



Optimisation et solution des problèmes de Pandora FMS



<https://pandorafms.com/manual/!775/>

parent link:

https://pandorafms.com/manual/!775/fr/documentation/pandorafms/complex_environments_and_optimization/08_optimization

2022/03/18 21:03

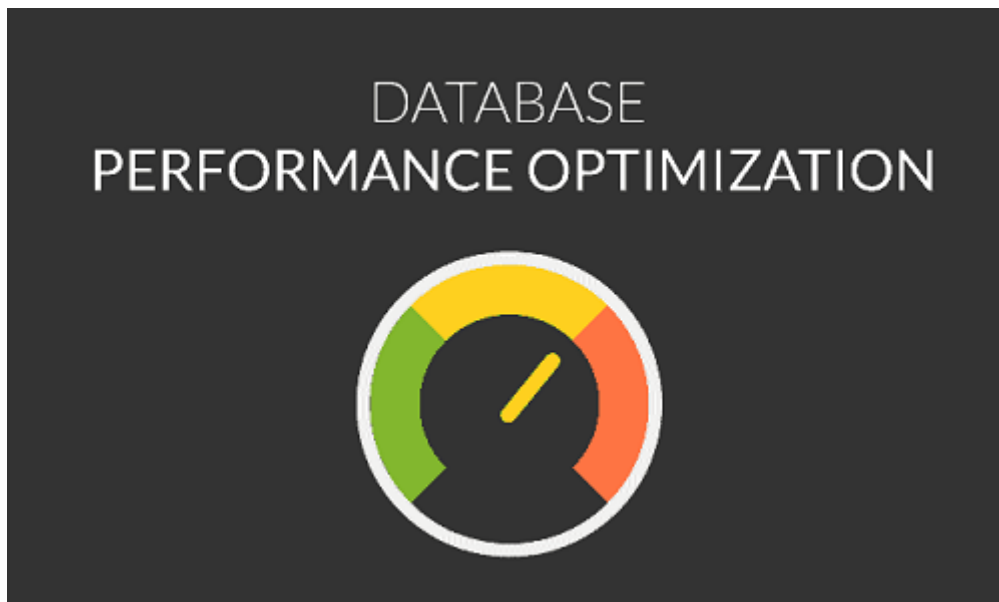


Optimisation et solution des problèmes de Pandora FMS

Optimisation et solution des problèmes de Pandora FMS

Introduction

Le serveur de Pandora FMS est capable de surveiller environ 2000 appareils (entre 5000 et 80 000 modules, [selon le matériel disponible](#)), ce qui nécessite également une configuration fine de la base de données.



De plus, dans ce chapitre, certaines techniques de détection et de dépannage des problèmes d'installation de Pandora FMS sont expliquées.

Optimisation de MySQL pour le Version Enterprise

Pour en savoir plus sur “Data backup and recovery in Pandora FMS”, [veuillez suivre ce lien](#).

Conseils généraux

- Sauf indication contraire, l'ensemble de cette rubrique fait référence à la version 5.7 de MySQL.

- Voir aussi « [Mise à niveau de MySQL 5.7 à MySQL 819_mysql_8](#) ».

Pour travailler avec des tableaux de plus de 2 GiB, il est nécessaire de suivre certaines directives :

- MySQL® recommande d'utiliser un système 64 bits. Les systèmes 32 bits pourraient connaître de sérieux problèmes à [partir de 2038](#).
- Plus la mémoire vive et le processeur sont importants, meilleures sont les performances. D'après notre expérience, la RAM est plus importante que l'UCT. Le minimum pour un système de niveau entreprise sera de 4 GiB. Un bon choix pour un grand système est 16 GiB. N'oubliez pas qu'une plus grande quantité de RAM peut accélérer les mises à jour des clés en conservant les pages clés les plus utilisées dans la RAM.
- Il est bon de pouvoir mettre le système hors service en cas de panne. Pour les systèmes où la base de données se trouve sur un autre serveur de base de données dédié, utilisez l'Ethernet Gigabit, de préférence avec de la fibre optique plutôt que du cuivre. La latence est aussi importante que les performances.
- L'optimisation du disque est très importante pour les très grandes bases de données : les bases de données et les tables doivent être réparties sur différents disques. Dans MySQL®, vous pouvez utiliser des liens symboliques pour cela. Utilisez des disques différents pour le système et la base de données.

L'utilisation de [disques SSD](#) est recommandée en raison de leur vitesse et de la latence améliorée du système.

- Si possible utilisez:

```
--skip-locking
```

- Ceci désactivera le verrouillage externe et fournira de meilleures performances (activé par défaut sur certains systèmes).
- Si le client et le serveur MySQL® se trouvent sur la même machine, utilisez des sockets au lieu de connexions TCP/IP lors de la connexion à MySQL® (cela peut donner une amélioration de 7,5 %). Vous pouvez le faire sans spécifier le nom d'hôte ou localhost lors de la connexion au serveur MySQL®. Désactivez la connexion binaire et la réplication si vous n'avez qu'un seul serveur hôte MySQL®.
- L'utilisation de versions récentes de MySQL® (5.5 ou plus) par rapport à des versions plus anciennes de MySQL® (5.0.x) peut offrir jusqu'à 20 % de différence en termes de performances.
- Il est recommandé d'utiliser la version modifiée de MySQL® ([Percona Server for MySQL®](#)), qui offre de meilleures performances. Par défaut, les plugins programmés sont pour Percona®.

Veuillez noter que les performances sont grandement affectées par les points suivants :

- N'utilisez les *logs* binaires que si vous utilisez une configuration MySQL® avec réplication.
- N'utilisez pas les *logs* de traçabilité des requêtes ou les journaux de requêtes lentes.

Outils de configuration automatique

Il existe de nombreux outils pour optimiser la configuration du serveur MySQL.

MySQL Tuning Primer, de Matthew Montgomery, est un outil en ligne de commande pour vérifier la configuration d'un serveur et voir ses performances et les améliorations possibles, en donnant quelques conseils et suggestions pour l'améliorer. C'est sur <https://bugs.launchpad.net/mysql-tuning-primer>.

Désactiver la réplication binaire

Si vous avez configuré un [système Pandora FMS HA](#), la réplication binaire est nécessaire. Cette recommandation n'est valable que si vous disposez d'un seul serveur Pandora FMS.

Par défaut, il est activé dans la plupart des distributions GNU/Linux. Pour le désactiver, éditez le fichier `my.cnf` généralement dans `/etc/my.cnf` et commentez les lignes suivantes :

```
# log-bin=mysql-bin
# binlog_format=mixed
```

Vous devez commenter les deux lignes, puis redémarrer le serveur MySQL®.

Performance d'accès au disque

Pour en savoir plus sur "Data backup and recovery in Pandora FMS", [veuillez suivre ce lien](#).

Avec d'autres paramètres clés, il y en a trois qui sont particulièrement pertinents quand il s'agit de la performance d'accès au disque, qui est souvent le goulot d'étranglement en ce qui concerne MySQL.

`innodb_log_file_size`

```
innodb_log_file_size = 64M
```

Cette valeur est fixée par défaut et peut être supérieure (même 512M) sans préjudice, sauf pour la récupération en cas de problème et d'occupation supérieure du disque. La valeur par défaut de MySQL est de 5M, ce qui est très faible pour les environnements de production avec des volumes de transactions élevés. Pour modifier cette valeur avec un système déjà en fonctionnement :

1. D'abord faites un DUMP complet et supprimer les fichiers d'index binaires d'Innodb.
2. Effacez les fichiers d'index binaires d'Innodb (généralement dans `/var/lib/mysql/ib*`).
3. Changez `my.cnf`, avec la valeur choisie.

4. Redémarrez MySQL.
5. Chargez le dump SQL.

Puisque le processus est le même que pour activer le jeton `innodb_file_per_table` (décrit un peu plus bas, nous recommandons de faire tout le processus simultanément) :

`innodb_io_capacity`

```
innodb_io_capacity = 100
```

Par défaut ce paramètre a la valeur 100, mais nous devons connaître au préalable les IOPS du disque système. Vous pouvez savoir exactement en cherchant IOPS et le modèle exact du disque dur (obtenu via `smartctl`), où les valeurs recommandées sont : 7500RPM → 100 IOPS, 15000 RPM → 190 IOPS, SSD → 1500 IOPS

Pour installer `smartctl`, vous devez demander le paquetage complet de `smartmontools` ; par exemple :

```
yum install smartmontools,
```

```
apt install smartmontools, etc.
```

`innodb_file_per_table`

Utiliser un espace de table pour chaque table ([extrait du manuel MySQL 5.0 en espagnol](#))

Dans MySQL 5.0, vous pouvez stocker chaque table InnoDB et ses index dans son propre fichier. Cette fonctionnalité est appelée “ *multiple tablespaces* ” car, en fait, chaque table a son propre espace de table.

L'utilisation de plusieurs espaces de table peut être bénéfique pour les utilisateurs qui veulent déplacer des tables spécifiques vers des disques physiques séparés ou qui veulent restaurer des sauvegardes de tables sans interrompre l'utilisation des autres tables InnoDB.

Plusieurs tablespaces peuvent être activés en ajoutant cette ligne à la section `mysqld` de `my.cnf` :

```
[mysqld]
innodb_file_per_table
```

Après le redémarrage du serveur, InnoDB stockera chaque nouvelle table créée dans son propre fichier `nom_de_la_table.ibd` dans le répertoire de la base de données à laquelle la table appartient. Ceci est similaire à ce que fait le moteur de stockage MyISAM, mais MyISAM divise la table en un fichier de données `tbl_name.MYD` et un fichier d'index `tbl_name.MYI`.

Pour InnoDB, les données et les index sont stockés ensemble dans le fichier `.ibd`. Le fichier `tbl_name.frm` est toujours créé comme d'habitude. Si la ligne `innodb_file_per_table` est supprimée de `my.cnf` et que le serveur est redémarré (voir instructions précédentes pour `innodb_log_file_size`), InnoDB créera à nouveau les tables à l'intérieur des fichiers de l'espace de table partagé.

`innodb_file_per_table` n'affecte que la création des tables. Si vous démarrez le serveur avec cette option, les nouvelles tables seront créées à l'aide de fichiers `.ibd`, mais vous pouvez toujours avoir accès aux tables existantes dans l'espace de tables partagé. Si vous supprimez cette option, les nouvelles tables seront créées dans l'espace partagé, mais il sera toujours possible d'avoir accès aux tables créées dans plusieurs espaces de tables.

Évitant l'écriture à chaque transaction

Par défaut, MySQL définit `autocommit = 1` pour chaque connexion. Ce n'est pas mauvais pour MyISAM, car ce que l'on écrit n'est pas garanti sur disque, mais pour InnoDB cela signifie que chaque insert / update / delete dans une table InnoDB entraînera une écriture sur disque (flush).

Qu'est-ce qui ne va pas avec l'écriture sur disque ? Rien du tout. Ils s'assurent que toute compromission garantit que les données sont présentes lorsque la base de données est redémarrée après un crash. Le problème est que le fonctionnement de la base de données est limité par la vitesse physique du disque.

Comme le disque doit écrire les données sur un disque avant la confirmation de l'écriture, cela prend du temps. En supposant même un temps de recherche moyen de 9ms pour l'écriture sur disque, nous sommes limités à environ 67 commits/sec¹, c'est très lent. Et pendant que le disque est occupé à essayer de faire écrire le secteur, il ne fait aucune lecture.

InnoDB peut éviter une partie de cette limitation en écrivant ensemble, mais la limitation existe toujours. On peut l'empêcher d'écrire à la fin de chaque transaction en le faisant mettre dans un système d'écriture " automatique ", qui écrit environ toutes les secondes. En cas d'échec, nous pouvons perdre les données de la dernière seconde, ce qui est plus que supportable si nous essayons de gagner en efficacité. Pour ce faire, nous utiliserons le jeton de configuration suivant : `innodb_flush_log_at_trx_commit = 0` La valeur par défaut est cette valeur dans la configuration.

Plus grande taille de KeyBuffer

En fonction de la RAM totale du système, c'est un paramètre global très important qui accélère les DELETE et INSERT.

```
key_buffer_size = 4M
```

C'est la valeur par défaut dans la configuration.

Autres paramètres importants

Il y a plusieurs tampons qui par défaut sur certaines distributions sont vides. La modification de ces paramètres peut donner de bien meilleures performances que celles par défaut. Il est important de s'assurer que ces jetons existent dans le fichier de configuration de MySQL.

```
query_cache_size = 64M
query_cache_limit = 2M
join_buffer_size = 4M
```

Pour MySQL version 8 et versions ultérieures, l'équipe de développement de MySQL ont fini le support pour *query cache*, si vous souhaitez obtenir plus d'informations visitez le lien suivant

[:https://dev.mysql.com/blog-archive/mysql-8-0-retiring-support-for-the-query-cache/](https://dev.mysql.com/blog-archive/mysql-8-0-retiring-support-for-the-query-cache/)

Amélioration de la concurrence dans InnoDB

Il y a un paramètre qui pourrait affecter beaucoup la performance du serveur MySQL avec Pandora. Ce paramètre est `innodb_thread_concurrency`. Ce paramètre est utilisé pour spécifier le nombre de *files concurrents* que MySQL peut exécuter. Une mauvaise configuration de ce paramètre pourrait faire qu'il soit plus lent que par défaut, il est donc particulièrement important de faire attention à plusieurs paramètres :

- Version MySQL. Dans les différentes versions de MySQL, ce paramètre se comporte TRES différemment.
- Nombre de processeurs réels (physiques).

Vous pouvez lire ici la [documentation officielle de MySQL](#).

La valeur recommandée est le nombre de CPU (physiques) multiplié par 2 plus le nombre de disques où se trouve InnoDB.

Dans les dernières versions de MySQL (> 5.0.21), la valeur par défaut est 8. Une valeur de 0 signifierait "ouvrir autant de fils que possible". Par conséquent, en cas de doute, il peut être utilisé :


```
innodb_thread_concurrency = 0
```

Différentes personnes (“ [Variable's Day Out #5: innodb_thread_concurrency](#) ”, “ [Do we still need innodb_thread_concurrency?](#) ”) ont fait des tests, et ont détecté des problèmes de performance dans des serveurs avec beaucoup de CPU physiques en utilisant un nombre très élevé, avec des versions relativement anciennes de MySQL (nous parlons de 2008).

Fragmentation de MySQL

Tout comme les systèmes de fichiers, les bases de données se fragmentent également, ce qui entraîne une perte de performance de l'ensemble du système. Dans un système à haute performance, tel que le Pandora FMS, il est essentiel que la santé de la base de données n'affecte pas le fonctionnement du système. Dans les systèmes surchargés, la base de données peut être bloquée à la fin, ce qui entraîne une chute de l'ensemble du système.

Une bonne configuration de MySQL pourrait rendre le travail de Pandora FMS plus rapide. Si vous avez des problèmes de performance, ce sera probablement parce que vous n'avez pas configuré MySQL correctement ou à cause d'un problème lié à la base de données.

Vérification du fichier my.cnf

Commencez par vérifier le fichier `my.cnf` et sa configuration de base pour MySQL. Ce fichier de configuration est écrit au [format INI](#) et son emplacement peut être déterminé avec la commande suivante :

```
mysqld --help --verbose | more
```

```
# euclides root ~ mysql -h localhost -u root -p --help --verbose | more
mysql Ver 5.7.35-38 for Linux on x86_64 (Percona Server (GPL), Release 38, Revision 3692a61)
Copyright (c) 2009-2021 Percona LLC and/or its affiliates
Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Starts the MySQL database server.

Usage: mysqld [OPTIONS]

Default options are read from the following files in the given order:
/etc/my.cnf /etc/mysql/my.cnf /usr/etc/my.cnf ~/.my.cnf
The following groups are read: mysql server mysql-5.7
The following options may be given as the first argument:
--print-defaults          Print the program argument list and exit.
--no-defaults            Don't read default options from any option file,
                          except for login file.
```

La configuration de votre `my.cnf` devrait être similaire à celle-ci (4 Go de RAM et en utilisant un

matériel de configuration moyenne). Vérifiez que vous avez bien saisi tous ces paramètres dans la section [mysqld] :

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
user=mysql
character-set-server=utf8
skip-character-set-client-handshake

max_allowed_packet = 64M
innodb_buffer_pool_size = 800M
innodb_lock_wait_timeout = 90
innodb_file_per_table
innodb_flush_log_at_trx_commit = 0
innodb_flush_method = O_DIRECT
innodb_log_file_size = 64M
innodb_log_buffer_size = 16M
innodb_io_capacity = 100
thread_cache_size = 8
thread_stack      = 256K
max_connections = 100

key_buffer_size=4M
read_buffer_size=128K
read_rnd_buffer_size=128K
sort_buffer_size=128K
join_buffer_size=4M

query_cache_type = 1
query_cache_size = 64M
query_cache_min_res_unit = 2k
query_cache_limit = 256K

sql_mode=""

[mysqld_safe]
log-error=/var/log/mysql.log
pid-file=/var/run/mysql/mysql.pid
```

Pour MySQL version 8 et versions ultérieures l'équipe de développement de MySQL ont conclu le support pour *query cache*, si vous souhaitez obtenir plus d'informations visitez le lien suivant

[:https://dev.mysql.com/blog-archive/mysql-8-0-retiring-support-for-the-query-cache/](https://dev.mysql.com/blog-archive/mysql-8-0-retiring-support-for-the-query-cache/)

Si vous utilisez MySQL 8 et que vous n'avez pas d'environnement HA, désactivez les journaux

binaires avec l'instruction suivante dans la section `[mysqld]`:

```
skip-log-bin
```

Toute modification que vous apportez au fichier `my.cnf` devra redémarrer le service MySQL.

- Vérifiez l'état du service avec `systemctl status mysqld.service`.
- Vérifiez la fin du fichier `/var/log/mysqld.log` pour voir si des erreurs se sont produites.
- Pour plus d'informations consultez le lien suivant [The Error Log](#) dans le site web de MySQL.

Reconstruire les bases de données

Pour en savoir plus sur la “ *Gestion et administration des serveurs* ”, [veuillez aller ver ce lien](#).

Lorsque le fichier `my.cnf` est modifié, l'un des “ problèmes ” les plus connus se produit, c'est la définition des nouvelles valeurs pour les enregistrements de transactions. Donc, si vous obtenez cette erreur, sauvegardez la base de données et restaurez la configuration précédente (utilisez les identifiants d'utilisateur racine ou `root`) :

```
InnoDB: Error: log file ./ib_logfile0 is of different size 0 5242880 bytes  
InnoDB: than specified in the .cnf file 0 67108864 bytes!
```

1. Une fois la sauvegarde créée, arrêtez le service MySQL avec la commande suivante :

```
systemctl stop mysql
```

2. Aller dans le dossier précédent où se trouvent les fichiers de données MySQL (`datadir`) (par défaut `/var/lib/mysql`) :

```
cd /var/lib/
```

3. Déplacez le dossier vers un autre emplacement (`/var/lib/mysql` → `/var/lib/mysql_backup`) :

```
mv mysql mysql_old
```

4. Créez un nouveau dossier (`/var/lib/mysql`) :

```
mkdir mysql
```

5. Affectez le propriétaire du dossier :

```
chown -R mysql. mysql
```

6. Initialisez le dossier avec les données MySQL :

```
mysql_install_db --datadir=/var/lib/mysql
```

7. Démarrez le service MySQL :

```
systemctl start mysql
```

Si vous recevez l'erreur suivant :

```
failed to retrieve rpm info for /var/lib/mysql/ibdata1
```

Il se peut que le SELinux soit en cours d'exécution. Vérifiez s'il a été nié (*denied*) en exécutant la commande suivante :

```
cat /var/log/audit/audit.log | grep /var/lib/mysql/ibdata1
```

- Pour désactiver SELinux, [consultez cette section](#).
- Si vous décidez de continuer en travaillant avec SELinux, [consultez cette section](#).

8. Lancez le configurateur et suivez l'assistant :

```
mysql_secure_installation
```

Vous pouvez sélectionner d'élérer le niveau de sécurité en utilisant des mots de passe complexes. Dans ce cas le mot de passe simple d'exemple utilisé *ice* dans la documentation (*pandora*) ne pourra pas être utilisé.

9. Reconstruire vos bases de données

```
mysql> create database pandora ;
```

Il se peut que vous avez besoin d'attribuer le mot de passe encore une fois pour l'utilisateur root dans MySQL. Pour ça utilisez cette commande, en remplaçant 'pandora' par le mot de passe établi dans l'étape 8 de cette section :

```
SET PASSWORD FOR 'root'@'localhost' =
password('pandora');
```

10. Assignez les permissions aux bons utilisateurs en remplaçant 'pandora' par le mot de pas établi pour l'étape 8 de cette section :

```
mysql> grant all privileges on pandora.* to pandora@'localhost' identified by
'pandora';
mysql> grant all privileges on pandora.* to pandora@'127.0.0.1' identified by
'pandora';
```

11. Chargez la sauvegarde de l'étape 1 :

```
mysql> use pandora;
mysql> source /path/to/your/backup.sql
```

Les systèmes avec MySQL/Percona ne chargent souvent pas correctement les paramètres de configuration de `my.cnf` (généralement parce que ces valeurs ont été entrées en dehors de la section `[mysqld]`)

Après avoir configuré le fichier `my.cnf` et redémarré le service MySQL, vous devez vérifier que ces modifications ont été correctement appliquées. Pour ce faire, utilisez la commande `SHOW VARIABLES` (le résultat peut contenir plus de cent éléments et être différent par rapport à la résumée suivante) :

```
mysql> show variables like 'innodb%';
+-----+-----+-----+
| Variable_name | Value |
+-----+-----+-----+
| innodb_adaptive_hash_index | ON |
| innodb_additional_mem_pool_size | 1048576 |
| innodb_autoextend_increment | 8 |
| innodb_autoinc_lock_mode | 1 |
| innodb_buffer_pool_size | 8388608 |
| innodb_checksums | ON |
| innodb_commit_concurrency | 0 |
| innodb_concurrency_tickets | 500 |
| innodb_data_file_path | ibdata1:10M:autoextend |
| innodb_data_home_dir | |
| innodb_doublewrite | ON |
| innodb_fast_shutdown | 1 |
| innodb_file_io_threads | 4 |
| innodb_file_per_table | OFF |
| innodb_flush_log_at_trx_commit | 1 |
| innodb_flush_method | |
```

innodb_force_recovery	0	
innodb_lock_wait_timeout	50	
innodb_locks_unsafe_for_binlog	OFF	
innodb_log_buffer_size	1048576	
innodb_log_file_size	5242880	
innodb_log_files_in_group	2	
innodb_log_group_home_dir	./	
innodb_max_dirty_pages_pct	90	
innodb_max_purge_lag	0	
innodb_mirrored_log_groups	1	
innodb_open_files	300	
innodb_rollback_on_timeout	OFF	
innodb_stats_method	nulls_equal	
innodb_stats_on_metadata	ON	
innodb_support_xa	ON	
innodb_sync_spin_loops	20	
innodb_table_locks	ON	
innodb_thread_concurrency	8	
innodb_thread_sleep_delay	10000	
innodb_use_legacy_cardinality_algorithm	ON	
+-----+-----+		

Consultez également les variables une par une :

```
mysql> show variables like 'innodb_log_file_size';
mysql> show variables like 'innodb_io_capacity';
mysql> show variables like 'innodb_file_per_table';
```

```
mysql> show variables like 'innodb_log_file_size';
+-----+-----+
| Variable_name      | Value      |
+-----+-----+
| innodb_log_file_size | 67108864  |
+-----+-----+
1 row in set (0.00 sec)

mysql> show variables like 'innodb_io_capacity';
+-----+-----+
| Variable_name      | Value      |
+-----+-----+
| innodb_io_capacity | 100        |
+-----+-----+
1 row in set (0.00 sec)

mysql> show variables like 'innodb_file_per_table';
+-----+-----+
| Variable_name      | Value      |
+-----+-----+
| innodb_file_per_table | ON         |
+-----+-----+
1 row in set (0.00 sec)

mysql>
```

Vérifier que les fichiers de données isolés pour chaque table sont ACTIFS

```
ls -lah /var/lib/mysql/pandora/*.ibd | wc -l
```

Il devrait avoir plus de 100 fichiers (selon la version de Pandora FMS). Chaque `.ibd` sera le fichier de données pour chaque table, lorsque vous aurez activé le paramètre `innodb_file_per_table` dans le fichier `my.cnf`. Si vous n'avez aucun de ces fichiers `.ibd`, cela signifie que vous n'utilisez qu'un seul fichier pour stocker toutes les informations. Cela signifie que la fragmentation des tableaux est commune au reste des tableaux, ce qui pourrait indiquer que la performance sera pire chaque semaine qui passe.

Si vous avez votre base de données fonctionnant sous une seule base de données, vous devrez d'abord recréer la base de données après avoir correctement configuré le fichier `my.cnf` et redémarré MySQL.

Vérification de la table de fragmentation par table

En utilisant le CLI de MySQL, vous devriez exécuter cette requête :

```
Select ENGINE, TABLE_NAME, Round( DATA_LENGTH/1024/1024) as data_length ,
round(INDEX_LENGTH/1024/1024)
as index_length, round(DATA_FREE/ 1024/1024) as data_free,
(data_free/(index_length+data_length))
as frag_ratio from information_schema.tables where DATA_FREE> 0 order by
frag_ratio desc;
```

Vous ne devriez voir que les tables avec un certain indice de fragmentation, par exemple :

ENGINE	TABLE_NAME	data_length	index_length	data_free	frag_ratio
InnoDB	tserver_export_data	0	0	5	320.0000
InnoDB	tagent_module_inventory	0	0	6	25.6000
InnoDB	tagente_datos_inventory	4	0	40	9.8842
InnoDB	tsession_extended	1	0	4	3.3684
InnoDB	tagent_access	2	7	27	2.9845
InnoDB	tpending_mail	2	0	4	2.6392
InnoDB	tagente_modulo	2	0	4	

2.1333						
InnoDB		tgis_data_history		24		11 67
1.9075						
InnoDB		tsession		2		0 4
1.7778						
InnoDB		tupdate		3		0 3
1.1852						
InnoDB		tagente_datos		186		194 399
1.0525						
InnoDB		tagente_datos_string		15		9 24
0.9981						
InnoDB		tevento		149		62 46
0.2183						
InnoDB		tagente_datos		2810		2509 65
0.0122						
InnoDB		tagente_datos_string		317		122 5
0.0114						
+-----+	-----+	-----+	-----+	-----+	-----+	-----+
-----+						

Cette requête ne fonctionnera que sur les tables ayant une fragmentation supérieure à 10 %.

Les tables qui sont trop grandes (comme tagent_donnees) pourraient prendre trop de temps à optimiser si elles sont trop fragmentées. Cela pourrait avoir un certain impact sur les systèmes de production. Nous recommandons donc de ne pas optimiser de si grosses tables, car cela pourrait provoquer un crash du système (le processus d'optimisation "verrouille" la table pour la réécrire)

Pour optimiser la table tagent_module_inventory :

```
optimize table tagent_module_inventory;
```

Un message apparaîtra pour vous le dire :

```
"Table does not support optimize, doing recreate + analyze instead".
```

Maintenant, si vous vérifiez à nouveau la fragmentation des tableaux, vous pouvez voir que la fragmentation a disparu :

ENGINE	TABLE_NAME	data_length	index_length	data_free	frag_ratio
+-----+	-----+	-----+	-----+	-----+	-----+
-----+					

InnoDB	tserver_export_data		0		0		5
320.0000							
InnoDB	tagente_datos_inventory		4		0		40
9.8842							
InnoDB	tsesion_extended		1		0		4
3.3684							
InnoDB	tagent_access		2		7		27
2.9845							
InnoDB	tpending_mail		2		0		4
2.6392							
InnoDB	tagente_modulo		2		0		4
2.1333							
InnoDB	tgis_data_history		24		11		67
1.9075							
InnoDB	tsesion		2		0		4
1.7778							
InnoDB	tupdate		3		0		3
1.1852							
InnoDB	tagente_datos		186		194		399
1.0525							
InnoDB	tagente_datos_string		15		9		24
0.9981							
InnoDB	tevento		149		62		46
0.2183							
InnoDB	tagente_datos		2810		2509		65
0.0122							
InnoDB	tagente_datos_string		317		122		5
0.0114							

Pour pouvoir effectuer cette optimisation, il sera nécessaire de disposer de l'espace nécessaire sur le disque dur pour effectuer l'opération. Sinon, une erreur apparaîtra et l'opération ne sera pas effectuée

Charge du système

Ceci est d'ordre général, mais nous devons nous assurer que le système d'E/S n'est pas un goulot d'étranglement (disques). Vous devrez exécuter la commande " vmstat " pour recueillir les informations du système :

```
vmstat 1 10
```

Maintenant, regardez les dernières colonnes (CPU-WA), une valeur supérieure à 10 signifie que

vosre disque d'E/S a un probl me qui doit  tre corrig .

Il est normal d'avoir quelques valeurs de CPU-US plus grands que c ero, sinon cela signifierait que vosre syst me utilise la m moire SWAP, ce qui est consid r  comme un tueur de performance. Vous devrez augmenter la m moire RAM ou diminuer la m moire RAM utilis e dans vos applications (threads Pandora FMS, buffers MySQL, etc).

Nous vous montrons l'exemple de sortie d'un syst me " normal " :

```
procs -----memory----- ---swap-- -----io----- --system-- -----cpu-----
r  b   swpd   free   buff  cache   si   so   bi   bo   in   cs  us  sy  id  wa  st
0  0   46248   78664 154644 576800   0   0    2  147    0   9  7 10 83  0  0
0  0   46248   78656 154644 576808   0   0    0    0   49  37  0  0 100  0
0
2  0   46248   78904 154648 576740   0   0    0  184   728 2484 63  6 31  0  0
0  0   46248   79028 154648 576736   0   0   16  616  363  979 21  0 79  0  0
1  0   46248   79028 154648 576736   0   0    0   20   35  37  0  1 98  1  0
0  0   46248   79028 154648 576736   0   0    0    0   28  22  0  0 100  0
0
1  0   46248   79028 154648 576736   0   0    0 3852  141  303  0  0 98  2  0
2  0   46248   78904 154660 576660   0   0    0  188  642 2354 56  4 40  0  0
1  0   46248   78904 154660 576680   0   0    0   88  190  634 13  0 86  1  0
1  0   46248   78904 154660 576680   0   0    0   16   35   40  0  0 100  0
0
1  0   46248   78904 154660 576680   0   0    0    0   26  21  0  0 100  0
0
0  0   46248   78904 154660 576680   0   0    0    0   27  27  0  0 100  0
0
1  0   46248   78904 154724 576616   0   0  112  192  608 2214 52  4 44  0  0
0  0   46248   78904 154724 576616   0   0    0   76  236  771 16  0 84  0  0
0  0   46248   78904 154724 576616   0   0    0   20   38   38  0  0 100  0
0
0  0   46248   78904 154724 576616   0   0    0    0   31  21  0  0 100  0
0
0  0   46248   78904 154740 576608   0   0    0 3192  187  322  1  0 96  3  0
1  0   46248   79028 154756 576544   0   0   16  192  632 2087 53  5 42  0  0
0  0   46248   79028 154760 576568   0   0    0   56  255  927 19  2 79  0  0
0  0   46248   79028 154768 576564   0   0    0   20   33   44  0  0 100  0
0
```

Partitionnement des tables MySQL

Pour utiliser le partitionnement des tables MySQL, vous devriez  galement utiliser le syst me de "multiples tablespaces" [d crit ci-dessus](#) (innodb_file_per_table).

MySQL 5.1 supporte le partitionnement des tables, ce qui vous permet de distribuer de très grandes tables en plus petits morceaux, comme des subdivisions logiques (vous pouvez consulter le [manuel de MySQL](#) pour plus de détails.)

E Si vous avez de grandes quantités de données dans votre base de données Pandora FMS et que vous pensez que beaucoup d'opérations de la console qui se réfèrent à ces données (par exemple drawing graph) sont assez lentes, vous améliorerez vos performances en utilisant le partitionnement des tables.

Vérifiez tout d'abord que le répertoire `/var/lib/mysql/pandora_history/*.ibd` contient beaucoup de fichiers (un par table), sinon, vous devrez faire un dump de la base de données, changer la configuration du fichier `my.cnf`, redémarrer MySQL, supprimer la base de données actuelle et la recréer à partir du dump.

Une fois que vous vous êtes assuré que `innodb_file_per_table` est activé, nous allons séparer les deux bases de données principales en différentes partitions. C'est un exemple de partitionnement de toute l'année 2015 jusqu'à présent et pour les mois à venir.

Cette opération nécessitera suffisamment d'espace disque pour être menée à bien. Nous devons vérifier la taille de `tagent_donnees.ibd`. Si par exemple il occupe 10gb, vous aurez besoin de 15gb d'espace libre pour démarrer l'opération.

Cette opération peut prendre beaucoup de temps, selon la taille de la table. Par exemple, il faudrait une heure et demie pour fractionner une table contenant environ 7 500 modules de données pour 100 jours (plus de 50 000 000 de lignes).

Pour démarrer le processus, vous devez exécuter la requête suivante dans le CLI MySQL :

```
ALTER TABLE tagente_datos PARTITION BY RANGE (utimestamp) (  
PARTITION Ene15 VALUES LESS THAN (UNIX_TIMESTAMP('2015-01-01 00:00:00')),  
PARTITION Feb15 VALUES LESS THAN (UNIX_TIMESTAMP('2015-02-01 00:00:00')),  
PARTITION Mar15 VALUES LESS THAN (UNIX_TIMESTAMP('2015-03-01 00:00:00')),  
PARTITION Apr15 VALUES LESS THAN (UNIX_TIMESTAMP('2015-04-01 00:00:00')),  
PARTITION May15 VALUES LESS THAN (UNIX_TIMESTAMP('2015-05-01 00:00:00')),  
PARTITION Jun15 VALUES LESS THAN (UNIX_TIMESTAMP('2015-06-01 00:00:00')),  
PARTITION Jul15 VALUES LESS THAN (UNIX_TIMESTAMP('2015-07-01 00:00:00')),  
PARTITION Ago15 VALUES LESS THAN (UNIX_TIMESTAMP('2015-08-01 00:00:00')),  
PARTITION Sep15 VALUES LESS THAN (UNIX_TIMESTAMP('2015-09-01 00:00:00')),  
PARTITION Oct15 VALUES LESS THAN (UNIX_TIMESTAMP('2015-10-01 00:00:00')),  
PARTITION Nov15 VALUES LESS THAN (UNIX_TIMESTAMP('2015-11-01 00:00:00')),  
PARTITION Dec15 VALUES LESS THAN (UNIX_TIMESTAMP('2015-12-01 00:00:00')),  
PARTITION pActual VALUES LESS THAN (MAXVALUE)  
);
```

Ensuite, la prochaine requête devrait être exécutée chaque mois pour réorganiser le partitionnement :

```
ALTER TABLE tagente_datos REORGANIZE PARTITION pActual INTO (  
PARTITION Feb16 VALUES LESS THAN (UNIX_TIMESTAMP('2016-02-01 00:00:00')),  
PARTITION pActual VALUES LESS THAN MAXVALUE);
```

En changeant “ Feb2022 ” par le mois dans lequel vous êtes.

Rappelez-vous que cette opération peut prendre des heures, selon la taille de la table tagent_donnees. Vous pouvez vérifier le processus en voyant la taille des fichiers de partitionnement en cours d'exécution :

```
[root@firefly pandora_history]# ls -lah | grep "#sql"  
  
-rw-rw---- 1 mysql mysql 424M dic 23 05:58 #sql-76b4_3f7c#P#Ago15.ibd  
-rw-rw---- 1 mysql mysql 420M dic 23 05:51 #sql-76b4_3f7c#P#Apr15.ibd  
-rw-rw---- 1 mysql mysql 128K dic 23 05:40 #sql-76b4_3f7c#P#Dec15.ibd  
-rw-rw---- 1 mysql mysql 840M dic 23 05:44 #sql-76b4_3f7c#P#Ene15.ibd  
-rw-rw---- 1 mysql mysql 440M dic 23 05:47 #sql-76b4_3f7c#P#Feb15.ibd  
-rw-rw---- 1 mysql mysql 10M dic 23 05:42 #sql-76b4_3f7c#P#Jan16.ibd  
-rw-rw---- 1 mysql mysql 404M dic 23 05:56 #sql-76b4_3f7c#P#Jul15.ibd  
-rw-rw---- 1 mysql mysql 436M dic 23 05:54 #sql-76b4_3f7c#P#Jun15.ibd  
-rw-rw---- 1 mysql mysql 400M dic 23 05:49 #sql-76b4_3f7c#P#Mar15.ibd  
-rw-rw---- 1 mysql mysql 408M dic 23 05:52 #sql-76b4_3f7c#P#May15.ibd  
-rw-rw---- 1 mysql mysql 72M dic 23 06:03 #sql-76b4_3f7c#P#Nov15.ibd  
-rw-rw---- 1 mysql mysql 404M dic 23 06:03 #sql-76b4_3f7c#P#Oct15.ibd  
-rw-rw---- 1 mysql mysql 416M dic 23 06:00 #sql-76b4_3f7c#P#Sep15.ibd
```

Reconstruire la base de données

Pour en savoir plus sur la “ Sauvegarde et restauration des données sur Pandora FMS ”, allez vers [ce lien](#).

Reconstruction partielle

Le système de gestion de base de données de MySQL, comme d'autres moteurs SQL tels qu'Oracle™, se dégrade avec le temps en raison de causes telles que la fragmentation des données produite par la suppression et l'insertion continue dans de grandes tables. Dans les grands environnements avec un volume de trafic élevé, il existe un moyen très simple d'améliorer les performances et d'empêcher leur dégradation : reconstruire périodiquement la base de données.

Pour ce faire, un arrêt de service doit être programmé, ce qui peut prendre environ 1 heure.

Dans cet arrêt de service, ce que vous devez faire est d'arrêter la console Pandora FMS WEB, et le serveur
Attention : laissez le serveur tentaculaire continuer à

recevoir des données, et celles-ci seront traitées dès que le serveur sera à nouveau opérationnel.

Une fois arrêté, faites un dump de la base de données (Export) ; dans cet exemple la base de données s'appelle pandora3 et l'utilisateur doit être root :

```
mysqldump -u root -p pandora3> /tmp/pandora3.sql  
Entrez le mot de passe :
```

On efface la base de données :

```
mysql -u root -p  
Enter password:
```

```
mysql> drop database pandora3;  
Query OK, 87 rows affected (1 min 34.37 sec)
```

Après créez la base de données et importez les données depuis l'exportation précédente :

```
mysql> create database pandora3;  
Query OK, 1 row affected (0.01 sec)  
mysql> use pandora3;  
mysql> source /tmp/pandora3.sql
```

Cela peut prendre quelques minutes, selon que le système est grand et que le matériel n'est pas trop important, pour un système avec 1500 agents et environ 100.000 modules.

Vous pouvez automatiser ce processus, mais en raison de sa nature délicate, il est préférable de le faire manuellement.

Reconstruction total

Ce chapitre ne concerne que les bases de données d'InnoDB. Pandora FMS est construit sur les bases de données InnoDB.

Malheureusement, MySQL se dégrade avec le temps, ce qui affecte les performances de l'ensemble du système. Il n'y a pas d'autre solution qui ne passe par la reconstruction de tous les schémas de base de données à partir de 0, en reconstruisant le fichier de données binaires que MySQL utilise pour stocker toutes les informations et les fichiers utilisés pour reconstruire les transactions.

Si vous regardez le répertoire `/var/lib/mysql` vous pouvez voir qu'il y a trois fichiers, qui sont

toujours appelés les mêmes et qui sont, selon la gravité du cas, des géants. Par exemple :

```
-rw-rw---- 1 mysql mysql 4.8G 2012-01-12 14:00 ibdata1
-rw-rw---- 1 mysql mysql 5.0M 2012-01-12 14:00 ib_logfile0
-rw-rw---- 1 mysql mysql 5.0M 2012-01-12 14:00 ib_logfile1
```

Le fichier `ibdata1` est celui qui contient toutes les données InnoDB du système. Dans un système très fragmenté, cela prend beaucoup de temps sans “ refaire ” ou “ réinstaller ” ce système sera gros et peu efficace. Le paramètre `innodb_file_per_table` dont nous avons parlé précédemment, régule une partie de ce comportement.

De plus, chaque base de données possède dans le répertoire `/var/lib/mysql` un répertoire pour définir sa structure. Vous devrez les supprimer aussi.

La procédure est simple :

- Dump (via `mysqldump`) tous les schémas et données sur le disque :

```
mysqldump -u root -p -A> all.sql
```

- Arrêtez MySQL.
- Supprimer les répertoires de la base de données InnoDB et `ibdata1`, `ib_logfile0`, `ib_logfile1`
- Démarrer MySQL.
- Créer à nouveau une base de données pandora (`create database pandora;`)
- Importer le fichier de sauvegarde (`all.sql`)

```
mysql -u root -p
mysql> source all.sql;
```

Le système devrait fonctionner ostensiblement plus vite maintenant.

Indices optionnels

Dans certaines situations, il est possible d'optimiser le fonctionnement de MySQL au détriment des ressources du système.

Cet indice, sert à optimiser la vitesse d'obtention des graphiques au prix d'une plus grande utilisation du disque et d'une légère diminution des performances des suppressions / insertions dans les tableaux de données :

```
ALTER TABLE `pandora`.`tagent_donnees` ADD INDEX `id_agent_module_utimestamp`
( `id_agent_module` , `utimestamp` );
```

Actuellement dans les tables plus lourdes de Pandora FMS dans MySQL vient cette optimisation par défaut. Il est

pratique de demander à des experts avant d'optimiser les tables MySQL

Requêtes lentes

Dans certains systèmes, selon le type d'information dont nous disposons, nous pouvons trouver des " requêtes lentes " qui aggravent le système. Nous pouvons activer le journal de ce type de requêtes pendant une courte période de temps (car cela nuit aux performances du système) afin d'étudier les requêtes pour essayer d'optimiser les tables avec des index.

Pour activer ce système, nous devons procéder comme suit :

- **Editez le fichier** my.cnf et ajoutez les lignes suivantes :

```
slow_query_log=1
long_query_time=2
slow_query_log_file=/var/log/mysql_slow.log
```

- Afin de l'utiliser créez-le et établissez les régulations administratives :

```
touch /var/log/mysql_slow.log
chown mysql:mysql /var/log/mysql_slow.log
chmod 640 /var/log/mysql_slow.log
```

- Redémarrer MySQL.
- Lorsque vous finissez d'analyser lesquels sont les slow queries, rappelez-vous de rétablir le fichier my.cnf en commentant les lignes ajoutées et en redémarrant le service MySQL encore une fois.

Optimisation de tableaux spécifiques

Une autre solution moins " radicale " pour pallier le problème de la fragmentation est l'utilisation de l'outil OPTIMIZE de MYSQL pour optimiser certaines tables FMS de Pandora. Pour cela, directement depuis MySQL, exécutez :

```
OPTIMIZE table tagente_datos;
OPTIMIZE table tagente;
OPTIMIZE table tagente_datos_string;
OPTIMIZE table tagent_access;
OPTIMIZE table tagente_modulo;
OPTIMIZE table tagente_estado;
```

Cela améliore les performances et il ne devrait pas être nécessaire de le lancer plus d'une fois par semaine, en pouvant le faire " HOT " pendant que le système fonctionne.

Dans les très gros systèmes, l'OPTIMIZE peut se *bloquer* et ne pas être une alternative, il est donc préférable de **reconstruire la base de données**.

Après avoir effectué ces opérations, il est pratique de les exécuter :

```
FLUSH TABLES;
```

D'après le manuel de MySQL :

For InnoDB tables, OPTIMIZE TABLE is mapped to ALTER TABLE, which rebuilds the table to update index statistics and free unused space in the clustered index.

(Pour les tables InnoDB, OPTIMIZE TABLE est mis en correspondance avec ALTER TABLE, qui reconstruit la table pour mettre à jour les statistiques de l'index et libérer l'espace inutilisé dans l'index en grappe).

Dans les installations à partir de 7.0 OUM715, la configuration suivante est appliquée par défaut

Note : Si votre installation de Pandora FMS a été faite avant la version 7.0 OUM 715, nous vous recommandons de faire les modifications suivantes dans votre base de données (principale et historique) :

```
alter table tagent_donnees add index (id_agente_modulo,utimestamp);
```

Cette action ajoutera un index à la table tagente_datos, permettant que les requêtes qui utilisent à la fois le système de reporting, la requête de données ou le module ou les graphiques personnalisés, renvoient des résultats dans un temps considérablement plus court que le précédent.

Jetons MySQL spéciaux

Il existe des jetons mySQL très *spéciaux*, qui peuvent aider ou nuire aux performances :

- innodb_thread_concurrency :

```
# Set to 0 in mysql 5.1.12 or higher  
innodb_thread_concurrency = 20
```

Ce paramètre (innodb_thread_concurrency) dans la version 5.1.12 ou supérieure, s'il est à 0 il signifie qu'il n'y a pas de limite à la concurrence, dans les versions inférieures s'il est à 20 ou plus, il implique la même chose : aucune limite à la concurrence.

- innodb_flush_method :

```
innodb_flush_method = 0_DIRECT
```


Ce paramètre affecte la façon dont il est écrit sur le disque.

- `innodb_lock_wait_timeout` :

```
innodb_lock_wait_timeout = 90
```

Il empêche qu'en cas de blocage occasionnel, MySQL *abandonne* (MySQL est parti) et s'arrête. Si elle dépasse 90 secondes, c'est un gros problème.

Références

Références :

- <http://dev.mysql.com/tech-resources/presentations/presentation-oscon2000-20000719/index.html>
- <http://jeremy.zawodny.com/mysql/mysql-optimization.html>

MySQL Percona XTraDB

Percona est une version améliorée de MySQL, notamment en ce qui concerne l'extensibilité. Il tire un meilleur parti des systèmes multi-CPU et accélère l'accès au disque.

Pour configurer votre percona MySQL, utilisez cet excellent assistant en ligne, qui va générer votre fichier `/etc/my.cnf` : [Percona Wizard Configurator](#)

Dimensionnement du Pandora FMS pour la haute capacité

Cette section décrit différentes méthodes pour configurer Pandora FMS dans un environnement à haute capacité. Il décrit également différents outils pour effectuer des tests de charge, utiles pour ajuster l'environnement à la plus grande capacité de traitement possible.

Pandora FMS a été configuré pour supporter une charge d'environ 2500 agents dans des systèmes où la base de données, la console et le serveur sont dans la même machine. Le chiffre recommandé est d'environ 2500 agents par système, mais ce chiffre varie beaucoup selon qu'il s'agisse d'agents XML, de modules distants, avec des intervalles élevés ou faibles, ou de systèmes avec beaucoup de capacité ou peu de mémoire.

Tous ces facteurs modifient grandement le nombre d'agents qu'un système peut gérer efficacement. Lors de tests en laboratoire, il a été possible d'exécuter 10000 agents dans un seul serveur avec un matériel de base, mais fortement optimisé.

Exemple de configuration d'un serveur de grande capacité

En supposant une machine CentOS avec 16 Go de RAM et 8 CPU que nous aimerions optimiser pour une capacité de traitement maximale du serveur de données (XML).

my.cnf

Seuls les paramètres les plus significatifs sont affichés).

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
user=mysql
character-set-server=utf8
skip-character-set-client-handshake
# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0
# Mysql optimizations for Pandora FMS
# Please check the documentation in http://pandorafms.com for better results
max_allowed_packet = 64M
innodb_buffer_pool_size = 800M
innodb_lock_wait_timeout = 90
innodb_file_per_table
innodb_flush_log_at_trx_commit = 0
innodb_flush_method = 0_DIRECT
innodb_log_file_size = 64M
innodb_log_buffer_size = 16M
innodb_io_capacity = 100
thread_cache_size = 8
thread_stack = 256K
max_connections = 100
wait_timeout = 900
key_buffer_size=4M
read_buffer_size=128K
read_rnd_buffer_size=128K
sort_buffer_size=128K
join_buffer_size=4M
query_cache_type = 1
query_cache_size = 64M
query_cache_min_res_unit = 2k
query_cache_limit = 256K
sql_mode=""
[mysqld_safe]
log-error=/var/log/mysql.log
pid-file=/var/run/mysql/mysql.pid
```

Pour MySQL version 8, et versions ultérieures, l'équipe de développement

de MySQL on conclut le support pour *query cache*, si vous souhaitez d'obtenir plus d'informations, visitez le site web suivant :

<https://dev.mysql.com/blog-archive/mysql-8-0-retiring-support-for-the-query-cache/>

pandora_server.conf

Seuls les paramètres les plus significatifs sont affichés.

```
verbose 3
server_threshold 5
dataserver_threads 1
max_queue_files 5000
```

Des choses à garder à l'esprit :

- Le nombre de `verbose` fait référence à la quantité d'informations écrites dans les journaux, il est conseillé de ne pas dépasser 10. Plus le nombre est élevé, moins la performance du Pandora FMS sera élevée en raison de la grande quantité d'informations à écrire dans les journaux.
- Une valeur haute (15) du paramètre `server_threshold` fait moins souffrir la base de données, tandis que l'augmentation du nombre maximum de fichiers traités fait que chaque fois que le serveur *cherche des fichiers* il remplit les tampons. Ces deux éléments de la configuration sont intimement liés. Dans le cas d'une optimisation du serveur réseau, il serait souhaitable d'abaisser le seuil du `server_threshold` à 5 ou 10.
- Le nombre très élevé de fils (plus de 5) dans `dataserver_threads` ne profite qu'aux processus avec une longue attente d'E/S, comme le réseau ou le serveur de plugins, dans le cas du serveur de données qui est tout le temps du processus, il pourrait même pénaliser la performance, c'est pourquoi nous en utilisons 5 ici. Dans les systèmes avec une DB lente, nous utiliserions encore moins de threads), essayer différentes combinaisons entre 1 et 10. Dans le cas d'une optimisation du système pour le serveur réseau, le nombre serait beaucoup plus élevé, entre 10 et 30.
- Certains paramètres de la configuration pourraient affecter fortement les performances de Pandora FMS, comme le paramètre `agent_access` (configurable depuis la Console).

Outils d'analyse de la capacité (Capacity)

Pandora FMS dispose de plusieurs outils qui vous aideront à dimensionner correctement votre matériel et vos logiciels en fonction du volume de données que vous vous attendez à obtenir. L'un d'eux est utile pour "attaquer" directement la base de données avec des données fictives (`dbstress`) et l'autre génère des fichiers XML fictifs (`xml_stress`)

Pandora FMS XML Stress

Il s'agit d'un petit script qui génère des fichiers de données XML comme ceux envoyés par les

agents Pandora FMS. C'est dans

```
/usr/share/pandora_server/util/pandora_xml_stress.pl
```

Les scripts lisent les noms des agents à partir d'un fichier texte et génèrent des fichiers de données XML pour chaque agent en fonction du fichier de configuration, où les modules sont définis comme des modèles.

Les modules sont remplis avec des données aléatoires. Il est possible de spécifier la valeur initiale et la probabilité de changement des données d'un module.

Exécutez le script de cette façon :

```
./pandora_xml_stress.pl <configuration file>
```

Exemple de configuration d'un fichier :

```
# Maximum number of threads, by default 10.
max_threads 10

# File containing a list of agent names (one per line).
agent_file agent_names.txt

# Directory where XML data files will be placed, by default /tmp.
temporal /var/spool/pandora/data_in

# Pandora FMS XML Stress log file, logs to stdout by default.
log_file pandora_xml_stress.log

# XML version, by default 1.0.
xml_version 1.0

# XML encoding, by default ISO-8859-1.
encoding ISO-8859-1

# Operating system (shared by all agents), by default Linux.
os_name Linux

# Operating system version (shared by all agents), by default 2.6.
os_version 2.6

# Agent interval, by default 300.
agent_interval 300

# Data file generation start date, by default now.
time_from 2009-06-01 00:00:00

# Data file generation end date, by default now.
time_to 2009-06-05 00:00:00
```

```
# Delay after generating the first data file for each agent to avoid
# race conditions when auto-creating the agent, by default 2.
startup_delay 2

# Address of the Tentacle server where XML files will be sent (optional).
# server_ip 192.168.50.1

# Port of the Tentacle server, by default 41121.
# server_port 41121

# Module definitions. Similar to pandora_agent.conf.

module_begin
module_name Module 1
module_type generic_data
module_description A long description.
module_max 100
module_min 10
module_exec type=RANDOM;variation=60;min=20;max=80
module_end

module_begin
module_name Module 2
module_type generic_data
module_description A long description.
module_max 80
module_min 20
module_exec type=SCATTER;prob=1;avg=40;min=0;max=80
module_end

module_begin
module_name Module 3
module_type generic_data
module_description A long description.
module_max 80
module_min 20
module_exec type=CURVE;min=20;max=80;time_wave_length=3600;time_offset=0
module_end

module_begin
module_name Module 4
module_type generic_data_string
module_description A long description.
module_max 100
module_min 10
module_exec type=RANDOM;variation=60;min=20;max=80
module_end

module_begin
module_name Module_5
module_type generic_proc
module_descricion Module 3 description.
```

```
# Initial data.  
module_data 1  
module_end
```

Envoyer et recevoir la configuration locale de l'agent

En activant dans votre `pandora_xml_stress.conf` la valeur de configuration `get_and_send_agent_conf` à 1, vous pouvez faire en sorte que les agents de test de charge se comportent comme des agents normaux, puisqu'ils envoient leur fichier de configuration et le md5.

E Depuis Pandora FMS Console Enterprise, vous pouvez modifier la configuration distante de sorte que lors des exécutions successives du `pandora_xml_stress`, il utilise la configuration personnalisée de Pandora FMS Console Enterprise au lieu de la définition de `pandora_xml_stress.conf`.

À part cela, vous pouvez configurer où sauvegarder localement les `.conf` de vos agents de test avec le jeton de configuration `directory_confs` dans le fichier `pandora_xml_stress.conf`.

Fichier de configuration

- `max_threads` : Nombre de threads dans lesquels le script va s'exécuter, cela améliore les E/S.
- `agent_file` : Chemin du fichier de la liste des noms, séparé par une nouvelle ligne.
- `temporal` : Chemin du répertoire où les fichiers de données XML factices seront générés.
- `log_file` : Chemin du fichier de log où le script informera de son exécution.
- `xml_version` : Version des fichiers de données XML (par défaut 1.0).
- `encoding` : Encodage des fichiers de données XML (par défaut ISO-8859-1).
- `os_name` : Nom du système d'exploitation des agents factices (Linux par défaut).
- `os_version` : Version du système d'exploitation des agents factices (par défaut 2.6).
- `agent_interval` : Intervalle des agents factices en secondes (par défaut 300).
- `time_from` : Date de début pour générer les fichiers de données XML fictifs, au format "ANNÉE-MOIS-JOUR HEURE:MIN:SEC".
- `time_to` : Date de fin pour générer les fichiers de données XML fictifs, au format "ANNÉE-MOIS-TIME:MIN:SEC".
- `get_and_send_agent_conf` : Valeur booléenne 0 ou 1, quand elle est active les agents factices vont essayer de télécharger par configuration à distance une version plus actuelle du fichier de configuration standard d'un agent. Et depuis la console d'entreprise de Pandora FMS, vous pouvez les éditer.
- `startup_delay` : Valeur numérique du temps en secondes avant que chaque agent ne commence à générer les fichiers, elle est utilisée pour éviter les conditions de course.
- `timezone_offset` : Valeur numérique de l'offset du fuseau horaire.
- `timezone_offset_range` : Valeur numérique qui est utilisée pour générer dans cette plage les fuseaux horaires de manière aléatoire.
- `longitude_base` : Valeur numérique, est la latitude où les agents fictifs apparaîtront.
- `longitude_base` : Valeur numérique, est la longitude où les agents factices apparaîtront.
- `altitude_base` : Valeur numérique, est l'altitude à laquelle les agents factices apparaîtront.
- `position_radius` : Valeur numérique, est la plage autour de la circonférence avec ce rayon que l'agent factice apparaît de façon aléatoire.

Définition des modules

La définition d'un module dans le fichier de configuration du script et si vous avez activé la configuration à distance sera également la même :

```
module_begin
module_name <nom_du_module>
module_type <type_de_donnees_du_module>
module_description <description>
module_exec
type=<type_generation_xml_stress>;<autres_options_separees_par_des_points_virgules>
module_unit <unites>
module_min_critical <value>
module_max_critical <value>
module_min_warning <value>
module_max_warning <value>
module_end
```

Et vous pouvez chacun le configurer comme :

- `tipo_generación_xml_stress` : vous pouvez prendre les valeurs **RANDOM** , **SCATTER** , **CURVE**.
- `module_attenuation <value>` : La valeur générée est multipliée par la valeur donnée, généralement entre 0.1 et 0.9.
- `module_attenuation_wdays <value> <value> ... <value>` : La valeur du module n'est atténuée que les jours donnés, qui vont du dimanche (0) au samedi (6). Par exemple, le module suivant simule une diminution de 50 % du trafic réseau les samedis et dimanches :

```
module_begin
module_name Network Traffic
module_type generic_data
module_description Incoming network traffic (Kbit/s)
module_exec type=RANDOM;variation=50;min=0;max=1000000
module_unit Kbit/s
module_min_critical 900000
module_attenuation 0.5
module_attenuation_wdays 0 6
module_end
```

- `module_incremental <value>` : Si elle est configurée à un, la valeur précédente du module est toujours ajoutée à une nouvelle valeur, ce qui donne une fonction croissante.
- autres : voir ci-dessous les options dont il dispose, en fonction du type de génération de module.

es jetons de configuration `min/max_critical` et `min/max_warning` ne sont disponibles qu'à partir de la version 5.0.

Aléatoire (RANDOM)

Ils ont les options suivantes :

- Probabilité de *variation* en % qu'elle varie par rapport à la valeur précédente.
- min valeur minimum que la valeur peut avoir.
- max valeur maximale que la valeur peut avoir.

Numérique

Génération de valeurs numériques aléatoires entre les valeurs min et max.

Booléens

Valeurs générées entre 0 et 1.

Chaîne

Une chaîne de longueur est générée entre la valeur min et la valeur max, les caractères sont aléatoires entre A et Z y compris les majuscules et minuscules et les chiffres.

Source externe de données (SOURCE)

Vous permet de choisir un fichier de texte brut comme source de données. Options :

- src : fichier source des données.

Le fichier contient une donnée par ligne, il n'y a pas de limite de ligne. Par exemple :

```
4
5
6
10
```

Il supporte tous les types de valeurs (chaînes numériques et textuelles). Ce type de modules va utiliser chacune des données du fichier pour générer les données des modules dans Pandora, les données sont prises de manière séquentielle. Par exemple, les données du module précédent seraient vues dans Pandora comme ceci :

```
4 5 6 10 4 5 6 10 4 5 6 10 4 5 6 10 4 5 6 10 4 5 6 10
```

Dispersion (SCATTER)

Elle ne s'applique qu'aux données numériques, et les graphiques générés sont semblables à ceux d'un battement de coeur, c'est-à-dire une valeur normale et occasionnellement un "battement de coeur".

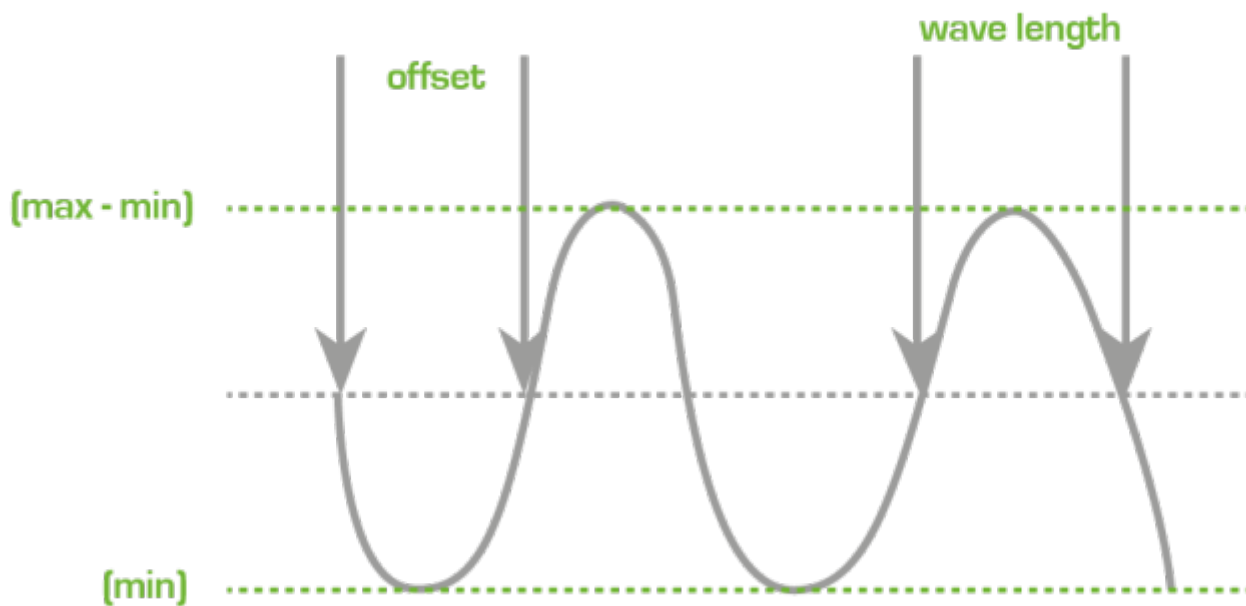
Et vous avez les options suivantes :

- min valeur minimale que la valeur peut avoir.
- max valeur maximale que la valeur peut avoir.
- Prob probabilité en % qu'il génère un *battement de coeur*.
- avg Valeur moyenne qui doit être affichée par défaut s'il n'y a pas de "*battement de coeur*".

Courbe (CURVE)

Il génère des données de module suivant une courbe trigonométrique. Ils ont les options suivantes :

- min valeur minimale que la valeur peut avoir.
- max valeur maximale que la valeur peut avoir.
- Valeur numérique en secondes de la durée du *pic* de l'onde.
- time_offset valeur numérique en secondes du début de l'onde à partir du temps zéro avec valeur zéro du module (similaire au graphique sinusoïdal).



Notes d'intérêt :

- Cet outil est préconfiguré pour chercher, dans tous les agents, les modules de nom " random ", " curve " ou "boolean ", et qu'utilisent une intervalle d'entre 300 secondes et 30 jours.

Comment mesurer la capacité de traitement du serveur de données

Il y a un petit script appelé `pandora_count.sh` qui se trouve dans le répertoire `/usr/share/pandora_server/util/` du serveur FMS Pandora. Ce script est utilisé pour mesurer le taux de traitement des fichiers XML par le serveur de données, et utilise comme

référence le total des fichiers en attente de traitement dans `/var/spool/pandora/data_in` ; pour pouvoir l'utiliser, il faut avoir des fichiers en attente de traitement pour plusieurs milliers de paquets (ou pour les générer avec l'outil mentionné précédemment). Une fois en exécution, vous pouvez l'arrêter en appuyant sur CTRL+C.

Ce script compte simplement les paquets existants maintenant et les soustrait des paquets qui étaient là il y a 10 secondes, puis divise le résultat par 10 et ce sont les fichiers qui ont été traités dans les 10 dernières secondes, montrant le taux par seconde. C'est une mesure un peu approximative, mais elle sert à ajuster les paramètres du serveur.

Pandora FMS DB Stress

C'est un petit outil pour tester la performance de votre base de données. Il peut également être utilisé pour "prégénérer" des données périodiques ou aléatoires (à l'aide de fonctions trigonométriques) et remplir des modules fictifs.

Vous devez créer un agent et lui affecter des modules pour l'injection automatique de données avec cet outil. Les noms doivent être appelés avec la notation suivante :

- *random* : pour générer des données aléatoires.
- *curve* : pour générer une courbe de coïncidences à l'aide de fonctions trigonométriques. Utile pour voir le travail d'interpolation avec différents intervalles, etc.
- *boolean* : pour générer des données booléennes aléatoires.

Vous pouvez donc utiliser n'importe quel nom contenant les mots aléatoire « *random* », « *curve* » et/ou « *boolean* », par exemple :

- *random_1*
- *courbe_autre*

Vous pouvez seulement choisir le type de module " *data_server* ".

Réglage fin de l'outil *Pandora FMS DB Stress*

Cet outil est préconfiguré pour rechercher, dans tous les agents, les modules nommés « *random* », « *curve* » ou « *boolean* », et qui utilisent un intervalle entre 300 secondes et 30 jours.

Si vous voulez modifier ce comportement, vous devez éditer le *script* `pandora_dbstress` et modifier certaines variables au début du fichier :

```
# Configure here target (AGENT_ID for Stress)
my $target_module = -1; # -1 for all modules of that agent
my $target_agent = -1;
my $target_interval = 300;
my $target_days = 30;
```

- La première ligne de la variable correspondant à `target_module`, doit être définie pour un module fixe ou -1 pour traiter toutes les cibles correspondantes.
- La deuxième ligne de la variable correspond à `target_agent`, pour un agent spécifique.
- La troisième ligne correspond à `target_interval`, défini en secondes et qui représente l'intervalle de périodicité prédéterminée du module.
- La quatrième ligne est `target_days` et représente le nombre de jours passés depuis la date, dans *timestamp*, actuelle.

Dépannage et outils de diagnostic dans Pandora FMS

Parfois, l'utilisateur a des problèmes et les développeurs de Pandora FMS ne peuvent pas l'aider sans avoir plus d'informations sur ses systèmes. Dans la version 3.0, nous avons développé quelques outils pour vous aider à résoudre certains problèmes :

Diagnostic Info

Dans les versions plus récentes du Pandora FMS, il existe une fonction permettant d'obtenir des informations de diagnostic de notre installation Pandora FMS.

Il est situé dans la section Admin tools → Diagnostic Info

- Operation
- Management
 - Discovery
 - Resources
 - Profiles
 - Configuration
 - Alerts
 - Servers
 - Setup
 - Admin tools
 - System audit log
 - Links
 - Diagnostic info
 - Omnishell
 - IPAM
 - Site news
 - File manager
 - DB Schema Check
 - DB Interface
 - DB Backup Manager
 - Elasticsearch Interface
 - Accountic console setup
 - API checker
 - Extension manager
 - Links
 - Warp Update
 - Module library
 - About

Enter keywords



Info status Pandora FMS

Pandora FMS Build	PC230427
Pandora FMS Version	v7.0NG.770
Minor Release	57
Homedir	/var/www/html/pandora_console
HomeUrl	https://munchkin.artica.es/pandora_console/
Enterprise installed	true
Update Key	ARTIC...F3MBZ
Updating code path	Path where the updated code is stored
Current Update #	412

Showing 1 to 9 of 9 entries [CSV](#)

PHP setup

PHP Version	8.0.26
PHP Max execution time	0
PHP Max input time	-1
PHP Memory limit	800M
Session cookie lifetime	0

Showing 1 to 5 of 5 entries [CSV](#)

Database size stats

Total amount of agents	200
Total modules	3479
Total groups	9
Total module data records	31233904
Total agent access record	2437
Total amount of events	499000
Total traps	0
Total users	19
Total sessions	72

Showing 1 to 9 of 9 entries [CSV](#)

Database health status

Total unknown agents	17
Total not-init modules	42
Pandora DB Last run	36 minutes 38 seconds Ago

Showing 1 to 3 of 3 entries [CSV](#)

Database status info

DB Schema Version (first installed)	6.0dev
DB Schema Version (actual)	6.0RC1
DB Schema Build	PD150908

Showing 1 to 3 of 3 entries [CSV](#)

System Info

CPU	Intel(R) Xeon(R) CPU E3-1230 v6 @ 3.50GHz x3
RAM	MemTotal: 3618520 kB
Os	Linuxmunchkin 3.10.0-314.el7.x86_64 #1 SMP Tue Nov 22 16:42:41 UTC 2016 x86_64 x86_64 GNU/Linux
Hostname	munchkin
IP	151.80.66.22, 127.0.0.1

Showing 1 to 5 of 5 entries [CSV](#)

MySQL Performance metrics

InnoDB buffer pool size	2048	It has to be 40% of the server memory not recommended to be greater or less
InnoDB file per table	ON	Recommended ON
InnoDB flush log at trx-commit	0	Recommended Value 2
InnoDB lock wait timeout	90	Min. Recommended Value 90s
InnoDB log buffer size	16	Min. Recommended Value 16M
InnoDB log file size	64	Min. Recommended Value 64M
Maximun allowed packet	64	Min. Recommended Value 32M
Maximun connections	100	Min. Recommended Value 90 connections
Query cache limit	0.25	Min. Recommended Value 8M
Query cache min-res-unit	2	Min. Recommended Value 2M
Query cache size	64	Min. Recommended Value 32M
Query cache type	ON	Recommended ON
Read buffer size	128	Min. Recommended Value 32K
Read rnd-buffer size	128	Min. Recommended Value 32K
Sort buffer size	128	Min. Recommended Value 32K
Sql mode		Must be empty
Thread cache size	8	Min. Recommended Value 8
Thread stack	256	Min. Recommended Value 256

Showing 1 to 18 of 18 entries [CSV](#)

Tables fragmentation in the Pandora FMS database

Tables fragmentation (maximum recommended value)	10%
Tables fragmentation (current value)	0.09%
Table fragmentation status	<input checked="" type="radio"/>

Dans cette fenêtre on peut voir l'information sur la configuration de Pandora FMS et MySQL, ainsi que les graphiques du système d'autocontrôle.

pandora_diagnostic.sh

```
# euclides root ~ /usr/share/pandora_server/util/pandora_diagnostic.sh
Pandora FMS Diagnostic Script v1.0 (c) ArticaST 2009-2015
http://pandorafms.org. This script is licensed under GPL2 terms

Please wait while this script is collecting data
cat: /etc/mysql/my.cnf: No such file or directory

Output file with all information is in '/tmp/pandora_diag.20211230_153859.data'
```

C'est un outil situé dans `/usr/share/pandora_server/util` et qui fournit beaucoup d'informations sur le système :

- Information de l'UCT.
- Uptime et avgload de l'UCT (temps de fonctionnement et moyenne de charge de travail).
- Information sur la mémoire.
- Information sur le noyau/la libération.
- Un fichier *dump* de configuration MySQL.
- Un fichier de vidage (filtrage des mots de passe) de la configuration du serveur Pandora FMS.
- Journaux d'information de Pandora FMS (mais pas le journal complet !).
- Information sur le disque.
- Information sur les processus de Pandora FMS.
- Une information complète du journal du kernel (dmesg).

Toutes les informations sont générées dans un fichier `.txt`, de sorte que les utilisateurs peuvent envoyer ces informations à toute personne qui souhaite les aider, par exemple, dans le forum des utilisateurs de Pandora FMS ou dans les listes de diffusion publiques de Pandora FMS. Cette information ne doit pas contenir de renseignements confidentiels. Notez que vous pouvez utiliser les privilèges `root` si vous voulez obtenir les fichiers `pandora_server.conf` et `my.cnf`.

C'est un exemple d'exécution :

```
$ ./pandora_diagnostic.sh

Pandora FMS Diagnostic Script v1.0 (c) ArticaST 2009
http://pandorafms.org. This script is licensed under GPL2 terms

Please wait while this script is collecting data

Output file with all information is in '/tmp/pandora_diag.20090601_164511.data'
```

Et voici quelques parties de la sortie du fichier :

Information gathered at 20090601_164511

Linux raz0r 2.6.28-12-generic #43-Ubuntu SMP Fri May 1 19:27:06 UTC 2009 i686
GNU/Linux

=====

CPUINFO

processor : 0
vendor_id : GenuineIntel
cpu family : 6
. .

Other System Parameters

Uptime: 16:45:11 up 5:27, 2 users, load average: 0.11, 0.12, 0.09

PROC INFO (Pandora)

slerena 11875 0.9 2.1 114436 44336 pts/0 Sl 13:14 1:56 gedit
pandora_diagnostic.sh
slerena 24357 0.0 0.0 4452 1524 pts/0 S+ 16:45 0:00 /bin/bash
./pandora_diagnostic.sh

MySQL Configuration file

The MySQL database server configuration file.

You can copy this to one of:
- "/etc/mysql/my.cnf" to set global options,
.
.
.

Pandora FMS Logfiles information

total 3032
drwxr-xrwx 2 root root 4096 2009-04-30 20:00 .
drwxr-xr-x 17 root root 4096 2009-06-01 11:24 ..
-rw-r----- 1 root sys 377322 2009-04-06 00:12 pandora_agent.log
-rw-r--r-- 1 root root 0 2009-04-06 00:15 pandora_agent.log.err
-rw-r--r-- 1 root root 13945 2009-04-02 21:47 pandora_alert.log
-rw-r--r-- 1 slerena slerena 2595426 2009-04-30 20:02 pandora_server.error
-rw-rw-rw- 1 root root 9898 2009-04-30 20:02 pandora_server.log
-rw-rw-rw- 1 root root 65542 2009-04-30 20:00 pandora_server.log.old
-rw-r--r-- 1 root root 94 2009-04-06 00:19 pandora_snmptrap.log
-rw-rw-rw- 1 root root 4 2009-04-03 14:16
pandora_snmptrap.log.index

System disk

```

S.ficheros          Tamaño Usado  Disp Uso% Montado en
/dev/sda6           91G   49G   37G  58% /
tmpfs               1003M     0 1003M   0% /lib/init/rw
varrun             1003M   260K 1002M   1% /var/run
varlock            1003M     0 1003M   0% /var/lock
udev               1003M   184K 1002M   1% /dev
tmpfs              1003M   480K 1002M   1% /dev/shm
lrm                1003M   2,4M 1000M   1% /lib/modules/2.6.28-12-
generic/volatile
-----
Vmstat (5 execs)
-----
procs -----memory----- ---swap--  -----io----- -system--  -----cpu-----
 r  b   swpd   free   buff  cache   si   so   bi   bo   in   cs  us  sy  id  wa
 2  0     0 684840 119888 619624   0   0   15   10  258  474  3  1 95  0
 0  0     0 684768 119888 619640   0   0    0    0  265  391  0  0 100  0
 0  0     0 684768 119892 619636   0   0    0   56  249  325  1  1 99  0
 0  0     0 684768 119892 619640   0   0    0    0  329  580  0  0 100  0
 0  0     0 684776 119892 619640   0   0    0    0  385 1382  1  0 99  0
-----
System dmesg
-----
[ 0.000000] BIOS EBDA/lowmem at: 0009f000/0009f000
[ 0.000000] Initializing cgroup subsys cpuset
[ 0.000000] Initializing cgroup subsys cpu
[ 0.000000] Linux version 2.6.28-12-generic (buildd@rothera) (gcc version
4.3.3 (Ubuntu 4.3.3-5ubuntu4) ) #43-Ubuntu SMP Fri May 1
19:27:06 UTC 2009 (Ubuntu 2.6.28-12.43-generic)
.
.
-----
END OF FILE
-----
560e8fa02818916d4abb59bb50d91f6a /tmp/pandora_diag.20090601_164511.data

```

[Retour à l'index de documentation du Pandora FMS](#)