PANDORAFMS

Optimización y solución de problemas

16-09-2021

Optimización y solución de problemas

Volver al Índice de Documentación Pandora FMS

Optimización y solución de problemas de Pandora FMS

Introducción

El servidor de Pandora FMS es capaz de monitorizar unos 2000 dispositivos, para ello es necesario afinar la configuración de la base de datos.

Además, en este capítulo, se explican algunas técnicas para detectar y solucionar problemas en la instalación de Pandora FMS.

Optimizando Pandora FMS

Optimización MySQL para //enterprise grade systems//

Consejos generales

Lo primero que se debe hacer si de verdad se desea tener un sistema ENORME con tablas mayores de 2GiB es seguir algunas pautas: MySQL recomienda usar un sistema de 64Bit. Además, se pueden sugerir estas recomendaciones generales: a más memoria RAM, y más CPU, mejor rendimiento. Acorde con nuestra experiencia, la memoria RAM es más importante que la CPU. Si está pensando usar 1GiB o una cantidad inferior de memoria para su sistema SQL, reconsidérelo. El mínimo para un sistema a nivel de empresa será 2GiB. Una buena opción para un gran sistema son 8GiB. Recuerde que más memoria RAM puede acelerar las actualizaciones clave mediante el mantenimiento de las páginas clave más usadas en la RAM.

Es buena idea ser capaz de retirar el sistema en caso de fallo. Para sistemas donde la base de datos está en un servidor específico, debería echar un vistazo a Ethernet Gigabit. La latencia es tan importante como el rendimiento.

La optimización del disco es muy importante para bases de datos muy grandes: habrá que partir las bases de datos y las tablas en diferentes discos. En MySQL se pueden usar enlaces simbólicos para ello. Utilice diferentes discos para el sistema y la base de datos, y muy importante: intente usar un disco duro de baja captura, ya que la aplicación se verá

comprometida por la velocidad de captura del disco, que aumenta en *N log N* según obtenga más datos.

Se recomienda el uso de discos SSD debido a su rapidez y la mejora de latencia en el sistema.

Use *–skip-locking* (activado de forma predeterminada en algunos sistemas) si es posible. Esto apagará el bloqueo externo y proporcionará un mejor rendimiento.

Si pone en marcha el cliente y el servidor MySQL en la misma máquina, use sockets en lugar de conexiones TCP/IP al conectar con MySQL (esto puede dar una mejora del 7.5%). Puede hacer esto sin especificar el nombre del anfitrión o el *localhost* al conectar al servidor MySQL. Deshabilite el inicio de sesión binario y la *replicación* si está lanzando sólo un servidor anfitrión MySQL.

Como aspecto general de cara a evaluar el rendimiento, tenga en cuenta que afecta mucho al rendimiento los siguientes puntos:

- NO utilice logs binarios si no va a utilizar una configuracion de MySQL con replicación.
- NO utilice logs de trazabilidad de queries o slowquery logs.

Versiones de MySQL

Se recomienda el uso de la versión modificada de MySQL (percona) [1)] que ofrecen un mayor rendimiento. Por defecto los plugins programados son para Percona.

Por otro lado, utilizar versiones recientes de MySQL (5.5) respecto a versiones más antiguas de MySQL (5.0.x) pueden ofrecer hasta una diferencia del 20% en rendimiento.

Herramientas automáticas de configuración

Existen bastantes herramientas para optimizar la configuración del servidor MySQL.

MySQL Tuning Primer, de Mattew Montgomery, es una herramienta de línea de comando para verificar la configuración de un servidor y ver su rendimiento y posibles mejoras, dando algunas pistas y sugerencias para mejorarlo. Está en https://bugs.launchpad.net/mysql-tuning-primer

Desactivar replicacion binaria

Por defecto viene habilitada en la mayoría de distros de Linux. Para desactivarla, editar el fichero *my.cnf*, habitualmente en */etc/my.cnf* y comentar las siguientes lineas:

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

Hay que comentar las dos líneas, y luego reiniciar el servidor, aunque por defecto estas líneas no vienen configuradas con la instalación en ISO de Pandora FMS.

Estas recomendaciones son para el caso de no tener configurado un sistema HA de Pandora FMS, el cual necesita la replicación binaria para su funcionamiento

Rendimiento de acceso a disco

Junto con otros parámetros clave, hay dos que son especialmente relevantes en cuanto a rendimiento de acceso a disco, que suele ser el cuello de botella respecto a MySQL.

```
innodb_log_file_size = 64M
```

Por defecto, se establece este valor, que puede ser mayor (incluso 512M) sin perjuicio, excepto para recuperación en caso de problema y una mayor ocupación de disco. El valor por defecto de MySQL es de 5M, el cual es muy bajo para entornos de producción con gran volumen de transacciones. Para alterar este valor con un sistema ya en funcionamiento primero tenemos que hacer un DUMP completo y borrar los ficheros índices binarios de Innodb (generalmente en /var/lib/mysql/ib*). Cambiar el my.cnf reiniciar el MySQL y cargar el dump de SQL. Dado que el proceso es el mismo que hay que hacer para activar el token **innodb_file_per_table** (descrito un poco más abajo, recomendamos hacer todo el proceso simultáneamente): cambiar todo el *my.cnf*, reiniciar y restaurar la BBDD una sola vez. Para conocer más acerca del proceso de backup y recuperación ir al siguiente enlace.

```
innodb_io_capacity = 100
```

Por defecto, este parámetro tiene el valor 100, pero debemos conocer previamente los IOPS del disco del sistema. Se puede conocer exactamente buscando IOPS y el modelo exacto del disco duro (obtenido via smartctl), donde los valores recomendados son: 7500RPM → 100 IOPS, 15000 RPM → 190 IOPS, SSD → 1500 IOPS

Evitando el Flush de disco en cada transacción

MySQL por defecto establece autocommit = 1 para cada conexión. Lo cual no es malo para MyISAM, ya que lo que uno escribe no está garantizado en el disco, pero para InnoDB significa que cada insert / update / delete en una tabla InnoDB se traducirá en una escritura en el disco.

¿Qué tiene de malo que escriba en el disco? Nada en absoluto. Se aseguran de que ante cualquier compromiso se garantiza que el dato esté allí cuando se reinicie la base de datos después de un accidente. El problema es que el funcionamiento de la BBDD está limitado por la velocidad física del disco. Dado que el disco tiene que escribir los datos en un disco antes de la confirmación de la escritura, esto toma su tiempo.

Suponiendo incluso un tiempo medio de búsqueda de 9ms por la escritura en disco, estamos limitados a aproximadamente 67 commits/sec1, esto es muy lento. Y mientras el disco está ocupado tratando de que el sector sea escrito, no está haciendo lecturas. InnoDB puede evitar parte de esta limitación realizando algunas escrituras juntas, pero aún así, la limitación existe.

Podemos evitar que escriba al final de cada transacción, haciendo que ponga un sistema "automático" de escritura, que escribe aproximadamente cada segundo. En caso de fallo, puede que perdamos los datos del último segundo, algo más que asumible si se trata de ganar eficiencia. Para ello, usaremos el siguente token de configuración:

```
innodb_flush_log_at_trx_commit = 0
```

Por defecto viene este valor en la configuración.

Mayor tamaño del KeyBuffer

Dependiendo de la RAM total del sistema, es un parámetro global muy importante que acelera DELETES e INSERT.

```
key_buffer_size = 4M
```

Este es el valor que viene por defecto en la configuración.

Otros buffers importantes

Hay varios buffer que por defecto en algunas distribuciones vienen vacíos. Modificar estos parámetros puede dar un rendimiento muy superior al que se obtiene por defecto. Es importante asegurarse de que existen estos tokens en el fichero de configuración de

MySQL.

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

Mejorando la concurrencia de innodb

Existe un parámetro que puede afectar bastante al rendimiento del servidor MySQL con Pandora. Este parámetro es *innodb_thread_concurrency*. Dicho parámetro sirve para especificar cuantos "hilos concurrentes" puede ejecutar MySQL. Una mala configuración de este parámetro puede hacer que vaya mas lento que por defecto, por lo que es especialmente importante prestar atención a varios parámetros.

- Versión de MySQL. En diferentes versiones de MySQL este parámetro se comporta MUY diferente.
- Número de procesadores reales (físicos).

Aquí se puede leer la documentacion oficial de MySQL [2)].

El valor recomendado es el número de CPU (físicas) multiplicado por 2 más el nº de discos donde se ubica InnoDB. En las últimas versiones de MySQL (> 5.0.21) el valor por defecto es 8. Un valor 0 significaría que "abra tantos hilos como le sea posible". Por lo tanto, si hay dudas, se puede usar:

```
innodb_thread_concurrency = 0
```

Diferentes personas [3) 4)] han hecho pruebas, y han detectado problemas con el rendimiento en servidores con muchas CPU físicas cuando se usa un número muy alto, con versiones de MySQL relativamente antiguas (del año 2008).

Usar un espacio de tablas para cada tabla

(Tomado del manual de MySQL en http://dev.mysql.com/doc/refman/5.0/es/multiple-tablespaces.html)

En MySQL 5.0, se puede almacenar cada tabla InnoDB y sus índices en su propio fichero. Esta característica se llama "multiple tablespaces" (espacios de tablas múltiples) porque, en efecto, cada tabla tiene su propio espacio de tablas.

El uso de múltiples espacios de tablas puede ser beneficioso para usuarios que desean mover tablas específicas a discos físicos separados o quienes deseen restaurar respaldos de tablas sin interrumpir el uso de las demás tablas InnoDB.

Se pueden habilitar múltiples espacios de tablas agregando esta línea a la sección mysqld de my.cnf:

```
[mysqld]
innodb_file_per_table
```

Después de reiniciar el servidor, InnoDB almacenará cada nueva tabla creada en su propio fichero nombre_tabla.ibd en el directorio de la base de datos a la que pertenece la tabla. Esto es similar a lo que hace el motor de almacenamiento MyISAM, pero MyISAM divide la tabla en un fichero de datos tbl_name.MYD y el fichero de índice tbl_name.MYI. Para InnoDB, los datos y los índices se almacenan juntos en el fichero .ibd. El fichero tbl_name.frm se sigue creando como es usual.

Si se quita la línea innodb_file_per_table de my.cnf y se reinicia el servidor, InnoDB creará nuevamente las tablas dentro de los ficheros del espacio de tablas compartido.

innodb_file_per_table afecta solamente la creación de tablas. Si se inicia el servidor con esta opción, las tablas nuevas se crearán empleando ficheros .ibd, pero aún se puede acceder a las tablas existentes en el espacio de tablas compartido. Si se remueve la opción, las nuevas tablas se crearán en el espacio compartido, pero aún se podrá acceder a las tablas creadas en espacios de tablas múltiples.

Fragmentación MySQL

Al igual que los filesystems, las bases de datos tambien se fragmentan, haciendo que todo el sistema pierda rendimiento. En un sistema de alto rendimiento, como Pandora FMS es vital que la salud de la BBDD no afecte al funcionamiento del sistema. En sistemas sobrecargados, al final la BBDD puede bloquearse, resultando una caída de todo el sistema.

Una buena configuración de MySQL podría hacer que Pandora FMS trabajase mas rápido. Si tiene problemas de rendimiento probablemente será porque no tiene MySQL correctamente configurado o por algún problema relacionado con la base de datos.

Comprobación del fichero my.ini/cnf

Empezaremos con el fichero my.cnf y su configuración básica para MySQL. La configuracón de su my.cnf debería ser similar a esta(4GB RAM y usando un hardware con una configuración media). Compruebe que tiene todos estos parámetros correctamente introducidos en la sección [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/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
```

Cualquier cambio que realices en el fichero my.cnf necesitará reiniciar MySQL. Comprueba al final del fichero /var/log/mysqld.log para ver si ha ocurrido algún error.

Reconstruir bases de datos

Uno de los “problemas” más conocidos cuando modificamos el fichero my.cnf es configurar los nuevos valores para los registros de transacciones. Por lo que, si le aparece este error:

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

Restaure la configuración previa, construya un backup de su base de datos y siga los siguientes pasos:

1. Una vez creado el backup, detenga el servicio MySQL

```
systemctl stop mysql
```

2. Vaya a la carpeta previa donde se encuentran los ficheros de datos de MySQL (datadir) (por defecto /var/lib/mysql)

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

3. Mueva la carpeta a otra ubicación (/var/lib/mysql → /var/lib/mysql_backup)

```
mv mysql mysql_old
```

4. Cree una nueva carpeta (/var/lib/mysql)

```
mkdir mysql
```

5. Asigne el propietario de la carpeta

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

6. Inicialice la carpeta con los datos de MySQL

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

7. Inicie el servicio

```
systemctl start mysql
```

8. Lance el configurador y siga el asistente

```
mysql_secure_installation
```

9. Reconstruya sus bases de datos

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

10. Asigne los permisos a los usuarios correctos

```
 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. Carge los backups

```
mysql> source /path/to/your/backup.sql
```

Muchas veces los sistemas con MySQL/Percona no cargan correctamente los parámetros de configuración del fichero my.cnf (normalmente porque estos valores han sido metidos fuera de la sección de [mysqld]

Después de haber configurado el fichero my.cnf y reiniciado, deberá comprobar que estos cambios han sido correctamente aplicados. Para hacer esto usaremos el comando SHOW VARIABLES:

```
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
|
+-----------------------------------------+-----------------------
+
```

Comprobación de que los ficheros de datos aislados para cada tabla están ACTIVADOS

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

Debería tener mas de 100 ficheros (dependiendo de la versión de Pandora FMS). Cada ".ibd" será el fichero de datos para cada tabla, cuando tiene activado el parámetro "innodb_file_per_table" en el fichero my.cnf. Si no tuviese ninguno de estos ficheros ".ibd" significará que utiliza un único archivo para almacenar toda la información. Esto significará que la fragmentación de las tablas es común al resto de tablas, lo que podría indicar, que el rendimiento será peor cada semana que pase.

Si tiene su base de datos corriendo bajo una única base de datos, necesitará en primer lugar re-crear la base de datos después de haber configurado correctamente el fichero my.cnf y reiniciar MySQL.

Comprobación de la fragmentación tabla por tabla

Usando el CLI de MySQL, deberá ejecutar esta consulta:

```
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;
```

Debería ver solamente las tablas con algún indice de fragmentación, por ejemplo:

```
+--------+-------------------------+-------------+--------------+--
---------+------------+
| 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 | 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 |
+--------+-------------------------+-------------+--------------+--
---------+------------+
```

Esta consulta funcionará solamente en las tablas que posean mas de un 10% de fragmentación.

Las tablas demasiado grandes (como tagente_datos) podrían tardar demasiado tiempo en optimizarse si están muy fragmentadas. Esto podría causar algún tipo de impacto en los sistemas de producción. Por lo que recomendamos no opmitizar este tipo de tablas tan grandes, ya que podría causar un bloqueo del sistema (el proceso de optimización “bloquea” la tabla para reescribirla)

Para optimizar la tabla “tagent_module_inventory”:

```
optimize table  tagent_module_inventory;
```

Aparecerá un mensaje diciéndonos:

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

Ahora, si comprueba de nuevo la fragmentación de las tablas, puede ver que la fragmentación ha desaparecido:

```
+--------+------------------------+-------------+--------------+--
---------+------------+
| 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 |
+--------+-------------------------+-------------+--------------+--
---------+------------+
```

Para poder realizar esta optimización será necesario tener el espacio necesario en el disco duro para realizar la operación. En caso contrario saldrá un error y no se realizará la operación

Carga del sistema

Ésto es de ámbito general, pero deberemos estar seguros de que el sistema de E/S no es un cuello de botella (discos). Necesitaremos ejecutar el comando "vmstat" para recoger información del sistema:

```
vmstat 1 10
```

Ahora, nos fijaremos en las últimas columnas (CPU WA), un valor por encima de 10 significará que tu disco de E/S tiene un problema que deberá ser arreglado. Tener unos valores de CPU US es normal, pero CPU SY no deberá ser mayor de 10-15. En condiciones normales debería tener SWAP si/so a 0, si no, significaría que su sistema esta usando memoria SWAP, considerado como un destructor de rendimiento. Necesitarás incrementar la memoria RAM o decrementar la memoria RAM usada en sus aplicaciones (hilos de Pandora FMS, buffers de MySQL, etc).

Le mostramos la salida de ejemplo de un sistema "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
```

Particionado de tablas MySQL

Para usar Particionado de tablas MySQL, debería usar también el sistema de "multiples tablespaces" descrito arriba (innodb_file_per_table).

MySQL 5.1 soporta particionado de tablas, lo que permite distribuir tablas muy grandes en trozos más pequeños, como subdivisiones lógicas. (Puede consultar el manual de MySQL para más detalles: http://dev.mysql.com/doc/refman/5.1/en/partitioning-overview.html)

Si tiene grandes cantidades de datos en la base de datos de su Pandora FMS y cree que muchas operaciones de la consola que se refieren a esos datos (por ejemplo drawing graph) son bastante lentas, mejorará su rendimiento utilizando particionado de tablas.

Comprobar en un primer momento que el directorio /var/lib/mysql/pandora_history/*.ibd tiene muchos ficheros (uno por tabla), si no, necesitará hacer un dump de la base de datos, cambiar la configuración del fichero my.cnf, reiniciar MySQL, borrar la base de

datos actual y re-crearla desde el dump.

Una vez que se haya asegurado que innodb_file_per_table está activado, separaremos las dos bases de datos principales en diferentes particiones. Éste es un ejemplo para hacer la partición de todo el 2015 hasta ahora y para futuros meses.

Esta operación necesitará espacio suficiente en el disco para se completada. Habrá que comprobar como es de grande “tagente_datos.ibd”. Si por ejemplo ocupa 10gb, necesitará 15gb de espacio libre para empezar la operación.

Esta operación puede llevarle mucho tiempo, dependiendo del tamaño de la tabla. Por ejemplo, llevaría una hora y media partir una tabla con aproximadamente 7500 modules data para 100 dias (más de 50,000,000 filas).

Para comenzar el procesó deberá ejecutar la siguiente consulta en el CLI de 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)
);
```

Luego habría que ejecutar cada mes la siguiente consulta para reorganizar el particionamiento:

```
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);
```

Cambiando “Feb16” por el mes en el que se encuentre.

Recuerde que esta operación podría tardar horas, dependiendo de lo grande que sea la tabla “tagente_datos”. Puede ir comprobando el proceso viendo el tamaño de los archivos de particionamiento ejecutando:

```
[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
```

Reconstruccion de la BBDD

Reconstruccion parcial

El sistema de gestión de base de datos de MySQL, al igual que otros motores SQL, como Oracle ™, se degrada con el tiempo por causas como la fragmentación de datos producida por el borrado y la inserción continuada en grandes tablas. En grandes entornos con mucho volumen de tráfico existe una forma muy sencilla de mejorar el rendimiento e impedir que el rendimiento se degrade: reconstruir la BBDD de forma periódica.

Para ello hay que programar una parada de servicio, que puede durar aprox. 1 hr.

En esta parada de servicio, lo que hay que hacer es parar la consola WEB de Pandora FMS, y el servidor (ojo, dejar el servidor Tentacle para que siga recibiendo datos, y esos se procesarán tan pronto como el servidor esté operativo de nuevo).

Una vez parados, hacemos un volcado de la BBDD (Export)

```
mysqldump -u root -p pandora3 > /tmp/pandora3.sql
Enter password:
```

Borramos la BBDD:

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

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

Creamos la BBDD y hacemos un import del export anterior:

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

Esto puede tardar unos cuantos minutos,dependiendo de si el sistema es grande y el hardware no demasiado portente, para un sistema con 1500 agentes y aproximádamente 100.000 módulos. Se puede automatizar este proceso, pero por su naturaleza delicada, es mejor hacerlo manualmente.

Reconstrucción total

Este capítulo solo afecta a bases de datos Innodb. Pandora FMS está construido sobre bases de datos Innodb.

Desgraciadamente MySQL se degrada con el tiempo, cosa que afecta a todo el rendimiento del sistema. No existe otra solución que no pase por reconstruir todos los esquemas de base de datos desde 0, reconstruyendo el fichero binario de datos que MySQL usa para almacenar toda la información y los ficheros usados para reconstruir las

transacciones.

Si nos fijamos en el directorio /var/lib/mysql se puede ver que hay tres ficheros, que siempre se llaman igual y son, en función de lo grave del caso, gigantes. En mi caso de ejemplo:

```
-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
```

El fichero ibdata1 es el que alberga todos los datos Innodb del sistema. En un sistema muy fragmentado, que lleva mucho tiempo sin "rehacer" o sin "reinstalar" este sistema será grande y poco eficiente. El parámetro **innodb_file_per_table** del que hablamos antes, regula parte de este comportamiento.

Así mismo, cada base de datos tiene dentro del directorio /var/lib/mysql un directorio para definir su estructura. Deberá borrarlos también.

El procedimiento es simple:

1. Volcar (via mysqldump) todos los esquemas a disco:

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

2. Parar MySQL.
3. Borrar ibdata1, ib_logfile0, ib_logfile1 y los directorios de bases de datos InnoDB
4. Levantar MySQL.
5. Crear la base de datos de pandora de nuevo (create database pandora;)
6. Importar el fichero de backup (all.sql)

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

El sistema deberia ir ostensiblemente más rápido ahora.

Indices opcionales

En algunas situaciones es posible optimizar el funcionamiento de MySQL a costa de recursos del sistema.

Este índice, sirve para optimizar la velocidad de obtención de gráficas a costa de un mayor uso de disco y un ligero descenso en el rendimiento de borrados / inserciones en las tablas de datos:

```
ALTER TABLE `pandora`.`tagente_datos`  ADD  INDEX
```

```
`id_agente_modulo_utimestamp`  (  `id_agente_modulo`  ,
`utimestamp`  );
```

Actualmente en las tablas más pesadas de Pandora FMS en MySQL viene dicha optimización por defecto. Es conveniente preguntar a expertos antes de optimizar las tablas de MySQL

Queries lentas

En algunos sistemas, en función del tipo de información que tengamos, podemos encontrarnos con algunas “queries lentas” que hacen que el sistema vaya peor. Podemos activar el log de este tipo de consultas durante un periodo CORTO de tiempo (ya que perjudica al rendimiento del sistema) a fin de estudiar las consultas a intentar optimizar las tablas con índices. Para activar este sistema hay que hacer lo siguiente:

Editar my.cnf y añadir las siguientes lineas:

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

Para poder utilizarlo:

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

Reiniciar mysql.

Optimización de tablas específicas

Otra solución menos “radical” para paliar el problema de la fragmentación es el uso de la herramienta OPTIMIZE de MYSQL para optimizar ciertas tablas de Pandora FMS. Para ello directamente desde MySQL, ejecutar:

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

```
OPTIMIZE table tagente_modulo;
OPTIMIZE table tagente_estado;
```

Eso mejora el rendimiento y no debería ser necesario lanzarlo mas que una vez a la semana, pudiendo hacerse “EN CALIENTE” mientras el sistema trabaja. En sistemas muy grandes el OPTIMIZE puede quedarse “bloqueado” y no ser una alternativa, para eso es mejor reconstruir la BBDD.

Después de hacer estas operaciones, conviene ejecutar:

```
FLUSH TABLES;
```

Del manual 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.

(Para las tablas de tipo InnoDB, OPTIMIZE TABLE se asigna a ALTER TABLE, que reconstruye la tabla para actualizar las estadísticas de índice y espacio libre no utilizado en el índice agrupado.)

En instalaciones a partir de 7.0 OUM715 la siguiente configuración viene aplicada por defecto

Nota: Si su instalación de Pandora FMS se realizó antes de la versión 7.0 OUM 715, recomendamos que realice las siguientes modificaciones en sus base de datos (principal e histórico):

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

Esta acción agregará un índice a la tabla tagente_datos, permitiendo que las consultas que utilizan tanto el sistema de informes, consulta de datos o gráficas de módulo o personalizadas, devuelvan resultados en un tiempo sensiblemente menor al anterior.

Tokens especiales de MySQL

Existen algunos tokens muy “especiales” de mySQL, que pueden ayudar o empeorar el rendimiento:

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

Este parametro (innodb_thread_concurrency) en versión 5.1.12 o superiores, si vale 0 significa que no hay límite a la concurrencia, en versiones inferiores a la señalada, si vale 20 o más, implica lo mismo: sin límite a la concurrencia.

```
innodb_flush_method = O_DIRECT
```

Este parámetro afecta a cómo se escribe en disco.

```
innodb_lock_wait_timeout = 90
```

Evita que ante un atasco ocasional el MySQL “se rinda” (MySQL has gone away) y se pare. Si pasa de 90 segundos, es que es un gran problema.

Referencias

Referencias:

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

MySQL Percona XTraDB

Percona es una versión mejorada de MySQL, sobre todo respecto a la escalabilidad. Aprovecha mejor los sistemas de múltiples CPU's y acelera el acceso a disco.

Para configurar su MySQL percona, utilice este excelente wizard online, que generará su fichero */etc/my.cnf*: Percona Wizard Configurator

Dimensionamiento Pandora FMS para alta capacidad

Esta sección describe diferentes métodos para configurar Pandora FMS en un entorno de alta capacidad. Así mismo describe diferentes herramientas para hacer pruebas de carga, útiles para ajustar el entorno a la mayor capacidad de proceso posible.

Se ha configurado Pandora FMS para soportar una carga de alrededor de 2500 agentes en sistemas donde base de datos, consola y servidor están en la misma máquina. La cifra recomendada está en torno a 2500 agentes por sistema, pero esta cifra varía enormemente en funcion de si son agentes XML, módulos remotos, con intervalos altos o bajos, o con sistemas de mucha capacidad o poca memoria, todos factores alteran enormemente el nº de agentes que un sistema puede gestionar eficientemente. En pruebas de laboratorio se han logrado ejecutar 10000 agentes en un solo servidor con

hardware básico, pero fuertemente optimizado.

Ejemplo de configuración de servidores de alta capacidad

Suponiendo una máquina CentOS con 16GB de RAM y 8 CPUs que quisieramos optimizar para la máxima capacidad de procesamiento del data server (XML).

my.cnf

(Solo se muestran los parámetros más significativos)

```
[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 = 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
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/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
```

pandora_server.conf

(Se muestran solo los parámetros relevantes)

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

Cosas a tener en cuenta:

- El número de *verbose* hace referencia a la cantidad de información con se escribe en los logs, siendo recomendable no sobrepasar de 10. Cuanto mayor sea el número, menos será el rendimiento de Pandora FMS debido a la gran cantidad de información a escribir en los logs.
- El nº de hilos muy alto (+5) solo beneficia a procesos con largas esperas de E/S, como el network o el plugin server, en el caso del dataserver que es todo tiempo de proceso, puede incluso penalizar el rendimiento, por eso usamos 5 aquí. En sistemas con una BD lenta, usaríamos incluso menos hilos), pruebe diferentes combinaciones entre 1 y 10. En el caso de optimizar el sistema para el networkserver, el nº sería mucho más alto, entre 10 y 30.
- El server threshold alto (15) hace que la BD se resienta menos, mientras que el incremento en el máximo número de ficheros procesados hace que cada vez que el servidor “busque ficheros” llene los buffers. Estos dos elementos de la configuración están íntimamente ligados. En el caso de optimizar el network server sería recomendable bajar el server threshold a 5 o a 10.
- Algunos parámetros de la configuracion pueden afectar mucho al rendimiento de Pandora FMS, como el parámetro agent_access (configurable desde la consola).

Herramientas de análisis de capacidad (Capacity)

Pandora FMS dispone de varias herramientas que le ayudarán a dimensionar adecuadamente su hardware y software para el volumen de datos que espera obtener. Una de ellas sirve para “atacar” directamente la base de datos con datos ficticios

(dbstress) y la otra genera ficheros XML ficticios (xml_stress)

Pandora FMS XML Stress

Este es un pequeño script que genera ficheros de datos XML como los enviados por los agentes de Pandora FMS. Está en /usr/share/pandora_server/util/pandora_xml_stress.pl

Los scripts leen los nombres de los agentes desde un fichero de texto y generan ficheros de datos XML para cada agente acorde con fichero de configuración, donde los módulos están definidos como plantillas.

Los módulos se rellenan con datos al azar. Se pueden especificar el valor inicial y la probabilidad de cambio de los datos de un módulo.

Ejecuta el script de este modo:

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

Ejemplo de configuración de un fichero:

```
# 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_descripcion Module 3 description.
# Initial data.
module_data 1
module_end
```

Enviar y recibir la configuración local del agente

Activando en su "pandora_xml_stress.conf" el valor de configuración "get_and_send_agent_conf" a 1, puede hacer que los agentes de prueba de carga se comporten como agentes normales, ya que envían su fichero de configuración y el md5. Y desde Pandora Console Enterprise puede modificar la configuración remota para que en sucesivas ejecuciones del pandora_xml_stress use la configuración personalizada desde Pandora Console Enterprise en vez de a través de la definición de "pandora_xml_stress.conf".

A parte de eso puede configurar donde guardar de forma local los conf de tus agentes de prueba con el token de configuración "directory_confs" en el fichero "pandora_xml_stress.conf".

Fichero de configuración

- **max_threads**: Número de threads en la que se ejecutará el script, esto mejora la E/S.
- **agent_file**: Ruta del fichero de lista de nombres, separados por nueva linea.
- **temporal**: Ruta del directorio donde se generaran los ficheros de datos XML ficticios.
- **log_file**: Ruta del fichero de log donde informara el script de su ejecución.
- **xml_version**: Versión del ficheros de datos XML (por defecto 1.0).
- **encoding** Codificación de los ficheros de datos XML (por defecto ISO-8859-1).

- **os_name**: Nombre del sistema operativo de los agentes ficticios (por defecto Linux).
- **os_version**: Versión del sistema operativo de los agentes ficticios (por defecto 2.6).
- **agent_interval**: Intervalo de los agentes ficticios en segundos (por defecto 300).
- **time_from**: Fecha de comienzo a generar los ficheros de datos XML ficticios, en formato "AÑO-MES-DIA HORA:MIN:SEC"
- **time_to**: Fecha de fin a generar los ficheros de datos XML ficticios, en formato "AÑO-MES-DIA HORA:MIN:SEC"
- **get_and_send_agent_conf**: Valor booleano 0 o 1, cuando esta activo los agentes ficticios intentaran descargar por configuración remota una versión mas actual del fichero configuración estandar de un agente. Y desde la consola Pandora FMS enterprise puedes editarlos.
- **startup_delay**: Valor numérico de tiempo en segundos antes de comience cada agente a generar los ficheros, se usa para evitar condiciones de carrera.
- **timezone_offset**: Valor numérico del offset de time zone.
- **timezone_offset_range**: Valor numérico que sirve para generar dentro de este rango los timezone de forma aleatoria.
- **latitude_base**: Valor numérico, es la latitud donde apareceran los agentes ficticios.
- **longitude_base**: Valor numérico, es la longitud donde apareceran los agentes ficticios.
- **altitude_base**: Valor numérico, es la altitud donde apareceran los agentes ficticios.
- **position_radius**: Valor numérico, es el rango alrededor, la circunferencia con este radio en que aparece el agente ficticio de forma aleatoria.

Definición de los módulos

La definición de un módulo dentro del fichero de configuración script y si tiene activada la configuración remota también será igual:

```
 module_begin
 module_name <nombre_del_módulo>
 module_type <tipo_de_dato_módulo>
 module_description <descripción>
 module_exec type
=<tipo_generación_xml_stress>;<otras_opciones_separadas_por_punto_y
_coma>
 module_unit <unidades>
 module_min_critical <value>
 module_max_critical <value>
 module_min_warning <value>
 module_max_warning <value>
 module_end
```

Y cada uno lo puede configurar como:

- **tipo_generación_xml_stress**: Puede tomar los valores RANDOM , SCATTER, CURVE.

- **module_attenuation <value>**: El valor generado se multiplica por el valor dado, normalmente entre 0.1 y 0.9.
- **module_attenuation_wdays <value> <value> ... <value>**: El valor del módulo se atenúa únicamente los días dados, que van de domingo (0) a sábado (6). Por ejemplo, el siguiente módulo simula una disminución del 50% en el tráfico de red los sábados y domingos:

```
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 se configura a uno, el valor previo del módulo se suma siempre a un nuevo valor, lo que da lugar a una función creciente.
- **otras**: Ver más abajo que opciones tiene, en función del tipo de generación del módulo.

Note que los tokens de configuración *min/max_critical* y *min/max_warning* están solo disponibles en la versión 5.0 o superior.

Aleatorios (RANDOM)

Los cuales tienen las siguientes opciones:

- **variation**: Probabilidad en % de que varie con respecto al valor anterior.
- **min**: Valor mínimo que puede tener el valor.
- **max**: Valor máximo que puede tener el valor.

Numéricos

Generada valores numéricos aleatorios entre el rango valor **min** y el valor **max**.

Boleanos

Generada valores entre 0 y 1.

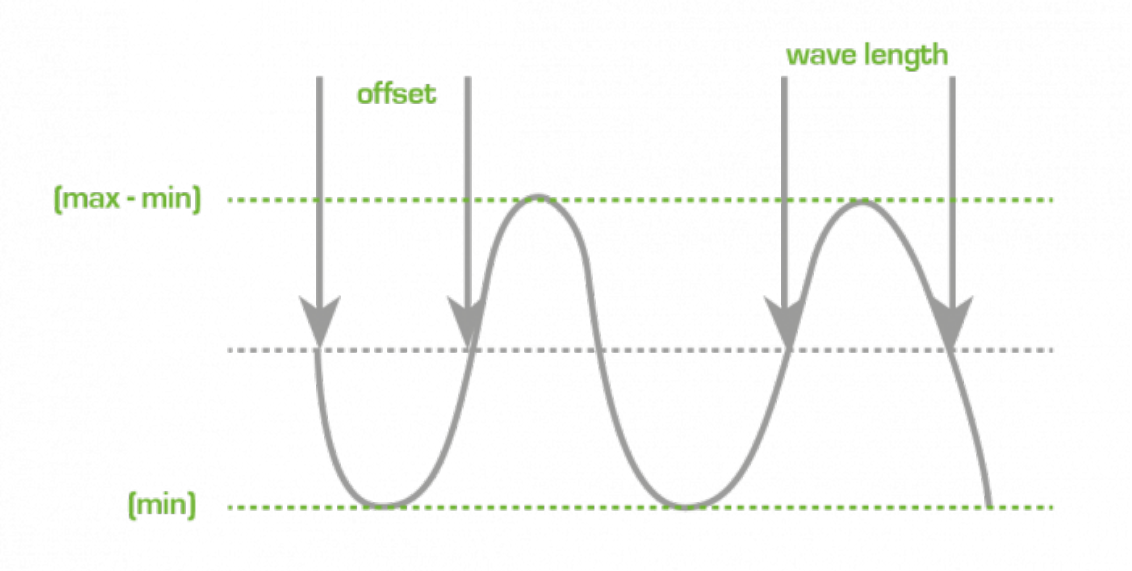
Cadena

Generada una cadena de longitud entre valor **min** y el valor **max**, los caracteres son aleatorios entre A y Z incluidas mayúsculas y minúsculas y cifras numéricas.

Fuente externa de datos (SOURCE)

Permite escoger un archivo de texto plano como fuente de datos. Opciones:

- src**: archivo fuente para los datos. El archivo contiene un dato por línea, no hay límite de líneas. Por ejemplo: <code> 4 5 6 10 </code> Admite todo tipo de valores (numéricos y cadenas de texto). Este tipo de módulos usará cada unos de los datos del archivo para generar los datos de los módulos en Pandora, los datos se cogen de forma secuencial. Por ejemplo, los datos del módulo anterior se verían en Pandora así: 4 5 6 10 4 5 6 10 4 5 6 10 4 5 6 10 4 5 6 10** Dispersión (SCATTER) **Solo vale para datos numéricos, y la gráficas generadas son parecidas a las de un latido de corazón, es decir un valor normal y de vez en cuando un *"latido"*. Y tiene las siguientes opciones:** * min**: Valor mínimo que puede tener el valor.** * max**: Valor máximo que puede tener el valor.** * prob**: Probabilidad en % de que genere un *"latido"*.** * avg**: Valor medio que debe mostrar por defecto si no hay ningún *"latido"*. == Curva (CURVE) == Genera datos de módulo siguiendo una curva trigonométrica. Los cuales tienen las siguientes opciones:** * min**: Valor mínimo que puede tener el valor.** * max**: Valor máximo que puede tener el valor.** * time_wave_length**: Valor numérico en segundos de la duración de la *"cresta"* de la onda.** * time_offset**: Valor numérico en segundos de comienzo de la onda desde el tiempo cero con valor cero del módulo (similar a la gráfica de seno).**



Notas de interes * **Tenga en cuenta que la cantidad de ficheros generados es la relación entre la fecha de comienzo (time_from) y fecha de final (time_to) y el intervalo seteado en el agente (agent_interval), por lo que a grandes tramos de tiempo o pequeño intervalo el script va a generar muchos ficheros de datos XML. == Como medir la capacidad de proceso del dataserver == Existe un pequeño script llamado "pandora_count.sh" que está en el directorio /útil en el directorio del servidor de Pandora FMS. Este script se usa para medir la tasa de procesamiento de ficheros XML por el data server, y utiliza como referencia el total de ficheros pendientes de procesar en /var/spool/pandora/data_in de forma que para poder usarlo se**

necesita tener pendiente de procesar varios miles de paquetes (o generarlos con la herramienta anteriormente mencionada). Este script simplemente cuenta los paquetes existentes ahora y los resta de los paquetes que habia hace 10 segundos, luego divide el resultado por 10 y esos son los fiheros que se han procesado en los ultimos 10 segundos, mostrando la tasa por segundo. Es una medida algo burda pero sirve para ajustar la configuracion del servidor. == Pandora FMS DB Stress == Esta es una pequeña herramienta para probar el rendimiento de su base de datos. También se puede usar para «pregenerar» datos periódicos o aleatorios (usando funciones trigonométricas) y rellenar módulos ficticios. Se debe crear un agente y asignarle módulos para inyección de datos automática con esta herramienta. Los nombres se deben llamar con la notación siguiente: * *random*: para generar datos aleatorios. * *curve*: para generar una curva de coincidencias usando funciones trigonométricas. Útil para ver el trabajo de interpolación con diferentes intervalos, etc. * *boolean*: generar datos booleanos aleatorios. De tal forma que se podría usar cualquier nombre que contenga las palabras «*random*», «*curve*» y/o «*boolean*», por ejemplo: * random_1 * curve_other Sólo se podrá elegir el tipo de módulo **«data_server».** Ajuste fino de la herramienta *Pandora FMS DB Stress***

Esta herramienta está preconfigurada para buscar, en todos los agentes, los módulos de nombre «*random*», «*curve*» o «*boolean*», y que usen un intervalo entre 300 segundos y 30 días.

Si se quiere modificar este comportamiento, se debe editar el *script pandora_dbstress* y modificar algunas variables al inicio del fichero:

```
# 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 primera línea de variable correspondiente con *target_module*, se debe establecer para un módulo fijo o -1 para procesar todos los objetivos que coincidan. La segunda línea de variable corresponde con *target_agent*, para un agente específico. La tercera línea corresponde con *target_interval*, definida en segundos y que representa el intervalo de periodicidad predeterminada del módulo. La cuarta línea es *target_days* y representa el número de días en el pasado desde la fecha, en *timestamp*, actual.

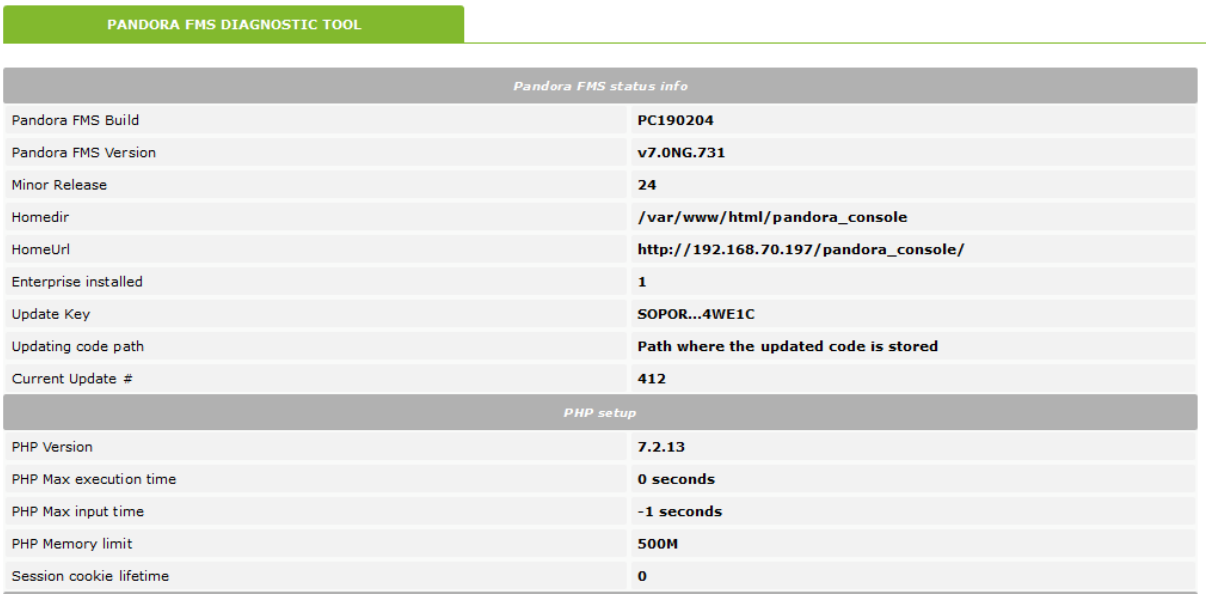
Solución de problemas y herramientas de Diagnóstico en Pandora FMS

A veces, el usuario tiene problemas y los desarrolladores de Pandora FMS no le pueden ayudar sin tener más información acerca de sus sistemas. En la versión 3.0, hemos desarrollado algunas herramientas con el fin de ayudarle a resolver algunos problemas:

Diagnostic Info

En versiones más actuales de Pandora FMS se encuentra una funcionalidad para poder obtener información de diagnóstico de nuestra instalación de Pandora FMS.

Se encuentra dentro de la sección de *Admin tools* → *Diagnostic Info*



PANDORA FMS DIAGNOSTIC TOOL

Pandora FMS status info	
Pandora FMS Build	PC190204
Pandora FMS Version	v7.0NG.731
Minor Release	24
Homedir	/var/www/html/pandora_console
HomeUrl	http://192.168.70.197/pandora_console/
Enterprise installed	1
Update Key	SOPOR...4WE1C
Updating code path	Path where the updated code is stored
Current Update #	412
PHP setup	
PHP Version	7.2.13
PHP Max execution time	0 seconds
PHP Max input time	-1 seconds
PHP Memory limit	500M
Session cookie lifetime	0

En esta ventana podemos observar información acerca de la configuración de Pandora FMS y MySQL, así como gráficas del sistema de auto-monitorización.

pandora_diagnostic.sh

Es una herramienta localizada en */usr/share/pandora_server/util* y proporciona mucha información acerca del sistema:

- Información de la CPU
- Uptime y avgload de la CPU
- Información sobre la memoria
- Información sobre Kernel/liberación
- Un fichero dump de configuración mysql
- Un fichero dump (filtrando contraseñas) de configuración del servidor de Pandora FMS

- Los logs de información de Pandora FMS (pero no el log completo!)
- Información del disco
- Información sobre procesos de Pandora FMS
- Una información completa del log del kernel(dmesg).

Toda la información se genera en un fichero .txt, con lo que los usuarios pueden enviar esta información a cualquiera que quiera ayudarles, por ejemplo, en el foro de usuarios de Pandora FMS o en las listas de correo públicas de Pandora FMS. Esta información no deberá contener ningún tipo de información confidencial. Tenga en cuenta que posiblemente usted quiera correr con privilegios de root si quiere obtener los ficheros*pandora_server.conf* y*my.cnf* parseados

Este es un ejemplo de ejecución:

```
$ ./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'
```

Y estas son algunas partes de la salida del fichero

```
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
```

Volver al Índice de Documentación Pandora FMS

1) http://www.percona.com/software/percona-server/

2) http://dev.mysql.com/doc/refman/5.0/en/innodb-parameters.html#sysvar_innodb_thread_concurrency

3)

http://optimmysql.blogspot.com/2008/04/variable-day-out-5-innodbthreadconcurre.html
4)
http://mysqlha.blogspot.com/2010/03/do-we-still-need-innodbthreadconcurrenc.html