Six-Axe Consultants
Groupe Six-Axe

Installation d'une solution open source PHP/MySQL sous IBM i

Session 13

Grégory Jarrige
gjarrige@six-axe.fr

zend CERTIFIED ENGINEER PHP 5

Solution open-source PHP pour IBMi

Six-Axe Consultants
Groupe Six-Axe

- Au sujet de l'auteur de cette présentation :
  - Développe sur plateforme IBM i (ex.AS/400) depuis 1991
  - Expert PHP5 certifié par Zend depuis février 2010
  - Consultant open-source chez Six-Axe Consultants
  - Domaines d'expertise : PHP 5, jQuery, HTML 5, RPG, SQL DB2 (incluant procédures stockées et SQLRPGLE)
  - Assure des sessions de formations, et de transfert de compétences, sur les technos précitées
  - Rédacteur d'un blog dédié à la diffusion de bonnes pratiques autour de PHP et de DB2 : http://www.gregphplab.com, et contributeur régulier du site XDocs400.com

Solution open-source PHP pour IBMi

- Pour tout savoir sur les activités de Six-Axe Consultants, je vous invite à consulter le site de la société :

http://www.six-axe.fr

Solution open-source PHP pour IBMi

Six-Axe Consultants
Groupe Six-Axe

Rappel des objectifs de cette session :

Avec le support de PHP et de MySQL via les solutions Zend, l'IBM i s'est ouvert à des milliers d'applications Web disponibles sur le marché. Et l'annonce du "DB2 Storage Engine for MySQL" a permis, de façon transparente pour les applications, de stocker les données de MySQL dans DB2. On abordera dans cette session un certain nombre de points sur lesquels vous devez être vigilants lorsque vous souhaitez installer une solution PHP open source sous IBM i.

Après une introduction aux aspects "sécurité" et "architecture applicative", l'accent sera mis sur la possibilité d'utiliser DB2 pour stocker les données de MySQL .Vous verrez que plusieurs solutions sont envisageables, mais que selon la manière dont l'application PHP est structurée, certaines solutions sont plus faciles à mettre en œuvre que d'autres. Le propos sera illustré par une installation "en live" de la solution e-commerce Opencart, et par son utilisation avec la base de données DB2 for i.

Solution open-source PHP pour IBMi

Six-Axe Consultants
Groupe Six-Axe

- Faire le choix d'une solution open-source implique de se poser un certain nombre de questions.
- Parmi les questions qui ne sont pas spécifiques à l'open source, on trouve…
  - Est-ce que la solution remplace une solution existante ? Si oui l'équipe projet en charge de l'ancienne solution est-elle en mesure de prendre en charge le pilotage de la nouvelle solution ?
  - Est-ce que la solution open-source s'appuie sur des langages qui sont maîtrisés au sein de votre organisation ? Faut-il mettre en place un plan de formation spécifique pour faire monter les équipes en compétence ? Faut-il recruter des ressources complémentaires ?

Solution open-source PHP pour IBMi

Six-Axe Consultants
Groupe Six-Axe

- Parmi les questions plus spécifiques à l'open source :
  - la compatibilité de la licence utilisée par la solution open source par rapport à la destination de votre projet ? Licence BSD, FreeBSD, GNU GPL, GNU LGPL, autre… pour y voir clair, il est peut être bon de vous faire conseiller par un juriste spécialisé dans les problématiques de licence open-source.
  - Quelques liens utiles pour vous aider à y voir plus clair :
    - http://fr.wikipedia.org/wiki/Licence_BSD
    - http://fr.wikipedia.org/wiki/LGPL
    - http://fr.wikipedia.org/wiki/Licence_publique_g%C3%A9n%C3%A9rale_GNU

Solution open-source PHP pour IBMi

- Quelques questions supplémentaires à se poser… (la suite):
  - La solution open-source qui vous intéresse dispose-t-elle d'une bonne documentation ?
  - A-t-elle une communauté d'utilisateurs active ? (passez du temps sur les forums dédiés au projet, pour vous faire une idée)
  - A-t-elle un support réactif ? Ou à défaut un support privilégié payant ?
  - Bénéficie-t-elle de mises à jour régulières ?
    - pensez à éplucher les changelogs : les dernières modifications recensées sont-elles d'ordre techniques, ou fonctionnelles, ou les deux ?
    - y a-t-il eu des failles de sécurité identifiées, ont-elles été corrigées rapidement ?
    - Prenez le temps de lire la feuille de route (roadmap), s'il y en a une...
  - Le rythme des mises à jour est-il élevé ? Certaines solutions bénéficient d'une à deux mises à jour par mois, est-ce compatible avec les contraintes de planning et d'exploitation de votre projet ?

Solution open-source PHP pour IBMi

Six-Axe Consultants
Groupe Six-Axe

- Quelques questions à se poser… (la suite):
  - La solution est-elle connue et reconnue par le marché ?
    - Trouve-t-on facilement des compétences sur le marché, pour renforcer l'équipe interne, soit ponctuellement, soit sur du long terme ?
    - Trouve-t-on facilement des sociétés capables d'intervenir sur cette solution pour une assistance, ou du développement spécifique par exemple ?
  - Quel est le modèle économique de l'équipe en charge du projet ? Semble-t-il pérenne ?
    - Si le projet est stratégique pour vous, votre société pourrait peut être devenir sponsor officiel du projet, afin de soutenir l'équipe de développement, et de pérenniser le projet (cela peut être très positif en termes d'image).

Solution open-source PHP pour IBMi

Six-Axe Consultants
Groupe Six-Axe

- Quelques questions d'ordre plus techniques, concernant la solution open-source envisagée:
  - Quelle est l'architecture de l'application (MVC, autre…) ?
  - Dans quelle version de PHP est-elle écrite (4.x, 5.2, 5.3, 5.4, …) ? Si c'est du PHP 4, stoppez tout !!!
  - Utilise-t-elle un framework PHP et si oui lequel ? Dans quelle version ?
  - L'application utilise très certainement du Javascript, mais dans quelles proportions ?
  - L'application utilise-t-elle un, voire plusieurs, framework(s) Javascript(s) ? Si oui, lequel, ou lesquels ? Et dans quelle(s) version(s) ?

Solution open-source PHP pour IBMi

Six-Axe Consultants
Groupe Six-Axe

- Questions d'ordre plus techniques (suite) :
  - Quelles sont les base de données supportées (MySQL, PostgreSQL, DB2, autre…) ?
  - Quelle extension (connecteur) base de données est utilisée (mysql, mysqi, PDO, autre…) ?
  - Les connecteurs sont-ils facilement interchangeables au moyen d'un wrapper par exemple ?
  - Le code SQL est-il disséminé un peu partout dans l'application, ou est-il au contraire centralisé dans un jeu de fonctions ou classes clairement identifiables ?
  - Le code source dans son ensemble (PHP, Javascript, autre...) est-il clairement documenté ?

Solution open-source PHP pour IBMi

Six-Axe Consultants
Groupe Six-Axe

L'un des éléments clés dans le choix d'une solution open-source, celui qui doit faire l'objet de toutes les attentions, c'est :

La sécurité !!!

L'audit de code effectué par un expert en sécurité ou en développement open-source permettra d'évaluer le nombre et la gravité des failles de sécurité d'une solution open-source.

Six-Axe Consultants Groupe Six-Axe

Parmi les failles les plus fréquemment rencontrées, il y en a une contre laquelle il est facile de se protéger, si on veut bien s'en donner la peine, il s'agit de la :

vulnérabilité face aux attaques par injection SQL

J'ai écrit un article sur ce sujet récemment, sur mon blog :

http://www.gregphplab.com

Pour de plus amples renseignements, prière de se reporter au dossier que j'ai intitulé :

Développeurs, arrêtez d'utiliser la vieille extension « mysql » !

Solution open-source PHP pour IBMi

Six-Axe Consultants
Groupe Six-Axe

Pour la suite de ma présentation, j'avais besoin...

- ... d'une solution open-source à installer sur un IBM i qui puisse intéresser fonctionnellement un maximum de monde... va pour une solution e-commerce !!!
- … d'une solution e-commerce de taille raisonnable, riche fonctionnellement mais pas trop complexe malgré tout, pour qu'elle puisse être prise en main par des équipes réduites comme celles que je rencontre souvent chez mes clients IBM i ?
- … d'une solution pas trop complexe donc relativement facile à personnaliser, si besoin...
- … d'une solution e-commerce qui bénéficie d'un rythme de mise à jour raisonnable (tous les 3 à 6 mois par exemple).

Six-Axe Consultants
Groupe Six-Axe

And the Winner is ....

opencart

Solution open-source PHP pour IBMi

Six-Axe Consultants
Groupe Six-Axe

opencart

- Opencart est une application développée pour une large part en PHP 5, par la société Opencart basée à Hong-Kong.
- Le projet est publié sous licence GNU GPL :
  - http://www.gnu.org/licenses/quick-guide-gplv3.html
- La solution est donc gratuite, et Opencart fonde son modèle économique sur la vente de services et modules additionnels, par l'intermédiaire du site internet Hostjars.
- Liens :
  - Site officiel d'opencart : http://www.opencart.com
  - Site officiel de Hostjars : http://hostjars.com
- Avertissement : *l'auteur de cette présentation, ainsi que la société Six-Axe Consultants, n'entretiennent à l'heure actuelle aucune relation commerciale avec Opencart et Hostjars. Le choix de présenter la solution opencart est dicté par les seuls critères évoqués sur la page qui suit.*

Solution open-source PHP pour IBMi

Six-Axe Consultants
Groupe Six-Axe

opencart

- Le choix d'opencart pour cette présentation est un choix pragmatique, dicté par plusieurs critères :
  - Développé en PHP 5, le projet est doté d'une architecture MVC (Modèle Vue Contrôleur) plutôt bien pensée et facile à comprendre (à noter que l'on retrouve la même architecture pour le front-office et le back-office, ce qui est très appréciable)
  - Doté d'une ergonomie agréable et conforme aux standards actuels
  - S'appuie pour les effets du front-office sur jQuery, framework javascript que l'on peut considérer comme un standard du marché
  - Couche d'accès base de données assez « lite » s'appuyant sur un « wrapper », facile à adapter à DB2 (la base d'origine étant MySQL)
  - Taille de code source raisonnable (cf. diapo suivante), permettant d'envisager une maîtrise rapide, si vous envisagez de personnaliser la solution pour l'interfacer avec un back-office RPG ou Cobol.
  - Une gestion de catalogue produit étoffée pouvant convenir à de nombreuses organisations (hormis la gestion de tarif, trop basique, qui nécessitera très certainement d'être personnalisée, par exemple en allant chercher le tarif des produits dans le back-office IBM i, via une UDF (User Defined Function) DB2.

Solution open-source PHP pour IBMi

Six-Axe Consultants
Groupe Six-Axe

opencart

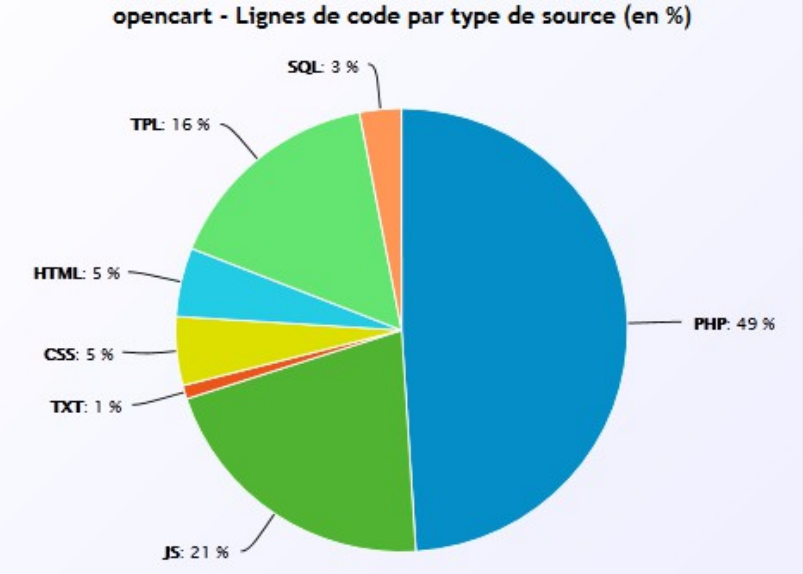
- La solution est soutenue par une communauté francophone qui communique via un forum et un site dédié :

Solution open-source PHP pour IBMi

Six-Axe Consultants
Groupe Six-Axe

opencart

Le nombre de lignes de code d'opencart est très raisonnable, surtout si on le compare à celui de certaines solutions concurrentes.



Extension	Nb.lignes(1)	Nb.fichiers(2)	(1)/(2)
PHP	83746	924	91
JS	38393	400	96
TPL	29458	294	100
HTML	9827	67	147
CSS	9562	103	93
SQL	7065	2	3533
TXT	3167	18	176
XML	53	5	11
INI	35	3	12
IMAGES	0	977	0
OTHER	0	149	0

Solution open-source PHP pour IBMi

Six-Axe Consultants
Groupe Six-Axe

opencart

Mise au point :

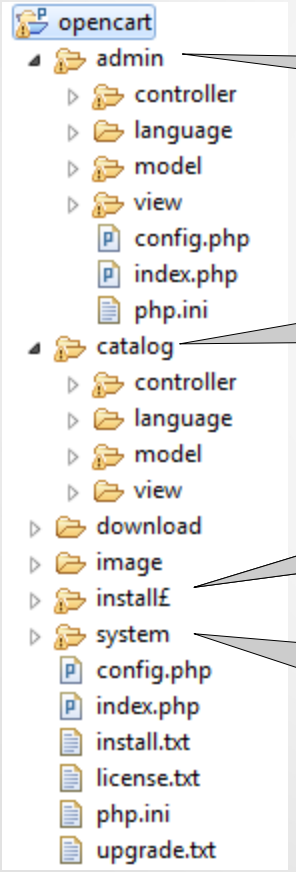
Durant la présentation, j'ai fait un parallèle entre la volumétrie (le nombre de lignes de codes) d'Opencart, et la volumétrie très supérieure de solutions e-commerces concurrentes.

Ces solutions alternatives présentent un niveau de fonctionnalité extrêmement intéressant. Mais leur niveau de complexité fait qu'il est peut être préférable de les envisager dans le cadre d'une approche de type « cloud », en recourant aux services d'hébergeurs spécialisés possédant le niveau d'expertise adéquat pour en assurer l'administration dans les meilleures conditions.

Solution open-source PHP pour IBMi

Six-Axe Consultants
Groupe Six-Axe

opencart



Architecture MVC pour le back-office

Architecture MVC similaire pour le front-office

Ne pas oublier de supprimer le répertoire « install » une fois l'installation terminée (mesure de sécurité)

Les éléments communs au back-office et au front-office sont regroupés dans le répertoire « system »

Solution open-source PHP pour IBMi

Six-Axe Consultants
Groupe Six-Axe

opencart

- Le seul point négatif dans Opencart, à mon avis, c'est que l'application utilise, pour les accès bases de données, un wrapper s'appuyant sur la vieille extension « mysql » (cf. le dossier consacré à ce sujet sur mon blog).
- Qu'est ce qu'un « wrapper » : le terme « wrapper » peut être traduit par « conteneur » ou « adaptateur », ici c'est le terme « adaptateur » qui est le plus indiqué. L'adaptateur base de données d'opencart est une classe contenant un jeu de méthodes permettant au développeur d'exécuter des requêtes SQL sans appeler directement les fonctions de l'extension « mysql ».
- Exemple ci-dessous de requête exécutée en faisant appel au « wrapper » d'opencart, plutôt qu'à la fonction équivalente de l'extension « mysql » :

```
$user_query = $this->db->query("SELECT * FROM " .
      DB_PREFIX . "user WHERE user_id = '" .
     (int)$this->session->data['user_id'] . "' AND status = '1'");
```

La technique de « casting » ci-dessus, utilisée de manière assez générale dans Opencart permet de se protéger contre les attaques par injection SQL,
mais elle n'est utilisable, et donc efficace, que sur les colonnes de type numérique entier.

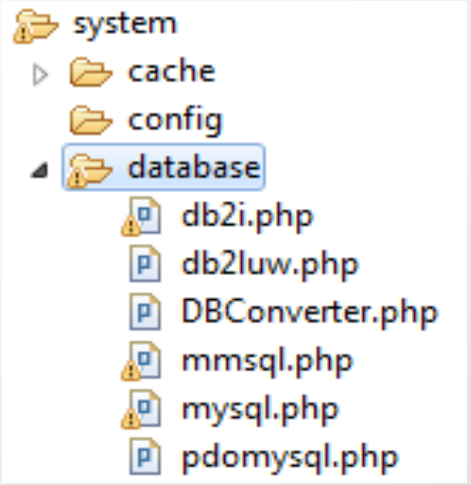
Solution open-source PHP pour IBMi

Six-Axe Consultants
Groupe Six-Axe

opencart

- Le « wrapper » d'opencart étant facile à comprendre, il est aussi facile à modifier, et à améliorer, du coup j'en ai écrit plusieurs :
  - un premier, pdomysql, utilisant l'extension « pdo_mysql », pour remplacer le wrapper standard, dans le cadre d'une utilisation avec MySQL
  - un second, db2luw, utilisant l'extension « pdo_odbc » (pour une utilisation avec DB2 pour Windows, c'est le fichier db2luw.php),
  - un troisième, db2i, utilisant l'extension « ibm_db2 » (pour une utilisation avec DB2 pour IBMi) :

J'ai également créé une classe DBConverter, dans laquelle j'ai regroupé quelques fonctions permettant de convertir à la volée certaines fonctions MySQL en des fonctions équivalentes adaptées à DB2.



Solution open-source PHP pour IBMi

Six-Axe Consultants Groupe Six-Axe

Nous allons quitter un moment ce support pour procéder à l'installation en live d'opencart.

Pour information, et pour gagner du temps, j'ai déjà effectué sur le serveur IBMi les actions suivantes :

- création d'une base de données MySQL vierge appelée « opencart2 » (avec encodage par défaut en « utf8_general_ci ».

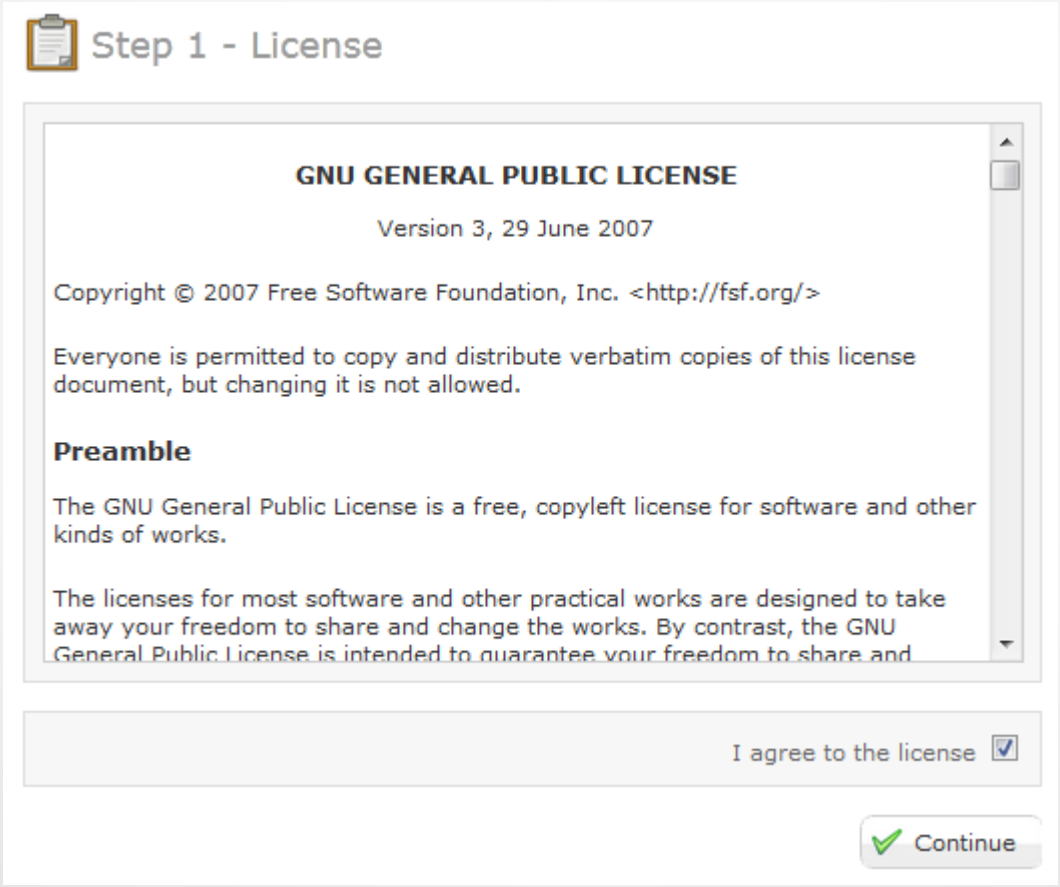
- installation dans l'IFS, dans le répertoire `/www/zendsvr/htdocs/opencart2`, du contenu du zip de l'application, récupéré sur le site officiel (étape déjà réalisée pour gagner du temps). La version installée est la v1.5.5.1 du 22 janvier 2013.

L'installation est très simple, elle se fait en 4 étapes, il suffit de suivre les instructions.

Solution open-source PHP pour IBMi

Six-Axe Consultants
Groupe Six-Axe

Une installation en 4 étapes :
Etape 1 : acceptation de la licence



Solution open-source PHP pour IBMi

Six-Axe Consultants
Groupe Six-Axe

Etape 2 : vérification de l'environnement

Step 2 - Pre-Installation

1. Please configure your PHP settings to match requirements listed below.

PHP Settings	Current Settings	Required Settings	Status
PHP Version:	5.3.14	5.0+	
Register Globals:	Off	Off	
Magic Quotes GPC:	Off	Off	
File Uploads:	On	On	
Session Auto Start:	Off	Off	

2. Please make sure the extensions listed below are installed.

Extension	Current Settings	Required Settings	Status
MySQL:	On	On	
GD:	On	On	
cURL:	On	On	
ZIP:	On	On	

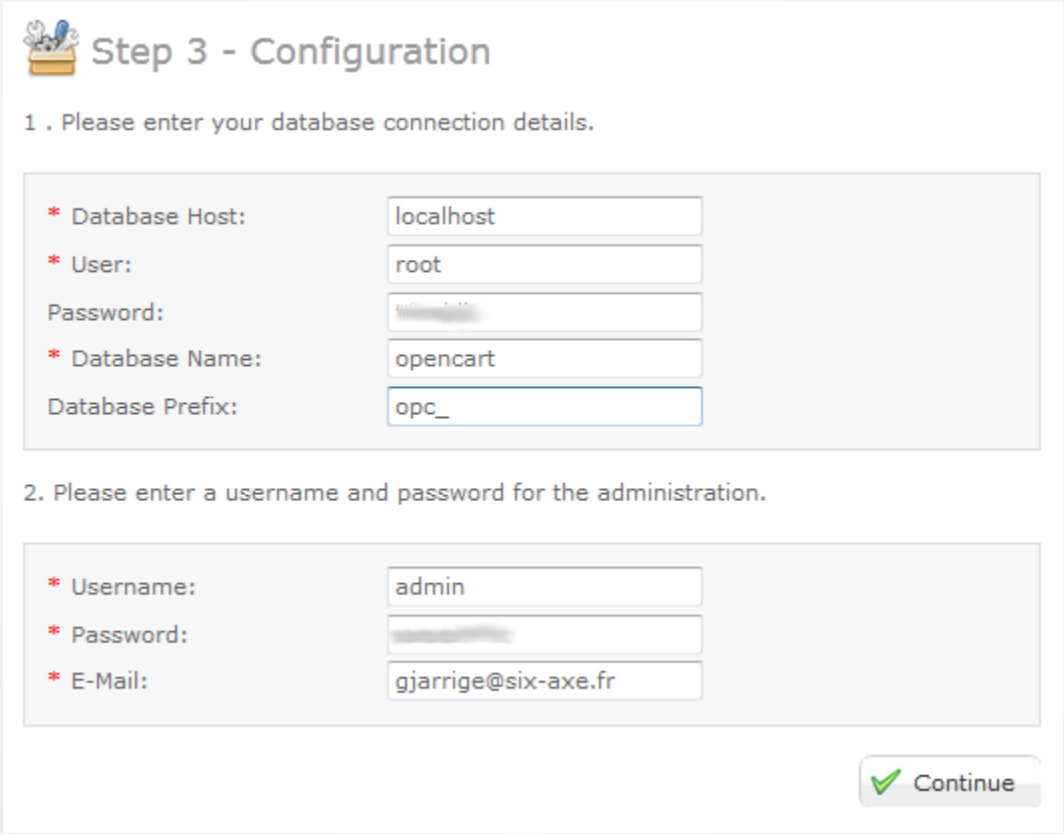
3. Please make sure you have set the correct permissions on the files list below.

Files	Status
C:\Program Files (x86)\Zend\Apache2\htdocs\opencartx/config.php	Writable

Solution open-source PHP pour IBMi

Six-Axe Consultants
Groupe Six-Axe

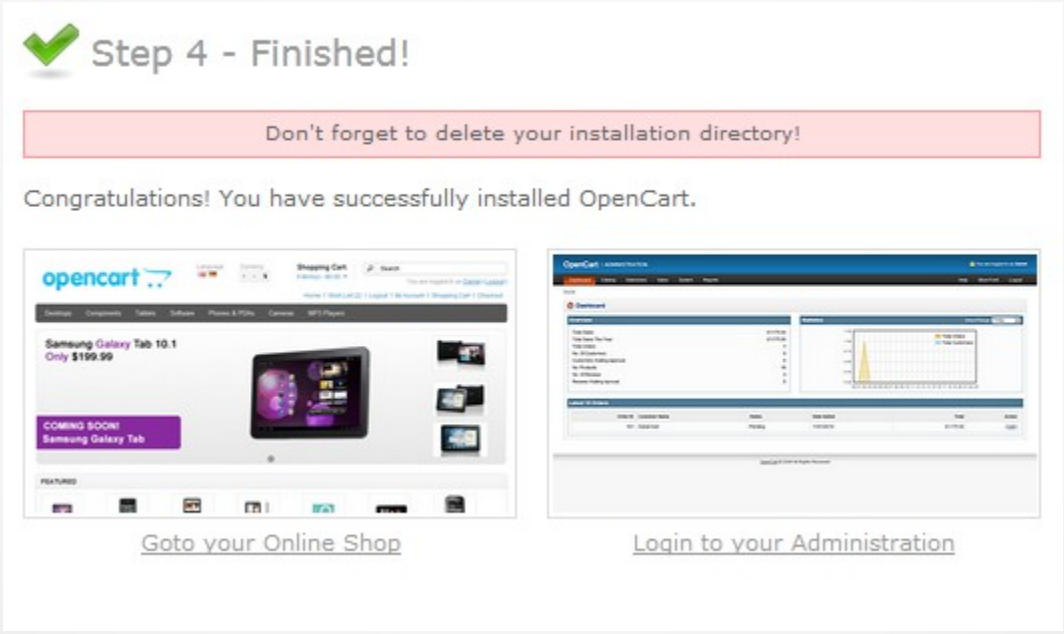
Etape 3 : configuration et définition des identifiants



Solution open-source PHP pour IBMi

Six-Axe Consultants
Groupe Six-Axe

Etape 4 : si tout s'est bien passé, vous devez obtenir un écran tel que celui ci-dessous :



Solution open-source PHP pour IBMi

Six-Axe Consultants
Groupe Six-Axe

Nous disposons maintenant d'une application PHP opérationnelle sur notre IBMi, alors penchons-nous sur la partie « base de données » :

Par défaut, opencart travaille avec la base MySQL. Ca fonctionne très bien, mais, puisqu'on est sur un IBMi, ne pourrait-on pas faire fonctionner opencart avec DB2 ?

YES, YOU CAN !!!

Et je dirais même plus... vous pouvez le faire de 2 manières différentes, que nous allons détailler dans la suite de la présentation.

Six-Axe Consultants Groupe Six-Axe

Comment faire fonctionner opencart avec DB2 ?

Solution n°1 : utiliser le moteur MySQL IBMDB2i fourni par IBM

Solution n°2 : recréer la base de données d'opencart en « full » DB2, et utiliser le wrapper que j'ai créé spécialement pour cette utilisation.

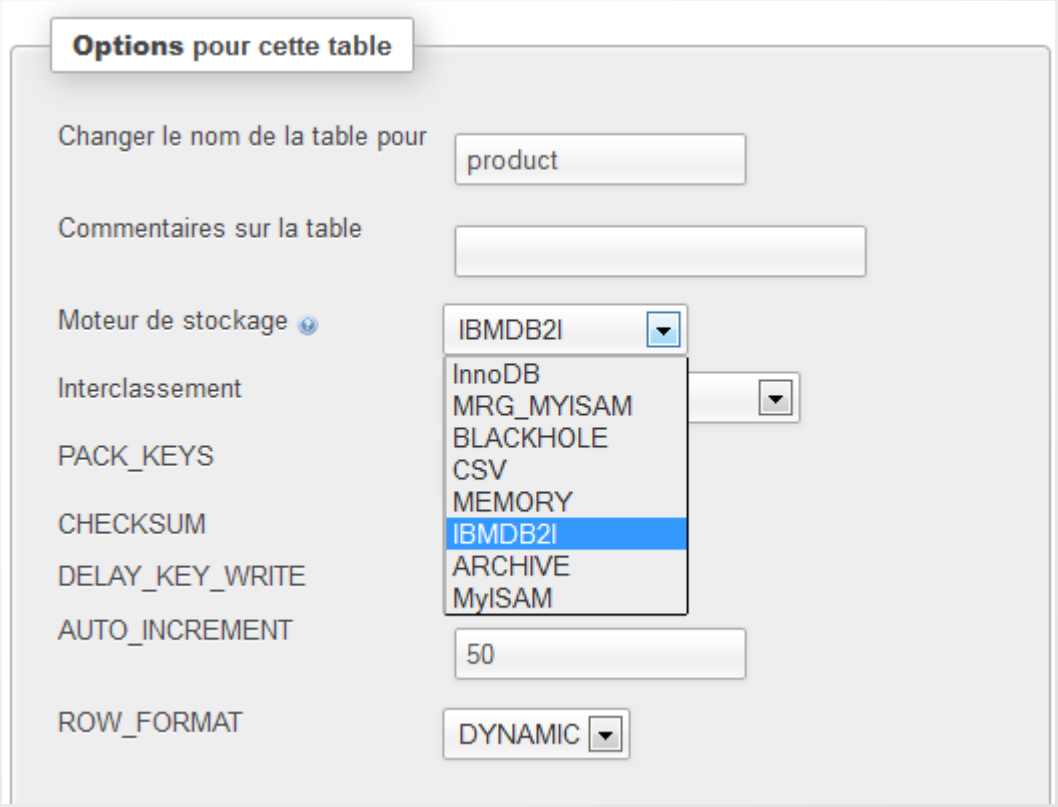
Chacune de ces solutions a des avantages et des inconvénients, nous allons les détailler ci-après.

Solution open-source PHP pour IBMi

Solution n°1 : moteur MySQL IBMDB2i

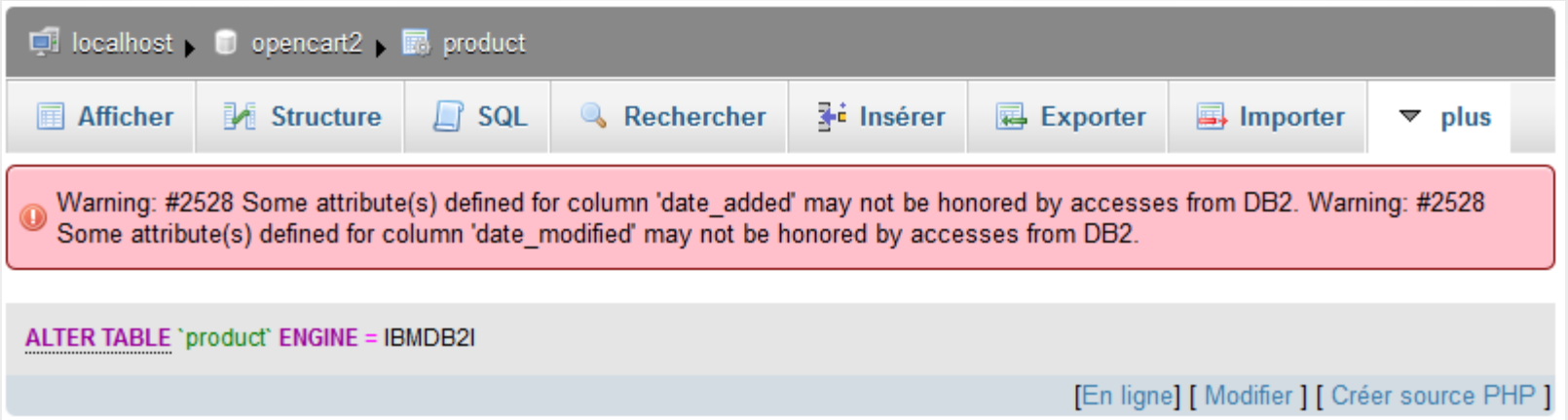
Le moteur MySQL IBMDB2i est un produit fourni gracieusement par IBM aux clients utilisant ZendServer sur la plateforme IBMi.

Sous PHPMyAdmin, sélectionnons dans notre base opencart2 la table product, puis cliquons sur le menu « opérations ».
On voit qu'il est possible de modifier le moteur de stockage (par défaut MyISAM) en le remplaçant par le moteur IBMDB2i.
Cliquons sur « exécuter » et regardons ce qui se passe.



Six-Axe Consultants Groupe Six-Axe

Solution n°1 : moteur MySQL IBMDB2i



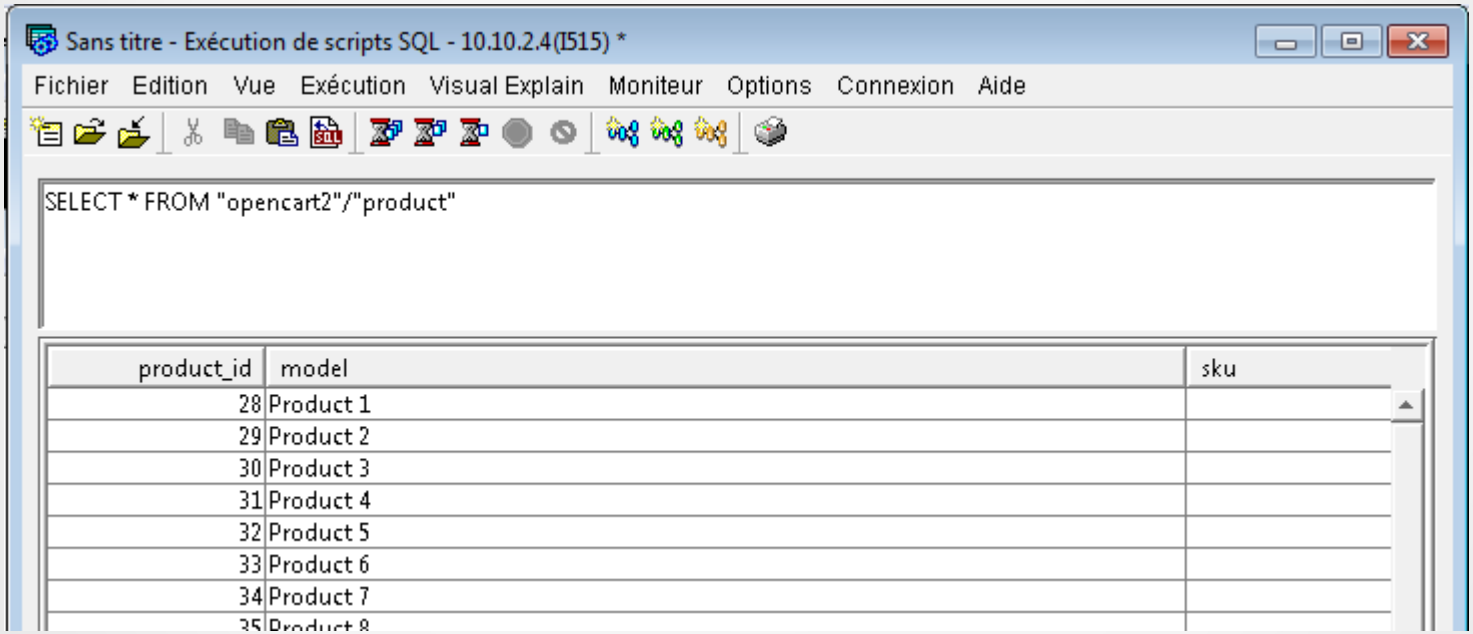
Et voilà ! Un simple ALTER TABLE a suffi pour créer sur l'IBMi une bibliothèque DB2 portant le nom de "opencart2" avec les guillemets devant et derrière le nom. La présence des guillemets est un effet de bord lié au fait que les objets d'origine ont des noms en minuscule (ce qui est le cas de 99 % des bases de données MySQL).
A l'intérieur de cette nouvelle bibliothèque, le moteur IBMDB2i a créé une table "product" (avec les guillemets également), dans laquelle il a importé les données provenant de la table MySQL d'origine.
A noter : si des indexs existent sur la table MySQL d'origine, ils sont recréés à l'identique sur la table DB2 créée par le moteur IBMDB2i.

Solution open-source PHP pour IBMi

Six-Axe Consultants
Groupe Six-Axe

Solution n°1 : moteur MySQL IBMDB2i

Requêtage sur la nouvelle table DB2 via le logiciel Systemi Navigator :



A noter : le logiciel Systemi Navigator d'IBM n'est pas très à l'aise avec les bibliothèques dont les noms contiennent des guillemets en première et dernière position. Il est par exemple très difficile de lui demander d'ajouter une bibliothèque de ce type à la liste des schémas affichables par défaut (j'y suis arrivé une fois... par accident, et je n'ai pas réussi à réitérer cet exploit).

Six-Axe Consultants Groupe Six-Axe

Solution n°1 : moteur MySQL IBMDB2i

Opencart est constituée d'environ 90 tables, s'il faut faire un ALTER TABLE sur chacune de ces tables pour la transférer sous DB2, vous risquez de faire une dépression.

Un conseil, utilisez les métadonnées de MySQL pour extraire la liste des tables et générer le script SQL qui vous permettra d'effectuer un changement de moteur en masse :

```
SELECT concat( 'ALTER TABLE ', table_schema, '.', table_name,
    ' ENGINE=IBMDB2I ;' )
FROM information_schema.TABLES
WHERE TABLE_SCHEMA = 'opencart2'
```

Ce qui donne à l'arrivée :

```
ALTER TABLE opencart2.oc_address ENGINE=IBMDB2I ;
ALTER TABLE opencart2.oc_affiliate ENGINE=IBMDB2I ;
ALTER TABLE opencart2.oc_affiliate_transaction ENGINE=IBMDB2I ;
ALTER TABLE opencart2.oc_attribute ENGINE=IBMDB2I ;
...
```

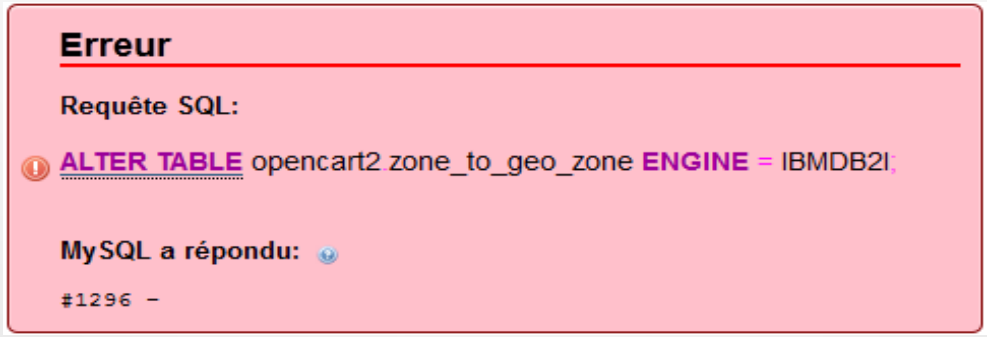
Six-Axe Consultants Groupe Six-Axe

Solution n°1 : moteur MySQL IBMDB2i

La migration des tables de MySQL vers DB2 a bien fonctionné pour la plupart des tables, sauf pour les 3 tables ci-dessous :

```
ALTER TABLE opencart2.product_discount ENGINE=IBMDB2I ;
ALTER TABLE opencart2.product_special ENGINE=IBMDB2I ;
ALTER TABLE opencart2.zone_to_geo_zone ENGINE=IBMDB2I ;
```

Le message d'erreur renvoyé par MySQL est à chaque fois très laconique, et je n'ai pas réussi pour l'instant à identifier l'origine du problème :



Six-Axe Consultants Groupe Six-Axe

Solution n°1 : moteur MySQL IBMDB2i

Les 3 tables pour lesquelles la conversion DB2 a été refusée sont des tables de paramétrage.

Etant demeurées sur MySQL, elles ne peuvent être utilisées en jointure avec des tables fonctionnant sous le moteur IBMDB2i, car le moteur ne sait pas gérer ce cas de figure.

Heureusement, dans la version actuelle d'Opencart, il semble qu'elles soient utilisées unitairement, sans passer par des jointures avec d'autres tables.

On pourrait donc considérer que ces tables peuvent demeurer sur MySQL, sans que cela ne pose de problème particulier.

Mais c'est un point à surveiller, voire un facteur de risque, car si dans une future version d'Opencart les développeurs décidaient d'utiliser ces tables dans des jointures avec des tables qui elles sont bien passées sous le contrôle du moteur IBMDB2i, ces requêtes ne fonctionneraient pas.

Six-Axe Consultants Groupe Six-Axe

Solution n°1 : moteur MySQL IBMDB2i

Avantages de la solution :

- une grande simplicité de mise en œuvre
- les tables étant sous DB2 (à l'exception des 3 précitées), elles deviennent accessibles aux applications IBMi qui peuvent les consulter et éventuellement les mettre à jour, et elles peuvent être administrées, sauvegardées, restaurées, de la même manière que toute autre table DB2.
- les performances : vous profitez des performances et de la capacité de montée en charge de DB2 pour IBM i.

Inconvénients de la solution :

- on l'a vu avec Opencart, le processus de conversion sous DB2 ne gagne pas à tous les coups
- les règles de nommage adoptées par le moteur IBMDB2i sont somme toute logiques, mais néanmoins contraignantes (obligation dans les requêtes DB2 SQL de placer des guillemets devant et derrière chaque objet DB2 (bibliothèques et tables).

Six-Axe Consultants Groupe Six-Axe

Solution n°2 : base opencart « full » DB2

Mais au fait ! Recréer la base sous DB2, est-ce que ça a du sens ?

Est-ce que les gains de performances obtenus justifient l'investissement ?

La diapo suivante présente un échantillon de quelques requêtes exécutées sur les 2 moteurs MySQL (MyISAM et InnoDB). J'ai indiqué leurs performances respectives, avec en regard les performances obtenues avec DB2 for i.

Les tables MyISAM et InnoDB sont strictement identiques, hormis le moteur. Elles contiennent le même nombre de lignes (500.000) et les mêmes indexs.

La table DB2 for i est de structure strictement équivalente, a les mêmes indexs que les tables MySQL, et les mêmes 500.000 lignes (données extraites à l'origine d'une table système DB2, en l'occurrence il s'agit de la table QSYS2/SYSCOLUMNS).

Les temps indiqués dans le tableau qui suit sont exprimés en secondes.

Solution open-source PHP pour IBMi

Six-Axe Consultants
Groupe Six-Axe

Requête SQL	MyISAM	InnoDB	DB2	commentaire concernant la version DB2
select * from syscolsm where table_name = ?	0,00193	0,00373	0,02339	idem MySQL
select count(*) as total from syscolsi where table_name = ?	0,00082	0,00217	0,01014	idem MySQL
select * from syscolsm where table_name = ? and column_name = ?	0,00151	0,00228	0,00449	idem MySQL
select p.* from syscolsm p where p.table_name = ? limit 1 , 10	0,00121	0,0018	0,00882	select foo.* from (select row_number() over () as rn, p.* from syscolsi p where p.table_name = ?) as foo where foo.rn between ? and ?
idem requête précédente (pagination) avec curseur scrollable	0,00148	0,00281	0,00487	requête identique à MySQL (scroll cursor)
select count(*) as nb from syscolsm2 p	0,00028	0,00053	0,01011	idem MySQL

Six-Axe Consultants Groupe Six-Axe

Solution n°2 : base opencart « full » DB2

J'avoue avoir été un peu déçu par les performances de DB2 for i.

Mais je pense très sincèrement que l'on ne peut pas tirer de conclusions des tests que j'ai effectués ici. Pour moi ces premiers résultats ne sont pas significatifs.

Pour obtenir des résultats réellement significatifs, il serait judicieux de chercher à « stresser » chaque moteur SQL en exécutant en parallèle un grand nombre de requêtes simultanées, de manière à se rapprocher le plus possible de conditions réelles d'utilisation.

Mais un vrai test de montée en charge, ça prend du temps, et le temps... comme chacun sait... donc je referme la parenthèse « performances » ici (tout en insistant sur le fait que le sujet reste ouvert).

Six-Axe Consultants Groupe Six-Axe

Solution n°2 : base opencart « full » DB2

Comment peut-on recréer la base de données MySQL d'opencart en DB2 SQL pur jus ? Est-ce facile à faire ? Quelles sont les contraintes ? C'est ce que nous allons voir maintenant.

Pour recréer la base de données MySQL en « full » DB2, il est préférable de se constituer quelques outils, histoire de pouvoir automatiser et industrialiser la solution.

Pour ce faire, il nous faudra :

- un peu de code SQL,
- beaucoup plus de code PHP,
- une bonne connaissance des types de données relatifs aux bases MySQL et DB2, et de leurs équivalences,
- et bien sûr un peu de temps de développement...

Six-Axe Consultants Groupe Six-Axe

Solution n°2 : base opencart « full » DB2

Si tous ces ingrédients sont réunis, il devient facile d'écrire un script PHP permettant d'effectuer la conversion d'une base vers l'autre.

Version MySQL

```
CREATE TABLE `oc_address` (
`address_id` int(11) NOT NULL AUTO_INCREMENT,
`customer_id` int(11) NOT NULL,
`firstname` varchar(32) NOT NULL,
`lastname` varchar(32) NOT NULL,
`company` varchar(32) NOT NULL,
`company_id` varchar(32) NOT NULL,
`tax_id` varchar(32) NOT NULL,
`address_1` varchar(128) NOT NULL,
`address_2` varchar(128) NOT NULL,
`city` varchar(128) NOT NULL,
`postcode` varchar(10) NOT NULL,
`country_id` int(11) NOT NULL DEFAULT '0',
`zone_id` int(11) NOT NULL DEFAULT '0',
PRIMARY KEY (`address_id`),
KEY `customer_id` (`customer_id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8
```

Version DB2

```
CREATE TABLE your_library/oc_address (
ADDRESS_ID integer GENERATED BY DEFAULT AS IDENTITY (
  START WITH 50 INCREMENT BY 1
  NO MINVALUE MAXVALUE 2147483647
  NO CYCLE NO ORDER
  CACHE 20 ) ,
CUSTOMER_ID integer not null default 0,
FIRSTNAME varchar(96) CCSID 1208 not null default '',
LASTNAME varchar(96) CCSID 1208 not null default '',
COMPANY varchar(96) CCSID 1208 not null default '',
COMPANY_ID varchar(96) CCSID 1208 not null default '',
TAX_ID varchar(96) CCSID 1208 not null default '',
ADDRESS_1 varchar(384) CCSID 1208 not null default '',
ADDRESS_2 varchar(384) CCSID 1208 not null default '',
CITY varchar(384) CCSID 1208 not null default '',
POSTCODE varchar(30) CCSID 1208 not null default '',
COUNTRY_ID integer not null default 0,
ZONE_ID integer not null default 0
,
CONSTRAINT your_library/Q_OC_ADDRESS_00001 PRIMARY
KEY( "ADDRESS_ID" )
) ;
```

Six-Axe Consultants Groupe Six-Axe

Solution n°2 : base opencart « full » DB2

Nous avons l'outil de conversion et de recréation des tables, mais il nous faut aussi les données. Puisque ces données existent dans la bibliothèque que nous avons créée via le moteur MySQL IBMDB2i, autant aller les chercher là où elles se trouvent.
Avec un peu de code SQL et de code PHP, on peut se créer un script qui va générer automatiquement les requêtes ci-dessous :

```
insert into yourlib/oc_address select * from "opencart2"/"address" ;
insert into yourlib/oc_affiliate select * from
 "opencart2"/"affiliate" ;
insert into yourlib/oc_affiliate_transaction select * from
 "opencart2"/"affiliate_transaction" ;
insert into yourlib/oc_attribute select * from
 "opencart2"/"attribute" ;
insert into yourlib/oc_attribute_description select * from
 "opencart2"/"attribute_description" ;
```

…

Il restera à traiter, de manière plus « manuelle » les 3 tables récalcitrantes que le moteur IBMDB2i n'a pas réussi à convertir sous DB2.

Six-Axe Consultants Groupe Six-Axe

Solution n°2 : base opencart « full » DB2

Nous avons les tables, et les données en full SQL DB2.
Pour que l'application opencart puisse les « attaquer » directement, sans passer par le moteur MySQL IBMDB2i, il nous faut un « wrapper » permettant de remplacer les appels à MySQL par des appels équivalents à DB2. Ces appels se feront via l'extension « ibm_db2 » qui est fournie en standard avec ZendServer pour IBMi.
Voici une version très abrégée du wrapper que j'ai écrit pour l'occasion :

```
final class DB2i {
    private $connection;
    private $statement;
    private $nbr_rows_affected;
    public function __construct($hostname, $username, $password, $database) {..
    public function query($sql, $params = array()) {..
    public function execute($sql, $params = array()) {..
    public function escape($value) {..
    public function countAffected() {..
    public function getLastId() {..
    public function __destruct() {..
    public function getSGBDName() {..
    public function getSGBDVersion() {..
    public function getSGBDServer() {..
}
```

Version améliorée autorisant le passage de paramètres, et donc l'exécution de requêtes SQL paramétrées

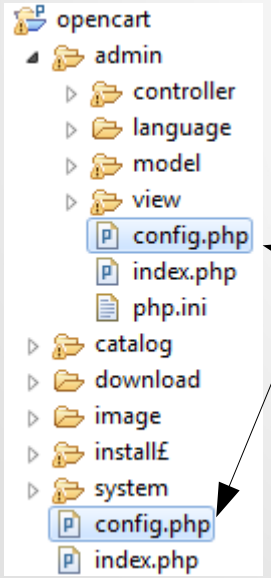
Nouvelle méthode ajoutée pour pouvoir exécuter des requêtes ne renvoyant aucun jeu de données.

Nouvelles méthodes que j'ai ajoutées pour traiter des cas particuliers.

Six-Axe Consultants Groupe Six-Axe

Solution n°2 : base opencart « full » DB2

Nous avons les tables, les données, le wrapper, il nous faut aussi indiquer à opencart que nous souhaitons remplacer le wrapper par défaut par notre wrapper spécifique.
Cette modification se fait très simplement, et à l'identique, dans les 2 fichiers config.php suivants :




```
define('DB_DRIVER', 'db2i');  // wrapper spécifique créé spécialement pour DB2
define('DB_HOSTNAME', '*LOCAL');  // base de données "locale", donc DB2 pour IBMi
define('DB_USERNAME', 'XXXXXX');   // profil de connexion
define('DB_PASSWORD', 'YYYYYY');  // mot de passe de votre profil de connexion
define('DB_DATABASE', 'opencart2'); // nom de le base de données MySQL que vous avez créée
define('DB_PREFIX', 'opc'); // à définir en fonction de ce que vous avez indiqué à l'installation
```

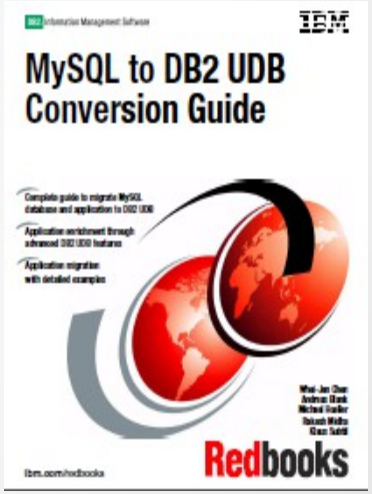
Six-Axe Consultants Groupe Six-Axe

Solution n°2 : base opencart « full » DB2

DB2 et MySQL ont de nombreuses similitudes, mais aussi des différences syntaxiques notables, qui vont nécessiter quelques adaptations.

Pour faire le point sur ces différences, IBM propose un intéressant redbook librement téléchargeable au format PDF (et EPUB),

http://www.redbooks.ibm.com/abstracts/sg247093.html



Autre source d'information intéressante :
http://fadace.developpez.com/sgbdcmp/fonctions/

Six-Axe Consultants Groupe Six-Axe

Solution n°2 : base opencart « full » DB2

Certaines différences syntaxiques, peuvent être traitées à la volée, au moyen d'un peu de code PHP placé dans le wrapper, avant l'exécution de chaque requête :

```
$sql_ori = array('`', 'LCASE', 'UCASE', 'LIMIT 1)', 'NOW()', "= '0000-00-00'");
$sql_des = array('', 'LOWER', 'UPPER', 'FETCH FIRST 1 ROW ONLY)', 'CURRENT DATE', 'IS NULL');
$sql = str_ireplace($sql_ori, $sql_des, $sql);
```

La fonction CONCAT ne fonctionne pas exactement de la même façon sous DB2 et MySQL

Sous MySQL	Sous DB2

`CONCAT(col1, col2, col3) as result`	Solution 1 :
	`CONCAT(CONCAT(col1, col2), col3) as result`
	Solution 2 :
	`col1 || col2 || col3 as result`

Un peu de code PHP placé à l'intérieur du wrapper va permettre de remplacer la syntaxe MySQL de CONCAT() par la solution 2 utilisant des doubles barres verticales (ou au choix par un ordre « concat » placé entre chacune des colonnes).

Six-Axe Consultants Groupe Six-Axe

Solution n°2 : base opencart « full » DB2

La pagination est beaucoup plus facile à gérer sous MySQL que sous DB2.

Sous MySQL	Sous DB2

`SELECT * FROM table`
`WHERE col1 = ?`
`ORDER BY colx`
`LIMIT 1, 10`
	Solution 1 : technique « full » SQL
	`SELECT foo.* FROM (`
	`  SELECT row_number() over (ORDER BY colx) as rn,`
	`  A.*`
	`  FROM table A`
	`  WHERE A.col1 = ?`
	`) AS foo`
	`WHERE foo.rn BETWEEN 1 AND 10`
	
	Solution 2 : curseur scrollable

J'ai présenté en détail les 2 techniques de pagination DB2 l'année dernière, dans la session consacrée aux « bonnes pratiques PHP et DB2 ». Vous pouvez consulter le slide sur mon blog : http://www.gregphplab.com

Dans le wrapper DB2, je détecte la présence de la clause LIMIT à l'intérieur de la requête, et la remplace à la volée par un curseur scrollable DB2.

Ces « adaptations » sont pilotées par la classe DBConverter que j'ai créée pour l'occasion.

Six-Axe Consultants Groupe Six-Axe

Solution n°2 : base opencart « full » DB2

D'autres différences syntaxiques plus subtiles sont difficiles, voire impossibles, à traiter à l'intérieur du wrapper. J'ai donc ajouté au wrapper la méthode getSGBDName(), permettant d'indiquer à l'application si la base de données sous-jacente est DB2 ou MySQL, ceci afin de pouvoir adapter certaines requêtes en amont du wrapper.
Par exemple, dans la requête ci-dessous, l'absence du préfixe «p» devant certaines colonnes ne posait pas de problème avec MySQL, mais provoquait un plantage sous DB2 :

```
    public function getCategory($category_id) {

        if ($this->db->getSGBDName() == 'DB2') {
          $query = $this->db->query("SELECT DISTINCT p.*, (SELECT keyword FROM " .
DB_PREFIX . "url_alias WHERE query = 'category_id=" . (int)$category_id . "') AS
keyword FROM " . DB_PREFIX . "category p WHERE p.category_id = '" . (int)
$category_id . "'");
        } else {
          $query = $this->db->query("SELECT DISTINCT *, (SELECT keyword FROM " .
DB_PREFIX . "url_alias WHERE query = 'category_id=" . (int)$category_id . "') AS
keyword FROM " . DB_PREFIX . "category WHERE category_id = '" . (int)$category_id .
"'")
        }
        return $query->row;
    }
```

Six-Axe Consultants Groupe Six-Axe

Solution n°2 : base opencart « full » DB2

Autre problème rencontré : sous MySQL, on est autorisé à effectuer un ORDER BY sur une colonne qui n'existe pas dans la liste des colonnes déclarées dans un GROUP BY. DB2 n'accepte pas ce type de syntaxe. Dans la plupart des cas, la colonne concernée était une colonne de tri, aussi pour me simplifier la vie, j'ai simplement supprimé la colonne concernée de l'ORDER BY, l'impact étant mineur (de mon point de vue).

Les requêtes de mise à jour ont nécessité elles aussi quelques ajustements :

```
if ($this->db->getSGBDName() == 'DB2') {
   $tmp_value = str_replace(',', '.', $value);
   $date_mod = $this->db->escape(date('Y-m-d H:i:s')) .'.000000';
   $tmp_params = array($tmp_value, $date_mod, $this->db->escape($currency)) ;
   $this->db->execute("UPDATE " . DB_PREFIX . "currency SET value = ?,
       date_modified = ? WHERE code = ?", $tmp_params);
} else {
   $this->db->execute("UPDATE " . DB_PREFIX . "currency SET value = '" .
      (float)$value . "', date_modified = NOW() WHERE code = '" .
      $this->db->escape($currency) . "'");
}
```

Six-Axe Consultants Groupe Six-Axe

Solution n°2 : base opencart « full » DB2

Après ces différents ajustements, une campagne de tests approfondie est nécessaire pour vérifier et stabiliser l'application.

On rappelle que l'objectif était de rendre l'application complètement opérationnelle sous DB2, sans passer par un moteur intermédiaire comme IBMDB2i. Le contrat est rempli, même si c'est au prix de quelques jours de travail.

Les techniques que j'ai présentées ici visaient à trouver un compromis acceptable entre simplicité et rapidité de mise en œuvre, en essayant de minimiser les impacts, mais sans faire de compromis inacceptable sur les performances ou la sécurité.

Pour améliorer la sécurité justement, il conviendrait de modifier les requêtes pour qu'elles fonctionnent sur le principe des requêtes SQL paramétrées, ce que le wrapper DB2 permet de faire. Encore quelques jours de travail en perspective, mais cela me semble économiquement acceptable au regard des fonctionnalités apportées par opencart.

Six-Axe Consultants Groupe Six-Axe

Solution n°2 : base opencart « full » DB2

Les techniques utilisées pour porter opencart sous DB2 sont reproductibles sur d'autres applications, avec plus ou moins de facilité.

Elles étaient relativement faciles à effectuer sur opencart :
- du fait de la bonne lisibilité de son code PHP et SQL,
- du fait de la taille raisonnable du code source,
- du fait de la présence d'une « couche » base de données fonctionnant sur le principe de « wrappers » assez facilement interchangeables

Les modifications effectuées dans certaines parties de l'application soulèvent la question de la mise à jour de l'application par rapport aux futures versions d'opencart. Je n'ai pas de réponse toute faite à cette question. L'idéal serait que ces modifications soient intégrées au standard, mais il n'est pas évident qu'elles soient compatibles avec les méthodes et les impératifs de l'équipe en charge du projet. Affaire à suivre...

Solutions alternatives

Entre l'approche consistant à utiliser Opencart uniquement en mode MySQL, avec ou sans l'aide du moteur IBMDB2i, et l'approche consistant à l'utiliser en mode « full DB2 », des approches alternatives peuvent être envisagées.

Lors des questions-réponses de fin de session, mon confrère Eric Saglier, expert IBM i et consultant senior chez Six-Axe Consultants, a proposé une approche que je trouve astucieuse et pragmatique, et que j'ai souhaité ajouter à la présentation. L'approche proposée par Eric consistait à utiliser le moteur IBMDB2i pour transférer périodiquement certaines tables MySQL sous DB2, afin de procéder à leur mise à jour via des programmes RPG par exemple.

Pour bien comprendre cette approche, il faut savoir que l'on peut très facilement ouvrir plusieurs connexions bases de données simultanées au sein d'un même script PHP (par exemple une connexion sur MySQL, et une autre sur DB2 for i).

Partant de ce postulat, un scénario comme celui présenté sur la diapo suivante, peut être développé très rapidement, et très facilement.

Six-Axe Consultants Groupe Six-Axe

Solutions alternatives

Soit un script PHP, exécuté toutes les nuits au moyen de la fonction Zend Jobqueue embarquée dans le Zend Server. Ce script effectuera les actions suivantes :

1. Connexion à la base de données MySQL d'Opencart

2. Altération sous MySQL de la table « product » pour la passer sur le moteur IBMDB2i, ce qui a pour effet de la déporter sur DB2 for i

3. Connexion à la base de données DB2 for i

4. Appel d'une procédure stockée DB2 ayant pour action de mettre à jour la table DB2 « product » à partir de données extraites du back office IBM i

5. Altération sous MySQL de la table « product » pour la repasser sur le moteur MyISAM, ce qui a pour effet de la retirer de DB2 for i et de la retransférer sur MySQL

A noter : la procédure stockée pourrait être une procédure « full SQL », ou une procédure externe encapsulant un programme RPGLE ou SQLRPGLE. La taille du script PHP ne devrait pas excéder une vingtaine de lignes, la taille du programme RPG dépendra de la complexité inhérente à votre application IBM i.

Six-Axe Consultants Groupe Six-Axe

En conclusion

On l'a vu au travers de cette présentation, l'association de PHP, de MySQL et du moteur IBMDB2i, constitue un atout important pour enrichir le système d'informations d'une entreprise avec des composants, ou des applications open source.

On a vu également qu'il était possible pour certaines applications PHP, avec plus ou moins de travail, de les faire fonctionner intégralement sous DB2, sans passer par le moteur IBMDB2i.

Les 2 approches sont aussi pertinentes l'une que l'autre, elles peuvent convenir à différentes organisations. L'utilisateur du moteur IBMDB2i peut aussi constituer une solution transitoire, permettant de démarrer rapidement, avec l'idée sous-jacente d'adapter la solution à DB2 à plus ou moins brêve échéance, de manière à pouvoir bénéficier des dernières avancées de DB2, qui sont nombreuses, aussi bien en V6 qu'en V7.

Six-Axe Consultants
Groupe Six-Axe

Merci de votre attention.

A bientôt.