



Optimización y solución de problemas



<https://pandorafms.com/manual/!779/>

Permanent link:

https://pandorafms.com/manual/!779/es/documentation/pandorafms/complex_environments_and_optimization/08_optimization
2022/01/22 19:13



Optimización y solución de problemas

El servidor de Pandora FMS es capaz de monitorizar unos 2000 dispositivos (entre 5 mil y 80 mil módulos, [dependiendo del hardware disponible](#)); para ello es necesario, además, *afinar la configuración de la base de datos* (BBDD).

En este capítulo de manera adicional se explican algunas técnicas para detectar y solucionar problemas en la instalación de Pandora FMS.

Optimización Percona

Todas las pruebas y validaciones se realizan con Percona Server for MySQL® 8 (opción recomendada). Debido a las similitudes entre Percona Server for MySQL 8 y MySQL® 8 existe una gran compatibilidad entre ambas soluciones. Para efectos prácticos de este tema se utiliza el término MySQL a secas, siempre teniendo en cuenta que las pruebas se realizan sobre Percona Server for MySQL.

Para conocer más acerca del “Respaldo y recuperación de datos en Pandora FMS”, [diríjase a este enlace](#).

Consejos generales

- A menos que se especifique otra cosa, todo este tema se refiere a MySQL versión 8 .
- Véase también [“Actualización de MySQL 5.7 a MySQL 8”](#).

Para trabajar con tablas mayores de 2 GB es necesario seguir algunas pautas:

- MySQL recomienda usar un sistema de 64 bits. Sistemas de 32 bits podrán tener serios problemas a partir del año 2038 debido al desbordamiento de las variables de fecha.
- A más memoria RAM y más CPU, mejor rendimiento. Acorde con nuestra experiencia, la memoria RAM es más importante que la CPU. El mínimo para un sistema a nivel de empresa será 4 GB. Una buena opción para un gran sistema son 16 GB. 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 otro servidor dedicado para la base de datos utilice Ethernet Gigabit, preferiblemente con fibra óptica en vez de cobre. 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, la base de datos y, (en su caso) los *logs* de replicación binaria.

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

- Si el cliente y el servidor MySQL están en la misma máquina, utilice *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 tiene solo un servidor anfitrión MySQL.
- Pandora FMS funciona sobre MySQL y precisamente se recomienda el uso de la versión modificada de MySQL (**Percona Server for MySQL**), la cual ofrece un mayor rendimiento. Los *plugins* programados son para Percona®.

Tenga en cuenta que los siguientes puntos afectan mucho al rendimiento:

- Solamente utilice *logs* binarios si utiliza una configuración de MySQL con replicación.
- No utilice *logs de trazabilidad de queries* o *slow query logs*. Esto solamente se recomienda **en casos puntuales**.

Desactivar replicación binaria

Si tiene configurado un **sistema HA de Pandora FMS** la replicación binaria es necesaria. Esta recomendación solamente es válida si tiene un servidor único de Pandora FMS.

Por defecto viene habilitada en la mayoría de distros de GNU/Linux. Para desactivarla, edite el fichero `my.cnf`, habitualmente en `/etc/my.cnf` y *comente* las siguientes líneas:

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

Hay que *comentar* las dos líneas, y luego reiniciar el servidor MySQL.

Rendimiento de acceso a disco

Para conocer más acerca del “Respaldo y recuperación de datos en Pandora FMS”, [diríjase a este enlace](#).

Junto con otros parámetros clave, hay tres 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

```
innodb_log_file_size = 64M
```

Por defecto se establece este valor, que puede ser mayor y llegar incluso a 512 Megabytes sin perjuicio alguno (*excepto para recuperación en caso de problema ya que tiene 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:

1. Primero se deberá hacer un DUMP completo de las bases de datos.
2. Borrar los ficheros índices binarios de InnoDB (generalmente en `/var/lib/mysql/ib*`).
3. Cambiar el fichero `my.cnf` con el valor escogido.
4. Reiniciar MySQL.
5. 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), se recomienda hacer todo el proceso simultáneamente.

innodb_io_capacity

```
innodb_io_capacity = 300
```

Por defecto, este parámetro tiene el valor 200, pero debe conocer previamente los IOPS del disco del sistema. Se puede conocer exactamente buscando IOPS y el modelo exacto del disco duro, donde los valores recomendados son: 7500RPM → 100 IOPS, 15000 RPM → 190 IOPS, SSD → 1500 IOPS. Más información en [este enlace](#).

innodb_file_per_table

Usar un espacio de tablas para cada tabla:

En Percona 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` del fichero `my.cnf`:

```
[mysqld]
innodb_file_per_table
```

Después de reiniciar el servidor, InnoDB almacenará cada nueva tabla creada en su propio fichero `name_table.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.

El parámetro `innodb_file_per_table` afecta solamente a 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 elimina 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.

Evitando escritura con cada transacción

MySQL por defecto establece `autocommit = 1` para cada conexión. Esto es inocuo para MyISAM, ya que lo que se escribe no está garantizado en el disco; para InnoDB significa que cada `insert / update / delete` en una tabla InnoDB se traducirá en una escritura en el disco (*flush*).

¿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 antes de la confirmación de la escritura, esto toma su tiempo. Suponiendo incluso un tiempo medio de búsqueda de 9 milisegundos por la escritura en disco, esto limita a aproximadamente *67 commits* por segundo, 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*. Se puede 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 se pierda los datos del último segundo, algo más que asumible si se trata de ganar eficiencia. Para ello, se usa el siguiente 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 las instrucciones DELETE e INSERT.

```
key_buffer_size = 4M
```

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

Otros parámetros importantes

Hay varios *buffer* que por defecto, en algunas distribuciones, faltan o están *comentados*. Modificar y/o agregar estos parámetros puede dar un rendimiento muy superior al que se obtiene por defecto. Es importante asegurarse de que existen estos *tokens*, con los siguientes valores, en el fichero de configuración de MySQL:

```
key_buffer_size=4M
read_buffer_size=128K
read_rnd_buffer_size=128K
sort_buffer_size=1M
join_buffer_size=4M
```

Para MySQL versión 8, y versiones posteriores, el equipo de desarrollo de MySQL ha retirado el soporte para *query cache* utilizado en versiones anteriores de PFMS; si desea obtener más información puede visitar el siguiente enlace web (en inglés):

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

Mejorando la concurrencia de InnoDB

Existe un parámetro que puede afectar bastante al rendimiento del servidor MySQL con Pandora FMS. Este parámetro es `innodb_thread_concurrency`. Dicho parámetro sirve para especificar cuántos "*hilos concurrentes*" puede ejecutar MySQL.

Este un parámetro avanzado y solamente se debe modificar de forma manual si hay que ajustar rendimiento en sistemas de alta concurrencia.

Una mal ajuste de este parámetro puede hacer que vaya más lento que por defecto, por lo que es especialmente importante prestar atención a varios aspectos:

- Versión de MySQL: En diferentes versiones de MySQL este parámetro se comporta muy diferente.
- Número de procesadores reales (físicos): Al respecto, puede acceder a la [documentación oficial de MySQL](#).

El valor recomendado es el número de CPU (físicas) multiplicado por 2 más el número de discos donde se ubica InnoDB.

El valor de `innodb_thread_concurrency` ha sido cambiado en varias versiones de MySQL, actualmente el valor por defecto es 0. Un valor de 0 significa que “abra tantos hilos como sean posibles”. Por lo tanto, si hay dudas, se puede usar:

```
innodb_thread_concurrency = 0
```

Fragmentación MySQL

Al igual que los *filesystems*, las bases de datos también 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 se tienen 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.cnf

Primero se debe verificar el fichero `my.cnf` y su configuración básica para MySQL. Dicho archivo de configuración está escrito en formato INI y su ubicación puede ser determinada con el siguiente comando:

```
mysql --help --verbose | more
```

La configuración de `my.cnf` debería ser similar a esta (4 GB de RAM y usando un hardware con una configuración media). Compruebe que tiene todos estos parámetros correctamente dentro en la sección `[mysqld]`:

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
user=mysql
character-set-server=utf8mb4
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=1M
join_buffer_size=4M

sql_mode=""

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

Si utiliza MySQL 8 y no tiene un entorno con HA, deshabilite los *log* binarios con la siguiente instrucción en la sección `[mysqld]`:

```
skip-log-bin
```

Con cualquier cambio que haga en el fichero `my.cnf` deberá reiniciar el servicio MySQL.

- Verifique el estado del servicio con `systemctl status mysqld.service`.
- Compruebe al final del fichero `/var/log/mysqld.log` para ver si ha ocurrido algún error.
- Para más información consulte el siguiente enlace [“The Error Log”](#) en el sitio web de MySQL.

Reconstruir bases de datos

Para conocer más acerca de la “Gestión y administración de los servidores”, [diríjase a este enlace](#).

Al hacer ciertas modificaciones al `my.cnf` (por ejemplo, al añadir el parámetro `innodb_file_per_table`), es probable que la base de datos no funcione al reiniciar el servicio. Si aparece el siguiente error deberá restaurar la configuración previa (deberá usar credenciales de usuario raíz o *root user*) y realizar un respaldo de la base de datos:

```
InnoDB: Error: log file ./ib_logfile0 is of different size 0 5242880 bytes
```

```
InnoDB: than specified in the .cnf file 0 67108864 bytes!
```

1. Haga un *backup* de la base de datos:

```
mysqldump -uroot -p pandora > /home/pandora/pandora.sql
```

2. Detenga el servidor MySQL y mueva los datos a una carpeta de *backup*:

```
systemctl stop mysql  
mv /var/lib/mysql /var/lib/mysql.bak
```

3. Cree una nueva carpeta para los datos MySQL (los permisos pertinentes serán asignados en el paso número cinco):

```
mkdir /var/lib/mysql
```

4. Inicialice el servidor MySQL indicando la carpeta de destino en el parámetro `--datadir`. Este proceso generará una contraseña temporal la cual se debe anotar (se mostrará por salida estándar o se almacenará en `/var/lib/mysql/mysql.log`):

```
mysqld --initialize --datadir /var/lib/mysql
```

5. Asigne los permisos pertinentes a la nueva carpeta:

```
chown -R mysql:mysql /var/lib/mysql  
chcon -R system_u:object_r:mysql_db_t:s0 /var/lib/mysql
```

6. Inicie el servicio MySQL y acceda con el cliente MySQL, usando la contraseña del paso número cuatro:

```
systemctl start mysql  
mysql -uroot -p
```

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

7. Cambiar la contraseña de usuario `root` por la que se desee (aquí se utiliza `Pandor4!`):

```
ALTER USER 'root'@'localhost' IDENTIFIED BY 'Pandor4!';
```

8. Salir del cliente MySQL y comprobar que se puede acceder de nuevo usando la nueva contraseña.

9. Acceder de nuevo a al cliente MySQL y crear la base de datos:

```
CREATE DATABASE pandora;  
USE pandora;
```

10. Cargar el *backup* de la base de datos:

```
SOURCE /home/pandora/pandora.sql
```

11. Crear los usuarios de acceso para Pandora FMS usando las mismas credenciales que en la instalación anterior (aquí se utiliza Pandor4!) y luego darles permisos sobre la base de datos:

```
CREATE USER 'pandora'@'localhost' IDENTIFIED WITH mysql_native_password BY  
'Pandor4!';  
CREATE USER 'pandora'@'127.0.0.1' IDENTIFIED WITH mysql_native_password BY  
'Pandor4!';  
GRANT ALL PRIVILEGES ON pandora.* TO 'pandora'@'localhost';  
GRANT ALL PRIVILEGES ON pandora.* TO 'pandora'@'127.0.0.1';
```

12. Después de haber configurado el fichero `my.cnf` y reiniciado el servicio MySQL, se deberá comprobar que estos cambios han sido correctamente aplicados. Para ello, puede consultar las variables una por una:

```
SHOW VARIABLES LIKE 'innodb_log_file_size';
```

```
SHOW VARIABLES LIKE 'innodb_io_capacity';
```

```
SHOW VARIABLES LIKE 'innodb_file_per_table';
```

O bien con una consulta general como:

```
SHOW VARIABLES LIKE "innodb%";
```

Una vez se haya comprobado que Pandora FMS (tanto Consola web como servidor) puede conectar correctamente a la base de datos y todo funciona correctamente, puede borrarse el directorio `mysql.bak`:

```
rm -rf /var/lib/mysql.bak
```

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

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

Debería tener más 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. Lo anterior implica que la fragmentación de las tablas es común al resto de tablas y esto podría anticipar 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 recrear la base de datos después de haber configurado correctamente el fichero `my.cnf` y reiniciar MySQL.

Optimización de tablas específicas

Se recomienda el uso de la herramienta `OPTIMIZE` de MySQL en ciertas tablas de Pandora FMS. Para ello directamente desde MySQL se debe 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;
```

Esto previene que el rendimiento se degrade con el tiempo mientras el sistema trabaja.

En tablas muy grandes el comando `OPTIMIZE` puede quedarse *bloqueado* y dejar de ser una alternativa: entonces es mejor [reconstruir la BB.DD.](#)

Después de hacer estas operaciones, conviene ejecutar:

```
FLUSH TABLES;
```

Cita 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.)

Tokens especiales de MySQL

Existen algunos *tokens* muy especiales de MySQL que pueden ayudar o empeorar el rendimiento.

- `innodb_flush_method`:

```
innodb_flush_method = 0_DIRECT
```

Este parámetro afecta a cómo se escribe en disco.

- innodb_lock_wait_timeout:

```
innodb_lock_wait_timeout = 90
```

Evita que ante un atasco ocasional el MySQL “se rinda” (*MySQL has gone away*) y se detenga. Si sobrepasa los 90 segundos es un gran problema.

Comprobación de la fragmentación tabla por tabla

Usando el CLI de MySQL, se 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 TABLE_TYPE = 'BASE TABLE' AND DATA_FREE > 0 ORDER BY frag_ratio DESC;
```

Se deberían ver solamente las tablas con algún índice 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 mucho tiempo en optimizarse si están muy fragmentadas. Esto podría causar algún tipo de impacto en los sistemas de producción.

Se recomienda precaución al optimizar estas tablas tan grandes. Un entorno normal podría optimizarse una vez al año y entornos más grandes cada seis meses.

Para optimizar la tabla tagent_module_inventory (en este ejemplo la base de datos es llamada pandora):

```
OPTIMIZE TABLE pandora.tagent_module_inventory;
```

Aparecerá un mensaje:

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

Ahora, si se comprueba de nuevo la fragmentación de las tablas, se podrá 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 imprescindible 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

Esto es de ámbito general, pero debe estar seguro de que el sistema de E/S (entrada y salida) no es un cuello de botella (escritura y lectura en dispositivos de almacenamientos/discos). Ejecute el siguiente comando para recoger la información del sistema:

```
vmstat 1 10
```

Ahora, observe en las últimas columnas (CPU-WA): un valor por encima de 10 normalmente significa que existe un problema de E/S que deberá ser arreglado.

Tener unos valores de CPU-US mayores a cero es normal, pero CPU-SY no deberá ser mayor de 10-15.

En condiciones normales debería tener SWAP-SI y SWAP-SO con valores en 0, de no ser así significa que el sistema está usando memoria SWAP, considerado como un destructor de

rendimiento. Necesitará incrementar la memoria RAM del equipo o reducir la cantidad de RAM utilizada en las aplicaciones (hilos de Pandora FMS, buffers de MySQL, etcétera).

Se muestra 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 en MySQL

Para usar Particionado de tablas MySQL, se debería usar también el sistema de *múltiples tablespaces* [descrito arriba](#) (innodb_file_per_table).

MySQL 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.)

Si se tienen grandes cantidades de datos en la base de datos (principal e *histórica*) Pandora FMS y estima que muchas operaciones de la Consola que se refieren a esos datos (por ejemplo *drawing graph*) son lentas, entonces mejorará su rendimiento utilizando particionado de tablas.

Particionado automático

Pandora FMS realiza automáticamente un particionado mensual en la base de datos histórica si se es configurado desde la Consola web, sección Historical database, de la configuración general. Para más detalles [consulte este enlace](#).

Particionado manual

Luego de haberse instalado Pandora FMS y habilitado su base de datos histórica, ésta será la que albergue la mayor cantidad de datos y es la recomendada para realizar un particionado de tablas. Precisamente las tablas indicadas para ello son `tagente_datos`, `tagente_datos_string`, `tagente_datos_inc` y `tevento`. A continuación se describe el proceso práctico manual para la tabla `tagente_datos` (véase también “[Particionado automático](#)”).

Se debe comprobar en un primer momento que el directorio `/var/lib/mysql/pandora_history/* .ibd` tiene muchos ficheros (uno por tabla). Si no es así, se 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 recrearla desde el *dump*.

Una vez que se haya asegurado que `innodb_file_per_table` está activado, separe las dos bases de datos principales en diferentes particiones.

- Esta operación necesitará espacio suficiente en el disco para ser completada. Habrá que comprobar qué tan grande es el archivo `tagente_datos .ibd`. Si por ejemplo ocupa 10 gigabytes, necesitará 15 GB de espacio libre para empezar la operación.
- Esta operación puede llevar 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 días (más de 50.000.000 filas).

Éste es un ejemplo para hacer la partición de todo el 2023 hasta ahora y para futuros meses. Para comenzar el proceso deberá ejecutar la siguiente consulta en el CLI de MySQL:

```
ALTER TABLE tagente_datos PARTITION BY RANGE (utimestamp) (  
PARTITION Jan23 VALUES LESS THAN (UNIX_TIMESTAMP('2023-02-01 00:00:00')),  
PARTITION Feb23 VALUES LESS THAN (UNIX_TIMESTAMP('2023-03-01 00:00:00')),  
PARTITION Mar23 VALUES LESS THAN (UNIX_TIMESTAMP('2023-04-01 00:00:00')),  
PARTITION Apr23 VALUES LESS THAN (UNIX_TIMESTAMP('2023-05-01 00:00:00')),  
PARTITION May23 VALUES LESS THAN (UNIX_TIMESTAMP('2023-06-01 00:00:00')),  
PARTITION Jun23 VALUES LESS THAN (UNIX_TIMESTAMP('2023-07-01 00:00:00')),  
PARTITION Jul23 VALUES LESS THAN (UNIX_TIMESTAMP('2023-08-01 00:00:00')),  
PARTITION Aug23 VALUES LESS THAN (UNIX_TIMESTAMP('2023-09-01 00:00:00')),  
PARTITION Sep23 VALUES LESS THAN (UNIX_TIMESTAMP('2023-10-01 00:00:00')),  
PARTITION Oct23 VALUES LESS THAN (UNIX_TIMESTAMP('2023-11-01 00:00:00')),  
PARTITION Nov23 VALUES LESS THAN (UNIX_TIMESTAMP('2023-12-01 00:00:00')),  
PARTITION Dec23 VALUES LESS THAN (UNIX_TIMESTAMP('2024-01-01 00:00:00')),  
PARTITION pActual VALUES LESS THAN (MAXVALUE)  
);
```

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

```
ALTER TABLE tagente_datos REORGANIZE PARTITION pActual INTO (  
PARTITION Jan24 VALUES LESS THAN (UNIX_TIMESTAMP('2024-02-01 00:00:00')),  
PARTITION pActual VALUES LESS THAN MAXVALUE);
```

Cambiando "Jan24" por el mes en el que se encuentre.

Recuerde de nuevo que esta operación podría tardar horas, dependiendo de lo grande que sea la tabla tagente_datos. Se puede ir comprobando el proceso viendo el tamaño de los archivos de partición ejecutando:

```
[root@histdb pandora_history]# ls -lah | grep "#sql"  
  
-rw-rw---- 1 mysql mysql 424M feb 24 05:58 #sql-76b4_3f7c#P#Jan23.ibd  
-rw-rw---- 1 mysql mysql 420M feb 24 05:51 #sql-76b4_3f7c#P#Feb23.ibd  
-rw-rw---- 1 mysql mysql 128K feb 24 05:40 #sql-76b4_3f7c#P#Mar23.ibd  
-rw-rw---- 1 mysql mysql 840M feb 24 05:44 #sql-76b4_3f7c#P#Apr23.ibd  
-rw-rw---- 1 mysql mysql 440M feb 24 05:47 #sql-76b4_3f7c#P#May23.ibd  
-rw-rw---- 1 mysql mysql 10M feb 24 05:42 #sql-76b4_3f7c#P#Jun23.ibd  
-rw-rw---- 1 mysql mysql 404M feb 24 05:56 #sql-76b4_3f7c#P#Jul23.ibd  
-rw-rw---- 1 mysql mysql 436M feb 24 05:54 #sql-76b4_3f7c#P#Aug23.ibd  
-rw-rw---- 1 mysql mysql 400M feb 24 05:49 #sql-76b4_3f7c#P#Sep23.ibd  
-rw-rw---- 1 mysql mysql 408M feb 24 05:52 #sql-76b4_3f7c#P#Oct23.ibd  
-rw-rw---- 1 mysql mysql 72M feb 24 06:03 #sql-76b4_3f7c#P#Nov23.ibd  
-rw-rw---- 1 mysql mysql 404M feb 24 06:03 #sql-76b4_3f7c#P#Dec23.ibd  
-rw-rw---- 1 mysql mysql 416M feb 24 06:00 #sql-76b4_3f7c#P#jan23.ibd
```

Reconstrucción de la BBDD

Para conocer más acerca del "Respaldo y recuperación de datos en Pandora FMS", diríjase a [este enlace](#).

Reconstrucción 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 continua 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 BB.DD. de forma periódica.

Para ello hay que programar una parada de servicio, que puede durar aproximadamente 1 hora.

En esta parada de servicio debe usted detener la Consola web PFMS y el servidor PFMS. Atención: puede dejar el servidor Tentacle funcionando para que siga recibiendo datos, y esos se procesarán tan pronto como el servidor esté operativo de nuevo.

Una vez detenidos, realizar un *volcado* de la BBDD (*Export*); en este ejemplo la base de datos es llamada pandora3 y el usuario debe ser root:

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

Ahora realice el borrado de la BB.DD.:

```
mysql -u root -p
```

```
DROP DATABASE pandora3;  
Query OK, 87 ROWS affected (1 MIN 34.37 sec)
```

El siguiente paso es recrear la BB.DD. y seguidamente hacer una importación de datos desde el *dump* realizado al principio:

```
CREATE DATABASE pandora3;  
USE pandora3;  
SOURCE /tmp/pandora3.sql
```

Esto puede tardar unos pocos segundos o varios minutos, dependiendo del tamaño de la base de datos y los recursos disponibles en la máquina.

Se puede automatizar este proceso, más debido su naturaleza delicada, es mejor realizarlo manualmente.

Reconstrucción total

Primero se debe detener el PFMS server:

```
/etc/init.d/pandora_server stop
```

O bien, si se dispone de systemd instalado:

```
systemctl stop pandora_ha
```

Segundo se deber hacer una exportación de todos los esquemas y datos. Escoger un directorio de la máquina, en una partición que cuente con el espacio suficiente para almacenar el *backup* de la base de datos en caso de ser una base de datos grande. Se ejecuta el siguiente comando:

```
mysqldump -uroot -p pandora --single-transaction > backup.sql
```

Tras terminar el proceso tendremos en el fichero `backup.sql` los datos, en el directorio donde nos encontremos.

Una vez hecho esto, se detiene MySQL con:

```
systemctl stop mysqld.service
```

Accederemos al directorio de instalación de MySQL en `/var/lib/mysql` y borraremos los ficheros llamados `binlog.000001`, `binlog.000002...` y `binlog.index`. Además, borraremos el directorio con nombre homónimo a la base de datos `pandora`.

```
rm -rf /var/lib/mysql/pandora
rm -rf /var/lib/mysql/binlog.0*
rm -rf /var/lib/mysql/binlog.index
```

Una vez realizados estos pasos, iniciaremos MySQL de nuevo con:

```
systemctl start mysqld.service
```

Accederemos al terminal de MySQL con el siguiente comando:

```
mysql -uroot -p pandora
```

Estará seleccionada la base de datos `pandora`, pero esta estará vacía. Se utiliza el siguiente comando para importar en la base de datos el *backup* que hicimos de la base de datos de Pandora FMS:

```
SOURCE backup.sql;
```

Por último se debe iniciar el PFMS server:

```
/etc/init.d/pandora_server start
```

O si `systemd` está instalado:

```
systemctl start pandora_ha
```

Índices 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 e 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 grandes de Pandora FMS en MySQL dicha optimización viene por defecto. Es conveniente contar con ayuda de personal experto para optimizar las tablas de MySQL.

Consultas lentas o "slow queries"

En algunos sistemas, en función del tipo de información almacenado, se pueden encontrar algunas *slow queries* (consultas lentas) que hacen que el sistema vaya peor. Puede activar el *log* de este tipo de consultas durante un período corto de tiempo (ya que perjudica al rendimiento del sistema) a fin de estudiar las consultas e intentar optimizar las tablas con índices.

Para activar este sistema hay que hacer lo siguiente:

- Editar el archivo `my.cnf` y añadir las siguientes líneas:

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

- Para poder utilizarlo debe crearlo y establecer las reglas administrativas:

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

- Reiniciar MySQL.
- Al finalizar de analizar cuáles son las *slow queries* recuerde restablecer el fichero `my.cnf` comentando las líneas agregadas y reiniciando de nuevo el servicio MySQL.

Referencias

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

html

- <http://jeremy.zawodny.com/mysql/mysql-optimization.html>

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 máxima recomendada está en torno a 2500 agentes (unos 60.000 módulos) por sistema, pero esta cifra varía enormemente en función 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 número de agentes que un sistema puede gestionar eficientemente. En pruebas de laboratorio se han logrado ejecutar 10.000 agentes en un solo servidor con hardware básico, pero fuertemente optimizado.

Ejemplo de configuración de servidores de alta capacidad

Suponiendo una máquina RHEL 8 con 16 GB de RAM y 8 CPU a optimizar para la máxima capacidad de procesamiento del Dataserver (XML).

my.cnf

Solmente se muestran los parámetros más significativos.

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
user=mysql
character-set-server=utf8mb4
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 = 6400M
```

```
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 = 300
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=1M
join_buffer_size=4M
sql_mode=""

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

pandora_server.conf

Se muestran solamente los parámetros más relevantes.

```
verbose 3
server_threshold 5
xxxxserver_threads 8
max_queue_files 5000
```

Aspectos a tener en cuenta:

- El número establecido en el parámetro `verbose` hace referencia a la cantidad de información que se escribe en los *logs*, siendo recomendable no sobrepasar de 3. Cuanto mayor sea el número, menor será el rendimiento de Pandora FMS debido a la gran cantidad de información a escribir en los *logs*.
- Un valor alto (15) del parámetro `server_threshold` 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* y llene los *buffers*. Estos dos elementos de la configuración están íntimamente ligados. En el caso de optimizar el *network server* puede ser interesante bajar el `server_threshold` a 5 ó 10.
- El número de hilos muy alto (más de 5) establecido en `xxxxserver_threads` solamente beneficia a procesos con largas esperas de E/S, como el *network server* o el *plugin server*. En el caso del *Dataserver*, el cual está todo el tiempo en proceso, configurar demasiados hilos puede incluso perjudicar el rendimiento. En sistemas con una BD lenta: pruebe diferentes combinaciones entre 1 y 10; con discos más rápidos y CPU con varios núcleos se podría incrementar. En el caso de optimizar el sistema para el *Networkserver*, el número puede ser bastante más alto, entre 10 y 30.

Herramientas de análisis de capacidad (Capacity)

Pandora FMS dispone de varias herramientas que ayudarán a dimensionar adecuadamente el hardware y software para el volumen de datos que se 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. Por defecto está ubicado 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 un fichero de configuración, donde los módulos están definidos como plantillas.

Estos módulos se rellenan con datos al azar. Se pueden especificar el valor inicial y la probabilidad de cambio de los datos de un módulo mediante valores escritos en un archivo.

Se puede ejecutar el *script* de este modo:

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

Ejemplo de configuración de un fichero (< configuration file >) llamado `pandora_xml_stress.conf`:

```
# 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 `pandora_xml_stress.conf` el valor de configuración `get_and_send_agent_conf` a 1 se puede hacer que los agentes de prueba de carga se comporten como agentes normales, ya que se envían el fichero de configuración y el `hash` MD5. Desde la Consola web PFMS se puede modificar la configuración remota para sucesivas ejecuciones del `pandora_xml_stress`.

Aparte de eso, se puede configurar dónde guardar de forma local los `.conf` de los agentes de prueba con el `token` de configuración `directory_confs` en el fichero `pandora_xml_stress.conf`.

Fichero de configuración

- `max_threads`: Número de hilos (*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 línea.
- `temporal`: Ruta del directorio donde se generarán los ficheros de datos XML ficticios.
- `log_file`: Ruta del fichero de *log* donde informará 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 GNU/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 `yyyy-MM-dd HH:mm:ss`.
- `time_to`: Fecha final a generar los ficheros de datos XML ficticios, en formato `yyyy-MM-dd HH:mm:ss`.
- `get_and_send_agent_conf`: Valor *booleano* 0 ó 1, cuando está activo los agentes ficticios intentarán descargar por configuración remota una versión actual del fichero configuración estándar de un agente. Por la Consola web Pandora FMS se podrá editarlos de manera remota y masiva.
- `startup_delay`: Valor numérico de tiempo en segundos antes de que se comience a generar los

ficheros de los agentes. Se utiliza 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 geográfica con la que se definirán los agentes ficticios.
- `longitude_base`: Valor numérico, es la longitud geográfica con la que se definirán los agentes ficticios.
- `altitude_base`: Valor numérico, es la altitud geográfica con la que se definirán los agentes ficticios.
- `position_radius`: define el radio de la circunferencia en el que se configurará (de manera aleatoria y en base a los parámetros `latitude_base` y `longitude_base`) la posición geográfica del agente ficticio.

Definición de los módulos

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

```
module_begin
module_name < module_name >
module_type < module_type_data>
module_description < description >
module_exec type =< xml_stress_type_generation >;<another_option >;<
another_option > ...
module_unit < units >
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** y **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 1, 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.

Aleatorios (RANDOM)

Los cuales tienen las siguientes opciones:

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

Numéricos

Genera valores numéricos aleatorios entre el rango valor min y el valor max.

Booleanos

Genera valores entre 0 y 1.

Cadena

Genera 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, sin límite de líneas. Por ejemplo:

```
4
5
6
10
```

Admite todo tipo de valores (numéricos y cadenas de texto). Este tipo de módulos usará cada uno de los datos del archivo para generar los datos de los módulos en Pandora FMS, los datos se toman de forma secuencial. Por ejemplo, los datos del módulo anterior se verían en Pandora FMS de esta manera:

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

Dispersión (SCATTER)

Solamente vale para datos numéricos y la gráfica generadas son parecidas a las de un latido de corazón, es decir un valor normal y de vez en cuando un *latido*.

Tiene las siguientes opciones:

- min: Valor mínimo que puede tener el valor.
- max: Valor máximo que puede tener el valor.
- prob: Probabilidad en porcentaje 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).

Nota: 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.

Cómo medir la capacidad de proceso del Dataserver

Existe un pequeño guión llamado `pandora_count.sh` que está en el directorio `/usr/share/pandora_server/util/` 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). Una vez en ejecución, se puede detenerlo pulsando las teclas CTRL+C .

Este *script* simplemente calcula la tasa de procesamiento de archivos del servidor. Es una medida algo burda, pero sirve para hacerse una idea de la efectividad de los ajustes en la configuración 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 generar 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 ajustar a 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, etcétera.
- *boolean*: generar datos verdadero y falso de forma aleatoria. De tal forma que se podría usar cualquier nombre que contenga las palabras «*random*», «*curve*» y/o «*boolean*», por ejemplo:
 - *random_1*
 - *curve_other*

Solamente 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;
```

1. 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.
2. La segunda línea de variable corresponde con *target_agent*, para un agente específico.
3. La tercera línea corresponde con *target_interval*, definida en segundos y que representa el intervalo de periodicidad predeterminada del módulo.
4. La cuarta línea es *target_days* y representa el número de días en el pasado desde la fecha, en *timestamp*, actual.

Herramientas de Diagnóstico en Pandora FMS

Algunas veces existen problemas para los que se necesita ayuda directa del soporte de Pandora FMS. Para facilitar la comunicación con el equipo de Soporte, los servidores Pandora FMS cuentan con algunas herramientas para ello.

Diagnostic Info

Esta herramienta se encuentra en la sección de Management → Admin tools → Diagnostic Info, y

está diseñada para proporcionar la información más importante acerca de la configuración de Pandora FMS y su base datos.

The screenshot displays the Pandora FMS web interface. The top navigation bar includes the Pandora FMS logo, the text "Pandora FMS the Flexible Monitoring System", a search bar, and user information for "admin". The left sidebar shows a menu with "Operation" and "Management" tabs, and a list of items including Discovery, Resources, Profiles, Configuration, Alerts, Servers, Settings, and Admin tools. The "Admin tools" section is expanded, showing "System audit log", "Links", "Diagnostic info" (highlighted), "Site news", and "File manager". The main content area is titled "Pandora FMS Diagnostic tool Admin tools" and displays a table of "MySQL Performance metrics".

MySQL Performance metrics		
InnoDB buffer pool size	✓ 512	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	✓ 2	Recommended Value 2
InnoDB lock wait timeout	✓ 120	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	✓ 32	Min. Recommended Value 32M
Maximun connections	✓ 130	Min. Recommended Value 90 conexions
Read buffer size	✓ 128	Min. Recommended Value 32K
Read rnd-buffer size	✓ 128	Min. Recommended Value 32K
Sort buffer size	✓ 256	Min. Recommended Value 32K
Sql mode	✓	Must be empty
Thread cache size	✓ 120	Min. Recommended Value 8
Thread stack	✓ 256	Min. Recommended Value 256

[Volver al índice de documentación de Pandora FMS](#)