

Debian vs Oracle, même pas peur !

Oracle, cela vous dit quelque chose ? Bien entendu c'est un géant dans le monde des logiciels propriétaires mais le produit certainement le plus connu à travers le monde est son SGBD, un monstre capable de gérer des millions de données avec des performances impressionnantes. Oui mais voilà, Oracle 10g (dernière version du SGBD) bien que compatible Linux n'est certifié que pour certaines distributions, à savoir RedHat et Suse.

Debian semble « condamné » à utiliser mysql et postgres, un sort qui satisfera le plus grand nombre mais qui est très réducteur, notamment si Debian Etch est utilisé en entreprise. Malgré ce que pourraient vous dire les personnes travaillant chez Oracle, ou les sociétés installant le SGBD, Oracle fonctionne sous Debian sans avoir à faire beaucoup plus de manipulations que lors d'une installation sous RedHat.

APT power

Comme pour chaque installation de logiciel, il y a lieu d'installer certains paquets nécessaires. Et Oracle demande beaucoup de choses, notamment des outils de compilation. Sous le profil de l'utilisateur root (/su root/), tapez la commande suivante :

```
apt-get install cpp-3.3 g++ g++-3.3 g++-4.1 gcc gcc-3.3 gcc-3.3-base gcc-4.1 lesstif2 libaiol libbeecrypt6 libc6-dev libdb3 libneon25 librpm4 libssp0 libstdc++5 libstdc++5-3.3-dev libstdc++6-4.1-dev linux-kernel-headers rpm unzip sudo
```

Sachez qu'une interface graphique (GNOME, KDE, XFCE ou autres, c'est sans importance) est nécessaire pour réaliser une partie de l'installation.

En effet, l'application utilise java pour se déployer sur votre serveur, plusieurs fenêtres apparaîtront comportant un certain nombre de questions auxquelles vous aurez à répondre. Les paquets que nous installons avec apt ne comprennent pas l'interface graphique. Votre serveur est maintenant prêt : affrontez l'installation d'Oracle.

La préparation par la simulation

Notre serveur Debian sera préparé pour être reconnu comme un serveur RedHat. Cette simulation permettra de contourner beaucoup de problèmes portant notamment sur les emplacements de certains outils nécessaires à l'application. Créez des liens symboliques pointant sur le véritable emplacement de ces outils. Voici tous les liens à ajouter pour rendre votre système opérationnel :

```
ln -s /usr/bin/awk /bin/awk
ln -s /usr/bin/rpm /bin/rpm
ln -s /usr/bin/basename /bin/basename
ln -s /etc /etc/rc.d
ln -s /lib/libgcc_s.so.1 /lib/libgcc_s.so
```

Créez maintenant un fichier indiquant à RedHat que le système sur lequel il s'installe est un RedHat. Ce fichier servira à Oracle lors de la détection des pré-requis.

```
echo « Red Hat Linux release 3.0 (drupal) » > /etc/redhat-release
```

Une des plus grosse différences entre RedHat et Debian est leur système de gestion des paquets : sous Debian, les paquets sont des *.deb* alors que sous redhat il s'agit de *.rpm*. Etch traite des rpm à certaines conditions, vous avez peut-être remarqué d'ailleurs qu'au début un paquet Debian en rpm a été installé avec apt. Néanmoins, la structure de gestion des rpm doit être initialisée :



Figure 1. Oracle 10g est un produit compatible linux, mais est-il compatible Debian ?



Figure 2. RedHat est certifié Oracle, mais est bien loin d'égaliser les performances de Debian

```
mkdir /var/lib/rpm
rpm --initdb
rpm --rebuilddb
```

La partie simulation est terminée, désormais votre Debian sera reconnu par Oracle comme un système RedHat.

Modification de fichiers systèmes

Cette partie est particulièrement délicate car nous allons paramétrer certains fichiers système. Soyez attentif à ce que vous modifiez sinon vous risquez d'avoir de très mauvaises surprises pouvant nécessiter jusqu'à la réinstallation du système, alors prudence.

D'abord, le module d'authentification PAM : vous devez retirer le commentaire de deux lignes (enlever le « # » en début de ligne) dans deux fichiers : `/etc/pam.d/su` et `/etc/pam.d/login`. La ligne à activer est la suivante (il se peut qu'elle soit déjà active dans un des deux fichiers) :

```
session required pam_  ↵
limits.so
```

Paramétrez le kernel ainsi que les sémaphores, ce paramétrage est in-

diqué dans la documentation d'Oracle, ne modifiez pas les valeurs sinon vous bloquerez l'installation ou le fonctionnement d'Oracle. Éditez le fichier `/etc/sysctl.conf` et ajoutez à la fin ces lignes :

```
kernel.shmall = 2097152
kernel.shmmax = 2147483648
kernel.shmmni = 4096
```

```
kernel.sem = 250 32000 100 128
fs.file-max = 65536
net.ipv4.ip_local_port_range = 1024 65000
net.core.rmem_default = 262144
net.core.rmem_max = 262144
net.core.wmem_default = 262144
net.core.wmem_max = 262144
```

La commande : `sysctl -p /etc/sysctl.conf` fera prendre en compte vos modifications par le système ; l'affichage des valeurs précédemment renseignées confirmera cette prise en compte.

Limitez maintenant les performances de votre système comme stipulé dans la documentation d'Oracle en éditant le fichier `/etc/security/limits.conf` et en ajoutant à la fin les lignes suivantes :

```
* soft nproc 2047
* hard nproc 16384
* soft nofile 1024
* hard nofile 6553
```

Redémarrez le serveur pour une prise en compte de ces dernières modifications. Éditez maintenant fichier `etc/profile` et ajoutez les lignes suivantes :

```
if [ $USER = «oracle» ]; then
    ulimit -u 16384 -n 65536
fi
```

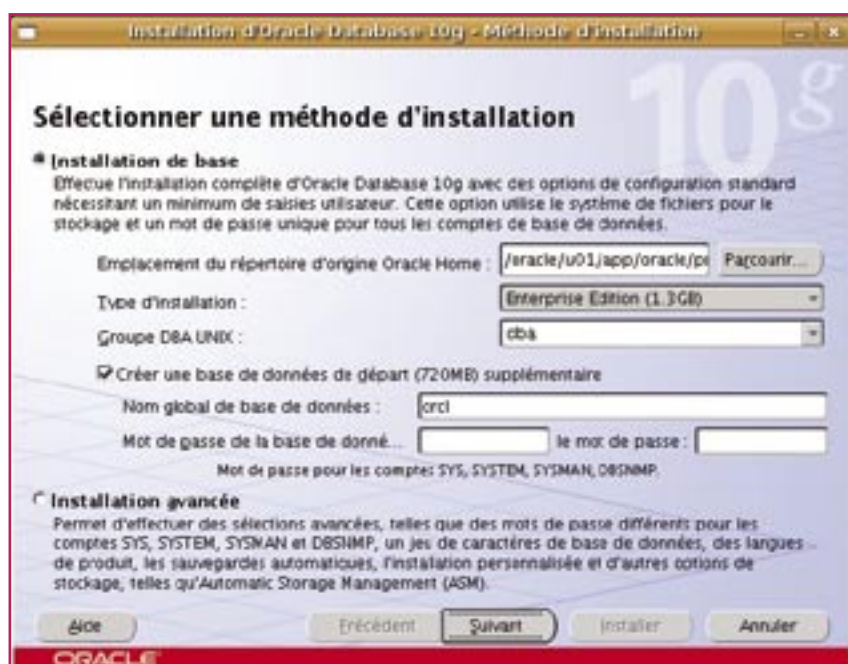


Figure 3. Installation d'Oracle database 10g avec Oracle Installer

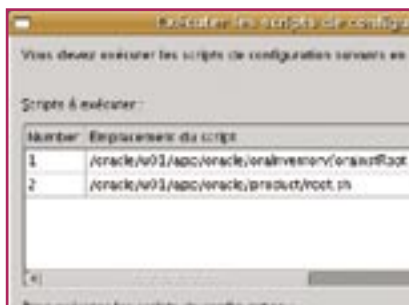


Figure 4. Les scripts administrateur à exécuter en mode console

Le paramétrage du système est terminé, si votre serveur fonctionne toujours malgré le traitement qu'il vient de subir, alors passez à l'étape suivante.

Un joli nid douillet

Il est temps de préparer l'arrivée d'Oracle sur notre système, le SGBD demande un certain nombre de choses en termes d'utilisateur et de groupe :

```
groupadd nobody
usermod -g nobody nobody
groupadd dba
groupadd oinstall
groupadd oper
useradd -d /oracle/home -g oinstall -G dba,oper -s /bin/bash oracle
```

Créez un certain nombre de répertoires pour accompagner ces utilisateurs. Adaptez-les à votre convenance mais Oracle étant une application assez spéciale, placez-la bien en évidence en créant directement sous la racine un répertoire oracle.

```
mkdir -p /oracle/home
mkdir -p /oracle/u01/app/oracle/product
mkdir -p /oracle/u01/app/oracle/oraInventory
mkdir -p /oracle/u02/oradata
mkdir -p /oracle/install
```

Téléchargez enfin Oracle en allant chercher les dernières versions sur <http://www.oracle.com/technology/software/products/database/oracle10g/htdocs/10201linuxsoft.html>. N'oubliez pas que Oracle 10g est un logiciel propriétaire, vous devez respecter la licence d'utilisation.

Deux paquets nous intéressent : la base de données bien entendu (database) et le client qui fournira quelques outils facilitant la gestion. Téléchargez-les et placez-les dans le répertoire `/oracle/install`. Placez-vous dans ce répertoire (`cd /oracle/install`) et décompressez les deux fichiers en adaptant correctement leur nom :

```
unzip 10201_database_linux32.zip
unzip 10201_client_linux32.zip
```

Une fois décompressés, effacez les fichiers zip (`rm *.zip`). Donnez les droits aux répertoires nouvellement créés :

```
chown -R oracle:oinstall /oracle
chmod -R 775 /oracle
chmod -R 755 /oracle/home
```

Pour faire l'environnement qui englobe plusieurs variables, utilisez le fichier `/oracle/home/.bashrc` qui sera lu à chaque connexion de l'utilisateur Oracle : `touch /oracle/home/.bashrc`. Éditez ensuite ce fichier et complétez-le avec les lignes ci-dessous (voir le Listing 1). La variable `ORACLE_SID` est à adapter à votre cas.

Comme d'habitude, affectez les bons droits et surtout le bon propriétaire à ce fichier :

```
chown -R oracle:oinstall /oracle/home/.bashrc
chmod 644 /oracle/home/.bashrc
```

Il ne vous reste plus qu'à attribuer un mot de passe à l'utilisateur Oracle (`passwd oracle`) et la création de l'en-

vironnement Oracle sera terminée. Déconnectez-vous et reconnectez-vous maintenant sous le profil de l'utilisateur Oracle (le fichier `.bashrc` sera pris en compte), l'installateur refuse de démarrer avec l'utilisateur root pour des raisons de sécurité.

Oracle installer

Ouvrez un terminal (pas administrateur !) et tapez la dernière commande suivante : `/oracle/install/database/runInstaller` pour afficher l'interface graphique de l'installateur d'Oracle.

Choisissez une installation de base avec les paramètres suivants : emplacement du répertoire d'origine Oracle Home `/oracle/u01/app/oracle/product`, type d'installation *Entreprise Edition (1.3 GB)*, et groupe `DBA unix dba`. Le nom de la base de donnée doit être identique à ce que vous avez renseigné dans le fichier `.bashrc`. Les mots de passe demandés vous permettront d'administrer votre base de données, alors conservez-les précieusement dans un endroit.

Validez juste les informations des écrans suivants, le dernier avant l'installation résumant ce que vous avez validé, vérifiez mais il ne devrait pas y avoir de problème. Cliquez sur *Installer* et laissez l'installation se dérouler, cela peut être plus ou moins long selon les capacités de votre ordinateur.

Au cours de l'installation, Oracle vous demandera de lancer deux scripts mais avec des droits de « root ». Ouvrez un terminal administrateur ou connectez-vous sous

Listing 1. Éditer et compléter le fichier Oracle

```
PS1='[\u sur \h] \w : '
umask 022
unset USERNAME
export ORACLE_BASE=/oracle/u01/app/oracle
export ORACLE_HOME=/oracle/u01/app/oracle/product
export ORACLE_SID=bdtest #bdtest étant le nom
de votre base de donnée
export ORACLE_TERM=xterm
export PATH=/oracle/u01/app/oracle/product/bin:/usr/sbin:$PATH
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib
export CLASSPATH=$ORACLE_HOME/JRE:$ORACLE_HOME/jlib:$ORACLE_HOME/
rdbms/jlib
export ORATAB=/etc/oratab
export ORACLE_HOME_LISTENER=$ORACLE_BASE
```



le profil root avec la commande `su root`. Ensuite, lancez les deux scripts indiqués (normalement `/oracle/u01/app/oracle/orainventory/orainstRoot.sh` et `/oracle/u01/app/oracle/product/root.sh`).

Le dernier message qui devrait apparaître est une adresse web pointant vers votre serveur à partir d'un port inhabituel (exemple : `http://votreserveur.com:1158/em`). En indiquant cette adresse dans un navigateur, vous administrerez votre serveur. Pas de message d'erreur, tout s'est bien passé, l'installation oracle database est terminée !

Starting Oracle

Vous imaginez bien que cela ne pouvait pas se finir comme ça, cela serait trop simple, il faut paramétrer le démarrage de votre base de données et créer un script de démarrage du SGBD. Ouvrez un terminal administrateur ou connectez-vous en tant que root puis éditez le fichier `//etc/oratab/` et modifiez la ligne concernant votre base de données (ici la base de données créée s'appelle `bdtest`) :

```
bdtest:/oracle/u01/app/oracle/  ↵
  product:N          devient     ↵
  bdtest:/oracle/u01/app/ ↵
  oracle/product:Y
```

Dans un second temps, réalisez un script de démarrage qui permettra de démarrer et d'arrêter proprement Oracle à chaque fois reboot du serveur : `touch /etc/init.d/oracle`. Éditez ce fichier et complétez-le, un exemple vous est fourni à cette adresse : <http://albatordb.blogspot.com/2007/04/oracle.html>. Rendez-le fonctionnel en lui attribuant les bons droits et surtout en l'intégrant au système de démarrage (`rc.d`). Pour cela trois petites commandes :

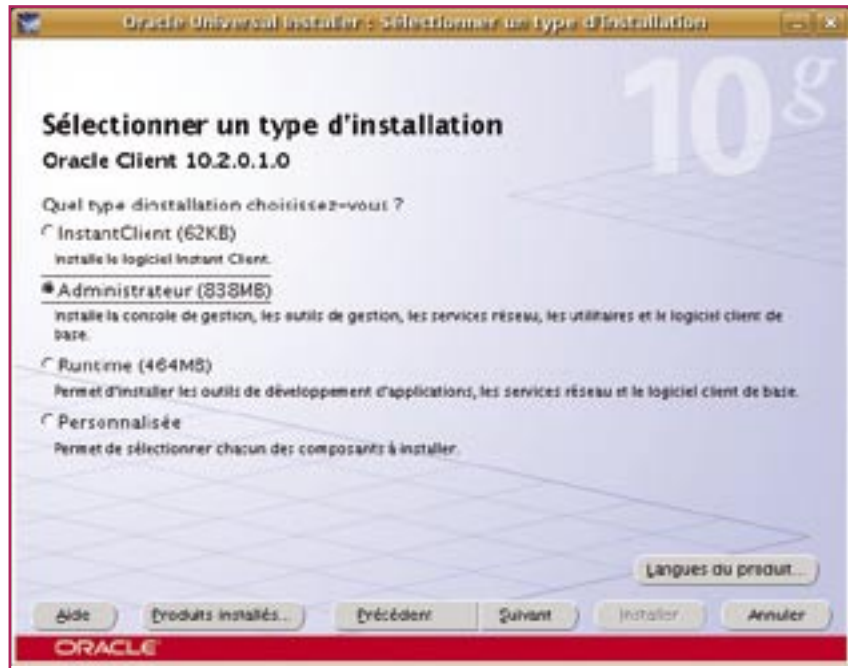


Figure 5. Le type d'installation « administrateur » du client Oracle est préférable

```
chown root:root /etc/init.d/ ↵
  oracle
chmod 755 /etc/init.d/oracle
update-rc.d oracle start 90 2 3 4 ↵
  5 . stop 10 0 1 6 .
```

Cette fois-ci, l'installation d'Oracle est effectivement terminée.

Un bon client

Quant au client oracle, notez qu'il n'est pas indispensable sur le serveur, il peut être installé sur un poste distant mais il est parfois pratique d'en avoir un en local pour faire un test par exemple.

Pour installer le client oracle, le principe est à peu près le même que pour la base de donnée, il y a un installateur graphique. Connectez-vous sous le profil oracle et tapez la commande : `/oracle/install/client/runInstaller`. Validez les écrans sauf celui relatif au type d'installation : sélectionnez *Administrateur*.

Une fois installé, l'outil le plus utile est l'Enterprise Manager que vous utilisez en lançant en tant qu'utilisateur Oracle, cette commande : `oemapp console`.

Connectez-vous à votre base de données en tant qu'utilisateur « sys » avec le mot de passe renseigné lors de l'installation de oracle database. Veillez à vous connecter en tant que « sysdba » pour administrer totalement votre base de données car cela ne sera pas possible sous un autre profil utilisateur.

Voilà, nous avons atteint notre objectif, est-ce toujours impossible d'installer Oracle sur Debian ?

Vous pouvez abandonner votre serveur RedHat que vous payez bien trop cher. De plus, la stabilité de Debian est bien meilleure et les possibilités bien plus grandes. ☺