2008 R2 VHD Base" -LiteralPath "<PATH-TO-VHD-FILE>" -Name "<ImageName.vhd>" -Location <HOSTED-SERVICE-LOCATION>`

Détails des paramètres : [Fig.2].

- <YOUR-SUBSCRIPTION-ID>: Identifiant du compte Windows Azure, à récupérer depuis le portail
- <YOUR-CERTIFICATE-THUMBPRINT>: Empreinte de votre certificat, à récupérer dans les propriétés de votre certificat via l'attribut "Thumbprint" (retirer les espaces entre les caractères pour la ligne de commande)
- <PATH-TO-VHD-FILE>: Chemin vers le disque virtuel
- <ImageName.vhd>: Nom que va porter ce disque virtuel dans le portail

- <HOSTED-SERVICE-LOCATION>: Localisation du Datacenter qui hébergera les instances de ce VM Role (choisir "North Europe" qui correspond à Dublin dans notre cas).

Cette opération peut durer plusieurs heures, voire même plus d'un jour en fonction de la taille du disque virtuel et de la connexion.

Création du Service Model

Une fois que l'image du VM Role est disponible depuis le portail et avant d'en créer des instances, il faut déclarer les paramètres d'instanciation auprès de Windows Azure, en renseignant le Service Model:

- Lancer Visual Studio 2010
- Créer un projet de type "Cloud\Windows Azure Project"
- Dans la fenêtre de sélection des rôles, ne rien sélectionner et valider
- Faire un clic droit sur "Roles", puis cliquer sur "Add" et enfin "New Virtual Machine Role"
- L'écran qui s'ouvre permet la configuration du VM Role [Fig.3]:

- **Virtual Hard Disk** : Certificat de communication avec le portail de gestion et Image (VHD) de base à utiliser

- **Configuration** : Nombre et taille des instances (ainsi que l'activation des "Diagnostics")

- **Settings** : Ajout des paramètres applicatifs et chaînes de connexions

- **EndPoints** : Ajout de règles d'ouverture/routage de ports (exemple d'ouverture du port 80 afin d'accéder au serveur IIS qui a été installé : Name: WebInEndpoint / Type: Input / Protocol: Http / ExternalPort: 80 / InternalPort: 80)

- **Virtual Network** : Activation possible de **Windows Azure Connect**

• Une fois la configuration terminée, faire un clic droit sur la solution et cliquer sur "Publish"

• Sur l'écran de publication, sélectionner "Deploy your Windows Azure project to Windows Azure" afin de pouvoir renseigner les informations :

- "Credentials" pour associer le certificat au compte Windows Azure
- "Remote Desktop" pour autoriser les connexions à distance à l'instance ou les instances du VM Role (il suffit de créer un certificat - à exporter avec la clé privée via le "View" -, un nom d'utilisateur, un mot de passe et une date d'expiration)

- Toujours sur l'écran de publication, sélectionner "Create Service Package Only" afin que la publication du package soit réalisée en local et non directement dans Windows Azure

Création du Service et déploiement du package

Une fois le package généré, il faut le déclarer auprès de Windows Azure grâce au portail, ce qui permettra alors d'instancier nos machines virtuelles :

- Se rendre sur le portail de gestion Windows Azure
- Sélectionner la rubrique "Hosted Services, Storage Accounts & CDN"
- Sélectionner la sous-rubrique "Hosted Services"
- Cliquer sur "New Hosted Service" puis renseigner l'écran comme suit [Fig.4] :

- **Name** : au choix (nom interne)
- **URL Prefix** : au choix (nom externe - url)
- **Region** : au choix (**North Europe** de préférence, pour Dublin)
- **Deployment Name** : au choix (numéro de version)
- **Package Location** : sélectionner le fichier du package (.cspkg)
- **Configuration File** : sélectionner le fichier de configuration (.cscfg)
- **Cliquer sur "Add Certificate"** : ajouter le certificat pour l'accès distant (si configuré)

• **Cliquer sur OK.**

Une fois le processus terminé, la ou les instances du VM Role sont lancées et disponibles comme si elles étaient dans votre Datacenter. Vous avez accès aux ports que vous avez définis dans le Service Model ainsi qu'à l'accès distant. Vous pouvez aussi joindre les instances à votre réseau d'entreprise en établissant une connectivité sécurisée de niveau IP grâce à la nouvelle fonctionnalité **Windows Azure Connect**.

Noter enfin que Windows Azure ne met pas à jour le système d'exploitation du VM role (contrairement aux Web et Worker role) : il faut donc mettre à jour le disque virtuel soi-même. Il en est de même pour les mises à jours des applicatifs et runtimes installés : la procédure

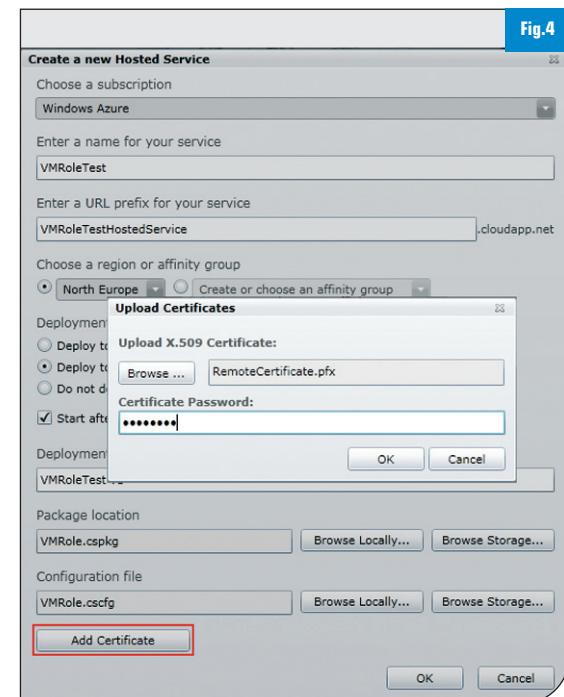
conseillée est alors d'installer les mises à jour dans un disque virtuel de différences, le mettre en ligne, l'associer au disque de base puis de mettre à jour le Service Model pour permettre de nouvelles instances.

Conclusion

Cet article a présenté une évolution remarquée de la plateforme Windows Azure : le VM role (Virtual Machine), disponible actuellement en bêta. Ce rôle complète les deux rôles existants Web et Worker en apportant un haut niveau de personnalisation de la machine virtuelle hébergeant les applications.

Plusieurs étapes sont nécessaires avant d'exploiter la machine virtuelle, de l'installation du système d'exploitation avec Windows Server 2008 R2 Hyper-V, jusqu'au transfert de l'image au format VHD (Virtual Hard Disk) dans Windows Azure et sa configuration... de nombreuses étapes manuelles qui gagneraient à être simplifiées et automatisées.

Le **VM role** est donc une option intéressante pour accélérer la migration d'applications existantes vers Windows Azure lorsque celles-ci nécessitent une configuration ou une installation particulière en dehors des standards proposés par Microsoft.



■ **Jean-Luc Bouchou**
Solutions Architect & Leader Windows Azure à Winwise
Blog : <http://blog.jeanluc-bouchou.com>



■ **Geoffrey Daniel**
Leader Technique à Winwise
Blog : <http://blogs.developpeur.org/neurocypher>



Se méfier du taux de disponibilité

Le cloud joue sur les aspects flexibilité et disponibilité. Cependant, un point sur lequel vous devez être très vigilant : quels sont les taux de disponibilité des services cloud que j'utilise ? Et la question est vitale si vous faites des offres SaaS. Ainsi, App Engine, version business, propose un taux de 99,9 % de disponibilité (SLA et service premium pour développeur).

La version standard n'est pas préconisée pour la production. Amazon EC2 propose un taux de 99,95 % (au niveau annuel). Par contre, Amazon S3 affiche un taux de 99,9 % (par mois). Côté Windows Azure, plusieurs taux sont affichés, selon la fonction / le service. Ils varient de 99,95 à 99,9. Avant toute chose : quel

niveau de SLA est disponible ? A quoi correspond exactement le taux de disponibilité du service ? Bref, lisez attentivement le contrat de votre fournisseur et quel niveau de support technique, il peut vous proposer.

La plupart des fournisseurs proposent un tableau de bord pour suivre l'évolution de la disponibilité des services, l'état du réseau, etc. Outil indispensable ! Autre tableau de bord indépendant : <http://www.cloudstatus.com/>

La latence est un problème important que vous devez gérer en optimisant l'utilisation des caches et des systèmes de basculements entre datacenters et zones géographiques. Si votre utilisateur se

situe en France et l'hébergement de votre application sur un datacenter aux USA ou en Asie, la latence sera importante car il faut que les données traversent des milliers de kilomètres, alors que si votre application se localise en Europe, la proximité avec l'utilisateur sera bien meilleure.

Des spécialistes comme Akamai proposent des outils et des mécanismes de très hautes performances pour trouver le meilleur chemin d'un point A à un point B. Windows Azure possède son propre système avec CDN (Content Delivery Network). Google App Engine a des mécanismes similaires comme le Memcached.

■ F.T.

Adapter une application PHP à Azure

PHP est une technologie bien maîtrisée sous Windows, et le récent développement d'un driver PDO pour SQL Server a apporté la dernière touche à l'édifice, en proposant aux applications PHP de s'intégrer sans (trop) de difficultés à SQL Server. Si l'intérêt de PHP sous Windows dépend principalement du contexte technique pré-existent, il en est tout autrement de Windows Azure. En effet, les plateformes cloud permettent une montée en charge à la demande, et il existe aujourd'hui très peu de solutions clé en main en PHP pour en tirer parti.

Comment aborder Azure quand on a une application PHP ? Cette discussion va d'ailleurs au-delà d'Azure, puisque les questions que pose cet article sont pertinentes pour toutes les plateformes cloud.

Architecture générale des applications cloud

La première chose à comprendre est que vous allez toujours tourner sur au moins deux machines : même si vous n'avez pas d'impératif de charge, cette architecture vous garantira une bonne disponibilité de votre application, et permet de faire des mises à jour à chaud. Dès lors, vous allez devoir renoncer à un certain nombre de ressources couramment utilisées dans une infrastructure mono-serveur :

- Le cache local (APC, etc.) ne peut plus contenir certaines données qui doivent être parfaitement synchronisées entre les N instances de votre site.
- Les sessions locales : à la différence d'une ferme de quelques serveurs, le répartiteur de charge de Windows Azure ne fait pas d'affinité serveur. En d'autres termes, vous n'avez pas la garantie que les utilisateurs reviennent toujours sur le même serveur de la ferme.
- Le système de fichiers local : d'une part il n'est pas partageable entre instances, d'autre part il n'est pas sauvegardé. Vous ne pourrez y mettre que des informations temporaires.

Les ressources mises à votre disposition dans Azure pour remplacer ces approches locales sont les suivantes :

- SQL Azure : une base de données relationnelle très proche de SQL Server.
- Azure Storage : trois entrepôts de données (un en mode blob, un en mode tabulaire, et un mécanisme de files d'attentes) qui permettent le partage simple des données entre instances et avec le monde extérieur.
- Azure XDrive : la possibilité de monter un fichier stocké dans les blobs Azure Storage comme un disque local (sur une machine Windows Azure).

Les problématiques de portage vers Azure sont grossièrement organisées autour de trois thématiques : la gestion des bases de données, la gestion des fichiers, et la gestion des caches. Bien entendu, il faut également prendre en compte le déploiement et le débogage des applications.

L'environnement de développement et de déploiement

La seule contrainte pour un développement Azure est d'avoir installé le SDK Azure, et les outils permettant le déploiement/débogage. Les solutions sont aujourd'hui assez nombreuses, et vous

pouvez soit utiliser Visual Studio (en le combinant par exemple à VS.PHP), soit les extensions Azure pour Eclipse, que vous trouverez sur www.windowsazure4e.org. Si vous avez d'autres environnements de développement, il existe un ensemble d'outils en ligne de commande permettant d'automatiser le déploiement vers Windows Azure (<http://azurephptools.codeplex.com/>).

A noter : le SDK Azure comporte un environnement de simulation qui reprend les principales briques de Windows Azure, et c'est la meilleure façon de commencer votre migration. Vous allez cependant devoir affronter quelques différences entre l'environnement de simulation et l'environnement cible, c'est pourquoi je vous recommande de travailler en trois phases : dans un premier temps dans l'environnement de simulation, puis à cheval entre Azure et l'environnement de simulation, en déployant les données dans le cloud et le code dans l'environnement de simulation, et enfin en faisant tout tourner sur Azure. Cette approche mettra en évidence les éventuels problèmes d'architecture liés à la distance qui sépare le code des données dans un environnement réparti.

Un mot pour terminer sur la version de PHP que vous allez exécuter dans Windows Azure : il existe aujourd'hui deux versions de PHP pour Windows (5.2 et 5.3). La différence entre les deux versions est importante, la version 5.3 est nettement préférable sur Azure (pour être précis, utilisez la version 5.3.x en mode Non Thread Safe, que vous pouvez télécharger sur <http://windows.php.net/>).

La gestion des bases de données

Le choix de la base de données est assez complexe à effectuer. Si votre application utilise déjà SQL Server, la solution la plus simple est de se connecter à SQL Azure. Cette connectivité est possible en PDO depuis août 2010, et les deux différences à connaître sont assez simples :

D'une part, toutes les tables dans SQL Azure doivent avoir une clé primaire.

D'autre part, les opérations DDL de manipulation de la structure des tables doivent être isolées des autres requêtes. Par défaut, toutes les requêtes transmises à SQL Azure sont préparées et donc dans un contexte DML, ce qui provoque une erreur pas toujours facile à comprendre. Voici un exemple du code requis pour exécuter une requête DDL :

```
$conn = new PDO(...);
$conn->setAttribute(constant('PDO::SQLSRV_ATTR_DIRECT_QUERY'), true);
$stmt1 = $conn->query("DROP TABLE ...");
```

L'article détaillé se trouve sur MSDN (<http://msdn.microsoft.com/en-us/library/ff754356%28v=sql.90%29.aspx>)

Autres bases de données :

Il est techniquement possible de faire tourner sur Azure n'importe quelle application, et il est même possible de disposer d'un disque dur persistant grâce à XDrive. Cependant, cette technique a plusieurs inconvénients : d'une part, le temps de latence est important, ce qui limite les performances (avec notamment des problèmes potentiels de performance), et surtout ce système n'est visible en lecture/écriture que d'une seule instance Windows Azure (les autres peuvent accéder à des instances en lecture seule), ce qui complique un peu les scénarios de déploiement et de montée en charge. Enfin, c'est à vous de fournir la logique d'administration de la base, alors que SQL Azure couvre tous les besoins en standard.

La gestion des caches

Caches d'opcodes

Les caches locaux en PHP ont surtout une fonction d'optimisation technique : ils servent par exemple de cache d'opcodes et de cache de résolution des fichiers. Dans ce cadre, vous pouvez probablement continuer d'utiliser votre cache habituel, ou le remplacer par WinCache (<http://sourceforge.net/projects/wincache/>), qui est un clone de APC développé par les équipes Microsoft et optimisé pour la plateforme Windows. Si vous désirez stocker des objets ou des données dans vos caches, voilà comment déterminer ce que vous pouvez faire. Les données d'une application sont en général de trois natures possibles : *Les données d'activité*, qui ne changent pas au cours du temps, et ne sont consultées qu'en lecture seule (un catalogue, des menus, etc.). Ces données peuvent être stockées dans n'importe quel type de cache sans risque pour la cohérence de vos données.

Les données d'activité : ce sont des données accédées par un seul utilisateur, que l'on met en cache pour accélérer leur mise à disposition. Vous ne pouvez pas stocker ces données sur la machine locale dans Azure, mais elles sont adaptées aux caches répartis.

Les données de ressource : ce sont des données accédées en lecture et écriture par plusieurs utilisateurs (par exemple le stock restant d'un produit). Ces données ne sont généralement stockables dans aucun cache, à moins d'avoir des stratégies d'invalidation de cache bien définies.

Caches répartis

Les caches répartis (le plus connu est memcached) permettent de mettre les objets en cache sur une ou des machines distinctes du serveur web. Les données sont donc visibles à l'identique depuis n'importe quel serveur web, ce qui règle le problème des données d'activité. Il existe plusieurs possibilités pour implémenter un cache réparti dans Azure :

- 1 Memcached
- 2 Azure AppFabric Caching
- 3 Sql Azure

Memcached est une bonne option, en ce sens qu'elle est familière à la plupart des développeurs PHP. Attention cependant, la gestion du firewall sur Windows Azure bloque les versions récentes, et pour le moment la seule version testée avec succès est la 1.2.1 (<http://www.urielkatz.com/archive/detail/memcached-64-bit-windows/>). En passant, il faut signaler une option intéressante : comme la plupart des applications PHP ont des besoins assez modestes en RAM, il est possible de faire tourner Memcached sur les instances PHP, et la technique est montrée ici : <http://www.davidaiken.com/2011/01/11/windows-azure-memcached-plugin/>

Azure AppFabric Caching a un ensemble de propriétés intéressantes, notamment en termes de persistance et de réplication. Il n'existe cependant pas encore de wrapper PHP pour le manipuler. Enfin, vous pouvez utiliser une base de données comme cache réparti (SQL Azure), mais n'oubliez pas que les performances d'une base de données pour des accès très simples (clé-valeur) est inférieure à celle d'un système mémoire comme Memcached, et que vous devez vous-même nettoyer les entrées de cache qui ne servent plus sous peine de voir la base grossir.

Gestion du filesystem

La gestion du filesystem est de loin la partie la plus délicate de l'adaptation d'une application PHP à Azure. En effet, même si on peut se donner un accès au filesystem local en cas de besoin, il n'est ni persistant ni répliqué, ce qui empêche de stocker ou de servir des contenus dès lors qu'on dépasse une instance. En fonction de la nature des fichiers, plusieurs solutions s'offrent à vous.

• 1/ filesystem local

Vous ne devez l'utiliser que pour des fichiers temporaires utilisés sur une seule requête. Il s'agit en général de l'upload d'une image qui sera retravaillée par votre code avant d'être sauvegardée, ou enchaînement d'opérations passant par le disque.

• 2/ Azure Blob Storage

C'est la solution recommandée, notamment pour les médias, fichiers uploadés par les utilisateurs, etc. La bonne façon de procéder est de stocker ces fichiers dans des blobs Windows Azure. Le SDK Azure pour PHP (<http://phpazure.codeplex.com/>) contient ce dont vous avez besoin, et vous pouvez retoucher votre code pour exploiter les blobs. Il existe cependant une solution intéressante : si vous utilisez déjà les stream wrappers pour PHP, sachez que le SDK contient un wrapper pour Azure blob storage, ce qui vous permet de ne pas retoucher votre code.

• 3/ XDrive

Le gros avantage de XDrive est que vous n'avez pas à retoucher votre code, mais n'oubliez pas que vous vous limitez à une seule instance de votre application Web, ce qui limite la montée en charge.

La gestion des sessions

La gestion des sessions sur une ferme de serveurs PHP se fait en déportant les sessions, par exemple dans une base de données. Cette solution n'est pas vraiment scalable (enfin, elle l'est sur le papier, mais la performance par machine en prend un coup). La solution la plus pragmatique sur Windows Azure consiste à déporter les sessions sur Memcached. L'extension PHP pour Memcached est aujourd'hui disponible sur la page de Pierre Joye, et la version que vous cherchez est là : http://downloads.php.net/pierre/php_memcache-2.2.6-5.3-nts-vc9-x86.zip

Au passage, si vous cherchez d'autres extensions et que vous êtes perdu dans la multitude de versions, prenez simplement les versions marquées 5.3 et NTS et vous serez dans la configuration recommandée pour Azure.

■ Pierre Couzy

Déployer une application PHP/MySQL dans le cloud avec Amazon EC2

Dans cet article, nous allons voir comment déployer une application PHP – MySQL sur une instance Amazon EC2. Mais avant de migrer une application vers le cloud “parce que c'est à la mode”, voyons quelles sont les raisons qui pourraient pousser à vouloir effectuer cette migration.

Les services d'Amazon (AWS), bien que relativement bon marché ne sont pas les plus accessibles. Un serveur low-cost chez OVH ou Dedibox reviendra souvent moins cher. L'intérêt principal d'Amazon est son élasticité. Si vous avez des besoins importants (nécessitant plus d'un serveur) pour votre application, et si vous avez une activité avec des pics marqués, utiliser Amazon EC2 prend du sens. Un cas d'utilisation typique est une campagne marketing à la télévision. Au moment de la diffusion des publicités, un grand nombre de visiteurs va affluer sur votre site. Vous allez multiplier les nombre de visiteurs moyen par 10 ou 100, en l'espace de 20 minutes.

Commençons la migration

Pour cette migration, nous allons créer 2 AMI. Une AMI est une image de machine virtuelle, qui peut être déployée et multipliée pour augmenter l'élasticité de la solution.

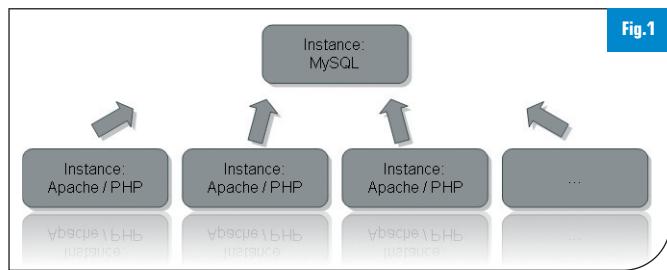
Nous allons donc créer :

- Une AMI pour MySQL
- Une AMI pour Apache/PHP

A partir de ces AMI, nous allons créer plusieurs instances. Une pour la base MySQL et plusieurs pour Apache/PHP. Nous pourrons directement rajouter des serveurs Apache/PHP lorsque la charge augmentera. Cependant, tous les serveurs se connecteront à la même base de données MySQL. Il ne faut donc pas longtemps pour comprendre qu'en cas de forte charge, nous n'aurons jamais de problèmes avec les serveurs PHP, mais que la base de données sera le goulot d'étranglement. Il existe des solutions pour augmenter les performances de MySQL (configuration master/slave, etc.) mais le sujet nécessiterait un article à part entière [Fig.1].

Commander une instance

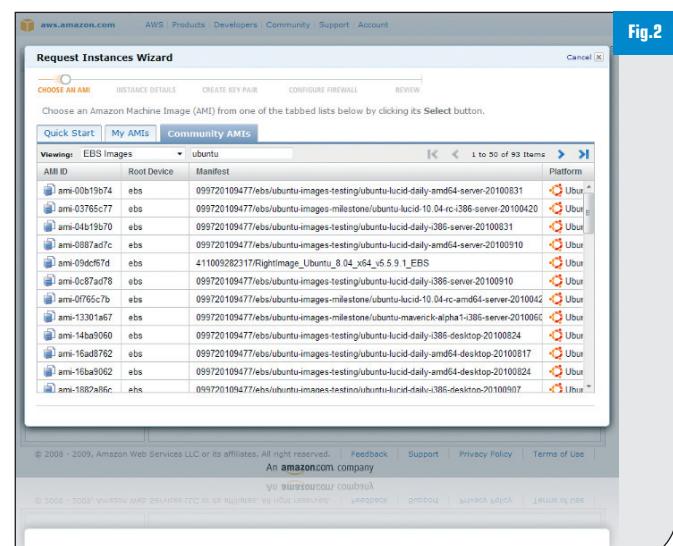
La première étape consiste évidemment à s'inscrire aux services Amazon. Une fois l'inscription effectuée, vous pouvez accéder à l'“AWS management console” qui permet de créer des instances.



Première étape importante: choisir la région. Amazon permet d'installer son serveur dans l'un de ses 4 datacenters (2 aux Etats-Unis, un en Irlande, un dans le Pacifique). Dans notre cas, nous choisirons l'Irlande pour être le plus proche possible de la France et donc bénéficier du meilleur ping possible.

Un clic sur "Launch instance" nous permet de sélectionner le type de machine virtuelle qui va accueillir notre serveur. Nous sélectionnons une image Ubuntu, mais vous pouvez choisir votre distribution préférée.

A savoir lorsque vous choisissez une AMI : si vous prenez une AMI 64 bits, vous ne pourrez pas prendre un serveur virtuel bas de gamme. Si vous n'avez pas besoin de la puissance d'une machine 64 bits, choisissez donc une image 32 bits. Il existe 2 types de stockage chez Amazon : EBS et instance. Avec un stockage EBS, le disque dur de votre machine virtuelle est "dans le cloud". Vous pouvez arrêter la machine virtuelle, la redémarrer sur un autre serveur et retrouver le contenu de votre disque dur. A l'opposé, le stockage instance est temporaire. Le contenu du disque dur disparaît dès que l'instance est arrêtée. On évitera évidemment de stocker une base de données sur ce type d'instance. Par contre, il peut être parfaitement adapté au démarrage d'un serveur web de soutien. Pour notre serveur MySQL, nous allons donc choisir une instance AMI Ubuntu Lucid Lynx 32 bits EBS. Le code est "ami-0c87ad78", nous pouvons chercher cette instance dans Amazon. En cliquant sur "Select", on accède à la création d'instances. Nous choisissons la création d'une instance de type "Small" [Fig.2].



Contrairement à une machine Linux classique, vous ne pourrez pas vous loguer sur une instance EC2 avec un login et un mot de passe. A la place, il faudra utiliser un couple clef publique/clef privée (fichier .pem) pour effectuer un SSH sur la machine (que EC2 vous permet de récupérer). L'étape suivante consiste à gérer le pare-feu de la machine virtuelle. Pour la machine MySQL nous allons ouvrir les ports SSH et MySQL. Les ports MySQL seront ouverts uniquement pour nos autres machines virtuelles PHP. Parce que nous sommes fainéants, nous ouvrons aussi le port HTTP pour installer PHPMyAdmin. Enfin, nous avons sélectionné toutes les options pour l'instance, nous pouvons la créer, en cliquant sur le bouton "Launch": *The instance will appear in the list of instances managed by Amazon* [Fig.3].

Se connecter au serveur

La connexion au serveur s'effectue en SSH, avec votre clef publique/privée. La manière d'utiliser votre clef publique/privée variera suivant le client SSH que vous utiliserez. Attention, si vous êtes sous Windows et utilisez Putty, le format de clef d'Amazon n'est pas supporté par défaut. Il faudra convertir la clef en utilisant ce tutorial : <http://docs.amazonwebservices.com/AmazonEC2/gsg/2006-06-26/putty.html>

Si vous utilisez OpenSSH, il suffit de taper :

```
ssh -i fichier.pem root@url du serveur
```

Note : pour Ubuntu, il faudra utiliser l'utilisateur "ubuntu" et non "root".

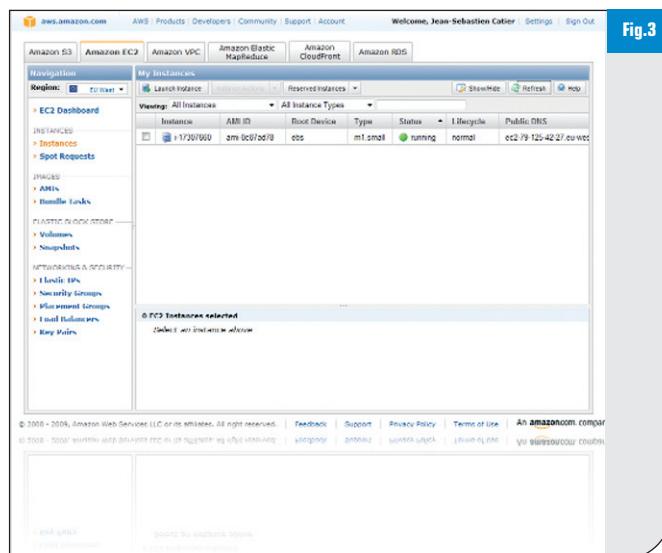
Installer MySQL

Installer MySQL se fait de manière classique. Si vous êtes sous Ubuntu :

```
sudo apt-get install mysql-server
```

Par défaut, MySQL refuse toute connexion ne venant pas de "localhost", c'est-à-dire de la machine sur lequel il est installé. Dans notre cas, les serveurs Amazon PHP vont tenter d'accéder à MySQL. Il faut donc autoriser MySQL à écouter les demandes venant de l'extérieur. Pour cela, il faut d'abord éditer le fichier /etc/mysql/my.cnf et modifier la valeur suivante:

```
bind-address = 127.0.0.1
```



Pour donner accès à n'importe quelle adresse IP, il suffit de commenter la ligne, ou indiquer l'adresse "0.0.0.0".

Cela ne suffit toujours pas à accéder à MySQL de l'extérieur. En effet, les utilisateurs MySQL ne sont autorisés que sur une adresse IP donnée (par défaut localhost). Il faut donc se connecter à la gestion des utilisateurs MySQL (à travers PHPMyAdmin par exemple), puis autoriser un utilisateur, que vous utiliserez, à être accédé depuis les serveurs extérieurs.

ATTENTION! Soyez-sûr d'avoir mis un mot de passe pour l'utilisateur!

Le cas des serveurs PHP

Passons maintenant aux serveurs PHP. Nous allons pour le monter prendre une nouvelle instance. Voici les points que nous modifions par rapport à la première instance :

Image AMI utilisée : une image Ubuntu 32bits ebs. Nous aurions pu utiliser une instance store, comme les serveurs PHP ne contiendront aucune donnée persistante (à part les logs). Cependant, nous verrons dans le 3e article que l'utilisation d'un store EBS facilite la mise en place de l'autoscaling.

Type d'instance : pour les serveurs PHP, nous voulions prendre une instance "Tiny". C'est l'instance la plus petite existant chez Amazon. L'idée était de jouer sur le nombre de machines plutôt que sur la puissance des instances pour dimensionner l'application. Cependant, 2 obstacles se sont mis en travers de notre chemin: la dernière version d'Ubuntu (10.04) n'était pas compatible avec les instances Tiny lorsque nous avons réalisé les tests. Nous nous sommes donc rabattus sur une instance "small".

Installons notre application PHP, donc les packets Apache, PHP, déployons l'application, etc. Evidemment, nous indiquons l'adresse IP du serveur de base de données MySQL que nous venons d'installer. A la fin de cette installation, vous devriez enfin avoir une application PHP qui fonctionne.

Load balancing de notre application

Avant d'activer l'autoscaling, nous devons disposer d'un load balancer (répartiteur de charge). Nous allons donc commencer par créer ce répartiteur de charge. Pour créer un load balancer, dans la console AVS, dans la section **Networking & Security**, sélectionnez **Load Balancers**, puis cliquez sur le bouton "Create Load Balancer". On donne un nom à notre load balancer ("http-load-balancer"), puis on choisit sur quel critère effectuer le load-balancing. Amazon nous propose de base de répartir la charge sur Apache, et c'est exactement ce que nous souhaitons [Fig.4].

Configurer le health check

L'étape suivante consiste à expliquer au load balancer comment savoir si les machines vers lesquelles il route les informations sont actives ou non. En effet, si une machine a une défaillance technique, il est important qu'il stoppe le routage vers cette machine. Pour cela, le load balancer effectue des health check, c'est-à-dire des requêtes régulières vers nos serveurs. Par défaut, il effectue sa requête vers la page d'accueil du site. Traditionnellement, la page d'accueil d'un site génère quelques requêtes, et peut être assez lourde. Nous allons donc plutôt diriger le "health check" vers une simple page HTML qui retournera "OK". Nous appelons cette page "load_balancer_ping.html". Nous gardons les options avancées par défaut: le load balancer effectuera une vérification toutes

les 5 secondes, considérera qu'une machine est défectueuse après 2 tests successifs défaillants et considérera qu'elle fonctionne de nouveau après 10 tests réussis [Fig.5].

L'étape suivante consiste à rajouter des instances dans notre load-balancer. Nous allons donc déployer dans ce load-balancer notre instance unique de serveur PHP.

Gérer les sessions

Le load balancer d'Amazon envoie les requêtes HTTP vers le serveur le moins chargé. Cela pourrait représenter un problème pour nous. En effet, si notre application PHP utilise des sessions, et si une session est démarrée sur un serveur, nous voudrions la garder sur les autres serveurs (et la partager). Dans les options Amazon, on peut trouver un paramètre "Session stickiness". Si on active ce paramètre, un utilisateur sera toujours routé vers le même serveur, et pourra donc garder sa session. Ce paramètre résoudrait donc notre problème si on souhaitait un simple load-balancer. Cependant, nous avons prévu d'effectuer de l'autoscaling. Dans le cadre de l'autoscaling, un serveur peut être dynamiquement ajouté, et surtout retiré. Si un serveur est retiré, nous voudrions que les sessions de ce serveur ne soient pas perdues. Le paramètre "Session stickiness" ne nous aidera pas dans ce cas.

Partager les sessions avec Memcached

Memcached est un gestionnaire de cache. Nous allons mettre en place le cache sur une machine centralisée. Dans notre cas, le serveur MySQL servira aussi de serveur Memcached.

Nous nous connectons donc sur le serveur MySQL et installons Memcached. Sous Ubuntu 10.04, il suffit de taper :

```
sudo apt-get install memcached libmemcache-dev
```

Nous nous connectons donc sur le serveur PHP et installons l'extension Memcached pour PHP :

```
sudo apt-get install php5-memcached
```

Avant de déporter les sessions vers le serveur Memcached/MySQL, nous devons autoriser les serveurs externes à accéder à Memcached. Par défaut, Memcached se comporte comme MySQL, c'est-à-dire qu'il n'écoute que les requêtes provenant de "localhost". Pour corriger cela, nous allons éditer le fichier /etc/memcached.conf et commenter la ligne suivante :

```
#-1 127.0.0.1
```

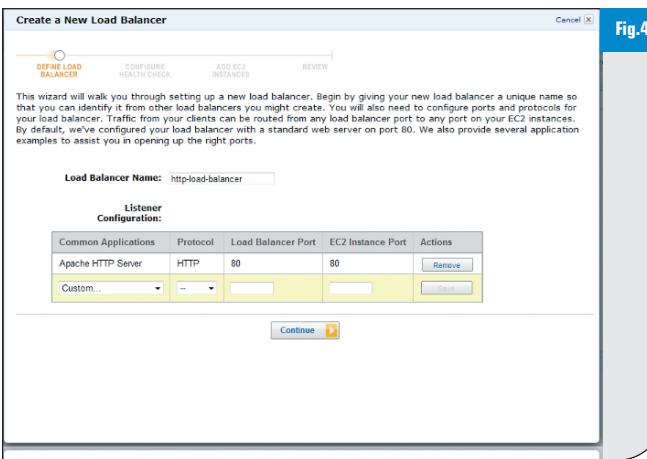


Fig.4

Nous devons maintenant redémarrer le service Memcached pour prendre les changements en compte :

```
sudo /etc/init.d/memcached restart
```

Memcached par défaut ne propose pas de mode d'authentification. Il faut donc absolument faire attention à ne pas ouvrir le port à tout le monde. En effet, ce serait une faille de sécurité énorme puisqu'elle donnerait accès depuis l'extérieur à toutes les sessions (et donc la possibilité de voler des sessions utilisateur). Heureusement, Memcached nous offre la possibilité de donner accès à un port uniquement aux machines d'un certain groupe. Et nous avons justement un groupe de sécurité pour les serveurs HTTP! Nous allons donc ouvrir le port Memcached du serveur MySQL (11211) mais uniquement pour les serveurs HTTP.

Nous pouvons maintenant déporter les sessions du serveur PHP. On peut le faire facilement à partir du php.ini, en remplaçant la directive "session.save_handler":

```
session.save_handler = memcached
session.save_path = "10.226.2.3:11211"
```

Il suffit de redémarrer le serveur PHP pour prendre en compte les modifications :

```
sudo /etc/init.d/apache2 restart
```

Ne pas oublier de tester l'application pour voir si des erreurs apparaissent dans le log d'erreur lors de l'ouverture des sessions :

```
tail -f /var/log/apache/error.log
```

Si aucune erreur relative aux sessions ne s'affiche, générez une image de serveur!

Générer une image de serveur

L'autoscaling consiste à rajouter dynamiquement de nouveaux serveurs à la demande. Pour cela, il faut bien sûr disposer d'une image de serveur! Dans cette section, nous allons voir comment faire une image de serveur (AMI) qui nous permettra de dupliquer notre serveur web à volonté.

Générer une image AMI est extrêmement simple. Dans la console AWS, il suffit d'accéder à la liste des instances, sélectionner l'instance à transformer en image, puis cliquer sur **Instance Actions > Create Image (EBS AMI)**. Il suffit de donner un nom à l'instance, éventuellement des détails. Un clic sur "Create this image" crée l'instance instantanément. Notez l'ID de l'instance dans la page de

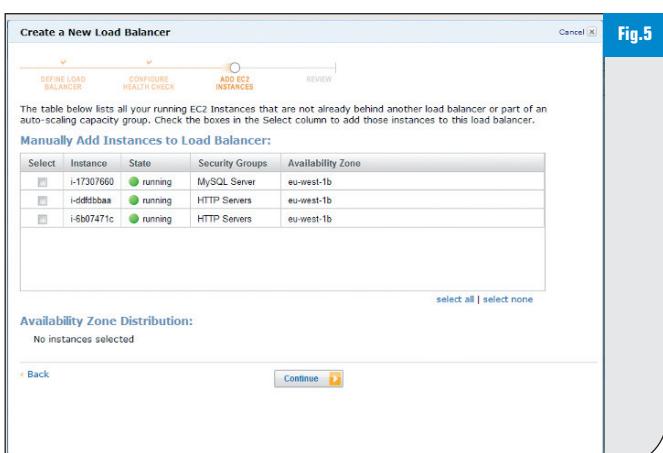


Fig.5

confirmation, nous allons en avoir besoin pour démarrer l'autoscaling [Fig.6].

Activer l'autoscaling

Pour activer l'autoscaling, nous devons utiliser l'API dédiée. Lors de l'écriture de cet article, la console AWS ne permet pas de faire cela depuis le web. Nous allons donc passer en ligne de commande.

Etape 1 : télécharger et installer l'API Autoscaling

La première étape consiste à télécharger l'API permettant d'effectuer l'Autoscaling. Il s'agit d'un fichier ZIP que vous pouvez décompresser. A l'intérieur du fichier ZIP, un fichier README.TXT contient les informations pour installer l'API sur son poste.

Note : lors de la procédure d'installation, vous allez avoir besoin d'une clef AWS pour vous connecter à votre compte. Vous pouvez trouver cette clef en vous connectant à la console AWS, puis dans la section **Account / Security Credentials**

Etape 2 : créer une configuration de lancement

Une fois les outils installés, nous allons créer une Launch configuration. Il s'agit d'une description de serveur qui sera créée automatiquement en cas de besoin. Une Launch configuration se crée par la ligne de commande suivante :

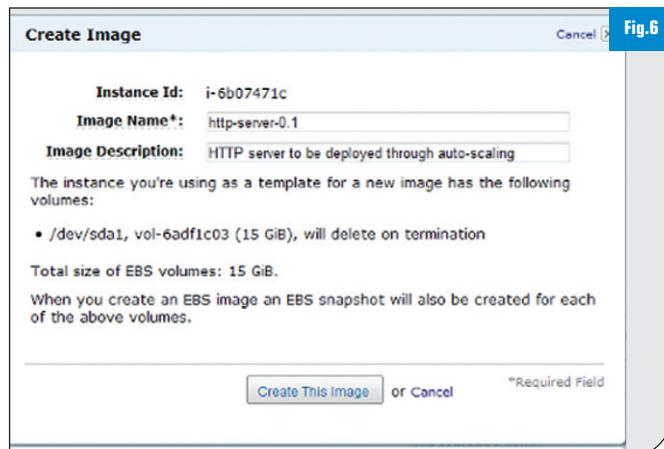
```
as-create-launch-config httpLaunch --image-id ami-8a95bffe --
instance-type m1.small --group "HTTP Servers" --region eu-west-1
OK-Created launch config
```

Etape 3 : créer un groupe d'autoscaling

Nous allons maintenant créer un groupe d'autoscaling. Ces groupes décrivent le nombre de machines minimales et maximales pouvant appartenir au groupe, la zone géographique, ainsi que les load-balancers auxquels les nouvelles machines doivent s'inscrire.

```
as-create-auto-scaling-group httpASG --launch-configuration
httpLaunch --availability-zones eu-west-1b --min-size 1 --max-
size 5 --load-balancers http-load-balancer --region eu-west-1
OK-Created AutoScalingGroup
```

Attention! Le load-balancer que vous avez précédemment créé est associé à une ou plusieurs zones géographiques (en fait, s'il contient 2 instances dans 2 zones géographiques, il est configuré pour ces 2 zones géographiques). Dans le groupe d'autoscaling que vous créez, vous ne devez pas indiquer de zone géographique



qui ne soit pas gérée par le load-balancer, autrement les instances créées ne seront pas utilisées!

Etape 4: créer un déclencheur

Nous avons défini la taille du groupe d'autoscaling, mais nous n'avons pas encore défini sous quels critères il faut rajouter des machines, et sous quels critères nous devons en retirer.

```
as-create-or-update-trigger httpTrigger --auto-scaling-group
httpASG --namespace "AWS/EC2" --dimensions "AutoScalingGroup
Name=httpASG" --measure CPUUtilization --statistic Average --
period 60 --breach-duration 120 --lower-threshold 30 --
lower-breach-increment"=-1" --upper-threshold 60 --upper-breach
-increment 1 --region eu-west-1
OK-Created/Updated trigger
```

Effectuer la maintenance

Il y a fort à parier que tôt ou tard, vous souhaiterez déployer une mise à jour. Sur le cloud, cela peut vite devenir plus complexe que sur un serveur simple, puisqu'il y a plusieurs machines à mettre à jour. Pour cela, 2 solutions. Soit votre application est capable de se mettre à jour automatiquement (c'est sûrement le scénario le moins probable), soit vous devrez redéployer tous les serveurs. Pour redéployer les serveurs, voici la marche à suivre.

Première étape : vider le groupe d'autoscaling

Nous allons commencer par supprimer le groupe d'autoscaling. Mais pour cela, nous devons d'abord supprimer toutes les instances à l'intérieur. Une manière simple de le faire est de réduire sa taille à 0 :

```
as-update-auto-scaling-group httpASG --min-size 0 --max-size
0 --region eu-west-1
OK-Updated AutoScalingGroup
```

Deuxième étape : recréer une machine virtuelle AMI

Il faut ensuite recréer une image de machine virtuelle AMI. Pour cette création, il vous suffit de reprendre les étapes indiquées plus haut dans cet article. Une fois la machine créée, il faut aussi recréer un groupe de lancement pour cette AMI. Dans l'exemple ci-dessous, nous supposons que ce nouveau groupe de lancement s'appelle "httpLaunch2".

Troisième étape : mettre à jour le groupe d'autoscaling

Une fois la machine créée, nous allons indiquer que le groupe d'autoscaling doit prendre en paramètre cette nouvelle machine, et nous allons lui redonner sa taille initiale (entre 1 et 5 machines).

```
as-update-auto-scaling-group httpASG --launch-configuration
httpLaunch2 --min-size 1 --max-size 5 --region eu-west-1
OK-Updated AutoScalingGroup
```

Tester!

Vous avez configuré l'Autoscaling aux petits oignons? Alors il est temps de tester la montée en charge! Pour cela, je vous conseille JMeter qui permet d'écrire facilement des tests de montée en charge. Enjoy !

■ David Negrier

Directeur technique - www.thecodingmachine.com

Refactoring : les meilleures pratiques

Tout au long du cycle de vie d'un logiciel, et plus particulièrement durant la phase initiale de développement, une question revient régulièrement : faut-il refactoriser le code ? La réponse, le plus souvent, n'est pas le fruit d'une réflexion éclairée et cohérente avec les véritables enjeux, mais plutôt une réaction instinctive visant à répondre à des problématiques à court terme. Pourtant, le refactoring est souvent la clé de la pérennité des applications...

Le terme de "refactoring" ne renvoie pas à une définition précise et universellement acceptée. Et ceci représente souvent le premier frein à son acceptation. En effet, on peut distinguer sans peine deux types de refactoring, n'ayant pas du tout ni les mêmes causes ni les mêmes conséquences.

Si le refactoring devrait toujours renvoyer à l'idée de factorisation au sens mathématique du terme, il n'est pas rare qu'il soit employé par abus de langage pour désigner la nécessité de réécrire du code trop vite conçu et implémenté. Autrement dit, le refactoring consiste à rationaliser des traî-

tements au sein du nombre le plus restreint possible de méthodes existantes ou à adapter, et non pas à réécrire tout ou partie d'une application.

Portée du refactoring

Une opération de refactoring, au sens strict, et telle que nous la considérerons dans cet article, doit se restreindre à un périmètre isofonctionnel, c'est-à-dire que l'application doit faire strictement la même chose avant et après le refactoring. S'il est question, en même temps, de modifier les fonctionnalités de l'application, il ne s'agit plus de refactoring, mais de refonte, ce qui sous-entend le début d'un nouveau cycle complet de conception/réalisation. Schématiquement, le refactoring d'une application 1.0 doit conduire à une version de maintenance 1.0.1, pas 1.1 et encore moins 2.0.

Il est extrêmement important de bien comprendre la différence entre ces deux objectifs, car la charge de travail en est proportionnelle. Or dans la plupart des situations, le plus grand ennemi du refactoring, ce sont les délais. Et ils sont infiniment plus longs dans le cas de l'intégration de nouvelles fonctionnalités que dans le cas d'un pur refactoring isofonctionnel. Et pour cause ! Dans ce dernier cas, les spécifications des fonctionnalités concernées sont déjà connues. L'opération de refactoring consiste à réorganiser le code de sorte à le minimiser, en rassemblant tous les traitements identiques et/ou similaires (selon leur degré de similitude) dans un jeu de méthodes restreint, plus générique et plus souple que dans la version précédente.

Quand procéder au refactoring ?

Parmi les erreurs courantes liées à la pratique du refactoring, la procrastination n'est pas la moins fréquente. Cette fâcheuse tendance que l'on a parfois à remettre une tâche au lendemain, quand ce n'est pas à jamais, conduit souvent à repousser les phases de refactoring le plus loin possible dans le temps.

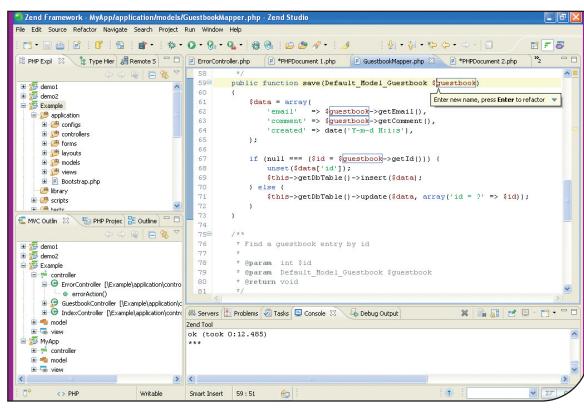
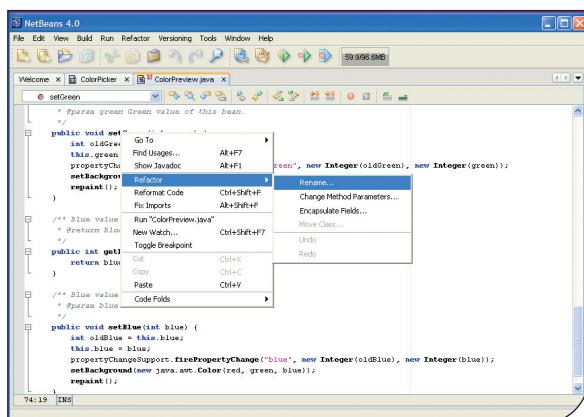
Typiquement, on envisage rarement de refactoriser avant qu'une première version publique de l'application n'ait été livrée.

En effet, avant qu'une première version fonctionnelle ne voit le jour, on se considère dans le « rush », et on admet l'idée qu'il vaut mieux foncer tête baissée, quitte à négliger certaines portions du code, en se disant que l'on reverra ça une fois la première version livrée.

Voilà comment on se retrouve avec des applications truffées « TODO » et autres « A CORRIGER !! », lesquels ne disparaissent en général jamais.

Au mieux, le code sera effectivement réécrit, les méthodes renommées, mais le code obsolète sera conservé...

Il est bien évident que psychologiquement, il n'est pas toujours facile quand on est dans l'urgence de se dire qu'il faut sacrifier une partie du temps imparti au projet pour réécrire ce qui vient tout juste d'être créé. C'est pourtant dans ces conditions qu'il est très important de procéder dès que possible aux phases de refactoring, qui n'en seront que plus courtes et plus sûres. La rigueur et la qualité du code ainsi obtenus sont les meilleurs atouts pour tenir des délais serrés.



Si on fait une analogie avec la construction, on peut dire qu'il est bien plus facile de desserrer une brique mal alignée dès qu'on s'en rend compte, plutôt que d'attendre que dix rangs supplémentaires aient été maçonnés par-dessus. Dans le premier cas, ça va très vite, et on s'épargne le risque de s'aligner sur un élément de travers. Dans le second, le travail de déconstruction est bien plus important, engendre beaucoup plus de conséquences potentielles et représente infiniment plus de travail ! Il se passe strictement la même chose dans le cas du refactoring. Si l'on tarde trop, l'ampleur de la tâche devient telle qu'on y renonce. Un processus entropique croissant s'installe alors, qui conduit inévitablement à une dégradation de plus en plus importante de la qualité générale du code. Finalement, la maintenance devient quasi impossible, et il ne reste alors plus que deux solutions : abandonner l'application ou la réécrire intégralement.

Les tests unitaires : compagnons indispensables du refactoring

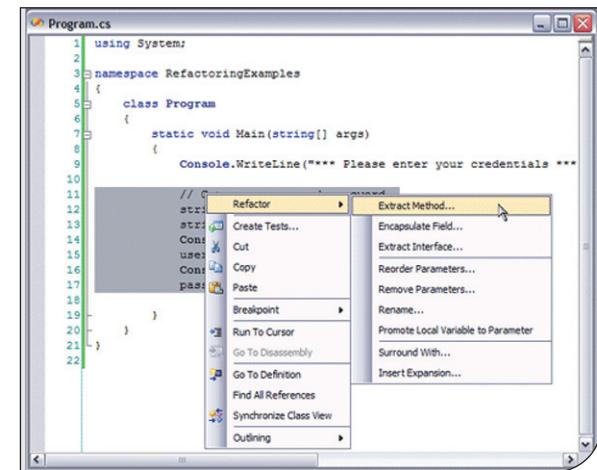
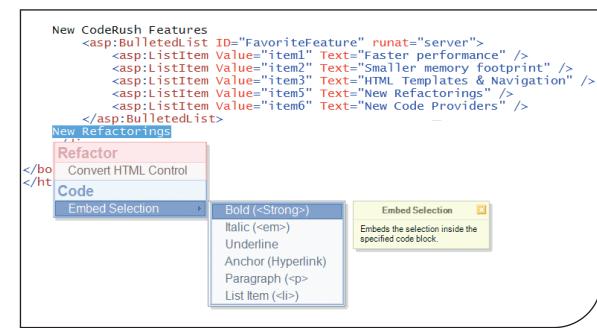
Bien entendu, vu sous cet angle, on se demande bien pourquoi toutes les équipes de développement du monde ne s'adonnent pas comme et quand il faut aux joies de la refactoring. Il y

a une raison simple à ce phénomène : les gains de productivité et de qualité qui sont promis ne sont pas « gratuits ». Pour que la refactorisation soit efficace, il est absolument nécessaire de l'encadrer par une procédure de tests automatisée, pour ne pas parler de tests unitaires (voir encadré)... Ainsi que nous l'avons rappelé précédemment, une opération de refactoring doit être neutre du point de vue des fonctionnalités de l'ensemble du système.

Après avoir modifié une partie du code de l'application, on doit donc être en mesure de garantir que les modifications n'ont entraîné aucune régression d'un point de vue fonctionnel, et ce quelle que soit la portée de l'impact de ces modifications.

Si l'on doit après chaque cycle de refactoring tester manuellement l'ensemble de l'application pour s'assurer qu'aucun effet de bord n'a été induit, beaucoup du bénéfice (notamment du bénéfice immédiat) du processus est perdu : non seulement le temps que l'on doit investir est démultiplié, mais en plus les risques d'introduction de nouveaux bugs sont considérables.

Une méthode de test automatisée est la seule solution pour que le refactoring ne tourne pas au cauchemar. Une fois les tests écrits, les exécuter permet de valider que le code effectue bien les traitements attendus. S'ils passent avant le refactoring, ils



doivent passer aussi après. Si c'est le cas, aucun autre test (manuel) n'est nécessaire.

Conclusion

Si toutes les conditions sont respectées – notamment un timing précoce et le recours à une méthode de test automatisée de l'application – un cycle de refactoring est non seulement court (cela peut ne prendre que de quelques minutes parfois à quelques heures au maximum) mais également sûr. Les pratiquer régulièrement tout au long du cycle de vie de l'application est un bon moyen de gagner du temps sur la durée totale de développement, et plus particulièrement lors des phases de maintenance fonctionnelle. S'en priver, à l'inverse, compromet significativement les possibilités d'évolution de l'application, de même que sa stabilité et sa cohérence. Aussi la décision de ne pas procéder au refactoring de l'application dont on a la responsabilité doit être prise en toute conscience, et ne peut se justifier, et encore, que dans le cas d'un développement événementiel n'ayant pas vocation à être pérennisé.

Tests automatisés et tests unitaires

On oppose bien souvent test unitaire à test manuel. Ce qui n'est pas tout à fait exact. Ou plus précisément, cette opposition est limitative, et empêche parfois les développeurs qui n'en sont pas coutumiers de bien comprendre ce que sont les tests unitaires. En réalité, il faut opposer les tests manuels aux tests automatisés.

Les tests automatisés permettent de décrire des scénarios de test sous forme de code utilisant le code de l'application. Dans ces scénarios, on instancie des classes, on en appelle des méthodes, et on définit les valeurs attendues en retour. Lors de l'exécution des tests, les retours réels sont comparés aux valeurs attendues, ce qui permet de valider ou non le bon fonctionnement du code de l'application.

Ce principe est poussé « à l'extrême » dans le cas des tests unitaires, qui prévoient que chaque portion du code testé ne dépende d'aucun autre élément de l'application (et surtout pas la base de données !). On peut donc les considérer comme les tests automatisés les plus aboutis possible, en notant bien que, même si l'on ne se conforme pas au concept de test unitaire, on utilisera le plus souvent un framework de tests unitaires pour implémenter un mécanisme de test automatisé. Ce qui ajoute naturellement à la confusion...

■ Gauthier Delamarre

Les 10 commandements des refactoring

Cet article présente les 10 commandements à respecter afin de mettre en place un processus de refactoring efficace au sein d'un projet informatique.

Connue de tous aujourd'hui, la notion de refactoring a été popularisée par Ward Cunningham et Kent Beck au cours des années 80 lorsqu'ils travaillaient avec le langage Smalltalk. Ce même Kent Beck en fit un élément central de la méthodologie agile Extreme Programming qu'il créa à la fin des années 90. Plus formellement, le refactoring est le processus qui consiste à changer la structure interne d'un logiciel, pour favoriser sa maintenabilité et son extensibilité, de telle sorte que son comportement externe reste identique. Pour être rentable, un logiciel doit vieillir en étant capable d'évoluer au cours du temps. Cette évolution s'accompagne d'une érosion du design initial qui conduit irrémédiablement à une qualité déclinante et donc à une maintenance complexifiée qui aura un coût supplémentaire. On comprend ici pourquoi la maintenance est la phase la plus critique dans le cycle de vie d'un logiciel. Le refactoring vient adresser cette problématique en permettant de maintenir le design du logiciel à un certain niveau de qualité mais également de venir rectifier un design initial moyen. Il s'agit d'un processus continu qui n'est pas anodin mais dont les bénéfices se mesurent sur le long terme au cours de la vie du logiciel. Très lié aux méthodes agiles, il peut (et doit) être mis en œuvre dans des démarches projets plus classiques. Pour être efficace, la mise en place d'un processus de refactoring doit respecter certaines règles qui relèvent bien souvent du bon sens mais qu'il est toujours bon de rappeler. Leur énumération est faite dans cet article sous la forme de 10 commandements à respecter pour réaliser un refactoring efficace.

1 TU TESTERAS LE CODE

Hantise de certains développeurs qui les voient encore trop souvent comme une contrainte plutôt qu'une

méthode de travail permettant un gain de temps notable et une meilleure compréhension de ce que doit faire le code, les tests unitaires constituent le pré-requis essentiel au refactoring. En effet, avant toute opération de refactoring il est nécessaire que la base de code à refactorer soit stabilisée et couverte de manière intelligente par des tests unitaires. Pour cela, des outils permettant de mesurer le pourcentage de couverture de code existent. On citera notamment les projets Cobertura et EMMA dans le monde Java. Les tests unitaires permettent de garantir que le comportement du code que l'on va refactorer est conforme à ses spécifications. Ainsi, il devient aisément après chaque opération de refactoring de vérifier que le code concerné a toujours le même comportement extérieur malgré les changements structurels qu'il a subis. Les tests écrits doivent permettre d'identifier tout de suite d'où vient le problème lorsqu'il y a échec d'une assertion afin de gagner un temps précieux dans la résolution d'éventuelles régressions apparues après des opérations de refactoring. Au cours du processus de refactoring, les tests unitaires ne doivent pas subir de modifications si ce n'est pour rajouter des cas de tests qui n'auraient pas été prévus initialement. Enfin, il est bon de noter que la pratique du développement piloté par les tests (TDD) facilitera grandement la mise en place d'un terrain propice au refactoring.

2 TU IMPLIQUERAS LE MANAGEMENT

Au sein d'un projet informatique, le manager a bien souvent une vision différente de l'équipe de développement pour peu qu'il n'ait pas un background technique derrière lui. Ainsi pour le management, le court terme fait loi et il privilégiera l'ajout de fonctionnalités visibles par l'utilisateur à une phase de refactoring qui aura une

plus-value plutôt abstraite à ses yeux. La notion de cette technique, métaphore lancée par Ward Cunningham dans les années 90, ne fait sûrement même pas partie de ses connaissances ... Dans ce cas de figure et afin d'instaurer un processus de refactoring efficace, tout l'enjeu va consister à convaincre le management de l'intérêt majeur que représente le refactoring pour son projet. Deux points d'entrée peuvent être envisagés pour convaincre le management. Le premier consiste à partir sur le domaine de la qualité et à montrer que cette dernière ne peut qu'être améliorée par un processus de refactoring continu. En effet, refactorer du code afin d'améliorer sa structure interne aura un effet bénéfique sur la qualité du code produit et permettra sûrement d'améliorer la satisfaction du client sur le produit final. Le second point va concerner les effets induits par le refactoring. La revue de code est au cœur du refactoring et il est prouvé que celle-ci favorise la découverte de bugs. Découvrir ces derniers au plus tôt permet d'améliorer le logiciel plus rapidement et de livrer un meilleur produit au client.

Enfin, refactorer un code permet une meilleure compréhension de celui-ci et va donc permettre d'ajouter de nouvelles fonctionnalités au logiciel plus rapidement par la suite. Des développements plus rapides grâce au refactoring, un argument à même de convaincre les managers les plus récalcitrants !

3 TU MOTIVERAS L'ÉQUIPE DE DÉVELOPPEMENT

Repasser sur le code d'un autre développeur, voilà bien quelque chose que les développeurs détestent faire. Et encore plus lorsque celui-ci est mal écrit ! Ce syndrome touche plus les développeurs débutants que les développeurs expérimentés cependant.

Ces derniers arrivent mieux à percevoir, de par leur expérience, que la valeur d'un programme se mesure plus sur le long terme et que certaines opérations de refactoring bien que contraignantes et peu gratifiantes sur le court terme apporteraient un réel plus pour le logiciel dans le futur. Ces derniers doivent donc jouer un rôle important pour motiver des développeurs moins chevronnés lors de phases de refactoring. En outre, il est important de bien faire comprendre l'intérêt du refactoring aux équipes de développement pour les impliquer et les faire adhérer à la démarche plutôt que de tenter d'imposer ce refactoring. On sait bien qu'un développeur non motivé ne fera rien de bon et au final l'effort de refactoring n'apporterait que peu de bénéfices dans un tel contexte.

4 TU T'INTÉGRERAS AU SEIN DU CYCLE DE VIE LOGICIEL

Le refactoring est un processus continu qui doit de fait être intégré au sein du cycle de vie logiciel. Ainsi sur des projets agiles, il peut être réaliste de placer 2 jours de refactoring à la fin de chaque sprint de 4 à 6 semaines. La fin des sprints étant liée à une phase de revue de code qui fera apparaître des besoins en refactoring. Une fois l'habitude de refactoring prise et intégrée dans le cycle de vie logiciel, les effets bénéfiques se feront vite ressentir et la structure du logiciel sera maintenue à un niveau de qualité suffisant pour sa pérennité sur le long terme.

Choisir le bon moment pour le refactoring est partie intégrante de son succès sur un projet. On prendra donc garde à ne pas placer une phase de refactoring juste avant une livraison client par exemple. Certains chantres du refactoring tels Martin Fowler préconisent quant à eux de ne pas prévoir de moment précis pour le refactoring mais plutôt de le faire au fur et à mesure quand celui-ci s'impose naturellement. L'ajout de nouvelles fonctionnalités sur le logiciel peut être un moment propice si l'on s'aperçoit qu'un refactoring facilitera son déve-

loppe. Quelle que soit l'approche retenue, il est primordial que le refactoring soit perçu et mis en place comme un processus continu et non comme une opération réalisée une fois par an par exemple. Le travail dans ce cas-là serait bien trop lourd et les résultats forcément décevants.

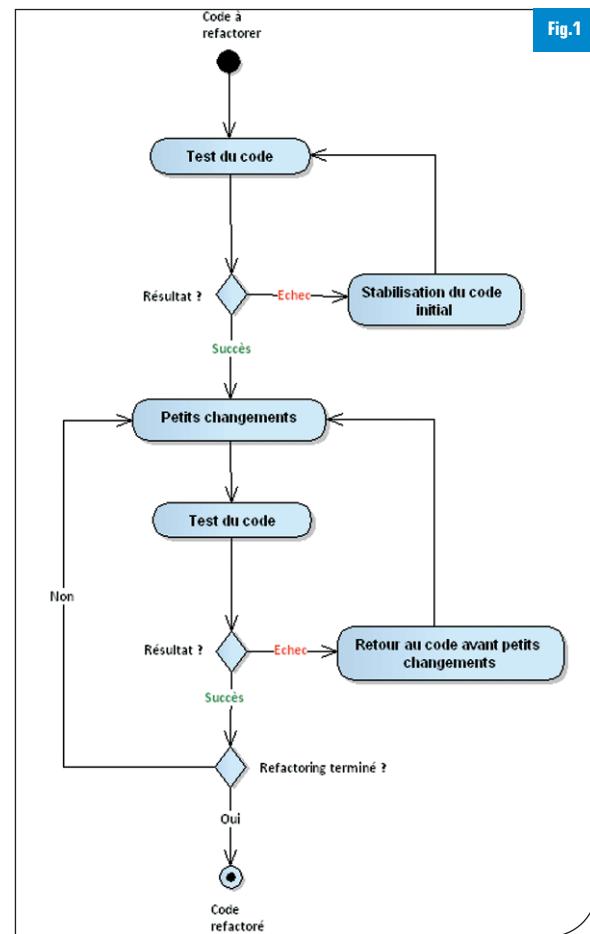
5 TU REFACTORERAS PAS À PAS

Le processus de refactoring doit s'effectuer pas à pas pour être optimal. Une fois la base de code à refacturer rendue stable et vérifiée par des tests unitaires joués avec succès, les opérations de refactoring peuvent commencer. Pour cela, on effectue des petits changements et au fur et à mesure de ces changements on va relancer les tests pour vérifier que le code n'a pas subi de régressions et que l'on n'a pas introduit de nouveaux bugs dans le code. Le refactoring est donc une technique fortement itérative comme le montre la [Fig.1]. Le mode pas à pas que l'on adopte en faisant du refactoring va garantir un changement rapide et sûr du code.

6 TU DÉFINIRAS UNE STRATÉGIE

Un programme simple et mal conçu n'est pas un problème en soi s'il ne doit pas évoluer. En effet, il ne sert à rien de refacturer pour refacturer. Le refactoring doit être un processus maîtrisé et nécessite donc la mise en place d'une stratégie afin d'identifier clairement ce que l'on va faire et donc ce que l'on attend du refactoring sur son projet. La définition d'une stratégie de refactoring consiste ainsi à décider ce que l'on va modifier. Les modifications du code peuvent intervenir à différents niveaux :

- **Présentation** : c'est le niveau le plus faible qui consiste à rajouter des commentaires dans le code, modifier son indentation, etc ...
- **Lisibilité du code** : on arrive déjà à un niveau plus élevé puisqu'ici il s'agira de détecter et de corriger ces fameux "code smells" qui bien souvent peuvent cacher des problèmes plus profonds.
- **Algorithmique** : ici, on s'intéresse à la refonte de certains algorithmes



qui n'auraient pas été écrits de manière suffisamment claire par exemple.

Processus de refactoring

- **Design** : c'est le niveau le plus complexe puisqu'on touche à l'architecture du code mais il va permettre d'améliorer son extensibilité s'il est réalisé correctement.

Ces différents niveaux ont été classés par ordre croissant de complexité et bien souvent, il est préférable de commencer à un niveau plus bas pour y faire le ménage, ce qui permettra d'avoir une meilleure lisibilité du code et de faire apparaître plus clairement les modifications de design devant être éventuellement apportées. La stratégie mise en place devra clairement définir le moment le plus opportun pour effectuer le refactoring du code. Comme expliqué précédemment, l'ajout d'une nouvelle fonctionnalité ou la correction d'une anomalie peuvent être des moments opportuns pour le refactoring de code.

Les revues de code permettant de faire apparaître les zones de code à modifier dans le cadre du refactoring,

la technique de programmation en binôme (Pair Programming) mise en avant par l'Extreme Programming sera bénéfique puisque le "copilote" (celui qui observe le code) fait ainsi une relecture constante du code et apporte un point de vue extérieur à celui du "driver" (celui qui développe).

7 TU UTILISERAS DE BONS Outils

L'outilage utilisé par le développeur est un élément clé pour la réussite du refactoring. Les IDE modernes offrent de grandes possibilités de ce point de vue là et il est donc essentiel de s'appuyer sur eux. En prenant l'exemple d'Eclipse Helios pour le développement en Java, on voit que l'outil met à disposition du développeur plus d'une vingtaine de fonctionnalités de refactoring venant corriger des problèmes allant de la présentation jusqu'au design [Fig.2]. Ces fonctionnalités s'accompagnent d'un historique des modifications permettant un retour arrière en cas d'échecs dans les tests unitaires. En cela, il entre parfaitement dans le processus de refactoring décrit au point 5.

L'utilisation d'un bon IDE comme Eclipse sera complétée par des outils de détection d'un certain nombre de problèmes dans le code qui nécessiteront un refactoring. L'écosystème Java regorge d'outils de qualité pour cela. Ainsi, l'utilisation d'un outil comme PMD permettra de détecter une complexité cyclomatique trop importante signe qu'il faut refactorer en faisant du polymorphisme. De même, un outil comme CPD recensera les morceaux de code pouvant être dupliqués. Enfin, l'utilisation de CheckStyle et de FindBugs donnera des indications précieuses sur des refactorings résolvant des "codes smells" ou des possibles bugs tels la présence de classes ayant un nombre de lignes de code trop élevé ou ayant des méthodes trop longues, utilisation d'une variable affectée à null, etc ... L'automatisation de ces outils sera un plus énorme dans la démarche de refactoring. Pour ce faire en Java, on pourra s'appuyer sur la pratique d'intégration continue avec par exemple le serveur Hudson. Un projet basé sur Maven

Fig.2		
Structure du code	Structure au niveau de la classe	Structure à l'intérieur de la classe
✓ Renommer	✓ Descendre vers une sous-classe	✓ Inliner
✓ Déplacer	✓ Remonter vers une super-classe	✓ Extraire la méthode
✓ Changer la signature d'une méthode	✓ Extraire une super-classe	✓ Extraire la variable locale
✓ Convertir une classe anonyme en classe imbriquée	✓ Extraire une interface	✓ Extraire une constante
✓ Convertir un type imbriqué au niveau supérieur	✓ Généraliser le type déclaré	✓ Introduire un paramètre
✓ Déplacer un type membre dans un nouveau fichier	✓ Utiliser le supertype si possible	✓ Introduire un paramètre en objet
		✓ Introduire une Factory
		✓ Introduire une indirection
		✓ Infère les arguments d'un type générique
		✓ Encapsuler la zone

Fonctionnalités de refactoring de Eclipse Helios

possède ainsi une intégration poussée avec les différents outils cités au paragraphe précédent. Les rapports produits par ces outils mettront en évidence les violations de certaines règles et permettront donc de mieux repérer les parties de code nécessitant un refactoring.

8 TU T'APPUIERAS SUR LES DESIGNS PATTERNS

Les designs patterns constituent un ensemble de solutions génériques répondant, via le paradigme objet, à des problèmes récurrents rencontrés dans le développement logiciel. Ils regroupent donc un ensemble de bonnes pratiques et de standards du domaine de l'industrie informatique. Les utiliser à bon escient s'avère crucial pour avoir un logiciel réutilisable et extensible. Dans le cadre du refactoring, ils constituent cependant des techniques complexes qui remanient

le code du logiciel en profondeur quelquefois. Leur utilisation doit donc être faite en mesurant les gains apportés et surtout les risques engendrés à les implémenter en refactorant l'existant. La littérature étant assez fournie sur le sujet, on ne s'attardera pas trop ici sur les différents designs patterns et leur formalisation. Ici, on s'intéresse plutôt sur les moyens permettant de détecter dans le code qu'un pattern précis pourrait être employé [Fig.3].

9 TU CONNAÎTRAS L'ÉTAT DE L'ART

Connaître l'état de l'art en matière de bonnes pratiques de développement amène des bénéfices importants parmi lesquels un refactoring plus efficace. En effet, en suivant les bonnes pratiques du domaine, le développeur favorisera le refactoring futur du logiciel en améliorant la maintenabilité et l'extensibilité du logiciel. Cette connaissance de l'état de l'art est favorisée par la veille technologique et doit s'appuyer sur des ouvrages référence en la matière. On citera ainsi le livre de Joshua Bloch "Java Efficace" qui ouvre les yeux au développeur sur un certain nombre de points importants à considérer dans le développement logiciel en Java. Ensuite, le livre de Martin Fowler sur le refactoring semble incontournable puisqu'il référence sous forme de catalogues un ensemble de bonnes pratiques de refactoring pouvant être appliquées.

Ce catalogue de bonnes pratiques existe en version internet à l'adresse suivante : <http://www.refactoring.com/>

Fig.3	
Design Pattern	Moyen de détection
Builder	Construction d'objets n'est pas visible et fluide
Facade	Couplage trop fort entre différents sous-systèmes
Observer	Classes ayant un couplage différent trop important
Proxy	Utilisation d'objets distribués
State	Complexité cyclomatique exagérée dans un code devant gérer des états différents
Strategy	Aucune séparation entre différents traitements pouvant être appliqués sur un même contexte
Visitor	Séparation inexistante entre traitements et données

Moyens de détection pour le refactoring par designs patterns

<catalog/index.html>. Les pratiques listées vont des rappels sur le nommage des variables et méthodes qui doit être explicite à des conseils sur l'utilisation de la délégation plutôt que l'héritage quand une classe n'utilise pas de données de sa classe mère actuelle. Elle passe aussi par l'utilisation du polymorphisme afin de remplacer élégamment des switchs et d'éviter l'utilisation du mot-clé *instanceof* en Java.

10 TU PRENDRAIS GARDE À CERTAINS PIÈGES

Jusqu'ici, cet article aura plutôt présenté les aspects positifs du refactoring mais il existe également un certain nombre de pièges dans sa mise en œuvre qu'il est bon de connaître. Avant tout refactoring en profondeur, il est nécessaire d'évaluer les gains apportés en regard des risques que cela comporte. Modifier en profondeur le design d'un composant central du logiciel peut quelquefois être plus pénalisant qu'autre

chose. En outre, il peut arriver qu'on sente qu'une partie du code nécessite du refactoring mais qu'on ne trouve pas de solution qui soit satisfaisante. Dans ce cas-là, il est préférable de ne pas refactorer à tout prix et d'attendre d'y voir plus clair jusqu'à une prochaine phase de refactoring. D'autre part, il faut prendre garde aux petites opérations de refactoring anodines en apparence et qui peuvent avoir des effets bien plus vastes. On pense ici à la modification du nom d'une méthode d'une interface publique d'une API qui est distribuée, par exemple. Chaque modification doit donc être réfléchie en amont afin d'éviter des effets négatifs inattendus en retour. Enfin, un bon équilibre doit être trouvé au niveau de la lisibilité du code afin d'éviter de sacrifier la performance du logiciel au profit d'un petit surplus de compréhension dans le code.

Conclusion

Cet article aura mis en avant 10 commandements pour mettre en place un processus de refactoring efficace.

Bien que les exemples d'outils cités concernaient le monde Java, le processus décrit ici est assez général et s'applique quel que soit le langage utilisé. A la lecture de cet article, certains lecteurs seront tentés de se demander : que faire avec du code existant non testé ? Cette question est effectivement délicate et elle n'a pu être abordée ici. Cependant, une piste simple pour démarrer consiste à mettre en place le processus de refactoring décrit ici en partant des évolutions et des ajouts de nouvelles fonctionnalités réalisés sur le logiciel pour mieux comprendre l'existant auquel on doit s'interfacer. Petit à petit, le refactoring pourra ainsi produire ses effets et se propager au code existant. Enfin, il est évident que le refactoring est une technique nécessitant de trouver le juste équilibre entre effets positifs et risques au sein de laquelle le pragmatisme doit rester roi.

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



Communauté Académique Intel

intel

Enseignez le développement parallèle avec nos ressources.

Avec l'essor des ordinateurs multi-coeurs, le développement parallèle est devenu une évidence et une évolution nécessaire.

Ces changements vont complètement bouleverser nos habitudes et nos méthodes de travail.

Faites partie de cette révolution :

- Téléchargez gratuitement des outils pédagogiques spécialisés pour le développement parallèle
- Intégrez le parallélisme dans votre programme grâce à l'ensemble des modules d'Intel et de sa communauté académique dans le monde.
La sélection s'étoffe de jour en jour et propose des études de cas, des démos, des conférences des brochures et des vidéos
- **LISTE DES COURS DISPONIBLES :**
 - Génie logiciel
 - Informatique visuelle
 - Techniques & base de programmation
 - Architecture et organisation
 - Algorithme & complexité
- Accédez à distance à l'une des machines les plus puissantes du marché disposant de 32 coeurs

Retrouvez nous sur :

<http://software.intel.com/fr-fr/academic> &
<http://software.intel.com/fr-fr/>

Swelen : le défi de l'information

Aujourd'hui, la grande majorité du contenu présent sur Internet est du contenu généré par ses utilisateurs. Tranches de vies, blogs, vidéos personnelles et encyclopédie éditée par tout le monde. Ce contenu croît à une vitesse phénoménale, il est d'une spécificité extrême, d'une qualité éditoriale de plus en plus variable, et d'une durée de vie de plus en plus courte.

Les outils de recherche et de tri d'information qui ont fait le succès d'Internet sont dépassés par ce phénomène monstrueux que connaît l'information et, bien que parmi les tenants majeurs du tri de l'information, peu encore l'admettent, ces outils sont de plus en plus inadaptés à l'évolution de ce phénomène. Pour comprendre cela, il faut revenir une décennie en arrière, et considérer la situation qui présidait à l'avènement de ces outils.

Les évolutions

Nous sommes passés d'un modèle de diffusion « broadcast » à un modèle « pair à pair » (Peer to Peer). Ainsi, l'information n'émane plus de quelques sources vers de multiples récipients, mais coule au gré de ses besoins, d'agent à agent, chacun étant à la fois producteur ET consommateur de l'information.

Parallèlement à cela, le typage de l'information est passé de celui de publication à celui d'expression. Cette information n'est plus générée dans une démarche rédactionnelle par des spécialistes, mais au gré des envies et aspirations de chacun, pour la plupart d'une manière automatique ayant plus trait à la verbalisation qu'à l'expression. Ainsi, nous communiquons pour communiquer, par extension naturelle de tous nos faits et gestes, plus que dans le but d'apporter une information spécifique à un point précis.

La typologie et la classe d'information ont explosé ces dernières années. L'avènement des réseaux sociaux, blogs et autres sites de publication rapide et simplifiée a ouvert l'espace des champs lexicaux, de ceux du discours à ceux de l'information périphérique. « Aujourd'hui, je porte des chaussettes rouges. », « A midi j'ai

repris de la salade. », « Le collier du chat de Gisèle amuse Martin. » etc. Cette information est fondamentalement différente de celle qu'on avait l'habitude de manipuler jusqu'à présent, et elle pose de véritables challenges de par sa quantité et son rapport signal/bruit.

En conjonction avec cette explosion de l'information en fragments épars, apparaît l'avènement d'un aspect révolutionnaire, qui est la méta-information. S'imposant naturellement dans un monde de petites entités éparses, la méta-information est l'information qui concerne l'information. Elle constitue une forme de classification, de mise en relation par le plus généralement des mots-clés (tags), ou bien de l'information contextuelle (géo-localisation, date, format) ou qualitative (votes, notations, pose de drapeaux). Cette méta-information est une grande révolution en elle-même, car elle change littéralement la potentialité des agrégats d'information en les structurant. Des arborescences et articulations en émergent, et l'information devient protéiforme. Cette protéiformité de l'information elle-même est la dernière composante de cette révolution. D'éléments unitaires classés et construits les uns sur les autres comme des pyramides de savoir, on est passé à des entités-outils d'informations changeantes et actives, formées de la complexité extrême de milliers de minuscules composantes liées entre-elles par l'ossature méta-informative, qui par la forme en donne la fonction.

Lorsqu'on saisit ce profond changement et les possibilités générées par cette nouvelle forme de savoir protéiforme, bruité et ubique, on comprend aisément pourquoi les outils d'aujourd'hui y sont totalement non préparés. Notre objectif est d'inventer et de

développer les nouveaux outils qui permettront de manipuler, trier, organiser et personnaliser cette connaissance protéiforme. C'est une importante mission dont nous nous sommes investis, l'enjeu principal est de fournir à chacun des outils puissants et adaptés à cette réorganisation perpétuelle de l'information.

Travaux de Recherche

Depuis ces 3 dernières années, nous effectuons des travaux de recherche sur un algorithme que nous avons baptisé Swelen Convergence. Il s'agit d'un outil d'analyse comportementale de corpus textuels qui va nous permettre de donner sens à un groupement d'informations méta-catégorisées. Cet outil a trois composantes principales :

1. L'évaluation des groupements d'information dans un espace virtuel. Ce groupement permet de reconnaître quels centres d'intérêts principaux émergent d'un corpus d'information touffu et changeant, et de recomposer à partir de ses fragments les plus représentatifs une description élaborée de ces centres d'intérêts.
2. L'association de chaque individu à ces centres d'intérêts principaux, dynamiquement, et en fonction de ses besoins estimés.
3. La sortie pour tout besoin spécifique d'un flux d'information personnalisé issu des informations groupées, en fonction des besoins identifiés pour les individus.

La principale force de cet outil est sa grande générativité et sa puissance. Il apporte une solution à la problématique de l'information protéique.

Nous travaillons aussi sur des méthodes de prédiction, de classification et d'organisation de l'information en fonction de l'Intelligence Collective des utilisateurs, des mots-clés saisis

lors de leurs recherches sur Internet, et des principaux indicateurs de satisfaction utilisateur. [Fig.1]

Etat d'avancement

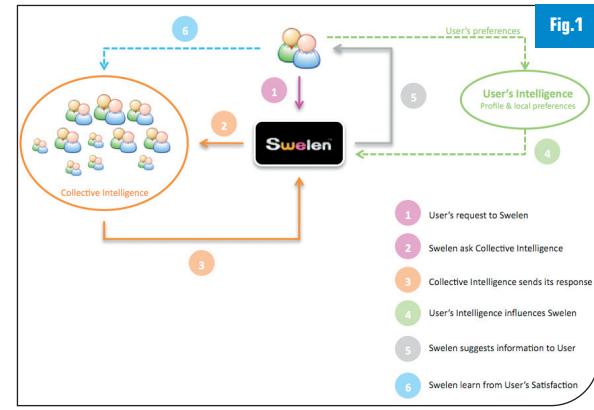
Bien qu'encore embryonnaire, Swelen Convergence est au cœur de l'architecture de nos produits. Un des premiers produits que nous avons développé est Swelen Suggest, un système de recommandation d'informations pour les éditeurs de sites internet. Suggest applique nos algorithmes de convergence au problème de la rétention du trafic internet, en proposant aux internautes d'un site, sur chacune des pages, une aide personnalisée et dynamique à la navigation, basée sur le corpus textuel des pages et sur les regroupements de comportements utilisateurs les plus proches du visiteur. L'outil est toujours en phase de bêta test auprès de clients grands comptes. [Fig.2]

Résultat : un nouvel indicateur de satisfaction des internautes

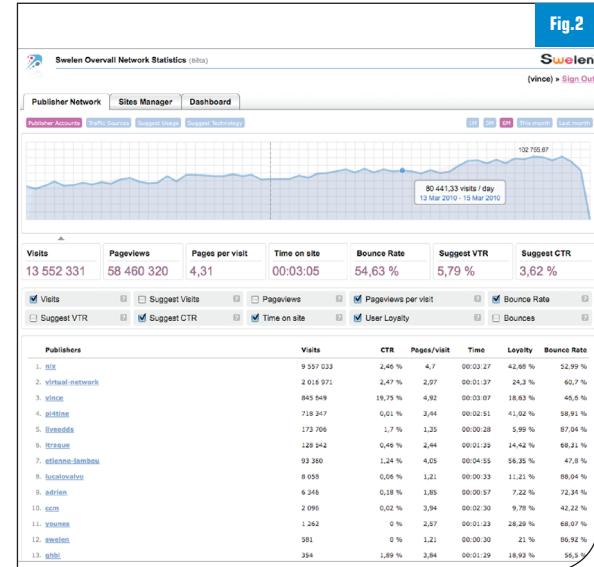
Avec Suggest, Swelen a relevé son pari de développer un système de recommandation basé sur l'étude comportementale des internautes lorsqu'ils sont confrontés à du contenu. Le système de filtrage collaboratif que nous proposons n'a pas besoin de la participation active des internautes. Les prédictions sont générées grâce à l'attribution automatique d'indices de satisfaction en fonction de leurs actions. La technologie est scalable, agnostique, et fonctionne sur tout type de contenu online. Lors de nos recherches sur l'Intelligence Collective, nous étions confrontés à un problème majeur : le plus souvent, les internautes ne votent pas et ils ne participent pas explicitement au système de notation. En effet, dans un système de recommandation, il est essentiel de recueillir les notes des internautes afin de pouvoir déterminer leur satisfaction (ou leur insatisfaction). Nous avons trouvé une solution élégante à cette problématique : les algorithmes se chargent eux mêmes d'attribuer des notes à la place des internautes. Nous effectuons cette attribution de notes grâce

à l'analyse en temps réel effectuée par l'indicateur de performance de visite, le VPI, détaillé plus loin dans ce document. De fait, nous n'avons pas besoin de leur participation active (les internautes n'ont pas besoin de voter ou noter les contenus que nous leur soumettons). Le système analyse automatiquement leur comportement pour déterminer s'ils sont satisfaits ou non. L'idée est de prédire le degré de satisfaction d'un internaute lorsqu'on lui soumet un contenu (une page web par exemple, ou une annonce publicitaire). De fait, pour chaque contenu et pour chaque internaute, nous attribuons un indice de satisfaction, comme si l'internaute avait voté explicitement, avec des étoiles par exemple. Le calcul de cet indice de satisfaction s'effectue en pondérant les taux de clics avec plusieurs dizaines d'autres facteurs que nous souhaitons garder strictement confidentiels.

Par exemple, nous considérons qu'un clic sur une annonce publicitaire est un vote pour cette annonce. Ce vote peut être négatif ou positif. Si après avoir cliqué sur l'annonce, une fois arrivé sur le site de l'annonceur, un internaute ferme directement le site, nous considérons qu'il a effectué un vote très négatif pour le site. Si en revanche l'internaute reste plusieurs minutes et visite plusieurs pages, nous considérons le vote comme étant un vote très positif. Pour chaque contenu, et pour chaque internaute, nous attribuons ainsi des notes qui représentent le degré de satisfaction et d'engagement de l'internaute. L'innovation est que nous avons réuni les technologies conventionnelles de mesure d'audience avec celles des Collaborative Filtering Systems (systèmes de filtrage collaboratif). Nous pensons que ce procédé peut venir enrichir considérablement l'industrie de l'information, et notamment l'industrie de la publicité en ligne. En utilisant l'analyse du comportement des internautes directement dans les processus de décision des algorithmes de gestion de l'information, nous pensons pouvoir augmenter la satisfaction des internautes lorsqu'ils sont confrontés à des conte-



Filtre à Intelligence Collective de Swelen



nus. Pour pouvoir observer et analyser ce nouveau phénomène, nous avons défini de nouvelles métriques, et nous avons conçu une solution de mesure d'audience à Intelligence Collective, Swelen Medusa, offerte gratuitement à la communauté des développeurs et des webmasters de sites Internet. [Fig.3]

Swelen Overall Network Statistics

Introduction sur la monétisation du trafic internet

Les flux de trafic, une fois décodés, nous permettent de comprendre les goûts et intérêts des internautes, et la façon dont ils perçoivent l'information. En analysant les grands mouvements de tendance, nous pouvons prédire à quels sites ces internautes vont réagir, et à quels contenus. En segmentant temporellement l'information, et en leur présentant au bon moment des contenus pertinents et personnalisés, c'est tout l'écosystème Internet qui s'améliore.

L'objectif de nos travaux de recherche sur la monétisation est de proposer un nouveau modèle basé principalement sur les indicateurs de satisfaction des internautes. Nos travaux n'ont pas vocation à ouvrir un débat ou une polémique sur les procédés industriels actuels. Cependant, nous pensons qu'ils peuvent aider considérablement à l'amélioration de « la satisfaction des internautes » et de l'ensemble de l'écosystème web.

Nous pensons que les procédés de recommandation représentent aujourd'hui les premières pierres des systèmes d'information de demain, notamment dans le secteur de l'analyse comportementale d'audience. Notre société concentre ses efforts de R&D sur des procédés et méthodes capables d'analyser l'intelligence collective des internautes. D'un point de vue industriel, la technologie qui intègre la plate-forme de Swelen permettrait de prédire si les annonces publicitaires proposées aux internautes ont une chance de les intéresser ou non.

L'idée sous-jacente est d'analyser le comportement des internautes et de segmenter ce comportement en différents types pour améliorer la performance des systèmes publicitaires. Notre technologie de suggestion permet d'isoler les internautes qui aiment un site Internet de ceux qui ne l'aiment pas. Grâce à cette segmentation de l'audience, nous pouvons isoler les internautes qui réalisent la plus grosse partie des revenus publi-

ciaires. En effet, les internautes n'ont pas le même comportement selon qu'ils découvrent un site grâce à un email d'un ami, ou par le biais d'une campagne publicitaire. Notre outil Swelen Suggest analyse les différentes sources de trafic et présente aux internautes les pages les plus pertinentes du site web. Les internautes qui utilisent nos suggestions apparaissent beaucoup plus engagés que ceux qui ne l'utilisent pas.

Les tests effectués avec l'outil Swelen Suggest nous ont amenés à nous intéresser aux différents aspects de la monétisation du trafic. En effet nos résultats prouvent que les internautes utilisant Suggest ont tendance à avoir un meilleur comportement sur le site des éditeurs, ils sont beaucoup plus satisfaits du site que ceux qui ne l'utilisent pas. Ces internautes, s'ils suivent nos suggestions, consultent en général beaucoup plus de pages, restent plus longtemps sur le site Internet, et ont tendance à se transformer plus rapidement en clients fidèles.

Une plateforme intégrée

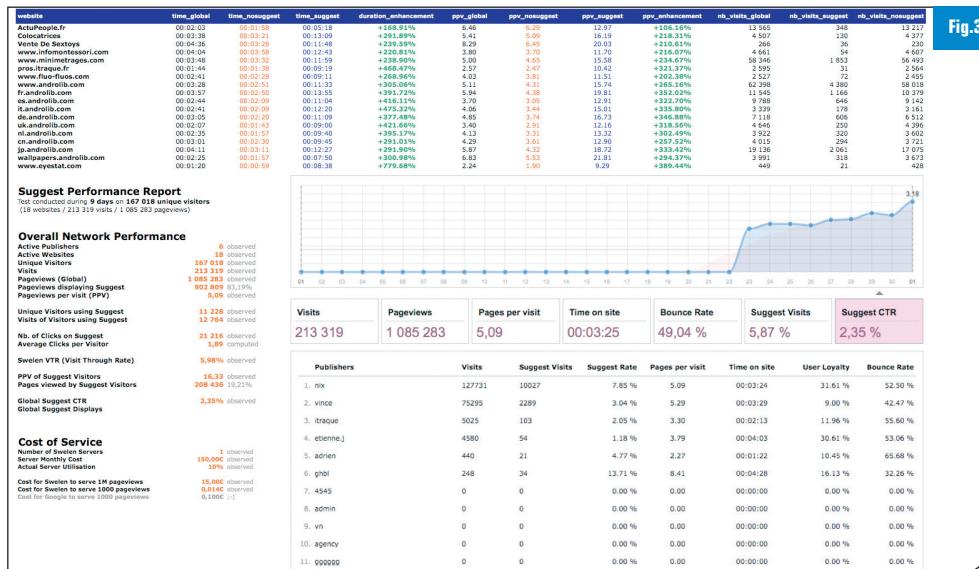
Nous travaillons depuis plusieurs années sur une plateforme publicitaire qui analyse l'intelligence collective des internautes afin d'améliorer les performances des campagnes publicitaires sur Internet. La technologie utilisée dans cette plate-forme est la même que celle utilisée dans notre outil Suggest. Les deux produits partagent la même approche, qui consis-

te à réduire la granularité à laquelle l'information est considérée, agrégée et traitée. En effet, le but fondamental d'une annonce publicitaire est de mettre en relation l'offre et la demande. Plus nous descendons à l'échelle des utilisateurs et des produits, plus nous pouvons comprendre et améliorer cette adéquation offre/demande. Nous pensons que la publicité en ligne doit être régie par des règles équilibrées qui profitent à tous les acteurs de l'écosystème : aux annonceurs, aux éditeurs mais en premier lieu aux internautes. Offrir un tel équilibre peut sembler chose simple, mais en réalité c'est une problématique très complexe à résoudre. Chez Swelen, nous travaillons activement sur la résolution de ce type de problématiques. Dans nos solutions publicitaires, nous souhaitons offrir de bonnes performances aux annonceurs en termes de retour sur investissement, des optimisations de revenus pour les éditeurs, et une meilleure expérience utilisateur chez les internautes. Pour ce faire, nous devons écouter attentivement les signaux émis par les internautes, analyser leur satisfaction et les grands mouvements de tendances dans leurs goûts et leurs intérêts, tout en respectant la plus scrupuleusement leur vie privée.

Notre plate-forme permet d'attribuer un indice de satisfaction lorsqu'un internaute est confronté à des contenus publicitaires. Elle utilise ce type d'indices de façon anonyme pour prédire si tel internaute va aimer tel type de contenu. Par la suite, des algorithmes d'apprentissage se chargent de vérifier si effectivement les prédictions ont été fructueuses ou non, et le système se corrige alors de lui-même en fonction des performances observées. Dans ce genre de problématique, il est crucial de comprendre que la majeure partie du travail consiste en la création, la sélection et l'amélioration des indicateurs de satisfaction utilisés par le système. Ce travail de longue haleine, entrepris très tôt dans la fondation de la plate-forme, est en constante évolution, et il occupe la majeure partie de nos travaux de recherche.

■ Vincent Fontaine & Vincent Borrel

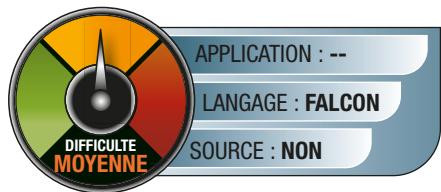
Evolution des taux de clics sur nos recommandations



Falcon, un langage de programmation procédural, OO, fonctionnel et OM



Falcon est basé sur une approche de programmation ouverte qui fusionne parfaitement la programmation procédurale, orientée-objet (OO), fonctionnelle et orientée-message (OM).



Falcon a été conçu à l'origine pour satisfaire aux exigences suivantes :

- échanger rapidement (ou utiliser directement) des données complexes avec C++;
- communiquer avec des applications et leur fournir les moyens de contrôler l'exécution de leurs scripts de manière dynamique;
- fournir plusieurs paradigmes de programmation sous couvert de grammaires tout à fait ordinaires;
- fournir un appui multilangage (UTF);
- fournir un moyen simple de construire des applications de type script, facilement et efficacement intégrées avec des bibliothèques tierces (en anglais : third-party librairies).

Le langage de programmation Falcon tourne sur Linux (32 et 64 bits), Win32 et Solaris (Intel). Des anciennes versions fonctionnent sur Mac OS X et FreeBSD.

1 LE LANGAGE

Falcon est un langage non-typé muni d'énoncés séparés par des fins de lignes et de code structuré en blocs énoncé/fin. Il supporte naturellement les entiers 64 bits, y compris les opérateurs de bit de champs, les virgules flottantes, les tableaux de chaînes de caractères, plusieurs types de dictionnaires, les listes et les *MemBuffers*, c'est-à-dire les zones de mémoire partagée parmi d'autres types de base et de classes-système. Morphologiquement, Falcon ne brise pas les conventions établies, par exemple :

```
function disbonjour()
    print1("Salut à tous!")
end

// Script principal
disbonjour()
```

Vous pouvez exécuter ce script en le sauvegardant dans un fichier-test et en l'introduisant dans Falcon via stdin ou en le lançant comme suit :

```
$ falcon <nomduscript.fal> [paramètres]
```

Dans Falcon on attache une grande importance au modèle multiparamétrique. On y ajoute une programmation tabulaire, une sorte de multicouche orientée objet, mais cela dépasse largement le cadre de cet article et donc nous ne l'aborderons pas. Chaque paradigme est généralement un bit "personnalisé" qui permet d'assouplir la programmation et de fusionner plus aisément avec d'autres paradigmes.

2 LA PROGRAMMATION PROCÉDURALE FALCON

La programmation procédurale Falcon est basée sur des déclarations de fonctions et des appels de paramètres variables. Par exemple :

```
function verifieParametres( premier, second, troisieme)
    > «— verifieParametres —»
    // «» en début de ligne signifie print1
    if premier
        > «Premier paramètre: », premier
    end

    // ... et des énoncés linéaires simples peuvent être abrégés avec «:»
    if second: > «Deuxième paramètre: », second
    if troisieme: > «Troisième paramètre: », troisieme
    > «—————»
end

// Script principal
verifieParametres( « Programmez! » )
verifieParametres( « Le Magazine du Développement » )
verifieParametres( « www.programmez.com », 8.6, 0xAB )
```

On peut utiliser les fonctions RTL pour rechercher les paramètres réels passés aux fonctions (ou méthodes). Les valeurs peuvent aussi être passées par référence (ou alias) et les fonctions peuvent avoir des blocs statiques et des variables.

```
function compteur( )
    static
        > «Compteur initialisé!»
        c = 0
    end
    return c++
end

> «Premier appel: », compteur( )
> «Second appel: », compteur( )
> «Troisième appel: », compteur( )

variable = «Valeur de départ»
variable_alias = $variable           // donne à variable un autre
nom qui est variable_alias
variable_alias = «Nouvelle valeur»
print1( variable )                  // affichera «nouvelle valeur»
```

```

function modifie_parametre( parametre )
    parametre = «Modifié!»
end

// Scripts principal
variable = «Valeur d'origine»
modifie_parametre(variable)      // passage par valeur
printl(variable)                // affiche toujours «valeur
d'origine»
modifie_parametre( $variable )  // passage par alias
printl(variable)                // affiche «Modifié»

```

A nouveau, les fonctions RTL peuvent être utilisées pour déterminer si un paramètre a été passé directement ou par référence. La directive stricte force les variables à être déclarées explicitement via le mot-clé **def** :

```

directive strict=on

def alpha = 10 // nous voulons vraiment déclarer alpha
test( alpha ) // un appel avant une déclaration est autorisé

function test( val )
    local = val * 2 // erreur : non déclaré avec def !
    return local
end

```

Falcon dispose d'un énoncé puissant pour traverser et modifier des séquences. L'exemple suivant affiche et modifie des valeurs dans un dictionnaire :

```

dictionnaire = [«un» => 1, «deux» => 2, «trois» => 3, «quatre»
=> 4, «cinq» => 5]
for key, value in dictionnaire

    // «>>» signifie «print»
    forfirst: >> «Le dictionnaire est: «

    // opérateur d'expansion de chaîne «@»
    >> @ «$key=$value»

    .= «Bang!»

    formiddle: >> «,»

    forlast: > «. «

end

```

“>>” et “>” sont respectivement équivalents à **print** et **printl**. “>>” affichera l'objet à l'écran et “>” le fera également avec un retour à la ligne. Notons l'opérateur d'expansion de chaîne dans le code ci-dessus. Falcon fournit une expansion de chaîne via des variables de nom et des expressions, et en appliquant l'opérateur unaire explicite @. Des expansions de chaîne peuvent contenir des spécificateurs de for-

mat, tels que @ «\${varnom:r5}», qui justifient à droite dans cinq espaces, mais on dispose aussi d'une classe de format pour cacher (“mémoire cache”) et utiliser plusieurs fois les mêmes formats.

Les ensembles définis par l'utilisateur et les séquences du langage fournissent tous deux des itérateurs qui peuvent être utilisés pour accéder à la liste de façon traditionnelle. Les opérateurs fonctionnels tels que **map**, **filter** et **reduce** sont également fournis.

```

// voyons ce qu'il y a maintenant dans le dictionnaire:
inspect( dictionnaire )

```

3 LA PROGRAMMATION FALCON ORIENTÉE OBJET

Un script Falcon peut définir des classes et instancier des objets à partir d'elles, créer des objets singltons (un objet singleton est un objet qui ne peut être instancié qu'une seule et unique fois dans le programme) avec ou sans classes de base, et appliquer des attributs transversaux aux instances. Le mot-clé **provides** vérifie les propriétés exhibées par les instances :

```

// une classe
class Quelqu_un( initval1, initval2 )
    // Une simple initialisation peut être effectuée directement
    prop1 = initval1
    prop2 = nil

    // init prend les paramètres de la classe
    // et exécute une initialisation plus complexe
    init
        self.prop2 = initval2
        > «Exécution de la classe Quelqu_un ...»

    end

    // une instance singleton
    object Identité
        function quiSuisJe( )
            > «Quelqu'un dit: «, self.prop1, « « , self.prop2
        end
    end

    // une instance
    instance = Quelqu_un( «qui», «es-tu ?» )
    instance.montreMoi( )

    // «Identité» est déjà une instance
    if Identité provides quiSuisJe
        Identité quiSuisJe( )
    end

```

Falcon a un *Basic Object Model* (BOM) qui est disponible dans tous les objets. Les objets et les classes peuvent *remplacer* certaines méthodes. Par exemple, passer un objet à l'opérateur d'impression ">" provoque l'appel de sa méthode BOM **toString**, et cet *overriding* peut s'exprimer ainsi :

```
object different
  function toString( )
    return «est différent...»
  end
end
> «l'objet... », different
```

Falcon supporte l'héritage multiple, mais il le clarifie en forçant linitialisation de l'héritage et la priorité selon l'ordre des déclarations d'héritage. Des classes supportent les membres statiques qui peuvent être partagés entre objets de la même classe, et des méthodes avec des blocs statiques qui peuvent agir comme des initialiseurs à grande(s) classe(s). Des méthodes peuvent être appelées directement à partir des classes lorsqu'elles n'ont pas besoin d'accéder à l'objet **self**, en fournissant la sémantique des méthodes statiques C++ / Java / C#.

Il est possible de fusionner des procédures normales avec des méthodes en initialisant des procédures à des propriétés :

```
function appelle_moi( )
  if self and self provides mon_nom( )
    > self.mon_nom
  else
    > «Désolé, appel incorrect.»
  end
end

object test
  prop1 = nil
  mon_nom = «Ceci est un test!»

  function bonjour( )
    > «Salut à tous de la part de », self.mon_nom
  end
end

//appels ordinaires
appelle_moi( )

// utilisant la procédure comme une méthode
test.propriété1 = appelle_moi

// ou bien une méthode comme une procédure
fond = test.bonjour
test.mon_nom = «Faucon»

// A noter : proc utilisera de manière dynamique le bon "self"
fond( )
```

4 LES ATTRIBUTS

Les attributs sont des propriétés binaires qui peuvent être présentes ou pas pour une instance spécifique ou objet, peu importe sa classe. Les attributs ont un grand pouvoir expressif, et dans Falcon, ils expriment ce qu'est un objet, ce qu'il possède et ce à quoi il appartient, selon le contexte. Par exemple, on peut définir un attribut "ready" qui dénote les objets prêts à être élaborés :

```
class Data( nom )
  attribut: «ready» // déclare un attribut "ready"
  nom = «générateur»

  function process( )
    > «Processus », self.nom, « lancé !...»
  end
end

processus = [ ]

for i in [0:10] // crée 10 "processus"
  processus += Data( i )
  printl(processus[i])
end

if nom = «ready»
  nom = Data.process( )
end
```

Les attributs sont des symboles arbitraires - au moment de la compilation - auxquels est associée une valeur statique. Des modules, fonctions, classes et objets peuvent être munis d'attributs qui peuvent exprimer leur fonctionnalité ou être employés pour différents objectifs.

```
programme_exemple: «en cours d'exécution»
auteur: «Rodrigue S. Mompelat»

function test( )
  groupe: «Les Fans de Falcon»
  activité: «Programmation avancée»
  lieu: «Danemark»

  // faire quelque chose...
end
// montre les attributs de ce module
inspect( attributes( ) )

// montre les attributs de la fonction test
inspect( test.attributes( ) )

class UneCertaineClasse
  attribut: «ready»
end
```

```
objet = UneCertaineClasse( )
if objet = «ready»
    > «Un objet ready est né !»
end
```

Des attributs de module peuvent être *inspectés* (mot-clé **inspect**) par d'autres modules par l'intermédiaire du *Compiler Feather Module*.

5 LA PROGRAMMATION FONCTIONNELLE

La construction de base de la programmation fonctionnelle Falcon est la *séquence accessible* (en anglais : *callable sequence*), connue aussi sous le nom de **Sigma**. Pour l'instant, la seule séquence reconnue est le tableau, mais d'autres types de séquences (comme les listes) devraient être bientôt reconnus. Schématiquement, un Sigma est un appel retardé qui peut fonctionner comme suit :

```
function test( a, b, c )
    > «Paramètres:»
    > a
    > b
    > c
end

// directe
test( «un», «deux», «trois» )

// indirecte
cached = [ test, «quatre», «cinq», «six» ]
cached( )
```

L'appel respecte le paradigme procédural (des paramètres variables) et le tableau est toujours un vecteur normal que l'on peut accéder et modifier par le biais des opérateurs standard du langage et les fonctions RTL. Cet appel retardé n'est toujours pas une pleine "évaluation de contexte fonctionnelle". Le processus adéquat d'évaluation fonctionnelle est appelé la **réduction Sigma**. Il résoud des Sigmas de manière récursive de l'intérieur vers l'extérieur et de gauche à droite lorsqu'ils sont au même niveau, en les remplaçant par leur valeur de retour. Des fonctions spéciales connues sous le nom d'**Etas** déclenchent et contrôlent l'évaluation fonctionnelle; la fonction Eta la plus simple est **eval** qui initialise et exécute une réduction basique Sigma. Par exemple, l'expression « $(a+b) * (c+d)$ » peut s'écrire sous la forme d'une séquence de type Lisp :

```
function add( a, b ) : return a+b
function mul( a, b ) : return a*b

> «(2+3)*(4+5)=», eval(.[mul .[add 2 3] .[add 4 5]])
```

La notation **[]** représente des déclarations de tableau dont les éléments sont séparés par des espaces blancs au lieu d'une **,** explicite.

Le RTL Falcon s'accompagne d'un ensemble riche d'**Etas**, tels que **iff** (if fonctionnel), **cascade** (qui unit des appels plus standard dans une séquence simple), **floop** et **times** (des modèles différents de boucles fonctionnelles), **map**, **filter**, **reduce** et beaucoup d'autres. Les séquences fonctionnelles peuvent être paramétrées par la clôture et les références. L'exemple ci-dessus peut être paramétrique de la façon suivante :

```
// add et mul comme précédemment
function calculateur( a, b, c, d )
    return .[eval .[mul .[add a b] .[add c d]]]
end

op = calculateur( 2, 3, 4, 5 )
> «(2+3)*(4+5)=», op( )
```

Les opérateurs fonctionnels traditionnels, tels que **map**, **filter** et **reduce**, sont supportés, mais le *système d'objets hors-bande* (out-of-band item system) étend leur fonctionnalité. Des *objets hors-bande* sont des objets marqués d'un drapeau par la fonction **oob**. Bien qu'ils soient des objets normaux sous tout autre aspect, cette marque spéciale indique qu'ils contiennent une valeur inattendue, particulière et d'une certaine façon extraordinaire, voyageant à travers les séquences fonctionnelles. Falcon supporte également les expressions Lambda et les *nested functions*.

La programmation Falcon s'apparente à la programmation orientée objet lorsque les Sigmas peuvent être définis comme des propriétés d'objets, et les méthodes d'objets peuvent être utilisées comme des Kappas (*Sigma-callable header symbols*) :

```
object UnCertainObjet
    une_propriete = 10
    function maProp( valeur )
        return self.une_propriete * valeur
    end
end

> «5*10=», eval(.[UnCertainObjet.maProp 5])
```

CE DONT NOUS N'AVONS PAS PARLÉ

Il ne nous était pas possible dans le cadre de cet article de décrire tout ce que Falcon peut faire, principalement parce que ce qui paraît utile à certains peut paraître inutile à d'autres. Nous n'avons pas parlé des co-routines, de l'opérateur d'indirection, de la programmation tabulaire à venir, du compilateur réflexif, du système de document modèle Falcon (Falcon Template Document), du module de multithreading etc.

■ Dr Rodrigue **Sabin Mompelat**

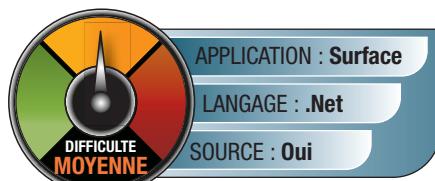
Professeur Associé

Langages de Programmation, Ingénierie Logicielle & Systèmes
Danmarks Tekniske Universitet - romo@imm.dtu.dk

Tous les jours : l'actu et le téléchargement
www.programmez.com

La gestuelle avec MVVM sur Microsoft Surface

Microsoft Surface est une plateforme intéressante à plusieurs niveaux, elle se distingue déjà par sa méthode de détection infrarouge qui permet d'avoir plus de précision et surtout supporte un grand nombre de contacts simultanés. Cela en fait un périphérique de choix pour la conception d'applications NUI (Natural User Interface), autrement dit une interface naturelle. Le choix des gestuelles et l'implémentation de celles-ci dans son application est donc essentiel.



Actuellement, la programmation sur Surface se fait par deux technologies Microsoft : WPF et XNA. La plupart des applications métiers seront conçues avec WPF, technologie de choix pour les interfaces riches sous Windows. Cependant il existe un point difficile à traiter, la gestion des gestuelles avec un pattern MVVM (Model / View / ViewModel).

1 PIQÛRE DE RAPPEL

MVVM est une solution permettant l'industrialisation du développement d'applications WPF. Elle est particulièrement bien adaptée pour des applications dont la couche d'accès aux données doit avoir une certaine indépendance des applications qui l'exploitent.

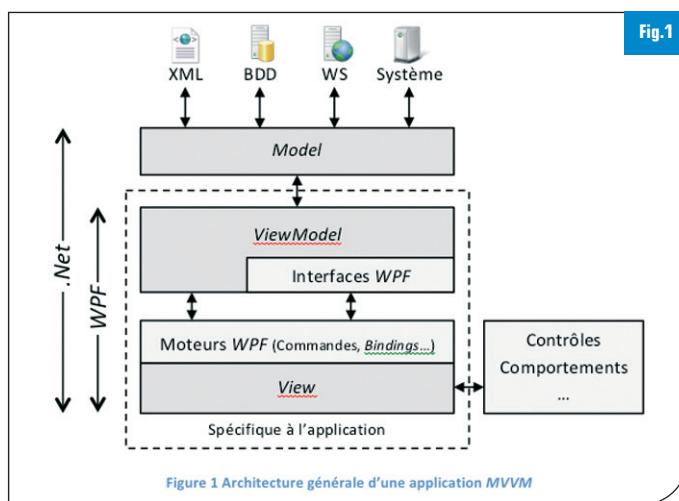
Une application MVVM repose sur la séparation de trois éléments :

- Le Model : les données métier.
- Le ViewModel : l'interface logique.
- La View : l'interface visuelle.

[Fig.1]

Le Model apporte les données métiers au monde .Net, c'est la couche d'accès aux données comme *Entity Framework* par exemple.

Le ViewModel est l'adaptation simultanée du Model pour le moteur de rendu *WPF*. La View mappe l'interface logique du ViewModel sur



La figure 1 représente la structure d'une application MVVM.

une application WPF par le mécanisme de *Binding*.

Dans la grande majorité des cas, la communication entre la *View* et le *ViewModel* se fait par l'intermédiaire de commandes (implémentées via l'interface *ICommand*) qui sont *bindées* sur la *View*. Il y a très peu de contrôles (Bouton, Menu, ...) dans WPF qui peuvent utiliser ces commandes en réaction à une action utilisateur, il faut la plupart du temps utiliser un *Behavior* (comportement) qui lie une commande à l'événement d'un contrôle.

Un *Behavior* est une classe statique qui possède des propriétés de dépendance attachées permettant d'ajouter un comportement non natif sur un *FrameworkElement*, voire *UIElement*. Un peu comme les propriétés attachées *Grid.Row* ou *Canvas.Left*, sauf qu'elles ne sont pas dépendantes d'un objet parent.

2 PROBLÉMATIQUE DES GESTUELLES

La grande problématique de Surface est que la gestion des contacts est déportée dans un *Behavior* (la classe *Contacts*) et que les quelques contrôles spécifiques à Microsoft Surface n'implémentent pas la gestion des commandes hormis le bouton et les menus. De plus, il est parfois difficile de conditionner une gestuelle complexe sans avoir à écrire beaucoup de code afin de contrôler et notifier celle-ci. Alors comment implémenter une gestuelle et pouvoir notifier notre *ViewModel* ?

Le plus simple serait d'avoir une propriété de dépendance de type *ICommand* qui se déclencherait à chaque fois que la gestuelle est effectuée par l'utilisateur. Il va donc falloir trouver un moyen pour exposer cette propriété mais aussi essayer de factoriser les gestuelles pour ne pas les réécrire sur chaque contrôle, pour cela je vous propose deux solutions simples.

Le CustomControl

La façon la plus simple serait d'embarquer cette gestuelle dans un contrôle de type *Container* afin que tous les contrôles contenus dans ce *Container* puissent hériter de la gestuelle. Concrètement, il suffit de créer un *CustomControl* de type *ContentControl* (ou *SurfaceContentControl*).

Au niveau du fichier *Generic.xaml*, on détermine un *Template* simple qui n'affiche aucun rendu mais qui permet de traiter les contacts.

Ensuite il faut créer une propriété de dépendance de type *ICommand* qui récupérera par *Binding* la commande du *ViewModel*.

Pour finir il suffit d'instancier la classe *Affine2DManipulationProces-*



sor, intercepter les nouveaux contacts sur le contrôle pour les envoyer vers l'*Affine2DManipulationProcessor*, il ne vous restera plus qu'à traiter la gestuelle dans l'*Affine2DManipulationDelta*.

```
// On crée le mécanisme de gestuelle

Affine2DManipulationProcessor processor;
processor = new Affine2DManipulationProcessor(Affine2DManipulations.TranslateX | Affine2DManipulations.TranslateY, this);

processor.Affine2DManipulationDelta += new
EventHandler<Affine2DOperationDeltaEventArgs>(this.Affine2DManipulationDelta);

// On récupère tous les contacts sur le contrôle afin de les
injecter dans l'Affine2DManipulationProcessor

Contacts.AddContactDownHandler(this, this.ContactEventHandler);
```

Cette solution est simple à mettre en place et fonctionne dans la majeure partie des cas. Une fois ce contrôle créé, on le déclare ainsi dans le XAML :

```
<local:MonControle Command=>{Binding MaCommande}>
<Rectangle Fill=>Red>
  Width=>500
  Height=>500/>
</local:MonControle>
```

Le Behavior

Une autre solution consiste à créer un comportement afin qu'il soit adaptatif, entendre par là qu'il peut se brancher sur n'importe quelle structure XAML. De plus on peut découper chaque gestuelle dans un *Behavior* différent. Dans ce cas, il est plus facile de créer une

classe avec des propriétés de dépendance adaptées à la gestuelle que l'on recherche, par exemple si l'on veut créer un *Behavior* de gestuelle qui capte une translation, on crée une propriété de dépendance pour définir le seuil de déplacement du contact avant le déclenchement de la commande.

Son utilisation dans le XAML se fait comme ceci :

```
<Rectangle Fill=>Red>
  Width=>500
  Height=>500
  local:GestureBehavior.Enable=>True
  local:GestureBehavior.DeltaStep=>100
  local:GestureBehavior.Command=>{Binding MaCommande}>/>
```

Dans les deux cas de figure, ces méthodes permettent vraiment une intégration facile dans un pattern *MVVM* tout en conservant la complexité d'une gestuelle. On peut imaginer aussi pouvoir ajouter une propriété *CommandParameter* sur le *CustomControl* et le *Behavior*. D'ailleurs, en ce qui concerne la déclaration des propriétés *Command* et *CommandParamter*, il est conseillé que le *Behavior* implémente l'interface *ICommandSource*. Cela permet de respecter les bonnes pratiques *WPF*, mais aussi de pouvoir indiquer un *Target*. En fait, tout est permis, il suffit juste d'adapter ces solutions à ses problématiques de développement.

Ressource :

CommandSource : [http://msdn.microsoft.com/fr-fr/library/system.windows.input.icommandsource\(VS.85\).aspx](http://msdn.microsoft.com/fr-fr/library/system.windows.input.icommandsource(VS.85).aspx)



■ Nicolas Calvi

MVP Surface - Consultant / Formateur chez Winwise - Blog : <http://blog.chaos-web.org/>

Maven - épisode III, la revanche du (plugin) site

Le mot build, sans traduction convaincante en français, n'a pas de définition clairement établie. Il regroupe toutes les étapes qui nous permettent de passer du code source à un logiciel livrable. En fonction des règles et des enjeux de chacun, cela ira de la simple compilation à l'analyse complète du code par des outils d'audit et de mesure de qualité, en passant par l'exécution d'une suite de tests et à l'intégration avec les outils de développement. Quelle que soit la définition qu'on donne du build, il n'est pas envisageable de réaliser ces opérations à la main.



APPLICATION : Maven

LANGAGE : --

SOURCE : non

De nombreuses solutions existent pour automatiser ce processus. Une large majorité d'entre elles utilisent des scripts pour figer les commandes nécessaires

et les enchaîner sans intervention de l'utilisateur. Selon la richesse et l'intelligence de l'outil, ces scripts peuvent être plus ou moins complexes et plus ou moins précis dans les commandes nécessaires pour répondre aux attentes du développeur. Ces solutions ont toutes ce même point commun : elles permettent de décrire les commandes à exécuter pour arriver à un objectif, et pas l'objectif lui-même, en laissant à l'outil le soin de faire le nécessaire. Ce manque d'abstraction aboutit rapidement à des structures complexes et difficiles à maintenir ou à mutualiser.



Depuis sa première version, Maven expérimente une approche différente dite déclarative, qui vise à indiquer en quoi un projet est différent d'un modèle type

et nécessite des étapes particulières pour le build. D'autres outils comme Gradle lui ont emboîté le pas, tout en conservant une dimension script qui laisse la boîte de Pandore grande ouverte. Le concept clé de Maven est de se baser sur des conventions, que chaque projet doit suivre pour bénéficier de la richesse de l'outil. Largement adopté, Maven est devenu un standard de fait, et un développeur qui arrive sur un projet Maven retrouve naturellement ses marques et exécute sans attendre la commande universelle « mvn install ».

ABOUTISSEMENT D'UNE TRILOGIE

Le développement de Maven 2 avait été une révolution, jetant l'éponge sur le code de Maven 1 devenu ingérable. Les migrations qui ont suivi ont parfois été douloureuses, mais l'objectif était de fermer la porte à tous les travers révélés par Maven 1 en proposant une base structurée et stricte. Malgré ces efforts, Maven 2 était loin de la perfection. Son socle technique vieillissant, ses défauts de conception interne, son manque de modularité rendaient les évolutions et certaines corrections difficiles.

La très large adoption de Maven 2 ne permettait pas d'envisager une nouvelle rupture. La compatibilité a donc été la toute première préoccupation dans le développement de Maven 3. Un énorme tra-

vail a été réalisé pour équiper le projet Maven d'un harnais de tests d'intégration, réunissant plus de 600 tests pour cerner chaque subtilité de sa gestion des dépendances, des plugins, du cycle de vie. L'élimination des défauts de conception ne s'est pas faite non plus sans mal, pas moins que la remise à niveau du socle technique pour des technologies actuelles. Ce travail de longue haleine a abouti en octobre 2010 avec la finalisation de Maven 3.0, signal de départ d'une nouvelle base de développement. Pour ne plus subir ce long effet tunnel, les développeurs ont annoncé un cycle de livraison de 6 semaines, assurant ainsi l'inclusion progressive de nouvelles fonctionnalités sans risquer une fuite en avant du périmètre comme on a pu le constater auparavant.

QUOI DE NEUF ?

Avant toute chose, Maven 3 se veut compatible avec l'existant Maven 2. Cette compatibilité n'est pas absolue, comme le résume la documentation (<http://cwiki.apache.org/MAVEN/maven-3x-compatibility-notes.html>), mais est très largement suffisante pour l'énorme majorité des projets. En premier lieu, Maven 3 effectue une validation plus stricte du POM, de l'origine des dépendances par rapport aux dépôts de bibliothèques, de la bonne déclaration des versions.

Un exemple concret : sur un projet, je déclare une dépendance vers Hibernate-envers. J'ai déjà utilisé cette bibliothèque précédemment et elle est disponible en local. Maven 2 ne va pas chercher plus loin et va donc résoudre mes dépendances sans sourciller. Cependant Hibernate-envers n'est pas disponible sur le dépôt central, aussi si je partage mon projet avec un collègue celui-ci aura une dépendance non résolue. Le build universellement reproductible promu par Maven est mis à mal !

Maven 3 conserve dans le dépôt local l'origine de chaque artefact. Il peut donc détecter que je tente d'utiliser une bibliothèque dont je n'ai pas déclaré le dépôt. L'ajout d'un élément `<repository>` dans mon POM résout le problème et rend mon projet plus fiable :

```
<repository>
  <id>jboss</id>
  <url>http://repository.jboss.org/nexus/content/groups/public-jboss</url>
</repository>
```

Il existe une petite série de cas similaires pour la gestion des dépen-

dances, des versions de plugins, des dépôts de bibliothèques et de plugins, ou l'utilisation de timestamps pour les SNAPSHOTS, pour lesquels Maven 3 se montre beaucoup plus regardant que son prédecesseur. Ne vous y trompez pas, il s'agit de rendre votre build plus fiable et d'éliminer les problèmes de dépendances introuvables et autres versions de plugin instables d'un poste de développement à un autre.

AETHER

La gestion des dépendances est une fonctionnalité clé de Maven. Si on a pu lui reprocher la qualité médiocre des métadonnées diffusées sur le dépôt central, Maven est le premier outil Java à voir fédérés autant de projets sur un format et un média de distribution commun, au point d'être cité par la JSR-294 comme exemple pour le futur mécanisme de modules de Java.

Le code historique de Maven 2 pour gérer les dépendances était, comme le reste, assez touffu et parfois obscur. Sonatype (société qui contribue largement au développement de Maven) a lancé un développement « feuille blanche » pour revoir complètement cette gestion, aboutissant au projet Aether (<http://aether.sonatype.org>).



Intégré à Maven 3, Aether porte désormais toute la résolution de dépendances. Le résultat par rapport à Maven 2 peut, dans certains cas, être différent. Un projet avec un graphe de dépendance particulièrement riche et des conflits de versions peut constater un changement dans le classpath résultant. Cela reste cependant assez marginal, avec au final un composant nettement plus fiable.

SITE ET CÔTÉ OBSCUR

Maven 3 abandonne aussi certaines fonctionnalités historiques. En premier lieu, le support des dépôts au format Maven1, qui appartient désormais à l'histoire.

Un autre changement notable est une refonte complète du mécanisme de génération de rapports et du site web du projet. Cette fonctionnalité de Maven 2 était clairement une verrou dans la structure de Maven, avec une gestion de version à part et un plugin site à la complexité croissante.

Maven 3 est désormais indépendant du mécanisme de reporting. Une nouvelle version du plugin site a été développée et supporte la majorité des plugins de production de rapports (<http://cwiki.apache.org/MAVEN/maven-3x-and-site-plugin.html>). Une migration reste donc nécessaire pour mettre à jour le plugin site (qui ne fonctionnera donc plus qu'avec Maven 3). Le plugin sait exploiter votre section `<reporting>` pour assurer la compatibilité ascendante, mais cet usage est déprécié et il vous gratifiera d'un warning.

[WARNING] The `<reporting>` section is deprecated, please move the reports to the `<configuration>` section of the new Maven Site Plugin. Ce bouleversement de la gestion du site est la seule, mais néanmoins significative, modification qui devra être apportée aux projets pour migrer vers Maven 3. Pour le reste, un projet qui gère déjà correctement ses dépendances et ses versions de plugin n'aura aucun souci à passer à la nouvelle version de Maven.

Mais quels bénéfices va en tirer le projet qui migrera ?

En premier lieu, Maven 3 est plus économique en consommation mémoire et également plus rapide. Evidemment, ce gain de temps ne concerne que les parties propres à Maven, à savoir la résolution

des dépendances et la configuration des outils, Maven ne sachant pas encore accélérer le compilateur Javac (peut-être pour une prochaine version ?)

Ensuite, lorsque quelque chose ne va pas, Maven 3 propose une gestion des messages d'erreur plus claire en cas de soucis – ça arrive – en proposant une pile d'erreurs plus détaillée et un rapport des modules exécutés ou en échec. Sur des projets de grande envergure l'analyse d'un problème en est largement facilitée.

DANS UN UNIVERS PARALLÈLE

Enfin, et c'est la fonctionnalité que vous serez nombreux à essayer en lisant ces lignes, Maven 3 propose de tirer parti du parallélisme des processeurs. Si vous avez un processeur Core i5 ou i7 vous disposez par exemple de quatre cœurs. Votre serveur d'intégration continue en a peut être huit ou plus sous le capot.

Les bibliothèques de test comme TestNG ou JUnit4 proposent déjà d'exécuter les tests en parallèle – je vous conseille d'essayer : ce sera aussi l'occasion de voir si vos tests sont bien indépendants les uns des autres, ce qui révèle souvent quelques surprises. Il suffit de spécifier dans la configuration du plugin surefire le paramètre parallel :

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-surefire-plugin</artifactId>
  <version>2.7</version>
  <configuration>
    <parallel>classes</parallel>
  </configuration>
</plugin>
```

Les classes de tests vont alors se répartir sur les différents cœurs disponibles, à moins de définir explicitement un nombre de threads à utiliser.

Qu'en est-il pour le reste du build ? Le build Maven 2 est un processus strictement séquentiel qui peine à tirer parti des multiples ressources CPU. Il en résulte que vous perdez en réactivité, et donc en productivité, alors que votre machine enchaîne les Nop en attendant qu'on lui donne quelque chose à faire. La génération de code source par wsdl2java dans mon module webservice pourrait très bien effectuer en parallèle de la compilation de code dans mon module businessDomain !

Maven 3 propose une nouveau paramètre, qui comme tout bon outil en ligne de commande se résume à un unique caractère « -T ». Ce petit flag supplémentaire est la porte d'entrée du parallel build (<http://cwiki.apache.org/MAVEN/parallel-builds-in-maven-3.html>)

En lançant « mvn -T 4 install » vous demandez à Maven de répartir le build sur 4 threads. Il va analyser les modules qui constituent le projet et les plugins impliqués, et si les plugins le supportent, va paralléliser le build des modules indépendants. En pratique, un gain de 20 à 50% peut être constaté sur la durée d'exécution, ce qui est loin d'être négligeable pour trois caractères tapés en plus !

« Si les plugins le supportent », voilà tout ce qui peut faire la différence. Pour que ce mécanisme puisse fonctionner, il faut que les plugins impliqués dans un build parallèle soient conçus pour accepter plusieurs exécutions concurrentes dans la même JVM. Certains plugins le supportent sans modification, d'autres y sont allergiques. Maven 3 introduit pour les développeurs de plugin l'annotation `@ThreadSafe` ; elle permet d'indiquer qu'un plugin peut s'exécuter

en parallèle sur plusieurs modules. Les plugins hébergés par le projet Apache Maven ont été testés et marqués dans leurs dernières versions, ainsi qu'un certain nombre de plugins de la communauté Mojo.

Si votre build utilise un plugin non marqué, Maven 3 vous en informera avec un warning. Dans de nombreux cas, cela signifie que le plugin considéré n'a pas encore été testé, pas nécessairement que le build parallèle n'est pas possible. Vous pouvez donc tenter le coup pour gagner en productivité, cependant n'utilisez pas cet argument pour produire votre binaire le jour de la livraison !

```
[WARNING] ****
[WARNING] * Your build is requesting parallel execution, but project *
[WARNING] * contains the following plugin(s) that are not marked as *
[WARNING] * @threadSafe to support parallel building. *
[WARNING] * While this /may/ work fine, please look for plugin updates *
[WARNING] * and/or request plugins be made thread-safe. *
[WARNING] * If reporting an issue, report it against the plugin in *
[WARNING] * question, not against maven-core *
[WARNING] ****
```

AU-DELÀ DE LA LIGNE DE COMMANDE

La refonte technique profonde dont fait l'objet Maven 3 apporte également une nouvelle voie d'évolution pour l'outil : celui-ci devient embarquable dans un autre contexte que la ligne de commande.

Vous l'utilisez peut-être d'ailleurs déjà sans le savoir au sein d'Eclipse : le plugin m2eclipse (<http://m2eclipse.sonatype.org>) est basé sur Maven 3 pour l'analyse des fichiers POM et la résolution des dépendances. Après des débuts chaotiques et de gros problèmes de performances dans l'intégration avec l'IDE le plus utilisé (mais pas forcément le meilleur), les versions 0.10 et 0.12 ont montré de nettes améliorations. Le plugin ne tourne plus en boucle comme on pouvait le constater précédemment, et apporte un réel confort pour une utilisation de Maven dans Eclipse.

Un autre exemple d'intégration de Maven dans un outil de développement nous vient à nouveau de Sonatype, avec le projet Maven-Shell (<http://shell.sonatype.org>). L'idée est de proposer une console dédiée à Maven, avec assistance syntaxique, coloration du log Maven pour plus de lisibilité et surtout mise en cache des dépendances résolues et des modules du projet. L'exécution de commandes Maven depuis cette console est plus rapide que de relancer une JVM Maven pour chaque opération.

Autre bénéfice de la refonte technique de Maven, le code en charge de la lecture du POM est désormais indépendant du cœur de Maven. En réponse aux critiques sur la syntaxe XML de Maven 2, Sonatype a lancé le projet *Polyglot Maven* (<http://polyglot.sonatype.org>) qui permet à ceux que l'aventure tente de définir leur build Maven en Clojure, Scala ou Groovy :

```
project {
    groupId 'com.programmez'
    artifactId 'sample-polyglot'
    version '1.0'

    dependency( 'fr.loof:artice:1.0' )
}
```

Cependant, si vous avez testé le support Maven dans Eclipse ou IntelliJ Idea, vous savez que le formalisme XML y est particulièrement bien outillé, aussi l'avenir de cette syntaxe alternative s'adresse-t-elle à une population réduite.

TENTÉ PAR LE CÔTÉ OBSCUR ?

Faut-il migrer vers Maven 3 ou rester confortablement avec un build Maven 2 ? L'approche conservative à l'avantage de rassurer, de ne rien coûter dans l'immédiat, et de n'obliger personne à prendre le moindre risque. Elle conduit à la colère de ne pas pouvoir profiter de nouvelles versions des plugins ou des correctifs, à l'agacement devant un build interminable, à l'exaspération devant des problèmes de dépendances manquantes ou de plugins inconsistants.

La migration nécessite de prendre le temps de tester le build Maven avec un dépôt local neuf, et d'analyser les éventuelles remontrances de l'outil. Ces minutes « perdues » aujourd'hui sont des heures gagnées demain. Migrer c'est aussi s'assurer le support de la communauté et la possibilité de bénéficier des correctifs qu'apporteront les prochaines versions de Maven 3 et de ses plugins. C'est encore obtenir un build plus rapide, moins consommateur de ressources, plus réactif et donc gagner en productivité et en confort. C'est aussi s'autoriser à utiliser de nouveaux outils de développement. Enfin, migrer, c'est avoir une chaîne de production cohérente, avec le même socle Maven en ligne de commande, dans l'IDE et sur l'intégration continue grâce au support natif de Maven 3 dans Hudson développé par notre compatriote Olivier Lamy.

QUE MAVEN 3 SOIT AVEC VOUS

Le meilleur de Maven 3 reste à venir : avec un socle rénové et modulaire, le développement de nouvelles fonctionnalités devient à nouveau possible alors qu'il était étouffé sous la complexité de Maven 2. Le support du build parallèle en est la démonstration.

Une évolution annoncée est le support de points d'extension pour les plugins. L'enrichissement d'un plugin Maven 2 nécessite de surcharger son classpath pour remplacer ses classes. Maven 3 emprunte le principe de plugins et de points d'extension d'Eclipse pour proposer des portes d'entrées dans les plugins Maven. La nouvelle politique de développement de plugins Maven est donc de proposer des points d'entrée clairement balisés sur lesquels les développeurs/utilisateurs pourront venir greffer des compléments propres à leurs besoin : manipulation du web.xml d'une archive web par exemple.

Autre évolution très attendue, la possibilité de composer le POM du projet en récupérant des fragments éprouvés. Actuellement, seul le mécanisme d'héritage permet de mutualiser de la configuration Maven, avec la limitation d'un héritage simple. On en revient donc assez inévitablement à des copier/coller de fragments XML. Les mixins permettront d'importer un fragment de POM pour prendre en charge un aspect du build. Le déploiement sur l'infrastructure de test, nos règles de codage et de suivi qualité, la configuration de notre générateur de code, pourront être mutualisés sous forme de fragments et assemblés par composition dans le POM. La syntaxe exacte et les limites du mécanisme ne sont pas encore établies,

mais cette fonctionnalité est attendue pour Maven 3.1 ... si la roadmap est respectée ;)



■ **Nicolas De Loof**
Architecte Java chez OBS IT&Labs, Committer Apache Maven et JUG-Leader Rennais

Concevoir une application PHP avec ZendFramework & Doctrine 2

Zend Framework est devenu une référence pour bâtir des applications web d'envergure en PHP. Doctrine est un objet relationnel mapper (ORM) pour PHP 5.2.3 + qui se trouve au sommet d'une couche d'abstraction de base de données (DBAL). Les versions ZF 1.11 et Doctrine 2.0-Beta4 sont sorties à la fin de l'été dernier et apportent leur lot de nouveautés. Dans le cadre d'une réalisation d'application web, l'intégration de Doctrine à ZF peut se révéler être une très bonne idée en termes de qualité de code, de facilité de maintenance, et de gain de temps de développement.



APPLICATION : Web

LANGAGE : PHP

SOURCE : Oui

ZF, ON RESTE EN VERSION 1

La version ZF 2 bien que prometteuse n'est prévue au plus tôt que pour le mois d'Avril 2011. Les rumeurs d'intégration de Doctrine 2 à ZF 2 courrent, mais pour l'instant aucune décision n'a été prise ; il semblerait que Zend_Db soit réécrit. Quoi qu'il en soit, la version 2 de ZF n'est pas prête et ne doit pas être envisagée pour la conception d'une application à court terme ; on nous promet de toutes façons des outils pour aider la migration future de nos applications vers la version 2.

Intégration de Doctrine 2 à ZF 2 courrent, mais pour l'instant aucune décision n'a été prise ; il semblerait que Zend_Db soit réécrit. Quoi qu'il en soit, la version 2 de ZF n'est pas prête et ne doit pas être envisagée pour la conception d'une application à court terme ; on nous promet de toutes façons des outils pour aider la migration future de nos applications vers la version 2.

DOCTRINE, PASSAGE À LA VERSION 2

La version 2 de doctrine dont l'architecture est basée sur JSR-317 (JPA v2) a été réécrite entièrement afin d'améliorer les performances (suppression des méthodes magiques, suppression de l'active record), et simplifier l'API. Notez cependant que Doctrine2 n'est supportée qu'à partir de PHP 5.3 (utilisation des namespaces etc.). Le langage DQL a été amélioré, il est plus rapide, plus propre, plus flexible. Cette bêta est proche de la version stable et peut être utilisée en environnement de production, d'autant qu'une migration Doctrine 1 à 2 peut s'avérer délicate.

PREMIERS PAS VERS UNE APPLICATION ZF

Commençons par récupérer la dernière version du Zend Framework. Depuis la version 1.8, Zend_Tools permet l'utilisation du ZF en ligne de commande pour les tâches répétitives ou un peu rebutantes (ie : créer une application, un controller, une vue, ...).

Pour créer notre application, il suffit donc maintenant de taper une seule ligne de commande :

```
/var/www/ZendFramework-1.11.0/bin$ ./zf.sh create project /var/www/monapplication
Creating project at /var/www/monapplication
Note: This command created a web project, for more information
setting up your VHOST, please see docs/README
```

L'arborescence d'un projet par défaut est automatiquement créée :

```
/var/www/monapplication$ ls -l
total 20
```

```
drwxr-xr-x 7 leto leto 4096 2010-11-29 14:29 application
drwxr-xr-x 2 leto leto 4096 2010-11-29 12:16 docs
drwxr-xr-x 4 leto leto 4096 2010-11-29 20:17 library
drwxr-xr-x 2 leto leto 4096 2010-11-29 14:23 public
drwxr-xr-x 4 leto leto 4096 2010-11-29 12:16 tests
```

Un simple lien symbolique ou une copie du répertoire Zend dans notre library termine l'installation :

```
/var/www/monapplication$ ln -s /var/www/ZendFramework-1.11.0/library/Zend library/Zend
```

Nous pouvons déjà lancer le navigateur pour un premier résultat, notre application ZF est prête ! [Fig.1]

INTEGRATION DE LA LIBRAIRIE DOCTRINE2 ET SON « PLUGIN »

Maintenant il nous faut télécharger et copier les librairies Doctrine (Comon, DBAL et ORM) dans le répertoire library de notre application :

```
/var/www/monapplication/library/
```

Archives à récupérer sur le site officiel : <http://www.doctrine-project.org/projects>

Pour faire fonctionner ZF et Doctrine, une librairie ZendX existe et permet de faire cohabiter les deux systèmes de manière harmonieuse.



se, il nous faut la télécharger et l'installer dans le répertoire library :

```
/var/www$ git clone https://github.com/nodevo/ZendX_Doctrine2.git
Initialized empty Git repository in /var/www/ZendX_Doctrine2/.git/
remote: Counting objects: 11, done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 11 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (11/11), done.
/var/www$ cp -R ZendX_Doctrine2/ZendX/ monapplication/library/
```



Le répertoire library compte maintenant 3 sous-répertoires « Doctrine », « Zend » et « ZendX » comme suit : [Fig.2].

CONFIGURATION DE NOTRE APPLICATION

Le fichier de configuration de notre application créé par défaut lors de la création du projet doit être enrichi afin de permettre l'utilisation du plugin ZendX_Doctrine2 et de configurer l'ORM. Eléments à ajouter à notre fichier /application/configs/application.ini :

```
;Plugin
pluginPaths.ZendX_Doctrine2_Application_Resource_ = «ZendX/Doctrine2/Application/Resource»

;EntityManager settings
resources.entitymanagerfactory.cache = «Doctrine\Common\Cache\ArrayCache»
resources.entitymanagerfactory.metadata.classDirectory = APPLICATION_PATH «/models/»
resources.entitymanagerfactory.metadata.driver = «annotation»
resources.entitymanagerfactory.proxyDir = APPLICATION_PATH «/proxies/»
resources.entitymanagerfactory.proxyNamespace = «Application_Proxies»
resources.entitymanagerfactory.autoGenerateProxyClasses = true
resources.entitymanagerfactory.useCExtension = false

;EntityManager Database settings
resources.entitymanagerfactory.connectionOptions.driver = «pdo_mysql»
resources.entitymanagerfactory.connectionOptions.host = «localhost»
resources.entitymanagerfactory.connectionOptions.user = «dbuser»
resources.entitymanagerfactory.connectionOptions.password = «secret»
resources.entitymanagerfactory.connectionOptions dbname = «monapplication_db»
```

Le fichier public/index.php doit également être enrichi juste avant le lancement du bootstrap afin d'accueillir la déclaration de la nouvelle resource :

```
$resourceLoader = new Zend_Loader_Autoloader_Resource(array(
    'basePath' => APPLICATION_PATH,
    'namespace' => 'Example'
));
```

```
$resourceLoader->addResourceType('model', 'models/', 'Entities');

$application->bootstrap()
    ->run();
```

CREATION DES ENTITIES ET REQUETAGE SIMPLE

Pour notre exemple, nous utiliserons la base suivante :

```
CREATE TABLE `auteur` (
    `au_id` int(10) unsigned NOT NULL AUTO_INCREMENT,
    `au_nom` varchar(45) NOT NULL,
    PRIMARY KEY (`au_id`)
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;

CREATE TABLE `livre` (
    `li_id` int(10) unsigned NOT NULL AUTO_INCREMENT,
    `li_titre` varchar(45) NOT NULL,
    `au_id` int(10) unsigned NOT NULL,
    PRIMARY KEY (`li_id`)
) ENGINE=MyISAM AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;
```

Le schéma apporte son lot de nouveautés côté syntaxe. Doctrine propose 3 méthodes : XML, Yaml et DocBlock (sous forme d'annotations PHP). A vous de faire votre choix; dans notre exemple la méthode DocBlock est retenue (à vous de configurer le bon Driver (setMetadataDriverImpl()), si vous souhaitez changer de méthode). La méthode recommandée par Doctrine est celle des annotations DocBlock. Voici ce que cela donne pour notre table auteur (fichier application/models/Auteur.php) :

```
<?php
/** @Entity @Table(name=»auteur») */
class Entities_Auteur
{
    /**
     * @Id @Column(name=»au_id», type=»integer»)
     * @GeneratedValue(strategy=»AUTO»)
     */
    private $id;

    /** @Column(name=»au_nom», type=»string», length=50) */
    private $nom;

    /**
     * Get id
     * @return integer $id
     */
    public function getId()
    {
        return $this->id;
    }
    /**
     * Set name
     * @param string $nom
     */
    public function setNom($nom)
    {
```

```

    $this->nom = $nom;
}

/**
 * Get name
 * @return string $name
 */
public function getNom()
{
    return $this->nom;
}
?
```

Tous les champs de la table sont déclarés en tant que propriétés *private* de la classe. Il faut donc pour chacune d'elles déclarer les setters/getters; à noter que la plupart des IDE permettent de les générer automatiquement.

Une classe nommée ActiveEntity qui se trouve dans le namespace DoctrineExtensions peut être étendue et permet d'étendre automatiquement les setters/getters magiques ainsi que les méthodes save(), toArray(), etc. Cependant, il n'est pas recommandé de l'utiliser et cela reste contre nature.

La connexion à la base de données est fournie par un EntityManager (em), celui ci est stocké dans notre cas dans le registre Zend, accessible de la façon suivante :

```
$em = Zend_Registry::get('em');
```

Cette classe va nous permettre de récupérer, sauver ou créer des entities; par exemple pour ajouter un auteur dans la base, je procède de la façon suivante :

```

$nouvelAuteur = new Entities_Auteur();
$nouvelAuteur->setNom('Victor Hugo');
$em->persist($nouvelAuteur);
$em->flush();
```

La synchronisation avec la base s'effectue lors de l'appel de la méthode flush(), les modifications des objets persistants stockés dans le UnifOfWork sont alors basculées dans la base de données. La construction d'une requête plus complexe (avec des jointures par exemple) demandera de faire appel au DQL. Le DQL de Doctrine 2 a été amélioré, mais le fonctionnement reste identique. C'est de toute façon la méthode la plus puissante et la plus flexible pour récupérer des objets depuis la base de données. Ce langage permet d'interroger votre bd dans un langage objet, avec les noms de classes, les champs, les héritages ... sans se soucier du nom des champs physiquement en base. Il est très similaire au SQL, mais ce n'est PAS du SQL.

Par exemple, listons les auteurs commençant par un 'a' dans notre base par une requête simple :

```

$dql = $em->createQuery('SELECT a FROM Entities_Auteur a WHERE
a.nom LIKE «a%»');
$ auteurs = $dql->getResult();
```

Les méthodes magiques findBy* quant à elles existent toujours, exemple :

```

$ auteur = $em->getRepository('Entities_Auteur')->findOneByNom
('Oscar Wilde');
```

LIER VOS OBJETS

L'intérêt majeur de Doctrine est aussi de pouvoir décrire les dépendances entre les tables dans les fichiers de mapping; exemple, lisons les tables auteur et livre : Ajouter dans le fichier models/Auteur.php :

```

/** @OneToOne(targetEntity="Entities_Livre", mappedBy="auteur",
cascade={"persist", "remove", "merge"}) */
private $livres;
```

Créer le fichier models/Livre.php et déclarer la dépendance avec Auteur de la manière suivante :

```

/** @ManyToOne(targetEntity="Entities_Auteur", cascade=
{"persist", "delete"})
 * @JoinColumn(name="au_id", referencedColumnName="au_id")
 */
private $auteur;
```

Nous pouvons maintenant lister les auteurs et leurs livres :

```

echo '<ul>';
foreach ($auteurs as $auteur) {
    echo '<li>' . $auteur->getNom() . '<ul>';
        foreach ($auteur->getLivres() as $livre) {
            echo '<li>' . $livre->getTitre() . '</li>';
        }
    echo '</ul></li>';
}
echo '</ul>';
```

Par les mêmes procédés, il est possible de créer des éléments et d'effectuer des suppressions en cascade.

■ Damien Camus – NODEVO
dcamus@nodevo.com



Nouveau : économisez jusqu'à 50%

Abonnement 2 ans au magazine + 1 livre numérique ENI

• **79€ 2 ans au magazine** au lieu de 130,90 (valeur de 22 numéros) Tarif France métropolitaine

+ un livre d'une valeur de 23,9 € à 31,9 €, soit un total de 154,8 € à 162,8 €

• **89€ 2 ans au magazine + archives sur Internet et PDF + 1 livre numérique ENI**

Livres à Choisir : • Visual Studio 2010 • PHP5.3 • Bing Maps • MySQL 5, Administration et optimisation

• Java et Spring, Concevoir, construire et développer une application Java/J2EE avec Spring.

Coupon d'abonnement p. 77 et sur www.programmez.com/abonnement.php



La dépendance binaire selon NuGet

Après une longue traversée du désert, le monde .NET va sans doute enfin disposer d'un nouveau standard en matière de gestion des dépendances binaires. Ce standard n'est autre que NuGet (anciennement NuPack), issu de l'initiative des équipes Web Platform & Tools de chez Microsoft et des membres du Nu Project. Il vient jusque dans votre Visual Studio pour vous apporter le meilleur des librairies du monde Open Source en quelques clics.



APPLICATION : binaire

LANGAGE : .Net

SOURCE : oui

Depuis de longues années, le monde Java possède un outil nommé Maven, qui, avec ses fichiers « pom.xml » permet de définir des packages et leurs dépendances (Maven fait aussi bien plus que cela). De plus il existe des implémentations reconnues de « repositories », véritables catalogues de composants publiquement accessibles avec Maven. D'un autre côté, on constate également le succès de la galerie Ruby Gems (<http://rubygems.org>) hébergeant pas moins de 1200 références. Pourquoi dans le monde Microsoft n'y-a-t-il pas d'équivalent ? Ce ne sont pas les tentatives qui ont manqué. Pour en citer quelques-unes : Nu, Refix, NPanday (intégration .NET dans le Maven de Java), HornGet, etc. Mais elles n'ont pas su s'imposer... Pourtant le besoin est là, dès lors que l'on veut industrialiser les développements, la question de la gestion des références et des packages se pose et chacun se retrouve à implémenter son propre système de gestion et de mutualisation des dépendances binaires, à un certain prix il faut le dire.

LE PROJET NUGET

NuGet est donc un outil de gestion des packages binaires orienté développement (on ne parle pas de packages livrables ici). Il est en partie inspiré de Ruby Gems et il permet d'incorporer des librairies tierces et leurs dépendances dans vos projets et ce aussi simplement que possible. Grâce à NuGet, vous n'aurez plus le réflexe de recopier les binaires depuis le dernier projet où vous les avez utilisés ! De même, tout est facilité afin que les auteurs de librairies créent leurs propres packages, et alimentent l'écosystème de librairies libres du monde .NET. Microsoft héberge ainsi une galerie par défaut qui contient à ce jour plus de 650 références (à peine 250 il y a deux mois), le projet est en plein boom ! A vous les NHibernate, Elmah, Castle, SQLCE, Mvc*, Fluent*, Reactive Fx, qui sont disponibles parfois en plusieurs versions. Il faut également noter que NuGet n'est pas un projet Closed Source comme les autres, c'est un projet Open Source sur CodePlex, où à la fois Microsoft et des personnes externes participent, or ce n'est pas dans les habitudes de Microsoft d'accepter les contributions ! La gestion des bugs est transparente sur CodePlex. Ce projet est pourtant stratégique car il est livré avec MVC3 et soutenu de manière visible par les ténors du développement Web chez Microsoft.

VOUS AVEZ DIT PACKAGE ?

Typiquement un package va contenir des assemblies à référencer. Un gestionnaire de package qui se respecte se doit également de gérer la transitivité des dépendances : une librairie peut dépendre

d'autres librairies dans des versions spécifiques. Il est alors beaucoup plus commode d'avoir l'outil qui est capable d'aller chercher les bonnes versions pour nous que de tout faire à la main [Fig.1].

Pour utiliser certaines librairies, il faudra apporter des déclarations dans les fichiers `web.config` ou `app.config`. D'autres librairies vont aussi avoir besoin de véhiculer des fichiers de type « contenu » (des modèles par exemple), ou bien des sources qui pourront être placés au sein de vos projets, en prenant soin de placer le bon Namespace dans les déclarations, c'est-à-dire celui qui correspond à votre projet. Le Package Manager de NuGet est capable d'automatiser toutes ces tâches depuis Visual Studio. On ne parle donc pas de packages livrables car il n'y a pas d'opération dans le GAC ni dans la base de registre, tout se passe dans les fichiers dans le scope de votre solution. NuGet se positionne ainsi en tant que complément du fameux Web Platform Installer et du gestionnaire d'extensions de Visual Studio. Lors de l'ajout d'un package, NuGet va créer un répertoire « packages » à côté de votre solution et y décompresser les packages qui ne sont autres que des fichiers zip à la norme OPC (Open Packaging Format). NuGet applique aussi les transformations nécessaires sur les sources et les altérations éventuelles du fichier de configuration, enfin, il rajoute en direct les références aux assemblies dans votre projet Visual Studio [Fig.2].

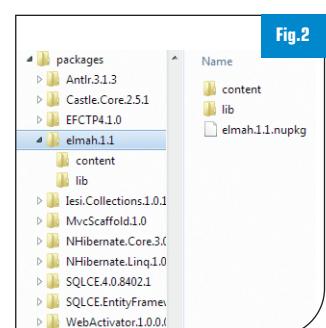
Cet ajout des binaires dans la structure de répertoires de votre solution est presque simpliste mais fonctionne bien avec les contrôleurs de source et notamment lorsque l'on travaille en équipe. Elle est également transparente pour les builds puisque quiconque récupère l'arborescence pourra compiler. Pour économiser de la place on pourra aussi placer les packages « plus haut » dans une arborescence de sources multi-solutions, mutualisant ainsi les packages pour toutes les solutions, mais cette fonctionnalité, bien qu'annoncée, n'est pas encore disponible à l'heure actuelle.

COMMENT UTILISER NUGET

Passons maintenant à la pratique, tout d'abord, passez par le gestionnaire d'extensions de Visual Studio 2010 afin de rechercher et installer l'extension « NuGet ». Ensuite, il y a deux manières d'utiliser



La pêche aux librairies n'est jamais une partie de plaisir



L'arborescence des packages créée par NuGet

le Package Manager de NuGet à savoir la console et la boîte de dialogue de référencement.

Du côté console, on dispose d'une nouvelle fenêtre de commandes nommée Package Manager Console, dans laquelle on pourra appeler des commandes PowerShell explicites telles que :

- List-Package : énumération de tous les packages installés
- List-Package -remote : énumération de tous les packages dans la source sélectionnée
- Install-Package : installation d'un package dans le projet sélectionné [Fig.3].

On notera les efforts fournis pour procurer de la convivialité : l'Intellisense est supporté pour les commandes, mais aussi pour compléter les noms des packages grâce à l'utilisation du standard OData

(AtomPub sur du REST) qui permet d'utiliser des requêtes riches et donc de filtrer côté serveur. En effet, avec le temps on peut s'attendre à voir des milliers de packages disponibles, chacun se déclinant potentiellement en plusieurs versions.

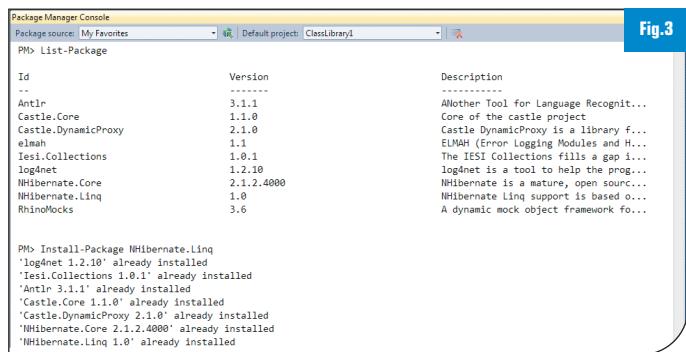
Les références sont mises à jour automatiquement dans Visual Studio. Ainsi, en une seule commande notre projet a été mis à jour avec toutes les dépendances de NHibernate.LINQ : un gain de temps non négligeable ! [Fig.4].

L'autre possibilité est d'utiliser l'IHM intégrée à Visual Studio, on trouvera une nouvelle entrée dans

le menu contextuel des projets nommée « Add Library Package Reference... ». La boîte de dialogue qui s'en suit reprend le design du gestionnaire d'extensions mais présente les packages NuGet. On peut facilement identifier les packages déjà installés, visualiser les mises à jours disponibles, rechercher un package ou tout simplement naviguer dans la liste [Fig.5].

Une des forces de NuGet est donc sa simplicité, et il s'intègre parfaitement dans nos habitudes de développement avec Visual Studio, il est en effet difficile de faire plus intuitif. Il est vraiment pratique de pouvoir détecter les mises à jour disponibles et de les appliquer d'un seul clic !

Une fois les packages installés par NuGet et archivés dans le contrôleur de sources, les autres développeurs n'auront pas besoin de refaire ces opérations. Par principe, la solution doit marcher sur



La console de gestion des packages NuGet.

tous les autres postes qui récupèrent la dernière version des sources, c'est en bonne partie pour cette raison très louable que NuGet s'interdit de toucher à d'autres zones du poste de travail, les autres développeurs n'ont dans l'absolu pas besoin d'installer NuGet !

SE CONNECTER À D'AUTRES SOURCES DE PACKAGES

Un des points clés avec NuGet est son ouverture à de multiples sources de packages. Vous trouverez dans les options la possibilité de rajouter vos propres galeries. Il pourra s'agir de « feeds » OData, ou bien d'un simple répertoire (local ou partage réseau) dans lequel vous aurez copié des packages [Fig.6].

NuGet fournit l'application ASP.NET qui permet d'héberger une galerie sous forme de feeds, vous n'avez plus qu'à copier vos packages dans le répertoire adéquat et le tour est joué, vous avez votre serveur de packages maison ! Cette ouverture est importante, car elle permet à des sociétés ayant une gestion stricte des librairies Open Source de ne mettre à disposition que les packages validés en interne. De même, le système permet typiquement d'héberger les composants maison que l'on ne veut surtout pas rendre publics.

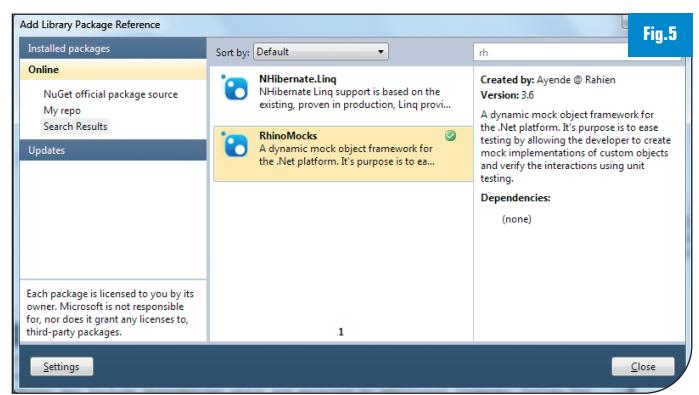
CRÉEZ VOTRE PACKAGE

Nous allons utiliser l'outil NuGet.exe qu'il faut télécharger à part (répérez simplement le package **NuGet.CommandLine**!). Vous pouvez par exemple déployer l'exécutable sur votre machine dans le répertoire %VS100COMNTOOLS% (en mode admin) :

```
copy nuget.exe %VS100COMNTOOLS%
```

Ensuite, rajoutez un répertoire « Packaging » à votre projet, puis ajoutez un fichier manifeste nommé « MyPublicLib.nuspec » :

```
<?xml version="1.0" encoding="utf-8"?>
<package>
  <metadata>
    <id>MyPublicLib</id>
    <version>1.0.8</version>
    <authors>Vincent Labatut</authors>
    <description>Librairie qui permet de faire plein de trucs</description>
    <language>fr-FR</language>
    <dependencies>
      <dependency id="log4net" version=">[1.2.10]</dependency>
    </dependencies>
  </metadata>
  <files>
    <file src="bin\Release\MyPublicLib.dll" target="lib\net40" />
  </files>
</package>
```



Recherche d'un package.

```

<dependency id="Unity" version="2.0" />
</dependencies>
</metadata>
</package>

```

Prenez soin de préciser les noms des packages dont vous dépendez ainsi que leurs versions souhaitées : ici log4net en version exactement 1.2.10 et Unity en version au moins 2.0.

Si votre librairie nécessite des références particulières au Framework .NET, ajoutez un script PowerShell nommé « Install.ps1 » dans « Packaging\Tools » :

```

param($installPath, $toolsPath, $package, $project)
$project.Object REFERENCES.Add("System.Security")
$project.Object REFERENCES.Add("System.ServiceModel")

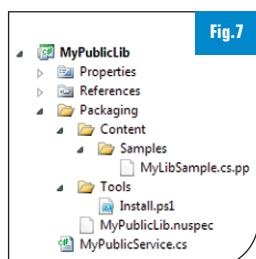
```

Enfin, admettons que vous vouliez fournir un exemple ou un modèle de classe type pour votre librairie (ce qui devrait faire l'objet d'un package à part pour ne pas polluer tous vos clients), vous pouvez rajouter un fichier source qui sera transformé automatiquement à l'installation du package dans le projet. En suffixant ainsi les sources de l'extension « .pp », ces derniers vont être interprétés et les propriétés du projet Visual Studio vont être remplacées :

```

namespace $rootnamespace$.MyLibSample
{
    public class MyLibSample : MyLibBase
    {
        public override void OverrideMe()
        {
            // etc.
        }
    }
}

```



Structure des répertoires du projet à packager

Pour chaque fichier du répertoire Packaging, assurez-vous qu'il soit en mode « Copy Always » dans ses propriétés Visual Studio. Voici la structure finale de l'exemple proposé : [Fig.7].

Afin de générer le package à chaque compilation, il suffit de rajouter cette ligne de commande dans l'événement Post-Build de votre projet :

```

md Packaging\lib
copy MyPublicLib.* Packaging\lib
"%VS100COMNTOOLS%\nuget" pack Packaging\MyPublicLib.nuspec -o c:\LocalPackages

```

Vous pouvez ensuite tester très facilement votre package en rajoutant c:\LocalPackages à vos sources de packages NuGet !

UTILISATION AVANCÉE DE NUGET

Les packages peuvent aussi contenir des définitions de commandes PowerShell pour déclencher des nouvelles actions. Ainsi la librairie **MvcScaffolding** contient la définition de la commande **Scaffold** qui permet de rajouter une vue dans un projet MVC. Ce système de déploiement simplifié de commandes PowerShell ouvre la porte à des scénarii d'entreprise et d'automatisation très prometteurs ! Autre astuce, on peut par exemple définir une meta-librairie : un package qui ne contient rien que des références, ce qui vous permettra d'installer une collection de packages et leurs dépendances en une seule commande. Cela peut être utile si vous utilisez toujours les mêmes librairies pour les nouveaux projets.

DES LIMITATIONS

Abordons maintenant les sujets qui fâchent : NuGet nécessite PowerShell 2.0, et donc ne tournera qu'à partir de Windows Vista, Windows 7 et Windows Server 2008, de plus il ne fonctionne que sur Visual Studio 2010, WebMatrix et Visual Web Developper 2010 Express (les autres éditions Express seront supportées). Enfin, c'est un projet encore en pleine ébullition, les formats de packages ont évolué entre la CTP1 et la CTP2 ce qui provoquait des plantages dans les commandes quand on manipulait des vieux packages, maintenant que la version 1.0 est sortie espérons que cela ne se reproduira plus !

UN AVENIR CERTAIN...

On est en droit d'espérer que NuGet vienne à s'imposer comme LE standard de gestion des dépendances binaires, simplifiant ainsi la vie de tous les jours des développeurs .NET et enlevant une épine du pied aux sociétés qui veulent industrialiser leurs développements à moindre coût.

Si NuGet ne permet pas de résoudre les conflits, il les met en évidence aux yeux de toute la communauté de façon telle que des solutions devraient être rapidement proposées, augmentant ainsi significativement la facilité d'accès à toutes ces librairies du monde libre.

Enfin, les *galleries* de composants NuGet voient le jour comme par exemple <http://nuget.org>.

Cette dernière s'appuie sur le moteur CMS Orchard (<http://orchardproject.net>) avec un connecteur vers une source NuGet.

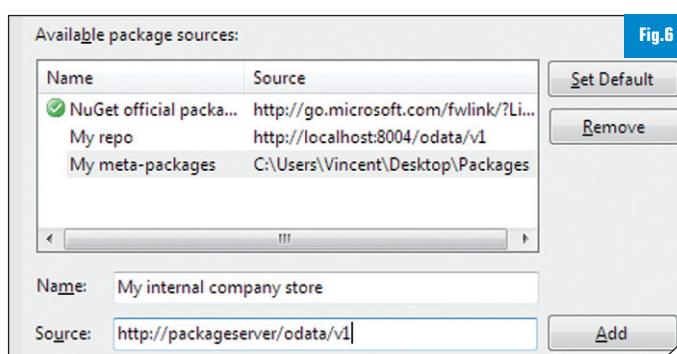
Pour terminer, retenez principalement cette URL : <http://nuget.codeplex.com>, où vous trouverez la documentation de NuGet ainsi que tous les liens nécessaires si vous voulez participer à l'aventure. A vos packages !



■ Vincent Labatut

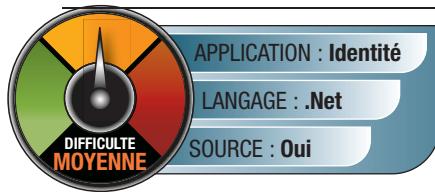
Microsoft MVP Visual Studio ALM et architecte .NET pour les Sogeti .Net Rangers
Blog : <http://blogs.codes-sources.com/vlabz>

Redéfinir les sources de packages



Windows Identity Foundation

La gestion de l'identité des utilisateurs est une tâche récurrente du développement des applications internet, qu'il s'agisse de réellement contrôler l'accès à certaines ressources ou simplement de personnaliser l'expérience utilisateur. Dans cette gestion, il convient de séparer deux parties : la vérification de l'identité du visiteur (authentification) et l'exploitation de cette identité (autorisations, personnalisation, etc.).



La gestion des autorisations ainsi que la personnalisation peuvent selon les cas relever d'une configuration technique assez simple ou de vraies règles applicatives mais la vérification des identités est quasiment toujours une tâche purement technique, n'ayant pas de rapport direct avec l'application qui exploite ensuite les identités « reconnues », aussi il est généralement bénéfique d'extraire cette vérification de l'application elle-même. Cette séparation présente des bénéfices à la fois pour le développeur de l'application et pour ses utilisateurs et elle permet en outre des scénarios de fédération d'identité impossibles à réaliser si chaque application définit ses propres mécanismes d'authentification. Dans le cas d'applications internes dans des environnements 100% Windows, l'authentification des utilisateurs est généralement confiée à Active Directory : cette approche est fiable, immédiatement disponible aux applications ASP.NET et permet aux utilisateurs de bénéficier du confort d'un Single Sign On. Dans le cas d'applications internet ou d'environnements hétérogènes, il faut trouver une alternative. Par ailleurs, les solutions b2b ont besoin de mécanismes d'authentification plus riches comme la fédération d'identité : imaginons une entreprise A qui offre des services aux utilisateurs de ses partenaires. Il peut s'agir par exemple d'une agence de voyage d'affaires qui s'occupe des réservations des employés de ses clients, d'un site de commerce en ligne offrant des réductions aux clients de certains partenaires, etc.

Imaginons que A authentifie ses utilisateurs avec une base qui lui est propre : son client/partenaire B doit alors lui transmettre périodiquement la liste de ses utilisateurs afin que ceux-ci puissent utiliser l'application de A et cela induit des délais à la fois lors des créations et lors des suppressions des utilisateurs venant de B, sans compter les éventuelles modifications si les services proposés ne sont pas identiques pour tous les utilisateurs. Pour ceux-ci mêmes, cette solution n'est pas idéale car ils doivent retenir un mot de passe spécifique à A (il est difficile d'imaginer que leur mot de passe propre à B soit aussi transmis à A !!!). La fédération d'identité consiste à établir une relation de confiance entre A et B et à appuyer le fonction-

nement de l'application de A sur une authentification réalisée par B : lorsqu'un utilisateur se présente sur A, son identité (ainsi que certains attributs associés comme par exemple l'appartenance à des groupes,) est « garantie » par B. Ces informations sont appelées des « Claims » et présentées à A directement par l'utilisateur : pour que A les accepte, il faut qu'ils soient signés par B (et que A soit configuré correctement pour reconnaître cette signature) ; généralement, ils sont véhiculés dans un token de sécurité au format SAML. La [Fig.1] illustre ce principe. En pratique, le schéma est généralement légèrement plus compliqué car l'utilisateur va d'abord s'adresser à A, qui va le rediriger vers B pour s'authentifier, tout en fournissant à B l'adresse de retour, la bascule A vers B puis le retour à A est alors automatique et quasi transparent pour l'utilisateur. Il faut noter que ce mécanisme d'externalisation de l'authentification est déjà largement répandu sur Internet, typiquement avec tous les sites qui acceptent les authentifications à l'aide de Windows Live ID. Windows Identity Foundation est un des composants Microsoft constituant la plate-forme « Geneva » avec ADFS 2.0 et Cardspace. Concrètement il s'agit d'un ensemble de composants .NET permettant la mise en place de ce type d'authentification dans les applications ASP.NET. Cet article d'initiation à WIF va présenter sa mise en œuvre au travers de plusieurs scénarios pratiques :

- Modification d'une application ASP.NET pour déléguer l'authentification des utilisateurs à une autre application, appelée STS (Security Token Service)
- Enrichissement du STS pour incorporer dans les Claims des données complémentaires comme la date de naissance de l'utilisateur
- Utilisation d'un STS pour authentifier l'utilisateur dans un scénario de type RIA appelant un service WCF.

DÉLÉGATION DE L'AUTHENTIFICATION À UN STS

WIF permet de déléguer l'authentification à un STS et de récupérer l'ensemble des « claims » renseigné par celui-ci tout en s'intégrant complètement aux mécanismes de sécurité habituels d'ASP.NET, c'est-à-dire :

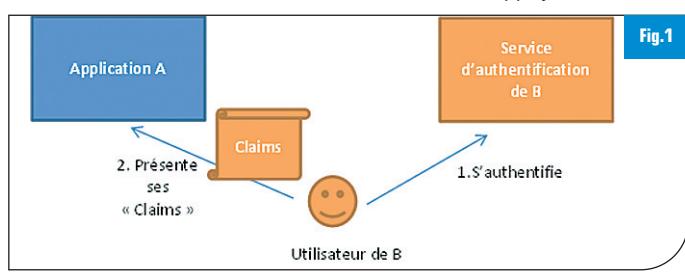
- Possibilité de conditionner l'accès à certaines parties du site par des entrées dans le fichier web.config comme par exemple :

```

<location path=>R1</>
<system.web>
  <authorization>
    <allow roles=>admin</>
    <deny users=>*</>
  </authorization>
</system.web>
</location>

```

Autorise l'accès au répertoire "admin" uniquement aux utilisateurs possédant le rôle « admin »



Externalisation de l'authentification

```

<system.web>
  <authorization>
    <deny users=>?</deny>
    <allow roles=>*</allow>
  </authorization>
</system.web>
</location>

```

```

<location path=>R3</>>
  <system.web>
    <authorization>
      <allow users=>*</allow>
    </authorization>
  </system.web>
</location>

```

- Possibilité d'accéder à l'identité de l'utilisateur courant à travers :

`Thread.CurrentPrincipal`
`Context.User`

- Dans le cas de la mise en place d'un STS avec WIF, ces deux propriétés retournent un objet de la classe `ClaimsPrincipal` qui implémente l'interface `IPrincipal` attendue. La propriété `Identity` de cette classe retourne un objet de type `ClaimsIdentity` qui permet à son tour de récupérer les Claims comme on le verra plus loin.

Pour ajouter le support d'un STS à un site AS.NET existant, il suffit de faire un clic droit sur le projet du site sous Visual Studio et de sélectionner l'option « Add STS Reference » : [fig.2]

Cette action lance un assistant qui va poser quelques questions puis mettre à jour le site web. La première étape permet de préciser le chemin du fichier `web.config` ainsi que l'URL de l'application qui est importante : en effet, elle permettra au STS de valider quelle application l'appelle (pour des raisons de sécurité, il ne doit pas envoyer les informations relatives aux utilisateurs à n'importe quelle application web l'invoquant) : [fig.3]. La deuxième étape est plus « intéressante » : elle permet en effet de choisir entre la création d'un STS dans l'application courante ou l'utilisation d'un STS existant : si on dispose par exemple d'un serveur AD FS 2.0 (Active Directory Federation Service), on va plutôt le référencer qu'en créer un nouveau. Dans le cadre de notre exemple, c'est néanmoins l'option de création qu'on va choisir : [fig.4]. La dernière étape est un récapitulatif des différentes modifications effectuées : [fig.5]. Elle précise en parti-

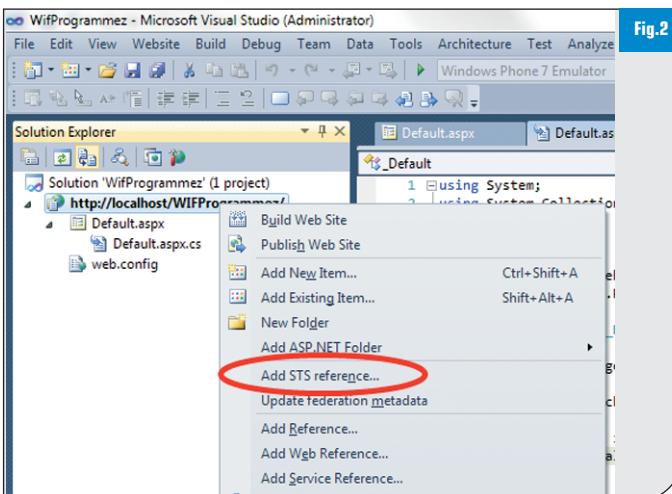


Fig.2

culier qu'un nouveau document, `FederationMetadata.xml`, a été ajouté au site initial : ce document est récupéré auprès du STS qui décrit ainsi quels claims il est capable de renseigner. Les claims insérés par défaut permettent l'utilisation de l'identité des utilisateurs et de la notion de rôle dans l'application ASP.NET :

```

...
<fed:ClaimTypesRequested>
  <auth:ClaimType Uri=>http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name...</auth:ClaimType>
  <auth:ClaimType Uri=>http://schemas.microsoft.com/ws/2008/06/identity/claims/role...</auth:ClaimType>
</fed:ClaimTypesRequested>
...

```

L'application ASP.NET est configurée par l'assistant pour interdire les connexions anonymes, sauf pour le fichier `FederationMetadata.xml`.

- Les modifications effectuées dans le `web.config` de l'application ASP.NET (l'ancien fichier est néanmoins sauvegardé par l'assistant sous le nom `web.config.backup`) sont les suivantes :
 - modification des autorisations comme décrit précédemment,
 - ajout de la référence à `Microsoft.IdentityModel`
 - ajout des modules http `WSFederationAuthenticationModule` et `SessionAuthenticationModule`
 - création d'une section `microsoft.identityModel` récapitulant entre autres : l'url du STS, les claims demandés, le « thumbprint » du

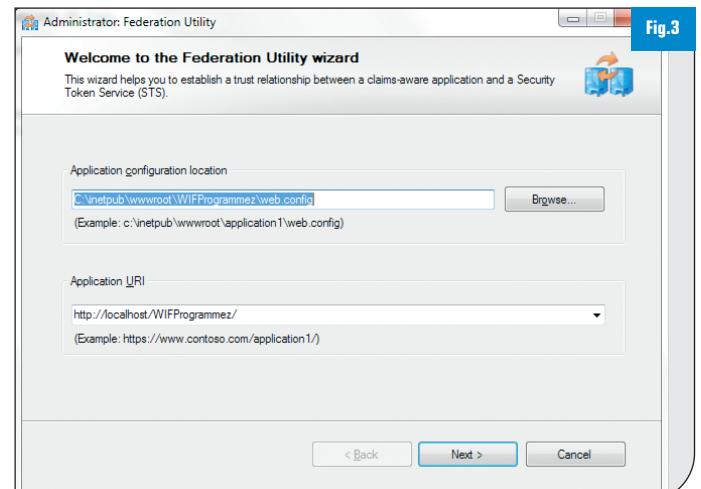


Fig.3

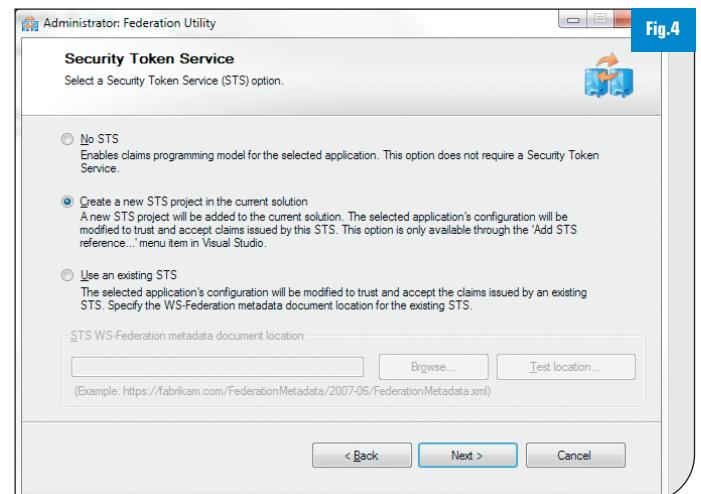


Fig.4

certificat du STS qui va permettre aux modules précédents de vérifier que les tokens de sécurité reçus ont bien été émis par le STS « de confiance ».

On peut maintenant vérifier le fonctionnement de l'authentification mise en place : quand on appelle l'application ASP.NET, on est immédiatement redirigé vers le STS, dont on peut bien évidemment personnaliser la page d'authentification : [Fig.6].

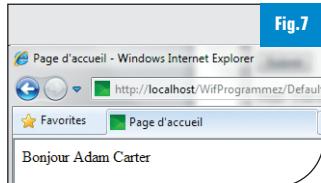


Fig.7

Une fois « Submit » cliqué, l'utilisateur est redirigé vers le site initial et on peut voir sur l'écran ci-dessous que la propriété Context.User est correctement renseignée : [Fig.7].

```
Default.aspx
Bonjour <asp:Literal ID=>NameLiteral</asp:Literal>
Default.aspx.cs
protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        IIdentity id = Context.User.Identity;
        NameLiteral.Text = (id.IsAuthenticated)
            ? id.Name : «Anonymous»;
    }
}
```

Remarque

Dans la première étape de l'assistant, nous avons passé sous silence l'option "No STS". Cette option permet de modifier le web.config de l'application ASP.NET pour incorporer l'utilisation du module ClaimsPrincipalHttpModule sans pour autant externaliser l'authentification des utilisateurs vers un STS externe.

Quel est l'intérêt de cette option ? Par exemple il sert à baser le code de l'application sur les claims à la place des rôles classiques sans nécessiter une authentification externe.

Pour exploiter le ClaimsPrincipalHttpModule inséré par l'assistant, il suffit par exemple d'activer l'authentification Windows au sein d'IIS Manager (le module fonctionne aussi avec une authentification par formulaire mais cela nécessite alors le développement de celui-ci).

ENRICHISSEMENT DU STS

Dans l'exemple précédent, les claims rentrés par le STS étaient assez simples (nom et rôles) mais permettaient déjà d'obtenir les principales fonctions d'autorisation d'ASP.NET associées à une authentification externe. Néanmoins, le STS peut retourner des informations plus complètes sur l'utilisateur authentifié si nécessaire : il pourrait s'agir par exemple de la durée d'un abonnement. Nous allons maintenant enrichir le STS de la solution précédente pour retourner de telles informations et les exploiter dans l'application ASP.NET. Du côté du STS, il faut d'abord modifier le fichier FederationMetadata.xml pour décrire quels sont les nouveaux claims que le STS peut publier ; pour ajouter la date de naissance et le code postal, on insère les lignes suivantes :

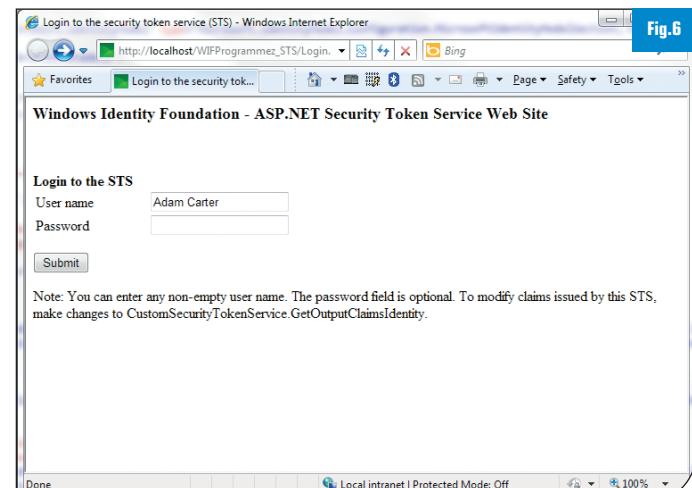
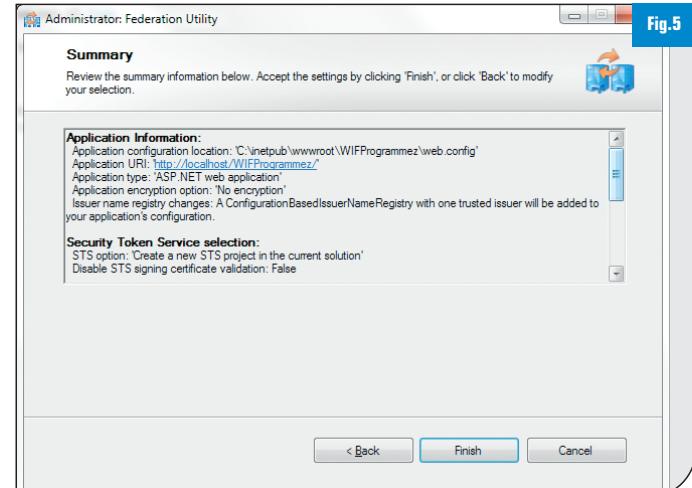
```
<auth:ClaimType Uri=>http://schemas.xmlsoap.org/ws/2005/05/identity/claims/dateofbirth</auth:ClaimType>
<auth:ClaimType Uri=>http://schemas.xmlsoap.org/ws/2005/05/identity/claims/postalcode</auth:ClaimType>
<auth:ClaimType Uri=>http://schemas.xmlsoap.org/ws/2005/05/identity/claims/postalcode</auth:ClaimType>
<auth:ClaimType Uri=>http://schemas.xmlsoap.org/ws/2005/05/identity/claims/postalcode</auth:ClaimType>
```

```
<auth:DisplayName>Date of Birth</auth:DisplayName>
<auth:Description>The date of Birth of the subject.</auth:Description>
</auth:ClaimType>
<auth:ClaimType Uri=>http://schemas.xmlsoap.org/ws/2005/05/identity/claims/postalcode</auth:ClaimType>
<auth:DisplayName>Postal Code</auth:DisplayName>
<auth:Description>The Postal Code of the subject.</auth:Description>
</auth:ClaimType>
```

Les principales Uri correspondant à ces claims sont déclarées dans la classe Microsoft.IdentityModel.Claims. ClaimTypes. On utilise d'ailleurs cette classe pour ajouter les claims dans la méthode GetOutputClaimsIdentity de la classe CustomSecurityTokenService du STS (valeurs « en dur » dans cet exemple) :

```
outputIdentity.Cards.Add( new Claim( System.IdentityModel.Claims.ClaimTypes.Name, principal.Identity.Name ) );
outputIdentity.Cards.Add( new Claim( ClaimTypes.Role, «Manager» ) );
outputIdentity.Cards.Add( new Claim(ClaimTypes.DateOfBirth, new DateTime(1900, 9, 13).ToString()) );
outputIdentity.Cards.Add( new Claim(ClaimTypes.PostalCode, «91210»));
```

Du côté de l'application ASP.NET sécurisée, on peut mettre à jour le



Les outils des Décideurs Informatiques

*Vous avez besoin d'info
sur des sujets
d'administration,
de sécurité, de progiciel,
de projets ?
Accédez directement
à l'information ciblée.*

Cas clients

Actu triée par secteur

Avis d'Experts



Actus | **Événements** | **Newsletter** | **Vidéos**

www.solutions-logiciels.com

OUI, je m'abonne (écrire en lettres capitales)

Envoyer par la poste à : Solutions Logiciels, service Diffusion, 22 rue rené Boulanger, 75472 PARIS - ou par fax : 01 55 56 70 20

1 an : 50€ au lieu de 60€, prix au numéro (Tarif France métropolitaine) - Autres destinations : CEE et Suisse : 60€ - Algérie, Maroc, Tunisie : 60€ , Canada : 80€ - Dom : 75€ Tom : 100€
10 numéros par an.

M. Mme Mlle Société

Titre : Fonction : Directeur informatique Responsable informatique Chef de projet Admin Autre

NOM Prénom

N° rue

Complément

Code postal : Ville

Adresse mail

Je joins mon règlement par chèque à l'ordre de **SOLUTIONS LOGICIELS** Je souhaite régler à réception de facture

LE MAGAZINE DES **DECIDEURS INFORMATIQUES**

SOLUTIONS & LOGICIELS

www.solutions-logiciels.com

N°17 FÉVRIER 2011 3^e ANNÉE

Pronostics 2011
À quoi les DSIs doivent s'attendre... p.10

Sécurité DLP
Éviter la fuite des données p.28

Décisionnel
Les nouveaux défis de la B.I. p.38

TechDays
Microsoft complète son infrastructure CLOUD p.14

Multifonctions
outils polyvalents de l'information p.20

DOSSIER

Michel De Bosschere, Président Ricoh France

Ricoh : du photocopieur au Cloud p.26



fichier FederationMetadata.xml par un clic droit sur la racine du site puis « Update Federation Metadata ».

Pour tester ces ajouts, il suffit de modifier la page default.aspx pour énumérer les claims reçus :

ClaimType	Value
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name	Adam Carter
http://schemas.microsoft.com/ws/2008/06/identity/claims/role	Manager
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/dateofbirth	27/07/1985 00:00:00
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/postalcode	91210

Default.aspx

```
<asp:GridView ID="ClaimsGridView" runat="server">
    AutoGenerateColumns="False">
        <Columns>
            <asp:BoundField DataField="ClaimType"
                HeaderText="ClaimType" />
            <asp:BoundField DataField="Value"
                HeaderText="Value" />
        </Columns>
    </asp:GridView>
```

Default.aspx.cs

```
ClaimsIdentity ci = id as ClaimsIdentity;
ClaimsGridView.DataSource = ci.Claims;
ClaimsGridView.DataBind();
```

SCÉNARIO RIA + WCF

Dans le cas d'une application cliente appelant un service WCF, il est là encore possible de s'appuyer sur WIF pour authentifier l'utilisateur puis exploiter cette authentification pour gérer les autorisations ou modifier le comportement du service WCF appelé. La différence avec le scénario précédent est que le STS ne doit pas offrir une page web d'authentification mais un service WCF : en effet, dans le paragraphe précédent, c'était le navigateur qui interprétaient les ordres de redirection http reçus d'abord de l'application puis du STS. Ici, la « négociation » est effectuée à l'aide du protocole WS-Federation implémenté par le binding WCF ws2007FederationHttpBinding. Le proxy WCF utilisé par le client dialogue à la fois avec le service WCF appelé et le service WCF du STS pour obtenir le token de sécurité qui est ensuite transmis dans l'en-tête du message envoyé au service WCF applicatif. La mise en œuvre de ce scénario est la suivante :

- référencer le service WCF depuis le client et vérifier que les appels fonctionnent correctement (même si le service ne dispose alors pas des claims du client),
- faire un clic-droit sur le site hébergeant le service WCF et sélectionner « Add STS Reference » : un assistant similaire à celui déjà utilisé pour les applications ASP.NET est alors démarré. Comme dans le cas précédent, on va choisir de créer un STS.

Le web.config du site WCF est alors modifié pour utiliser le binding ws2007FederationHttpBinding et sécuriser les échanges WCF au

niveau message (i.e. le transport n'est pas modifié mais les en-têtes des messages SOAP contiennent des données cryptées et/ou signées) :

```
<ws2007FederationHttpBinding>
    <binding>
        <security mode="Message">
            <message>
                <issuerMetadata address="http://localhost/WIFProgrammez
WCFService_STS/Service.svc/mex" />
            <claimTypeRequirements>
                <add claimType="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name"
                    isOptional="true" />
                <add claimType="http://schemas.microsoft.com/ws/2008/06/identity/claims/role"
                    isOptional="true" />
            </claimTypeRequirements>
            </message>
        </security>
    </binding>
</ws2007FederationHttpBinding>
```

On peut ensuite mettre à jour la référence du service WCF dans le projet client (un « Update Service Reference » classique) : le binding passe évidemment à la même valeur du côté serveur (ws2007FederationHttpBinding) mais la configuration de ce binding référence maintenant aussi le serveur STS fournissant le token de sécurité. Celui-ci est appelé à l'aide d'un binding de type ws2007FederationHttpBinding qui est paramétré afin que le client « demande » les bons claims au serveur.

Références

La page de référence de Windows Identity Foundation sur MSDN est <http://msdn.microsoft.com/en-us/security/aa570351.aspx>, elle contient entre autres :

- le lien vers la page de téléchargement de WIF,
- le lien vers la page de téléchargement de ADFS 2.0,
- le Developer Training Kit qui contient comme d'habitude des présentations, des vidéos ainsi que des labs.
- Un certain nombre de whitepapers

Ce qu'il faut retenir

Les principaux points à retenir de WIF sont le principe et les avantages d'une authentification externe basée sur les claims qui permet d'implémenter des scénarios de single sign-on ou de fédération d'identité entre partenaires et/ou clients. On a vu que les accès à des applications ASP.NET ou des services WCF pouvaient ainsi tirer parti de ces principes, même si les scénarios présentés dans cet article sont restés très simples ; WIF permet d'aller bien au-delà (par exemple, appels WCF effectués par l'application Web pour le compte de l'utilisateur initial) mais aussi s'intègre parfaitement dans les applications Azure. Enfin, ADFS V2 est un composant à investiguer pour l'implémentation des STS utilisés dans les différents scénarios basés sur WIF.

■ Alain Zanchetta - Consultant Architecte - Microsoft I Services

Donnez votre avis sur ce numéro www.programmez.com/magazine_satisfaction.php

A la découverte de FreeBSD

Moins connu que son cousin Linux, FreeBSD est un système d'exploitation libre de type Unix. Pas très facile à prendre en main, ses qualités et sa stabilité séduiront le geek. Découverte.



Lorsqu'on parle de "système d'exploitation libre", on pense généralement Linux. Mais il en existe de nombreux autres, notamment dans la famille Unix. Dans Programmez! 130 nous avons découvert OpenSolaris qui malheureusement n'a pas survécu au rachat de Sun Microsystems par Oracle. Nous nous intéressons aujourd'hui à FreeBSD, un système d'exploitation totalement libre, et ne dépendant d'aucune entreprise. Sa réputation en termes de performances, de qualité et stabilité n'est plus à faire, et il est très volontiers utilisé dans les serveurs pour cette raison. Pas très facile à installer et à prendre en main en comparaison d'un Linux Ubuntu par exemple, FreeBSD est très séduisant pour le geek. Comme le terme anglais free le suggère, FreeBSD est totalement libre et gratuit y compris pour des usages commerciaux. Il s'agit d'un système d'exploitation Unix de la famille BSD. BSD signifie Berkeley Software Distribution. FreeBSD est donc un descendant de la famille des Unix développés à la fin des années 70 à l'Université américaine de Berkeley.

1 FREEBSD EN QUELQUES MOTS

En tant que système Unix, FreeBSD présente de nombreuses similitudes avec Linux, son cousin plus populaire. Mais il présente aussi de nombreuses différences. Une de ses spécificités qui peut se révéler très intéressante est la façon dont le projet est géré. Comme pour le défunt OpenSolaris, il n'existe qu'une seule équipe de développement, donc un seul FreeBSD. Cette équipe de développement est constituée d'un noyau d'environ 15 personnes. Autour de ce noyau s'organise une équipe plus large d'environ 150 personnes. Alors que les distributions Linux ont souvent comme objectif une installation facile, FreeBSD a d'abord pour but un système stable, extrêmement robuste même, performant, sécurisé et cohérent, sans composants propriétaires. Pour ses composants, priorité est donnée à la licence BSD. Du côté de la technique, le noyau de

FreeBSD est doté d'une architecture multiprocesseurs multi-threadée capable de faire tourner le noyau en parallèle sur plusieurs processeurs, et permettant aux tâches à haute priorité de passer avant tout autre activité du noyau, ce qui réduit ainsi les temps de latence. Cette architecture comprend une pile réseau multi-threadée et un sous-système de mémoire virtuelle multi-threadé. Les files d'attentes du noyau (Kernel Queues) permettent aux programmes de répondre très efficacement à divers événements asynchrones comme des entrées/sorties sur des fichiers ou des sockets, augmentant alors les performances du système et de l'application. Nous pourrions encore mentionner le gestionnaire de disque virtuel et la possibilité d'augmenter la taille du système de fichiers, ainsi que la couche de compatibilité avec les binaires Linux, ou la couche de compatibilité avec les pilotes Windows NT pour les périphériques WiFi. Arrêtons-nous là, le lecteur ayant compris qu'avec FreeBSD nous avons quelque chose de très intéressant entre les mains. Ce système est réputé difficile à installer.

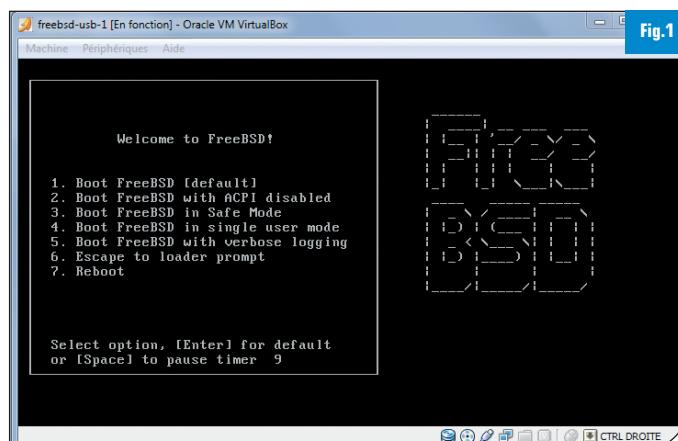
Il l'est certainement plus qu'un Ubuntu Linux d'aujourd'hui mais pas plus qu'une distribution Linux d'il y a quelques années. D'ailleurs, sur ce point et au niveau de la configuration, FreeBSD n'est pas sans rappeler Gentoo Linux.

2 MATÉRIEL ET PRÉ-REQUIS

Nous allons maintenant aborder l'installation de ce système. La suite de cet article ne posera pas de problème à un lecteur qui connaît bien Linux. Une bonne façon d'aborder FreeBSD est de l'installer sous VirtualBox. Ainsi aucun risque de casser un système d'exploitation résidant déjà sur la machine de travail. Pour cet article nous avons travaillé avec un VirtualBox sous Windows et un FreeBSD 8.1. La machine physique est un ordinateur portable. A terme cependant, VirtualBox n'est pas viable pour utiliser FreeBSD, faute d'extensions *invité*, ce qui implique l'impossibilité d'avoir une grande résolution d'écran en environnement graphique. Une fois familiarisé avec FreeBSD, si on apprécie ce système, le mieux est de l'installer sur un disque dédié.

3 INSTALLATION

La première chose à faire est de télécharger une image ISO du système sur www.freebsd.org. On gradera ensuite cette image pour démarrer depuis le lecteur de DVD. Sous VirtualBox il est possible de booter directement l'image, ce qui économisera une gravure. Dès que le système est amorcé, nous arrivons à un menu [Fig.1]. Choisir le boot par défaut avec la touche [Enter] ou avec la touche 1, sachant qu'à ce stade vous serez en clavier Qwerty, donc que vous devrez en fait appuyer sur &. Sélectionnez, dans la fenêtre qui suit, la France, parmi les différents pays, ce qui nous amènera immédiatement au choix du clavier Azerty. La fenêtre courante, et toutes celles qui vont suivre fonctionnent sur le même principe. La touche [Enter] valide la sélection en surbrillance. Les flèches permettent de modifier la sélection dans la zone principale. La touche Tab permet de basculer



Le menu d'accueil de FreeBSD

entre la fenêtre principale éventuellement et toujours entre les zones 'Ok' et 'Cancel'. Nous arrivons ensuite au choix du type d'installation. [Fig.2] Tout simplement choisir une installation standard. Puis nous arrivons au partitionnement du disque. Nous nous plaçons dans un cas simple: soit vous êtes sous VirtualBox et alors vous allouerez tout le disque virtuel au système, soit vous installez sur un disque dédié et vous allouerez de même le disque entièrement pour le système. Simplement pressez la touche A, vous êtes en clavier Azerty à ce stade, puis la touche Q. Nous arrivons maintenant à l'installation du boot pour notre futur système [Fig.3]. Puisque nous sommes dans le cas simple évoqué plus haut, installer un secteur de boot standard, en sélectionnant le choix proposé par défaut: 'Install a StandardMBR'. Là encore, si vous êtes sous VirtualBox, aucun danger. Par contre si vous êtes sur une machine qui de fait doit devenir dual-boot, parce que vous installez sur un disque dédié ou sur un disque partitionné dont la première partition contient un Windows, sachez que vous allez écraser le secteur de boot de Windows. Dans ce cas de figure, il convient de choisir l'option 'Install the FreeBSD Boot Manager' et de procéder en suivant attentivement les instructions. Si tout va mal et qu'il devient impossible de démarrer Windows, pas de panique. Lancez le CD-Rom de ce dernier, demandez à avoir une console DOS et là saisissez :

Bootrec.exe/FixMbr

Puis validez. Retirez le CD de Windows puis redémarrez, votre Windows est revenu à la vie, et vous pouvez retenter l'aventure FreeBSD

::) Continuez à valider jusqu'à arriver à une fenêtre intitulée 'FreeBSD Disklabel Editor' [Fig.4]. Faites simple en choisissant l'option de partitionnement par défaut avec la touche A, puis validez en pressant la touche Q. Vous arrivez ensuite à une fenêtre vous demandant quelle distribution installer. N'hésitez pas, choisissez 'All system sources and binaries'. [Fig.5] et pressez [enter] On vous demandera dans la fenêtre qui va suivre, si vous voulez installer la 'FreeBSD ports collection'. Répondez oui sans hésiter. Les ports sont une des spécificités et une des richesses de FreeBSD. Nous en parlerons à la fin de cet article. Validez, ce qui vous ramène à la fenêtre précédente. Mettez alors 'Ok' en surbrillance via la touche Tab, puis validez. On vous demandera alors depuis quel média vous installez. Dans notre cas de figure, il s'agit du CD/DVD, l'option par défaut. Validez. Validez ensuite dans la fenêtre 'de la dernière chance' : l'installation démarre. Une fois les paquetages installés, on vous demandera si vous voulez configurer une carte réseau. Répondez oui et suivez la procédure qui est automatique. On vous demandera ensuite si vous voulez faire une passerelle de votre système. Répondez non à cette question et aux suivantes, jusqu'à arriver à la configuration d'un accès SSH. Répondez alors oui. Un accès SSH est toujours bien pratique, surtout sous VirtualBox. Répondez ensuite non à toutes les questions, jusqu'à arriver à la configuration du fuseau horaire. Effectuez cette configuration. Ensuite signalez que vous possédez une souris. Ca y est, votre système est installé ! Vous arrivez toutefois à une fenêtre qui vous proposera l'installation de paquetages supplémentaires.

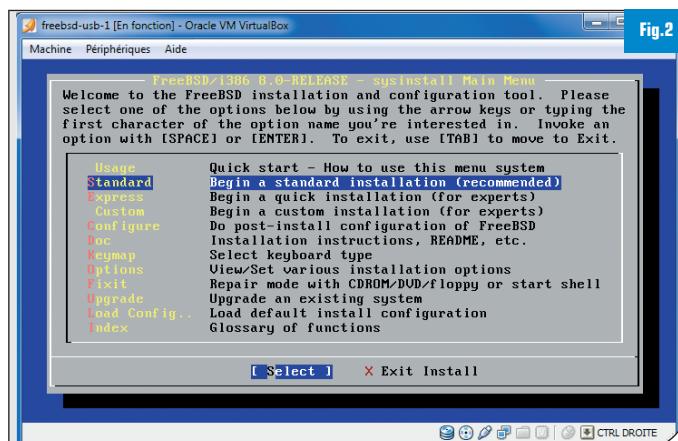
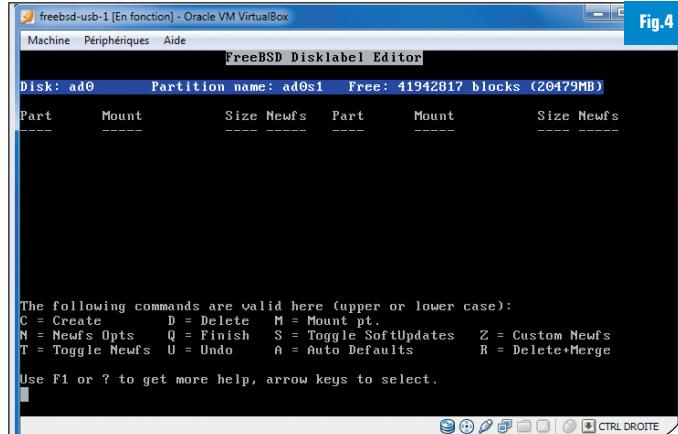


Fig.2

Choisir une installation standard



Etape de partitionnement du disque

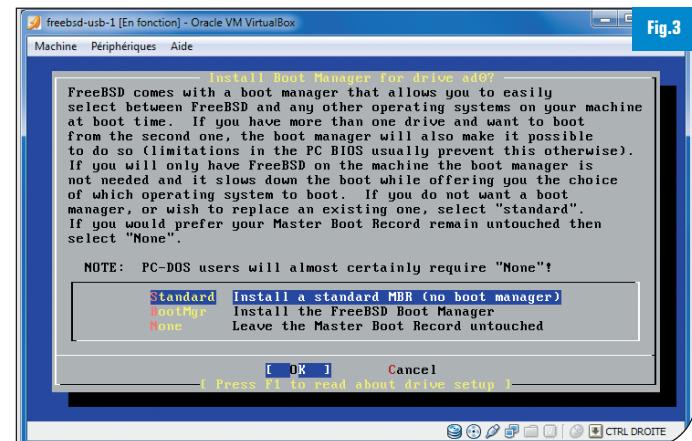


Fig.3

Choisir l'option de boot qui convient à votre configuration.

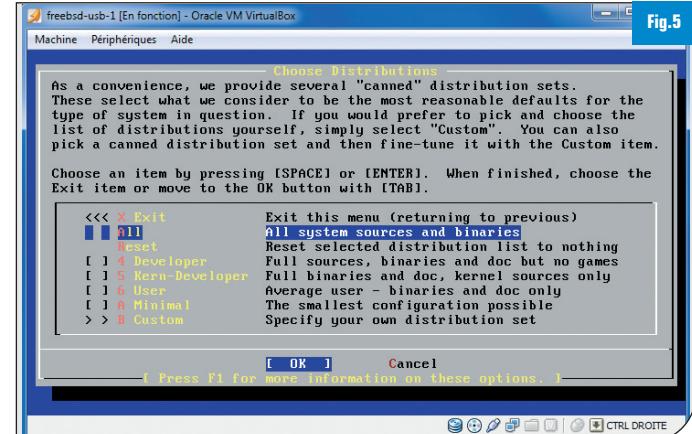


Fig.5

N'hésitez pas, demandez une installation complète.

4 INSTALLER GNOME, X-WINDOWS ET FRANCISER LE TOUT

Demandez la liste de tous les paquetages. La première chose à faire est de sélectionner un éditeur de texte qui va bien et sans lequel il n'y a pas de survie sous un système Unix. Choisissez par exemple Emacs :-) ou Vi... Vous verrez alors que le gestionnaire de paquetage gère les dépendances et sélectionne automatiquement les paquetages nécessaires. Sélectionnez ensuite Gnome (ou KDE si vous préférez ce dernier) puis Xorg. Validez, ce qui vous ramène à la fenêtre précédente. Sélectionnez alors Install et lancez l'installation. Une fois cette opération effectuée, il vous sera proposé de créer un utilisateur courant et on vous demandera de donner un mot de passe pour root. Autant d'opérations classiques en administration Unix/Linux, que vous exécuterez facilement. L'installation complète est terminée. A ce stade vous verrez que l'installateur vous propose d'effectuer éventuellement d'autres opérations post-installation, comme créer un autre utilisateur. Sachez enfin qu'en tant qu'administrateur, vous pourrez toujours revenir dans cet installateur, par exemple pour ajouter des paquetages à votre système en tapant :

```
sysinstall
```

dans un terminal. Attention ici à ne pas relancer une installation qui écraserait votre système.

5 LES FICHIERS DE CONFIGURATION DE FREEBSD

Redémarrez votre FreeBSD flambant neuf et connectez-vous en root. Il est temps de découvrir un aspect agréable de ce système. Lors du boot, vous aurez vu le lancement du démon SSH. Sous FreeBSD il est toujours possible de lancer un démon de façon traditionnelle :

```
/etc/rc.d/sshd start
```

Cependant on préférera travailler avec le fichier /etc/rc.conf. Ainsi, si vous éditez le vôtre, vous y verrez :

```
sshd_enable="YES"
```

Cette ligne suffit pour que le système lance le démon au démarrage. Gnome lui-même nécessite des démons. *Dbus* par exemple. Et *Xorg* nécessite *hald*. En effet, par défaut vous ne trouverez pas de fichier /etc/X11/xorg.conf. Xorg analysera votre système lors de son lancement et s'auto-configure pourvu que le démon *hald* soit actif. (remarque: il est toujours possible en cas de matériel récalcitrant de suivre une procédure à l'ancienne : Xorg -configure, puis adaptation du fichier xorg.conf.new ainsi généré) Pour tous ces démons nécessaires à Gnome, une seule ligne suffit dans rc.conf

```
gnome_enable="YES"
```

et au prochain démarrage vous vous retrouverez sous le gestionnaire de connexion graphique. Mais ce n'est peut-être pas ce que vous voulez. Vous préférerez peut-être taper explicitement la commande classique :

```
startx
```

Dans ce cas, supprimez la ligne `gnome_enable="YES"` du fichier /etc/rc.conf, et mettez à la place :

```
dbus_enable="YES"
hald_enable="YES"
```

```
#gnome_enable="YES"
```

Par contre, vous devrez ajouter au fichier .xinitrc de chaque utilisateur cette ligne :

```
exec ck-launch-session gnome-session
```

On le voit, configurer une séquence démarrage sous FreeBSD est tout ce qu'il y a de simple et agréable. Pour en savoir plus sur ce fameux fichier /etc/rc.conf, tapez tout simplement `man rc.conf` dans un terminal. Une fois sous Gnome, vous retrouverez un clavier Qwerty. Ceci provient de Gnome lui-même. Vous devez configurer votre clavier dans les préférences de Gnome.

Ceci fait, vous avez un clavier Azerty, mais l'interface de Gnome restera en anglais. Nous nous intéressons maintenant à un fichier capital sous FreeBSD: /etc/login.conf. N'hésitez surtout pas à faire un petit coup de `man login.conf`. Ce fichier permet de définir des classes de sessions. Ainsi c'est ici qu'on réglera les classiques quotas de disques. Mais il y a des choses beaucoup plus excitantes à faire ici, comme par exemple définir quelle quantité de mémoire un utilisateur ne peut dépasser, ou même la charge CPU limite pour un utilisateur. C'est ici que dire que FreeBSD est remarquable pour un serveur prend tout son sens :-) Quant à nous, nous configurons notre système afin que les utilisateurs français travaillent en français, en ajoutant dans ce fichier :

```
french:FRENCH Users Accounts:\
:charset=UTF-8:\
:lang=fr_FR.UTF-8:\
lang=fr_FR:\
lc_all=fr_FR:\
lc_collate=fr_FR:\
lc_ctype=fr_FR:\
lc_messages=fr_FR:\
lc_monetary=fr_FR:\
lc_numeric=fr_FR:\
lc_time=fr_FR:\
:tc=default:
```

Si l'on veut que root devienne français on mettra la même chose pour lui :

```
root:\
:#ignorenologin: # cette ligne en commentaire
:charset=UTF-8:\
:lang=fr_FR.UTF-8:\
# etc, etc.
```

FreeBSD ne lit pas ce fichier directement mais travaille avec une petite base de données qu'il faut maintenant reconstruire :

```
cap_mkdb /etc/login.conf
```

Dès maintenant, l'interface de votre Gnome sera francisée :-)

6 GÉRER LES UTILISATEURS AVEC LE COUTEAU SUISSE PW

Vous avez sans doute remarqué que le Shell sh est attribué par défaut aux utilisateurs. Sans doute préférerez-vous le Shell Bash. Sous Linux on modifierait cet attribut d'utilisateur avec la commande usermod. Celle-ci n'existe pas sous FreeBSD. A la place on utilisera la commande pw, qui est un véritable couteau suisse de gestion des

utilisateurs, et dont la page de man mérite d'être étudiée :-) Succinctement pw s'utilise avec une commande 'à la Linux', suivi de commutateurs. Ainsi pour modifier un utilisateur de nom fred et lui attribuer le Shell Bash, on donnera :

```
pw usermod -s /bin/bash -n fred
```

7 LES PÉRIPHÉRIQUES

Sous FreeBSD, les noms de périphériques sont différents de ce que l'on a sous Linux. Voici quelques-uns de ces noms de périphériques, parmi les plus utiles

Nom	Périphériques
ad0	Premier disque dur ATAPI (ATA/IDE)
ad1	Second disque dur ATAPI (ATA/IDE)
acd0	Premier lecteur CDROM/DVD ATAPI (ATA/IDE)
acd1	Second lecteur CDROM/DVD ATAPI (ATA/IDE)
da0	Premier disque dur SCSI
cd0	Premier lecteur CDROM SCSI
fd0	Premier lecteur de disquette
usb0	Premier port USB
usb1	Second port USB

Ainsi, si vous voulez monter un CDROM inséré dans le premier lecteur et sur le répertoire /cdrom a priori destiné à cela, la commande sera :

```
mount -t cd9660 /dev/acd0 /cdrom
```

Pour en apprendre plus sur les points de montage, faites un petit :

```
cat /etc/fstab
```

```
# Device      Mountpoint  FStypeOptions      Dump      Pass#
/dev/ad0s1b    none        swap      sw      0 0
/dev/ad0s1a    /           ufs       rw      1 1
/dev/ad0s1e    /tmp        ufs       rw      2 2
/dev/ad0s1f    /usr        ufs       rw      2 2
/dev/ad0s1d    /var        ufs       rw      2 2
/dev/acd0     /cdrom      cd9660ro,noauto 0 0
```

Dans cette sortie, l'omniprésent ufs est le système de fichiers par défaut de FreeBSD.

8 CONFIGURER LE SON

Si FreeBSD fait fort pour les cartes réseau ou pour la configuration automatique de X-Windows, on ne peut toutefois pas en dire autant pour le son. Vous avez sans doute remarqué que Gnome est resté muet jusqu'à maintenant. Voici comment procéder pour configurer votre carte son. Commencez par saisir cette commande dans un terminal :

```
cat /dev/sndstat
```

Vous obtiendrez une sortie semblable à celle-ci :

```
FreeBSD Audio Driver (newpcm: 32bit 2009061500/i386)
Installed devices:
pcm0: <Intel ICH (82801AA)> at io 0xd100, 0xd200 irq 5 bufsz 16384
kld snd_ich [MPSAFE] (lp:1v/1r:1v channels duplex default)
```

Dans ce charabia se trouve le nom du driver de la carte son, ici snd_ich. Il vous suffit d'ajouter une ligne dans le fichier /boot/loader.conf :

```
snd_ich_load="YES"
```

Bien entendu, pour votre machine ce ne sera peut-être pas snd_ich qui conviendra. Regardez bien la sortie de cat /dev/sndstat, et adaptez en conséquence. Le driver sera chargé automatiquement à chaque démarrage du système avec notre manipulation. Toutefois, un administrateur Unix détestant en général rebooter inutilement ;), il est possible de charger le driver directement :

```
kldload snd_ich
```

9 LE DÉVELOPPEMENT ET LES PORTS

Sous FreeBSD comme sous Linux, vous pouvez installer et utiliser tous les outils de développement que la communauté open source met à votre disposition, et tous les langages que vous voulez: Java, Ruby, Python, etc. Comme pour tout environnement Unix qui se respecte, l'outil central de développement de FreeBSD est le compilateur C/C++, le compilateur GNU en l'occurrence, que vous invoquerez par cc, gcc ou g++. Ici tout se passe comme sous Linux. Mais cela nous amène aux ports de FreeBSD. Un port est un ensemble de fichiers de configuration pour assurer la compilation d'un logiciel. En cela, les ports sont semblables aux ebuild de Gentoo Linux. L'ensemble des ports assure la belle cohérence dont jouit FreeBSD. Si vous avez besoin de recompiler une application, par exemple après une mise à jour de sécurité, c'est tout simple. Supposons que nous voulons recompiler le Shell Bash. Son port est dans /usr/ports/shells/bash. Nous faisons donc tout simplement :

```
cd /usr/ports/shells/bash
```

puis :

```
make
```

Le système va commencer par télécharger une archive de sources qui sera déposée dans le répertoire /usr/ports/disfiles. Les archives dont dépend l'application en cours de construction seront également téléchargées. Le tout sera décompressé automatiquement puis la compilation sera lancée. Ensuite vous ferez :

```
make install
```

pour installer l'application et enfin :

```
make clean
```

pour nettoyer tous les fichiers objets issus de la compilation, y compris la compilation des applications faisant partie des dépendances. Cette opération est importante pour éviter des problèmes subtils lors de la compilation de mises à jour et pour économiser de la place sur le disque dur. On notera que les archives de sources déposées dans /usr/ports/disfiles restent en place.

10 CONCLUSION

Un tout petit peu difficile à prendre en main, FreeBSD est un système cohérent, très solide, et doté de caractéristiques techniques très séduisantes. Il a tout ce qu'il faut pour passionner le geek lecteur de Programmez! :-)

■ Frédéric Mazué - fmazue@programmez.com

Economisez jusqu'à 50%



Programmez ! est le magazine du développement. Langage et code, développement web, carrières et métier : Programmez !, c'est votre outil de veille technologique.

Pour votre développement personnel et professionnel, abonnez-vous à Programmez ! www.programmez.com

1 -25%

Abonnement 1 an

49€ au lieu de 65,45 € tarif au numéro - Tarif France métropolitaine

2 +0,8€ par mois

Abonnement Intégral : + archives

1 an au magazine + archives sur Internet et PDF

59€ Tarif France métropolitaine

3 jusqu'à -50%

Abonnement 2 ans + 1 livre numérique ENI

• **79€** au lieu de 130,90 (valeur de 22 numéros) Tarif France métropolitaine + un livre d'une valeur de 23,9 € à 31,9 €, soit un total de 154,8 € à 162,8 €

• **89€ 2 ans au magazine + archives sur Internet et PDF + 1 livre numérique ENI**



OUI, je m'abonne

Vous pouvez vous abonner en ligne et trouver tous les tarifs www.programmez.com

Abonnement 1 an au magazine : **49€** (au lieu de 65,45 € tarif au numéro) Tarif France métropolitaine

Abonnement Intégral : 1 an au magazine + archives : **59€** Tarif France métropolitaine

Abonnement 2 ans au magazine + livre numérique ENI : **79€** Tarif France métropolitaine

Abonnement 2 ans au magazine + livre numérique ENI + archives : **89€** Tarif France métropolitaine

Livres à Choisir : Visual Studio 2010 PHP5.3 Bing Maps MySQL 5, Administration et optimisation

Java et Spring, Concevoir, construire et développer une application Java/J2EE avec Spring. Détails sur www.programmez.com/abonnement.php

M. Mme Mlle Entreprise : _____ Fonction : _____

Prénom : _____ Nom : _____

Adresse : _____

Code postal : _____ Ville : _____

Tél : _____ (Attention, e-mail indispensable)

E-mail : _____ @ _____

Je joins mon règlement par chèque à l'ordre de Programmez ! Je souhaite régler à réception de facture

A remplir et retourner sous enveloppe affranchie à : Programmez ! - Service Abonnements - 22 rue René Boulanger - 75472 Paris Cedex 10.
abonnements.programmez@groupe-gli.com

Offre limitée,
valable jusqu'au
31 mars 2011

Le renvoi du présent bulletin implique pour le souscripteur l'acceptation pleine et entière de toutes les conditions de vente de cette offre. Conformément à la loi Informatique et Libertés du 05/01/78, vous disposez d'un droit d'accès et de rectification aux données vous concernant. Par notre intermédiaire, vous pouvez être amené à recevoir des propositions d'autres sociétés ou associations. Si vous ne le souhaitez pas, il vous suffit de nous écrire en nous précisant toutes vos coordonnées.

Petite prise de contact avec IPv6

La pénurie d'adresses IPv4 est désormais imminente. Le nouveau protocole de communication IPv6 va donc se généraliser rapidement, et les développeurs vont devoir le manipuler. Voici quelques notions de base.



APPLICATION : réseau

LANGAGE : C++

SOURCE : oui

Internet est un réseau informatique, qui en tant que tel nécessite un protocole de communication afin que les machines connectées puissent communiquer entre elles. Ce protocole est le protocole IP, pour Internet Protocol. Il est actuellement utilisé dans sa version 4, d'où son nom d'IPv4. IP est un protocole «bout en bout», ce qui signifie que deux machines arbitraires peuvent communiquer directement l'une avec l'autre, l'infrastructure du réseau leur étant aussi transparente que possible. Une communication bout en bout implique un système d'adressage tel que chaque machine se voit attribuer une adresse unique et puisse établir une communication avec une machine quelconque simplement en connaissant son adresse. En IPv4 une telle adresse est codée sur 4 octets et est généralement représentée en notation pointée. Par exemple : 192.198.1.10

4 octets, soit 32 bits, permettent de définir un peu plus de 4 milliards d'adresses, ce qui a paru énorme, et même inépuisable, aux concepteurs du protocole à l'époque. Mais c'est un nombre dramatiquement petit aujourd'hui. 4 milliards c'est moins d'une adresse par habitant de la terre, alors que nous allons vers un monde où chaque habitant disposera de plusieurs machines et autres appareils connectés, fixes ou mobiles. C'est d'autant moins, en raison d'un système de préfixage qui fait que certaines plages d'adresses sont réservées à certains usages et donc indisponibles pour le réseau. Ainsi 192.168.1.10 est une adresse de préfixe 192.168.x.x réseau aux Intranet. L'épuisement des adresses IP est très proche, même si la date exacte fait débat, certains organismes, par exemple la défense américaine, détenant un très grand nombre d'adresses qui seront peut-être mises à disposition et même si des techniques comme le NAT permettent de tricher momentanément avec le problème, non sans violer le principe du bout en bout. Mais épuisement il y aura, la parade étant IPv6, qui sort peu à peu de son état expérimental pour progressivement se généraliser. Familiarisons-nous donc avec ce protocole qui fera très vite partie de notre quotidien

1 IPv6, NOTIONS DE BASE

IPv6 définit un système d'adressage sur 128 bits, soit 16 octets. Ce système contient donc un nombre d'adresses colossal: 2^{128} . Un nombre si grand qu'on le tient pour inépuisable, ainsi qu'on le disait fièrement pour IPv4 il y a peu de temps :-) Ce nombre étant inépuisable, un système de préfixe a été défini pour réservier des plages d'adresses tout comme cela avait été fait pour IPv4 :-) En fait IPv6 et IPv4 se ressemblent beaucoup, et un réel effort a été fait par les concepteurs afin de limiter le travail de portage des applications de IPv4 vers IPv6. Comme avec IPv4 les adresses peuvent être de plusieurs types :

- Unicast est l'équivalent de l'Unicast IPv4. Dans ce cas, une machine envoie des informations vers une seule machine
- Multicast est l'équivalent de Broadcast d'IPv4. Dans ce cas une machine envoie des informations vers un groupe de machines.
- Anycast permet l'envoi d'informations vers une machine appartenant à un groupe.

2 LES ADRESSES IPv6

Les adresses IPv6, bien que définies par 16 octets sont concrètement représentées par 8 mots de 16 bits, soit 16 octets. Les valeurs sont hexadécimales. Cette représentation ayant sans doute été adoptée car plus humainement lisible qu'un enfilage de 16 octets. Il convient toutefois d'être attentif à ce point, susceptible de générer des erreurs de programmation lorsqu'on manipule ces adresses directement. Voici à quoi ressemble une adresse IPv6 :

FEDC:BC97:7604:3210:EDBC:A987:6543:293F

Les mots sont écrits en «Network byte order», c'est-à-dire finalement en Big Endian, donc de façon intuitive, heureusement. Cette notation peut être simplifiée. Ainsi si par exemple nous avons :

FEDC:0000:0000:0000:0800:A987:6543:293F

L'adresse sera simplifiée en :

FEDC:0:0:0:800:A987:6543:293F

Ainsi nous avons retiré tous les octets inutiles. Mais les pages de champs nuls peuvent aussi être retirées. Finalement, l'adresse de départ apparaîtra ainsi sous vos yeux :

FEDC::800:A987:6543:293F

Etant entendu qu'il ne peut y avoir qu'une seule page de mots nuls délimitée par :: dans une adresse pour ne pas engendrer d'ambiguité. L'adresse indéterminée, équivalente à 0.0.0.0 de IPv4 est :

0:0:0:0:0:0:0:0

Soit finalement ::. L'adresse de bouclage, ou loopback qui en IPv4 est 127.0.0.1 est ici :

0:0:0:0:0:0:0:1

Soit finalement ::1. Abordons maintenant quelques manipulations de base.

3 MANIPULATIONS DE BASE

Pour pouvoir se faire la main avec IPv6 il faut naturellement un système d'exploitation qui le supporte, ce qui est le cas des Windows récents, y compris Windows XP si l'installation a été faite spécifiquement par la commande 'ipv6 install'. C'est également le cas des Linux récents et d'autres systèmes 'Unix like'. Sous Linux un noyau 2.36.x

est recommandé. Si vous faites des essais avec OS virtualisé, par exemple sous VirtualBox, le NAT que vous utilisez couramment pour accéder au réseau est en IPv6 et donc ne fonctionnera plus. Pour avoir IPv6 dans ce cas, il faut, en lieu et place du NAT, ponter (bridge) l'interface de la VirtualBox avec l'interface physique du système hôte. Enfin il faut que votre fournisseur d'accès supporte IPv6, faute de quoi vous serez limité à des essais en local entre machines personnelles. Sous Linux, outre les fichiers de configuration propres à votre système, votre noyau doit avoir été compilé avec le support IPv6. Pour vérifier que tout fonctionne, il suffit de taper :

```
ifconfig eth0
```

dans un terminal. Comme illustré [Fig.1] une adresse IPv6 doit apparaître. Dans le cas de votre serveur :

```
FE80::a00:27FF:FE32:DE27
```

On remarque le préfixe FE80 qui est le préfixe des adresses locales. L'interface doit répondre à un ping :

```
ping6 FE80::a00:27FF:FE32:DE27
```

Si vous possédez une autre machine connectée à votre réseau, vous pouvez connaître son adresse IPv6 de la même façon et elle aussi répondra à un ping. Si vous êtes sous Windows tout se passe de la même façon. On remarque le même préfixe local sur l'illustration [Fig.2]. Les commandes à utiliser sont ipconfig /all et ping -6. Une adresse IPv6 quelconque sur Internet peut être touchée de la même manière par un ping. Le tout étant d'en connaître une... Les sites sous IPv6 sont encore peu nombreux, et très probablement votre fournisseur d'accès ne vous propose pas encore un service DSN en IPv6. Il existe quelques sites en IPv4 qui listent des sites en IPv6. Le site sixy.ch (<http://www.sixy.ch>) en est un. Pour expérimenter, j'ai retenu l'adresse IPv6 de debian-fr.org. L'adresse est 2001:41d0:1:ed34::3. Vous pouvez accéder à la page d'accueil en saisissant cette URL dans votre navigateur :

[http://\[2001:41d0:1:ed34::3\]/](http://[2001:41d0:1:ed34::3]/)

Bien remarquer les crochets qui entourent l'adresse IP. Leur rôle est d'éviter une confusion entre le : qui délimite les mots de l'adresse et le : qui précède conventionnellement le numéro de port. Nous sommes en HTTP, le numéro de port est 80. Nous aurions donc pu donner cette URL : [http://\[2001:41d0:1:ed34::3\]:80/](http://[2001:41d0:1:ed34::3]:80/)

4 PROGRAMMER AVEC IPv6

Comme il a été dit plus haut, les concepteurs de IPv6 se sont efforcés de minimiser le travail des programmeurs sur le code existant. Ainsi pour ouvrir et fermer un socket en Python, nous écrirons :

```
import socket

# adresse IPv6 de debian-fr.org
host = '2001:41d0:1:ed34::3'

port = 80

s = socket.socket(socket.AF_INET6, socket.SOCK_DGRAM)
s.connect((host, port))

s.close()
```

Le seul changement par rapport à l'équivalent IPv4 est un AF_INET qui se transforme en AF_INET6. Même s'il aurait été préférable qu'il se transforme en PF_INET6 (voir plus loin). Le format d'adresse est compris, et le socket est ouvert.

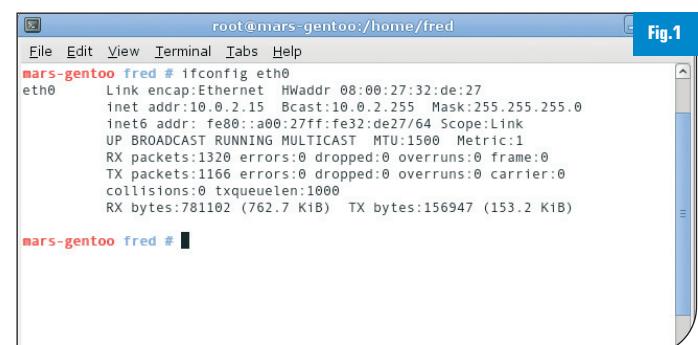
5 AVEC JAVA

Faute de serveur DNS, nous devons manipuler notre adresse IPv6 directement, ce qui est un peu lourd. Avec l'exemple ci-dessous nous ouvrons un socket avec le serveur debian-fr.org, sur son adresse IPv6. Puis nous écrivons une requête GET dans le socket. Le serveur nous renvoie alors la page :

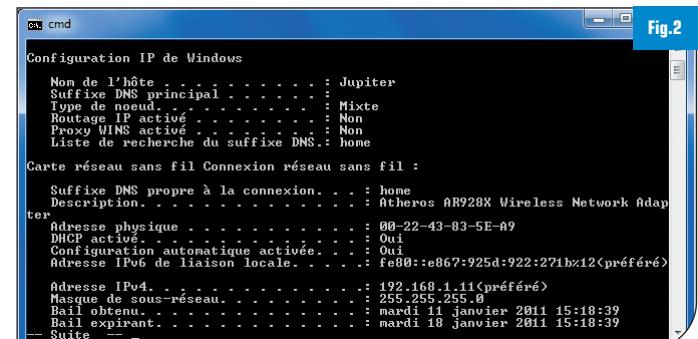
```
package demosocket;
import java.io.*;
import java.net.*;

public class ClientIPv6 {
    public static void main(String[] args) {
        try {
            byte[] raw_addr = new byte[16];
            raw_addr[0] = 0x20;
            raw_addr[1] = 0x01;
            raw_addr[2] = 0x41;
            raw_addr[3] = (byte)(0xd0 & 0x7F);
            raw_addr[3] |= 0x80;
            raw_addr[4] = 0x0;
            raw_addr[5] = 0x01;
            raw_addr[6] = (byte)(0xed & 0x7F);
            raw_addr[6] |= 0x80;
            raw_addr[7] = 0x34;
            raw_addr[15] = 0x03;
            //Inet6Address i6addr = Inet6Address.getByAddress("2001:

```



Une adresse IPv6 locale sous Linux



Une adresse IPv6 locale sous Windows

```
41d0:1:ed34::3", raw_addr, 0);
        Inet6Address i6addr = Inet6Address.getByAddress("debian-fr.
org", raw_addr, 0);
        Socket s = new Socket(i6addr, 80);
        BufferedReader in = new BufferedReader(
            new InputStreamReader(s.getInputStream()));
        DataOutputStream out = new DataOutputStream(s.getOutputStream());
        out.writeBytes("GET / HTTP/1.0\r\n\r\n");
        String responseLine;
        while ((responseLine = in.readLine()) != null) {
            System.out.println(responseLine);
        }
        in.close();
        out.close();
        s.close();
    } catch (UnknownHostException uex) {
        uex.printStackTrace();
    } catch (IOException e) {
        System.out.println("Erreur" + e);
    } catch (Exception ex) {
        ex.printStackTrace();
    }
}
```

```

#include "stdafx.h"
#define WIN32_LEAN_AND_MEAN
#include <windows.h>
#include <winsock2.h>
#include <ws2tcpip.h>
#include <iostream>
using namespace std;

int main(int argc, char* argv[]) {
    WSADATA wsadata;
    SOCKADDR_STORAGE addr;
    SOCKET sock;

    if(::WSAStartup(MAKEWORD(2, 2), &wsadata)) {
        cerr << "Impossible d'initialiser la Winsock" << endl;
        return EXIT_FAILURE;
    }
    cout << "Winsock initialisee" << endl;
    sock = socket(PF_INET6, SOCK_STREAM, 0);
    if(sock != INVALID_SOCKET)
        cout << "Socket cree" << endl;
    else {
        cerr << "Erreur a la creation du socket" << endl;
        ::WSACleanup();
        return EXIT_FAILURE;
    }
    // Préparer l'adresse à laquelle se connecter
    memset(&addr, 0, sizeof(SOCKADDR_STORAGE));
    addr.ss_family = PF_INET6;
    struct sockaddr_in6* addr6 = (struct sockaddr_in6*)&addr;
    addr6->sin6_scope_id = 0;
    addr6->sin6_port = ::htons(80); // port 80 HTTP

    /*
    addr6->sin6_addr.u.Byte[0] = 0x20;
    addr6->sin6_addr.u.Byte[1] = 0x01;
    addr6->sin6_addr.u.Byte[2] = 0x41;
    addr6->sin6_addr.u.Byte[3] = 0xd0;
    addr6->sin6_addr.u.Byte[4] = 0x00;
    addr6->sin6_addr.u.Byte[5] = 0x01;
    addr6->sin6_addr.u.Byte[6] = 0xed;
    addr6->sin6_addr.u.Byte[7] = 0x34;
    addr6->sin6_addr.u.Byte[15] = 0x03;
    // ou bien
    */
    const char* host = "2001:41d0:1:ed34::3";
    ::inet_pton(AF_INET6, host, (void *)&addr6->sin6_addr);
    if(::connect(sock, (sockaddr*)addr6, sizeof(sockaddr_in6))) {
        cerr << "Impossible de connecter le socket" << endl;
        ::closesocket(sock);
        ::WSACleanup();
        return EXIT_FAILURE;
    }
    else
        cout << "Socket connecte" << endl;
}

const char* commande = "GET / HTTP/1.0\r\n\r\n";
if(::send(sock, commande, strlen(commande), 0) == SOCKET

```

6 AVEC C++

Avec ce langage (ou avec C, Pascal, etc.) il est nécessaire de gérer la mémoire. Et bien évidemment une adresse Ipv6 est plus volumineuse qu'une adresse IPv4, ce qui entraîne de petites modifications dans le code. Le code suivant a été écrit sous Visual Studio 2008 pour Windows mais est quasiment directement transposable pour Linux :

```

ERROR) {
    cerr << "Erreur lors de l'écriture dans le socket" << endl;
    ::closesocket(sock);
    ::WSACleanup();
    return EXIT_FAILURE;
}
else
    cout << "Commande envoyée" << endl;

// recevoir les données du serveur
const int bufsize = 1;
char buffer = 0;
int result = 0;
cout << "Données lues:" << endl << endl;
do {
    result = ::recv(sock, &buffer, bufsize, 0);
    if(result > 0)
        cout << buffer;
} while(result > 0);
cout << endl;
::closesocket(sock);
::WSACleanup();
return EXIT_SUCCESS;
}

```

Sous Windows, ce code doit être lié à la librairie ws2_32.lib. L'appel de la fonction d'initialisation WSAStartup est spécifique à Windows et n'existe pas sous Linux. Apparaît désormais un type générique SOCKADDR_STORAGE (sockaddr_storage sous Linux) pour contenir une adresse IP 4 ou 6. Un transtypage du pointeur sur cette structure sera fait quand cela est nécessaire. Nous utilisons cette structure dans notre code, mais un peu « à vide ». Comme nous ne travaillons qu'avec IPv6 elle n'était pas nécessaire. Nous transtypons donc vers une structure sockaddr_in6 que nous initialisons. Bien noter le type de socket: PF_INET6. Mettre AF_INET6 va probablement fonctionner sur la plupart des implémentations, mais pas nécessairement sur toutes et se serait une faute de programmation que de faire ainsi. Bien remarquer l'appel à htons, comme en IPv4, pour transformer la valeur du port en Network order. Sans cela cette valeur, sous architecture x86, serait écrite en Little Endian et le programme échouerait. Le code propose ensuite deux possibilités. La première, en commentaire dans l'exemple, consiste à manipuler les octets un à un comme en Java. Nous y rencontrons évidemment les mêmes pièges tendus par IPv6. L'autre possibilité,

meilleure, consiste à utiliser l'API inet_pton qui va prendre la représentation textuelle de l'adresse, pour faire le travail que nous avons fait manuellement. Cette fonction a un nom standard et existe bien entendu sous Linux. Curieusement, sauf erreur de la part de votre serveur, la documentation qui vient avec Visual Studio 2008 ne la mentionne pas. Elle est présente sous Visual Studio 2010 en revanche. Le reste est simple. Après la création du socket et sa connexion nous y écrivons la même requête GET que précédemment, et bien sûr ne recevons pas la mauvaise page :-) [Fig.3]. L'exemple lit un à un les octets émis par le serveur. Nous pourrions bien sûr les lire par paquets. En fait une fois la connexion établie, le travail avec un socket est similaire au travail avec un fichier, les fonctions send et recv remplaçant write et read. A la fin du travail, le socket est fermé avec la fonction closesocket. Mais sous Linux, on utilisera simplement close.

7 SI UN DNS EST DISPONIBLE

Dans ce cas, on ne manipulera pas les adresses directement, mais on obtiendra celles-ci au moyen de la fonction getaddrinfo. Voici l'extrait intéressant de l'exemple que le lecteur trouvera en entier sur notre site.

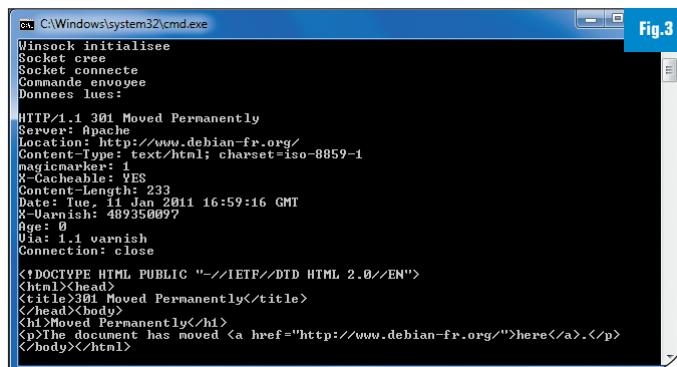
```

struct sockaddr_in6* addr6 = 0;
struct addrinfo hints;
memset(&hints, 0, sizeof(addrinfo));
hints.ai_family = PF_INET6;
hints.ai_socktype = SOCK_STREAM;
hints.ai_protocol = IPPROTO_IPV6;
const char* host = "debian-fr.org";
struct addrinfo* addrinfo = 0;
int res = ::getaddrinfo(host, 0, &hints, &addrinfo);
if(res) {
    if(res == WSAHOST_NOT_FOUND)
        cerr << "Pas de résolution de noms..." << endl;
    else
        cerr << "Erreur à l'appel getaddrinfo" << endl;
    ::WSACleanup();
    return EXIT_FAILURE;
}
Else {
    addr6 = (sockaddr_in6*)addrinfo->ai_addr;
    addr6->sin6_scope_id = 0;
    addr6->sin6_port = ::htons(80); // port 80 HTTP
}

```

L'usage n'est pas très simple. Elle reçoit en argument la représentation textuelle d'une adresse ou un nom d'hôte. On lui passe une structure addrinfo (hints dans l'exemple) partiellement définie. C'est par ce moyen que l'on demande à la fonction de travailler avec IPv6, IPv4 ou les deux. Les résultats de la résolution de nom peuvent donc être multiples. Ces résultats sont fournis dans une liste chaînée. On passe un pointeur sur une structure addinfo en troisième argument à la fonction et celle-ci initialise le pointeur avec l'adresse de la première structure de la liste chaînée. Au programmeur de parcourir cette liste selon ses besoins, puis de libérer la mémoire avec la fonction freeaddrinfo.

■ Frédéric Mazué - fmazue@programmez.com



Lecture d'une page Web à travers un socket IPv6 avec JC++ sous Windows

AGILITÉ

Scrum en action

Difficulté : **

Editeur : Pearson

Auteur : Guillaume Bodet

Prix : 17,50 €

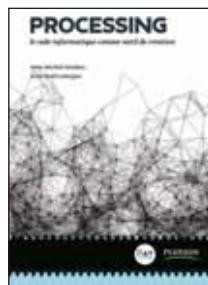
Comment évoquer



Scrum sans faire dans la traditionnelle présentation ? L'auteur propose une approche toute à fait étonnante : le récit. Nous suivons le récit d'une équipe, des responsables, des développeurs appliquant la méthode agile, avec les bons conseils, les points à retenir. C'est une approche originale et qui fait passer l'objectif de Scrum ! Une bonne lecture avant de dormir...

AGILITÉ

Processing le code informatique comme outil de création



Difficulté : ***

Editeur : Pearson

Auteur : collectif

Prix : 28 €

Voici un outil de création multimédia multi-facette possédant une richesse fonctionnelle rare. Ce livre vous aidera

à prendre en main le logiciel et à concevoir vos premières créations. Il aborde les différents aspects du langage, depuis l'installation du logiciel jusqu'à la fabrication de documents PDF et de vidéos ou le traite-

ment dynamique de données XML et le pilotage d'appareils électroniques tels que les cartes de prototypage Arduino et Wiring. De nombreux exemples permettent de mieux comprendre l'outil et la philosophie de l'environnement.



été simples à comprendre et à maîtriser surtout pour un débutant en développement web. CSS3 rajoute une nouvelle couche de complexité même si la puissance justifie l'investissement que l'on fera dessus. L'auteur offre ici une approche très design de CSS : animation, transition, formulaire, les couches, les effets, etc. Une véritable mine d'informations !

WEB

CSS3 pour les web designers

Difficulté : **

Editeur : Eyrolles

Auteur : Dan Cedreholm

Prix : 12 €

Les feuilles de styles n'ont jamais

matiser les tests avec JUnit et pourquoi il faut le faire et quelle démarche globale il faut appliquer pour améliorer la qualité logicielle. Vous verrez comment insérer l'outil dans votre développement. Une excellente approche pragmatique et assez facile à mettre en pratique rapidement.



WEB MOBILE

Relever le défi du web mobile

Difficulté : **

Editeur : Eyrolles

Auteur : collectif

Prix : 29,90 €

Le dossier webmaster du mois a parfaitement illustré l'intérêt de cet ouvrage : relever le défi du web mobile. A la fois tactique, stratégique, technique, l'ouvrage propose une vision large et pragmatique du web mobile, comment un développeur doit l'appréhender et les pièges à éviter.

La partie format (et langages) est à lire impérativement pour comprendre toute la difficulté actuelle.

Car même pour un « simple » site web, le développeur devra faire des tests, des adaptations sur l'ensemble des plateformes. Et ce n'est pas HTML5 qui va nous sauver ! Sans oublier les problématiques d'optimisation, d'extensions...

Bref, vous l'aurez compris, nous avons été conquis par l'ouvrage ! Un excellent travail même si nous aurions apprécié une partie technique plus costaude, peut être dans une seconde édition.

TESTS

JUnit

Difficulté : ***

Editeur : Eni

Auteur : Benoit Gantama

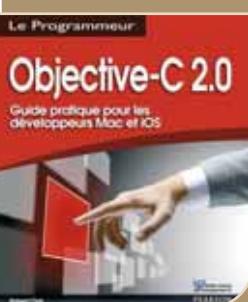
Prix : 39 €

Vous aimez les tests ? Pourquoi ne pas essayer JUnit surtout si vous développez en Java ? Ici, vous allez savoir comment auto-

riser chaque aspect fondamental du langage, qu'il s'agisse de l'envoi de messages aux objets et aux classes, des frameworks ou des protocoles. Chaque concept est illustré par des exemples simples et la plupart des chapitres proposent des exercices pratiques. Pour aborder en douceur la programmation Mac et iOS, il s'agit d'une bonne approche.

SPÉCIAL MAC / IOS

Objective-C 2.0



Difficulté : ***

Editeur : Pearson

Auteur : Robert Clair

Prix : 32 €

Objectif : être précis et concis. Voilà la mission de Robert Clair. Dans

le monde Apple, Objective-C est le langage de référence depuis l'arrivée de Mac OS X et le langage historique de NeXT. Après un rappel sur le C et la programmation orientée objet, vous créerez un premier programme Objective-C avec Xcode. Vous apprendrez ensuite à maîtriser

Objective-C

Difficulté : **** - Editeur : Eni

Auteur : collectif - Prix : 54 €

Vous voulez créer des applications pour iOS que ce soit pour iPhone et iPad ? Vous devrez apprendre à maîtriser Objective-C, et ses outils. Les auteurs misent beaucoup sur le concret : Du code et encore du code !

Objective-C

Créez vos applications iPhone et iPad

(Mac OS, Linux, Windows)



Les auteurs décrivent tout d'abord des applications simples (threads, services web, base de données...) et évoluent ensuite progressivement vers des applications plus complexes combinant plusieurs techniques (IHM, modèles de conception dont MVC et KVO, Cocos2D...). L'apprentissage du langage s'appuie sur des notions de développement logiciel pour améliorer les compétences générales du développeur. Une très bonne introduction, dommage que le livre soit proposé à 54 €.

egilia®

LEARNING

LE SPÉCIALISTE DE LA
FORMATION CERTIFIANTE
EN **INFORMATIQUE**
ET **MANAGEMENT**

Faire de vos succès
notre réussite

www.egilia.com

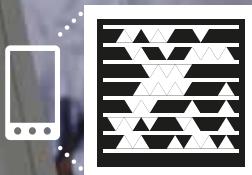
CONTACTEZ NOS CONSEILLERS FORMATION

N° National 0 800 800 900

APPEL GRATUIT DEPUIS UN POSTE FIXE

ANVERS . LIEGE . PARIS . LYON . LILLE . AIX-EN-PROVENCE .
STRASBOURG . RENNES . BRUXELLES
TOULOUSE . BORDEAUX . GENEVE . LAUSANNE . ZURICH .

**L'INFINI EST MA SEULE LIMITE
DE DEVELOPPEMENT.
C'EST ÇA LA PUISSANCE DU CLOUD.**



Pour utiliser ce flashcode,
téléchargez l'application
gratuite sur :
<http://gettag.mobi>

Windows Azure est une plateforme qui vous permet de déployer et exploiter vos applications dans le Cloud avec vos outils et compétences d'aujourd'hui. En outre, c'est une plateforme d'une élasticité infinie qui épouse parfaitement les fluctuations de votre activité. Avec Windows Azure, pensez à votre potentiel avant de penser à vos contraintes ; et n'ayez comme seule limite que votre imagination.

La simplicité - une des nombreuses vertus du Cloud.



Microsoft

Cloud Power

LA PLUS LARGE GAMME DE SOLUTIONS DANS LE NUAGE. SUR TERRE.

Microsoft® Office 365 • Windows Azure™ • Windows Server® Hyper-V® • Microsoft Dynamics™ CRM Online.

En savoir plus : www.microsoft.fr/cloud