

Configuration et utilisation de dbmon

Gilles Lamiral <mailto:lamiral@linux-france.org> \$Date: 2001/12/30 22:51:38 \$ \$Revision: 1.18 \$

Test, configuration et utilisation de DBMON.

Contents

Vous trouverez la dernière version de ce document à l'adresse :

1 Introduction

DBMON est une suite logicielle. Le script `dbmon` contacte un ou plusieurs serveurs MON, collecte des données établies par les moniteurs (`*.monitor`), crée et met à jour des bases de données (`*.rrd`) ou une base mysql, génère plusieurs index (`dbmon-*.html`) permettant de visualiser sur un butineur les données sur une période quelconque (`rrdview.cgi`), s'endort un peu et recommence.

2 Prérequis

Vous devez avoir lancé le ou les serveur(s) MON avant de lancer `dbmon`, sinon `dbmon` attendra 5 minutes avant de réessayer de recontacter le(s) serveur(s) MON.

L'ordonnanceur MON classique n'est pas très précis quand il lance les moniteurs. Il prend toujours un peu de retard. La version suivante corrige ce défaut:

```
/apps/dbmon-0.08/bin/mon
```

3 Comportement par défaut

Pour connaître le comportement par défaut de `dbmon`, il suffit de l'invoquer avec l'option `"-help"` et de lire la section "DEFAULT VALUES":

```
/apps/dbmon-0.08/bin/dbmon -h
```

4 Tests

La totale se fait par les commandes suivantes:

```
cd /apps/dbmon-0.08/  
gmake test
```

"gmake test" est équivalent à la commande:

```
gmake testu && gmake testm && gmake testf
# testu : tests unitaires
# testm : tests des moniteurs
# testf : tests fonctionnels (mon + mysql + rrdtool)
```

4.1 Tests unitaires

La totale des tests unitaires se fait avec les commandes suivantes:

```
cd /apps/dbmon-0.08/
gmake testu
```

Si vous avez des problème avec gmake, laissez tomber les tests.

4.2 Tests des moniteurs

La totale des tests des moniteurs se fait avec les commandes suivantes:

```
cd /apps/dbmon-0.08/
gmake testm
```

Il y a cependant de fortes chances que certains tests des moniteurs échouent car ils nécessitent la présence de serveurs spécifiques (LDAP, DNS etc.)

4.3 Tests fonctionnels (et apprentissage)

Pour vérifier que dbmon fonctionne bien, lancez les tests fonctionnels:

```
cd /apps/dbmon-0.08/
gmake testf
```

et pour vous familiariser avec lui, je vous propose de travailler dans le répertoire /tmp/.

```
rm -rf /tmp/dbmon/test/
mkdir -p /tmp/dbmon/test/
cd /tmp/dbmon/test/
/apps/dbmon-0.08/bin/dbmon
```

Vous constatez que dbmon a besoin du répertoire ./rrdbases/

```
mkdir ./rrdbases
/apps/dbmon-0.08/bin/dbmon -Dgen
```

Ceci produit la sortie suivante :

```
can sleep for 150 secondes
sleeping for 139 secondes
```

Pour arrêter dbmon, il suffit de taper Ctrl-c. Par défaut, dbmon n'est bavard que si les choses se passent mal (il existe des modes debug, confer l'option `-h`).

Si le serveur MON n'a pas pu être contacté, il vous le dit. Si tout s'est bien passé, dbmon a créé toutes les bases rrd dans le répertoire `rrdbases/` ainsi que le fichier `dbmon.html` dans le répertoire courant.

Pour vérifier que la création des bases rrd s'est bien passée, vous lancez la commande suivante :

```
rrdtool last rrdbases/localhost~2583.rrd
```

Ce qui produit le temps epoch de la mise à jour de la base:

```
1008758400
```

Si vous n'avez pas le logiciel rrdtool, installez le, ou passez à la suite (qui permettra de vérifier aussi que tout fonctionne).

Vous pouvez visualiser le fichier avec un navigateur quelconque. Les liens de la page renvoie vers un script cgi nommé `rrdview.cgi` et dont le chemin complet est :

```
/apps/dbmon-0.08/cgi-bin/rrdview.cgi
```

Vérifions qu'il fonctionne:

```
echo |/apps/dbmon-0.08/cgi-bin/rrdview.cgi 2>/dev/null
```

La précédente commande doit générer une page html.

Nous allons configurer un serveur web définissant ce script (`rrdview.cgi`) comme un script CGI (Common Gateway Interface) car vous devrez le faire plus tard pour vos utilisateurs.

Pour un Apache classique (`/usr/local/apache/conf/httpd.conf` ou `srm.conf`), vous copiez-coller la configuration suivante en INCLUANT les commentaires.

Pensez à SAUVEGARDER votre fichier de configuration apache sous un autre nom AVANT de le modifier.

Vous trouverez une copie de cette configuration dans le fichier `/apps/dbmon-0.08/etc/apache.conf`

```
#####
# Debut de configuration pour DBMON #
#####
```

```
Alias /dbmon "/apps/dbmon-0.08/"
```

```
<Location /dbmon>
    Options Indexes FollowSymLinks MultiViews
</Location>
```

```
<Location /dbmon/cgi-bin>
    AddHandler cgi-script cgi
    Options Indexes ExecCGI
</Location>
```

```
#####
# Fin de configuration pour DBMON #
#####
```

Pour en savoir plus sur les "Handler" Apache, consultez la section *handler* de son *manuel* [<http://httpd.apache.org/docs/>](http://httpd.apache.org/docs/).

Ensuite relancez le serveur web:

```
/usr/local/apache/bin/apachectl configtest
/usr/local/apache/bin/apachectl restart
```

ou bien (pour les vieilles versions)

```
/usr/local/apache/sbin/apachectl configtest
/usr/local/apache/sbin/apachectl restart
```

Utilisez votre butineur pour visualiser l'URL

Dans votre butineur graphique, vous devez obtenir une page contenant un tableau avec une image contenant le message "Error !".

Ce n'est pas grave, c'est même normal. En haut de la page, vous observez un champ textuel contenant la valeur `foo.rrd`. C'est une valeur par défaut, qui n'est là que pour vous rappeler que ce champ doit contenir le nom d'un fichier de type "base de données rrdtool". Remplacez cette valeur avec la suivante:

```
/tmp/dbmon/test/rrdbases/localhost~2583.rrd
```

Cliquez ensuite sur l'image "Error !". L'image est en fait le bouton de l'interface. À chaque que vous utilisez cette interface, la prise en compte des nouveaux paramètres est lancée par cet unique bouton-image.

Vous devez obtenir une image en 600x150 pixels. Si aucun message d'erreur n'apparaît, tout fonctionne bien, sinon reprenez ce chapitre au début et vérifiez bien chaque étape.

Vous pouvez effacer le répertoire de test:

```
cd
rm -rf /tmp/dbmon/
```

Pour tester complètement la configuration du serveur web ainsi que dbmon:

```
/apps/dbmon-0.08/tests/etc/dbmon-test.init start
```

Ce script ne vous rend pas la main mais un Ctrl-c fera l'affaire après l'apparition du " sleeping for xxx secondes".

Visualiser alors, dans un butineur, l'url
:

Tous les liens doivent fonctionner sauf le lien RRDTOOL si vous n'avez pas accès à internet.

5 Script de démarrage

Le script dbmon peut fonctionner en mode démon (fork, détachement du terminal, rattachement au processus patriarche init). Pour faciliter le démarrage, l'arrêt ou l'éveil de dbmon, un script, nommé `/apps/dbmon-0.08/etc/dbmon.init`, conforme au standard UNIX system V, est là pour vous aider.

Ce script fonctionne parfaitement sans modification et correspond à la configuration préconisée dans la présente documentation. Néanmoins, Vous pouvez le copier, éditer la copie et changer les valeurs. Vous remarquerez que les bases rrd et le fichier index html sont définis comme étant dans le répertoire `/data/dbmon/` (qui est créé automatiquement par le script `dbmon.init`).

5.1 Usage de dbmon.init

Usage :

```
/apps/dbmon-0.08/etc/dbmon.init
```

Lancement :

```
/apps/dbmon-0.08/etc/dbmon.init start
```

Arrêt :

```
/apps/dbmon-0.08/etc/dbmon.init stop
```

Relance :

```
/apps/dbmon-0.08/etc/dbmon.init restart
```

Réveil :

```
/apps/dbmon-0.08/etc/dbmon.init reload
```

5.2 Plusieurs serveurs MON

Le script `dbmon.init` est adapté pour contacter plusieurs serveurs MON (jusqu'à 10). Il vous suffit de remplacer les variables dans le (ou votre) fichier `dbmon.init` :

```
SERVER='localhost:2583'    # MON server:port
SERVER_01='osm1:2583'     # MON server:port
SERVER_02='osm2:2583'     # etc.
SERVER_03='osm3:2583'
SERVER_04='osm4:2583'
SERVER_05=''
SERVER_06=''
SERVER_07=''
SERVER_08=''
SERVER_09=''
```

5.3 Lancement au boot

Le paquet d'installation de `dbmon` effectue un lien symbolique de `/apps/dbmon-0.08/etc/dbmon.init` dans le répertoire `/etc/init.d/`.

Ne pas oublier de mettre proprement à jour les répertoires "rcN.d" (notamment `/etc/rc0.d/` pour l'extinction du processus)

De préférence, `dbmon` doit être lancé après MON, `dbmon` doit être arrêté avant MON.

5.4 Essais

Je vous invite à lancer chacune des commandes suivantes:

```
/apps/dbmon-0.08/etc/dbmon.init
/apps/dbmon-0.08/etc/dbmon.init stop
/apps/dbmon-0.08/etc/dbmon.init start
/apps/dbmon-0.08/etc/dbmon.init stop
/apps/dbmon-0.08/etc/dbmon.init restart
/apps/dbmon-0.08/etc/dbmon.init reload
/apps/dbmon-0.08/etc/dbmon.init stop
```

Visualiser avec votre navigateur la page index localisée à l'URL
<<http://localhost/dbmon/html/dbmon.html>>

Remarque: le fichier `dbmon.init` contient les définitions:

```
SERVER='localhost:2583'
TOPFILE=/apps/dbmon-0.08/html/dbmon.html
```

`TOPFILE` définit la page d'index générée par DBMON.

Vu que `http://localhost/dbmon/` est un alias pour le répertoire `/apps/dbmon-0.08/`, je vous laisse comprendre la correspondance de la variable `TOPFILE` avec l'URL `http://localhost/dbmon/html/dbmon.html`

5.5 Eviter l'usage de MySQL

Pour éviter l'usage de MySQL, il suffit de commenter la ligne définissant la variable `SQL` dans le fichier `etc/dbmon.init` en ajoutant un dièse devant sa définition.

```
#SQL="--sql"
```

6 Pour de vrai

Vous pouvez utiliser les anciens moniteurs avec `dbmon`, comme cela a été fait dans le chapitre précédent. Cependant, vous n'aurez que le comportement global de l'ensemble des serveurs surveillés, par service. Pour bénéficier du cas par cas, des temps de réponse par serveur, il faut utiliser de nouveaux moniteurs. Ils sont tous situés dans le répertoire `/apps/dbmon-0.08/monitors/`

Ils portent tous un nom différent des anciens noms pour éviter toute ambiguïté. Je vous invite à exécuter la commande suivante:

```
ls -l /apps/dbmon-0.08/monitors/
```

Les moniteurs portent l'extension `".monitor"`. Exemple: `dns.rrdrt.monitor`

Puisque de nouveaux moniteurs sont à utiliser, il faut modifier le fichier de configuration du serveur `MON`.

Ne JAMAIS MODIFIER un fichier de configuration de serveur `mon` sans en faire une SAUVEGARDE.

Ne JAMAIS RENSEIGNER un fichier de configuration de serveur `mon` SANS tester au préalable l'appel des MONITEURS EN LIGNE DE COMMANDE.

Pourquoi ? Pour éviter de compliquer la mise en place.

Ajoutons d'abord le répertoire des nouveaux moniteurs. La variable à renseigner est `mondir`:

```
# mondir      = mon.d
mondir        = mon.d:/apps/dbmon-0.08/monitors/
```

La variable `mondir` peut contenir une liste de répertoires quelconques séparés par des deux-points (comme la classique variable `$PATH`). Je rappelle que toute ligne commençant par un dièse est interprétée comme un commentaire (En fait, elle n'est PAS interprétée, elle est tout simplement ignorée). Dans l'exemple précédent, l'ancienne définition de `mondir` a été commentée.

Ensuite, il faut changer ou bien ajouter les moniteurs appelés pour chaque service. Exemple :

```
watch dns-root-servers
  service dns
    interval 15m
    monitor dns.rrdrt.monitor --rrdrt
  service ping
    interval 15m
    monitor fpingv.rrdrt.monitor --rrdrt
```

```
watch Partitions
  service DiskSpace
  interval 5m
  monitor freespace.rrdfs.monitor --rrd
```

Notez l'ajout des options `--rrdrt` ou `--rrd`, nécessaires pour obtenir le traitement cas par cas des cibles.

Relancer le serveur MON.

Pensez à LIRE le document intitulé "monitors" pour parfaire cette mise À jour.

N'ESSAYER PAS de faire des appels TROP COMPLIQUÉS tout de suite, faites les modifications EN DOUCEUR, de façon INCRÉMENTALE.

J'insiste. Ne JAMAIS RENSEIGNER un fichier de configuration de serveur mon SANS tester au préalable l'appel des moniteurs en LIGNE DE COMMANDE. Vous gagnerez ainsi beaucoup de temps et de confiance.

Rappel: la ligne de commande est la même que celle du fichier de configuration à laquelle il faut rajouter les serveurs testés. Par exemple, voici un extrait de configuration:

```
...
hostgroup LOCAL
  localhost
  127.0.0.1
...
watch LOCAL
  service DNS
  interval 15m
  monitor dns.rrdrt.monitor --rrdrt
```

La ligne de commande correspondante est :

```
/apps/dbmon-0.08/monitors/dns.rrdrt.monitor --rrdrt localhost 127.0.0.1
```

Le code retour doit valoir 0

```
echo $?      # en shell bourne et assimilés (ksh bash tcsh ash etc.)
echo $status # en C shell et assimilés (tcsh etc.)
```

Voici l'algorithme de mise à jour:

0a) lancer le serveur mon

```
/apps/dbmon-0.08/etc/mon.init start
```

0b) lancer dbmon

```
/apps/dbmon-0.08/etc/dbmon.init start
```

1) réveiller le serveur mon

```
kill -HUP PID  
ou bien /apps/dbmon-0.08/etc/mon.init reload
```

2) réveiller dbmon

```
kill -HUP PID  
ou bien /apps/dbmon-0.08/etc/dbmon.init reload
```

3) visualiser le fichier index avec votre butineur

```
http://localhost/dbmon/html/localhost:2583.html
```

(Ne vous sentez pas obligés de relancer mozilla à chaque fois; un rechargement de la page suffira amplement (Alt-r)

4) tester les nouveautés en ligne de commande

5) modifier la configuration du serveur mon

```
vi /apps/dbmon-0.08/etc/mon.cf
```

(Ne vous sentez pas obligés de quitter vi à chaque fois une sauvegarde :w suffira amplement)

6) Goto 1) jusqu'à pleine satisfaction.

Voici un moyen de connaître le PID des processus dbmon:

```
ps -edf |grep bin/dbmon
```

Un exemple complet de fichier de configuration de serveur MON utilisant les nouveaux moniteurs est:

```
/apps/dbmon-0.08/etc/mon.cf
```

Je vous souhaite une bonne supervision.

7 Problèmes connus

7.1 quotewords core dump

Si un groupe contient beaucoup de cibles, le dialogue client-serveur peut échouer à cause d'un bug du module perl `Text::ParseWords`. Aucun remède miracle n'existe à ce jour. Vous devez diviser le groupe en plusieurs parties.

8 Liens utiles

MON home page <<http://www.kernel.org/software/mon/>>

MON mailing list <<http://www.kernel.org/software/mon/list.html>> **Inscrivez-vous !**