



# Оптимизация и устранение неисправностей



from:

<https://pandorafms.com/manual/!781/>

Permanent link:

[https://pandorafms.com/manual/!781/ru/documentation/05\\_big\\_environments/08\\_optimization](https://pandorafms.com/manual/!781/ru/documentation/05_big_environments/08_optimization)

2025/05/12 09:50



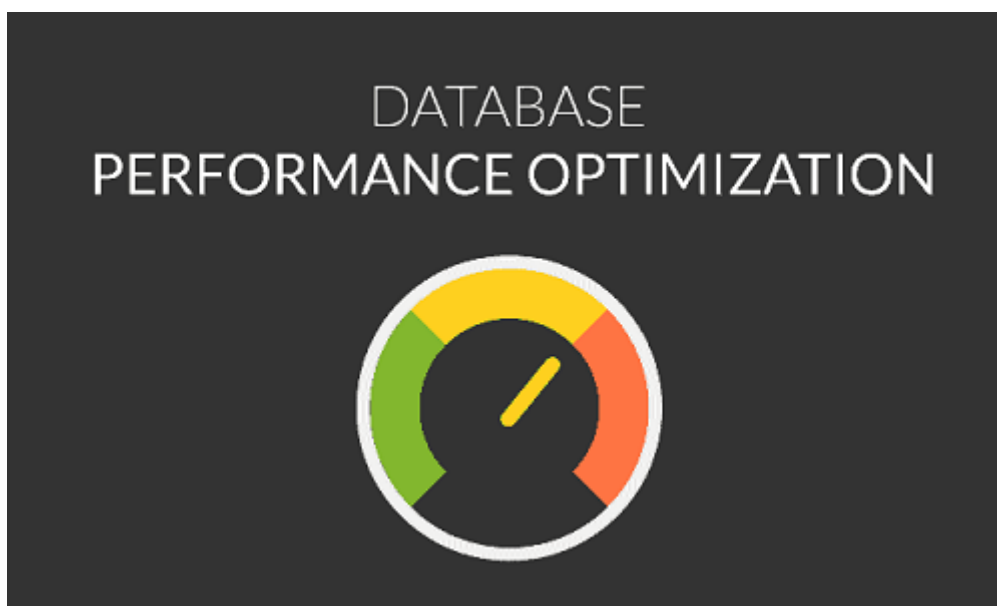
# Оптимизация и устранение неисправностей

[Вернуться в оглавление Документации Pandora FMS](#)

## Оптимизация и устранение неисправностей Pandora FMS

### Введение

Сервер Pandora FMS способен контролировать около 2000 устройств (от 5000 до 80 000 модулей, [в зависимости от имеющегося оборудования](#)); это также требует тонкой настройки конфигурации базы данных.



Кроме того, в этой главе объясняются некоторые методы обнаружения и устранения проблем Pandora FMS установки.

### Оптимизация MySQL для Версия Enterprise

Чтобы узнать больше о «Резервное копирование и восстановление данных в Pandora FMS», [перейдите по этой ссылке](#).

### Общие советы

Для работы с таблицами размером более 2 Гб необходимо соблюдать некоторые рекомендации:

- MySQL® рекомендует использовать 64-битную систему. У 32-битных систем могут возникнуть серьезные проблемы, начиная с 2038 года.
- Чем больше оперативная память и больше процессор, тем выше производительность. По нашему опыту, оперативная память более важна, чем процессор. Минимум для системы уровня предприятия составит 4 Гб. Хороший выбор для большой системы - 16 Гигабайт. Помните, что увеличение объема оперативной памяти может ускорить обновление ключей за счет хранения наиболее используемых ключевых страниц в оперативной памяти.
- Хорошо иметь возможность отключить систему в случае сбоя. Для систем, где база данных находится на другом выделенном сервере баз данных, используйте Gigabit Ethernet, предпочтительно с оптоволоком, а не медью. Задержка так же важна, как и производительность.
- Оптимизация дисков очень важна для очень больших баз данных: базы данных и таблицы должны быть разделены на разных дисках. В MySQL® для этого можно использовать символические ссылки. Используйте разные диски для системы и базы данных.

Рекомендуется использовать твердотельные накопители благодаря их скорости и улучшенной задержке системы.

- По возможности используйте `–skip-locking` (включенный по умолчанию в некоторых системах). Это отключит внешнюю блокировку и обеспечит более высокую производительность.
- Если клиент и сервер MySQL® находятся на одной машине, используйте сокеты вместо TCP/IP соединений при подключении к MySQL® (это может дать улучшение на 7,5%). Это можно сделать, не указывая имя хоста или localhost при подключении к серверу MySQL®. Отключите двоичный вход и репликацию, если у вас только один хост-сервер MySQL®.
- Использование последних версий MySQL® (5.5 или более поздних) вместо старых версий MySQL® (5.0.x) может обеспечить до 20% разницы в производительности.
- Мы рекомендуем использовать модифицированную версию MySQL® ([Percona Server for MySQL®](#)), которая обеспечивает лучшую производительность. По умолчанию программируемые плагины предназначены для Percona®.

Обратите внимание, что на производительность сильно влияют следующие моменты:

- Используйте двоичные журналы только в том случае, если вы используете конфигурацию MySQL® с репликацией.
- Не используйте журналы отслеживания запросов или журналы медленных запросов.

## Автоматизированные инструменты конфигурирования

Существует довольно много инструментов для оптимизации конфигурации сервера MySQL.

MySQL Tuning Primer, созданный Мэттью Монтгомери, это инструмент командной строки для проверки конфигурации сервера и просмотра его производительности и возможных улучшений, дающий некоторые подсказки и предложения по улучшению. Его можно найти

здесь <https://bugs.launchpad.net/mysql-tuning-primer>

## Отключить двоичную репликацию

Если у вас настроена **система Pandora FMS HA**, требуется бинарная репликация. Эта рекомендация действительна только в том случае, если у вас один сервер Pandora FMS.

Он включен по умолчанию в большинстве дистрибутивов GNU/Linux. Чтобы отключить его, отредактируйте файл `my.cnf`, обычно находящийся в `/etc/my.cnf`, и закомментируйте следующие строки:

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

Вам необходимо закомментировать эти две строки, а затем перезапустить сервер MySQL®.

## Производительность доступа к дискам

Чтобы узнать больше о «Резервное копирование и восстановление данных в Pandora FMS», [перейдите по этой ссылке](#).

Наряду с другими ключевыми параметрами, есть три, которые особенно важны с точки зрения производительности доступа к диску, который часто является узким местом в работе с MySQL.

`innodb_log_file_size`

```
innodb_log_file_size = 64M
```

По умолчанию установлено это значение, которое может быть больше (даже 512M) без ущерба, за исключением восстановления в случае проблемы и большей занятости диска. Значение по умолчанию в MySQL составляет 5M, что очень мало для производственных сред с большим объемом транзакций. Чтобы изменить это значение в работающей системе, сначала нужно сделать полный DUMP и удалить файлы бинарных индексов InnoDB (обычно в `/var/lib/mysql/ib*`). Изменение `my.cnf`, перезапуск MySQL и загрузка дампа SQL. Поскольку этот процесс аналогичен процессу активации токена `innodb_file_per_table` (описанному ниже), мы рекомендуем выполнять весь процесс одновременно: изменить весь `my.cnf`,

перезагрузиться и восстановить БД только один раз. Чтобы узнать больше о процессе резервного копирования и восстановления, перейдите по следующей [ссылке](#).

innodb\_io\_capacity

```
innodb_io_capacity = 100
```

По умолчанию этот параметр установлен на 100, но IOPS системного диска должен быть известен заранее. Вы можете точно узнать IOPS и точную модель жесткого диска (полученную через smartctl), где рекомендуемые значения: 7500RPM → 100 IOPS, 15000 RPM → 190 IOPS, SSD → 1500 IOPS

Чтобы установить smartctl, необходимо запросить полный пакет smartmontools; например:

```
yum install smartmontools,
```

```
apt install smartmontools, и т.д.
```

innodb\_file\_per\_table

Используйте табличное пространство для каждой таблицы ( [Взято из руководства на испанском языке MySQL 5.0](#)):

В MySQL 5.0 вы можете хранить каждую таблицу InnoDB и ее индексы в собственном файле. Эта функция называется “multiple tablespaces” (несколько табличных пространств), поскольку, по сути, каждая таблица имеет свое собственное табличное пространство.

Использование нескольких табличных пространств может быть полезно для пользователей, которые хотят переместить определенные таблицы на отдельные физические диски или восстановить резервные копии таблиц, не прерывая работу с другими таблицами InnoDB.

Несколько табличных пространств можно включить, добавив эту строку в раздел [mysqld](#) от my.cnf:

```
[mysqld]
innodb_file_per_table
```

После перезапуска сервера InnoDB будет хранить каждую новую созданную таблицу в собственном файле table\_name.ibd в каталоге базы данных, к которой принадлежит таблица. Это похоже на то, что делает механизм хранения MyISAM, но MyISAM разделяет таблицу на файл данных tbl\_name.MYD и индексный файл tbl\_name.MYI. Для InnoDB данные и индексы хранятся вместе в файле .ibd. Файл tbl\_name.frm создается как обычно. Если вы

удалите строку `innodb_file_per_table` из `my.cnf` и перезапустите сервер, InnoDB заново создаст таблицы в файлах общего табличного пространства. `innodb_file_per_table` влияет только на создание таблиц. Если сервер запущен с этой опцией, новые таблицы создаются с помощью файлов `.ibd`, но доступ к существующим таблицам по-прежнему возможен в общем табличном пространстве. Если опция удалена, новые таблицы будут создаваться в общем пространстве, но таблицы, созданные в нескольких табличных пространствах, будут по-прежнему доступны.

## Избегание промывки диска при каждой транзакции

MySQL по умолчанию устанавливает `autocommit = 1` для каждого соединения. Что удобно для работы с MyISAM, так мы знаем, что обычно запись на диск не гарантирована, но для InnoDB это означает, что каждая вставка / обновление / удаление в таблице InnoDB означает запись на диск. Что плохого в записи на диск? Совсем ничего. Есть гарантия, что при любой ситуации данные сохранятся при перезапуске базы данных после сбоя. Проблема в том, что производительность БД ограничена физической скоростью диска. Поскольку перед фиксацией записи, диск должен записать данные на другой диск, это занимает время. Предполагая даже, что среднее время поиска будет 9 мс для записи на диск, мы ограничены примерно 67 коммитами/сек<sup>1</sup>, что очень медленно. И пока диск занят попытками записать сектор, он не выполняет чтение. InnoDB может избежать части этого ограничения, выполняя некоторые записи одновременно, но все же ограничение существует. Мы можем предотвратить запись в конце каждой транзакции, заставив его установить «автоматическую» систему записи, которая пишет примерно каждую секунду. В случае сбоя мы можем потерять данные последней секунды, что более чем приемлемо, если мы хотим добиться эффективности. Для этого мы будем использовать следующий токен конфигурации:

```
innodb_flush_log_at_trx_commit = 0
```

По умолчанию это значение установлено в конфигурации.

## KeyBuffer большего размера

В зависимости от общего объема оперативной памяти системы, это очень важный глобальный параметр, который ускоряет DELETES и INSERT.

```
key_buffer_size = 4M
```

Это значение по умолчанию в конфигурации.

## Другие важные буферы

В некоторых дистрибутивах есть несколько буферов, которые по умолчанию пусты. Изменение этих параметров может дать гораздо лучшую производительность, чем по умолчанию. Важно убедиться, что эти маркеры существуют в конфигурационном файле MySQL.

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

## Улучшение параллелизма innodb

Есть один параметр, который может повлиять на производительность сервера MySQL с Pandora. Этот параметр - `innodb_thread_concurrency`. Он используется для указания того, сколько «параллельных потоков» может запускать MySQL. Неправильная настройка этого параметра может привести к тому, что он будет работать медленнее, чем по умолчанию, поэтому особенно важно обращать внимание на него.

- Версия MySQL. В разных версиях MySQL этот параметр ведет себя очень по-разному.
- Количество фактических (физических) процессоров.

Вы можете прочитать официальную документацию по MySQL здесь <sup>1)</sup>.

Рекомендуемое значение - это количество CPU (физических), умноженное на 2, плюс количество дисков, на которых расположен InnoDB. В последних версиях MySQL (> 5.0.21) значение по умолчанию равно 8. Значение 0 будет означать «открыть как можно больше потоков». Поэтому, если есть сомнения, его можно использовать:

```
innodb_thread_concurrency = 0
```

Разные люди <sup>2) 3)</sup> проводили тесты и обнаружили проблемы с производительностью на серверах с большим количеством физических CPU, когда используется очень большое количество, с относительно старыми версиями MySQL (от 2008 года).

## Фрагментация MySQL

Как и filesystems, базы данных также становятся фрагментированными, что приводит к снижению производительности всей системы. В высокопроизводительных системах, таких как Pandora FMS, очень важно, чтобы состояние БД не влияло на производительность системы. В перегруженных системах база данных может в конечном итоге выйти из строя,

что приведет к общесистемному сбою.

Хорошая конфигурация MySQL может ускорить работу Pandora FMS. Если у вас возникли проблемы с производительностью, это, вероятно, связано с неправильной настройкой MySQL или с какой-то проблемой, связанной с базой данных.

## Проверка файла my.cnf

Мы начнем с файла my.cnf и его базовой конфигурации MySQL. Этот файл конфигурации записан в **формате INI**, и его местоположение можно определить с помощью следующей команды:

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

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

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

Starts the MySQL database server.

Usage: mysqld [OPTIONS]

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

Конфигурация вашего my.cnf должна быть похожа на эту (4GB RAM и использование средней аппаратной конфигурации). Проверьте, правильно ли введены все эти параметры в разделе [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
```

Любые изменения, которые вы внесете в файл `my.cnf`, потребуют перезапуска MySQL. Проверьте конец файла `/var/log/mysqld.log`, чтобы узнать, не возникли ли какие-либо ошибки.

## Восстановление баз данных

Одна из наиболее распространенных проблем при изменении файла `my.cnf` - установка новых значений для журналов транзакций. Поэтому, если вы получите эту ошибку:

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

Восстановите предыдущую конфигурацию, создайте резервную копию базы данных и выполните следующие действия:

1. После создания резервной копии остановите службу MySQL.

```
systemctl stop mysql
```

2. Перейдите в предыдущую папку, где находятся файлы данных MySQL (`datadir`) (по умолчанию `/var/lib/mysql`)

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

3. Переместите папку в другое место (/var/lib/mysql → /var/lib/mysql\_backup)

```
mv mysql mysql_old
```

4. Создайте новую папку (/var/lib/mysql)

```
mkdir mysql
```

5. Назначьте владельца папки

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

6. Инициализируйте папку с данными MySQL

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

7. Запустите службу

```
systemctl start mysql
```

8. Запустите конфигуратор и следуйте указаниям мастера

```
mysql_secure_installation
```

9. Перестройте свои базы данных

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

10. Назначьте разрешения правильным пользователям

```
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. Загрузите резервные копии

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

Много раз системы MySQL/Persona неправильно загружают конфигурационные параметры файла my.cnf (обычно потому, что эти значения были введены вне секции [mysqld]).

После настройки файла `my.cnf` и перезагрузки следует проверить правильность применения этих изменений. Для этого мы воспользуемся командой `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          |
+-----+-----+
```

**Проверка того, что изолированные файлы данных для каждой таблицы являются АКТИВНЫМИ**

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

У вас должно быть более 100 файлов (в зависимости от версии Pandora FMS). Каждый «ibd»

должен быть файлом данных для каждой таблицы, если в файле my.cnf включен параметр «innodb\_file\_per\_table». Если у вас нет ни одного из этих файлов «ibd», это означает, что вы используете один файл для хранения всей информации. Это означает, что фрагментация таблиц является общей для остальных таблиц, что может указывать на то, что производительность будет ухудшаться с каждой неделей.

Если ваша база данных работает под управлением одной базы данных, вам нужно будет сначала заново создать базу данных после правильной настройки файла my.cnf и перезапустить MySQL.

### Проверка фрагментации каждой таблицы

Используя MySQL CLI, вы должны выполнить этот запрос:

```
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;
```

Вы должны видеть только таблицы с некоторым индексом фрагментации, например:

```
+-----+-----+-----+-----+-----+-----+
-----+
| ENGINE | TABLE_NAME          | data_length | index_length | data_free | frag_ratio |
+-----+-----+-----+-----+-----+-----+
-----+
| InnoDB | tserver_export_data  | 0           | 0           | 5         | 320.0000  |
| InnoDB | tagent_module_inventory | 0           | 0           | 6         | 25.6000   |
| InnoDB | tagente_datos_inventory | 4           | 0           | 40        | 9.8842    |
| InnoDB | tsession_extended    | 1           | 0           | 4         | 3.3684    |
| InnoDB | tagent_access        | 2           | 7           | 27        | 2.9845    |
| InnoDB | tpending_mail       | 2           | 0           | 4         | 2.6392    |
| InnoDB | tagente_modulo      | 2           | 0           | 4         | 2.1333    |
| InnoDB | tgis_data_history    | 24          | 11          | 67        | 1.9075    |
| InnoDB | tsession             | 2           | 0           | 4         | 1.7778    |
```

InnoDB   tupdate		3		0		3	
1.1852							
InnoDB   tagente_datos		186		194		399	
1.0525							
InnoDB   tagente_datos_string		15		9		24	
0.9981							
InnoDB   tevento		149		62		46	
0.2183							
InnoDB   tagente_datos		2810		2509		65	
0.0122							
InnoDB   tagente_datos_string		317		122		5	
0.0114							
+-----+	-----+	-----+	-----+	-----+	-----+	-----+	-----+
-----+							

Этот запрос будет работать только для таблиц с фрагментацией более 10%.

Оптимизация слишком больших таблиц (например, tagente\_datos) может занять слишком много времени, если они слишком фрагментированы. Это может оказать некоторое влияние на производственные системы. Поэтому мы не рекомендуем оптимизировать такие большие таблицы, так как это может привести к сбою системы (процесс оптимизации «блокирует» таблицу для ее перезаписи)

Для оптимизации таблицы «tagent\_module\_inventory»:

```
optimize table tagent_module_inventory;
```

Появится следующее сообщение:

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

Теперь, если вы снова проверите фрагментацию таблиц, вы увидите, что фрагментация исчезла:

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	



увеличить память RAM или уменьшить память RAM, используемую в ваших приложениях (потoki Pandora FMS, буферы MySQL и т.д.).

Мы покажем вам пример вывода «нормальной» системы:

```
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
```

## Разделение таблиц MySQL

Чтобы использовать разделение таблиц MySQL, необходимо также использовать систему «нескольких табличных пространств», описанную выше (*innodb\_file\_per\_table*).

MySQL 5.1 поддерживает разделение таблиц, которое позволяет распределять очень большие таблицы на более мелкие фрагменты, например, логические подразделы. (Для получения более подробной информации обратитесь к руководству по MySQL:

<http://dev.mysql.com/doc/refman/5.1/en/partitioning-overview.html>)

Если в базе данных Pandora FMS хранятся большие объемы данных и вам кажется, что многие операции консоли, относящиеся к этим данным (например, построение графика),

выполняются довольно медленно, повысьте производительность с помощью разбиения таблиц.

Сначала проверьте, что в каталоге `/var/lib/mysql/pandora_history/*.ibd` есть много файлов (по одному на таблицу), если нет, то вам придется сделать дамп базы данных, изменить конфигурацию файла `my.cnf`, перезапустить MySQL, удалить текущую базу данных и заново создать ее из дампа.

Убедившись, что функция `innodb_file_per_table` включена, разделим две основные базы данных на разные разделы. Это пример разделения для всего 2015 года на данный момент и для будущих месяцев.

Для выполнения этой операции потребуется достаточно места на диске. Вы должны проверить, насколько велик «`tagente_datos.ibd`». Если, например, он занимает 10 Гб, то для начала работы вам потребуется 15 Гб свободного места.

Эта операция может занять много времени, в зависимости от размера таблицы. Например, чтобы разделить таблицу с данными примерно 7500 модулей за 100 дней (более 50 000 000 строк), потребуется полтора часа.

Чтобы запустить процесс, необходимо выполнить следующий запрос в MySQL CLI:

```
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)  
);
```

Затем каждый месяц необходимо будет выполнять следующий запрос для реорганизации сегментирования:

```
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);
```

Измените «Фев16» на месяц, в котором вы находитесь.

Помните, что эта операция может занять несколько часов, в зависимости от размера таблицы «tagente\_datos». Вы можете проверить процесс, просмотрев размер файлов сегментирования, запустив:

```
[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
```

## Реконструкция БД

### Частичная реконструкция

Система управления базами данных MySQL, как и другие SQL-движки, такие как Oracle &trade;, со временем деградирует из-за фрагментации данных, вызванной постоянным удалением и вставкой в большие таблицы. В больших средах с большим объемом данных существует очень простой способ повысить производительность и предотвратить снижение производительности: периодически перестраивать базу данных.

Для этого необходимо запланировать отключение службы, которое может длиться прим. 1 ч.

При остановке службы необходимо остановить WEB-консоль Pandora FMS и сервер (будьте осторожны, оставьте сервер Tentacle продолжать получать данные, и они будут обработаны, как только сервер снова начнет работать).

После остановки мы делаем дамп базы данных (Export).

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

Мы удаляем базу данных:

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

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

Мы создаем базу данных и импортируем предыдущий экспорт:

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

Это может занять несколько минут, в зависимости от того, насколько велика система и от мощности оборудования, для системы с 1500 агентами и примерно 100 000 модулей. Этот процесс можно автоматизировать, но из-за его деликатного характера лучше всего выполнять его вручную.

## Общая реконструкция

Эта глава касается только баз данных InnoDB. Pandora FMS построена на базе данных InnoDB.

К сожалению, MySQL со временем деградирует, что влияет на общую производительность системы. Нет другого решения, кроме как перестроить все схемы базы данных с нуля, перестроить двоичный файл данных, который MySQL использует для хранения всей информации, и файлы, используемые для перестройки транзакций.

Если мы посмотрим на каталог `/var/lib/mysql`, то увидим, что там есть три файла, которые всегда называются одинаково и являются, в зависимости от тяжести случая, огромными. В этом случае, пример:

```
-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
```

Файл `ibdata1` - это файл, в котором хранятся все данные InnoDB системы. В очень фрагментированной системе, которая долгое время не подвергалась «ремаппингу» или «переустановке», эта система будет большой и неэффективной. Параметр `innodb_file_per_table`, рассмотренный выше, регулирует некоторые из этих действий.

Аналогично, каждая база данных имеет внутри каталога `/var/lib/mysql` каталог, определяющий ее структуру. Они также должны быть удалены.

Процедура проста:

1. Сбросьте (с помощью `mysqldump`) все схемы на диск:

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

1. Остановка MySQL.
2. Удалите каталоги ibdata1, ib\_logfile0, ib\_logfile1 и базы данных InnoDB
3. Восстановление MySQL.
4. Снова создайте базу данных pandora (create database pandora;)
5. Импорт файла резервной копии (all.sql)

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

Теперь система должна работать заметно быстрее.

## Дополнительные индексы

В некоторых ситуациях можно оптимизировать производительность MySQL за счет системных ресурсов.

Этот индекс служит для оптимизации скорости получения графиков ценой увеличения использования диска и небольшого снижения производительности удаления/вставки в таблицах данных:

```
ALTER TABLE `pandora`.`tagente_datos` ADD INDEX `id_agente_modulo_utimestamp`  
( `id_agente_modulo` , `utimestamp` );
```

В настоящее время в самых тяжелых таблицах Pandora FMS в MySQL эта оптимизация проходит по умолчанию. Перед оптимизацией таблиц MySQL рекомендуется обратиться к специалистам

## Медленные запросы

В некоторых системах, в зависимости от типа информации, мы можем столкнуться с некоторыми «медленными запросами», которые ухудшают работу системы. Мы можем включить протоколирование таких запросов на короткий период времени (поскольку это вредит производительности системы) для изучения запросов, чтобы попытаться оптимизировать таблицы с индексами. Чтобы активировать эту систему, сделайте следующее:

Отредактируйте файл my.cnf и добавьте следующие строки:

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

Для того чтобы иметь возможность использовать его:

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

Перезапустите mysql.

## Оптимизация конкретных таблиц

Другим менее «радикальным» решением для облегчения проблемы фрагментации является использование инструмента MYSQL OPTIMIZE для оптимизации определенных таблиц Pandora FMS. Чтобы сделать это непосредственно из MySQL, выполните

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

Это улучшает производительность и не требует запуска более одного раза в неделю, и может выполняться «WHILE HOT» во время работы системы. В очень больших системах OPTIMIZE может «заблокироваться» и не быть альтернативой, для этого лучше перестроить базу данных.

После выполнения этих операций рекомендуется выполнить:

```
FLUSH TABLES;
```

Из руководства 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.*

(Для таблиц типа InnoDB, OPTIMIZE TABLE приписывается ALTER TABLE, которая перестраивает таблицу для обновления статистики индекса и неиспользуемого свободного пространства в сгруппированном индексе.)

В установках начиная с версии 7.0 OUM715 по умолчанию применяется следующая конфигурация

Примечание: Если ваша установка Pandora FMS была выполнена до версии 7.0 OUM 715, мы

рекомендуем вам внести следующие изменения в ваши базы данных (основную и историческую):

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

Это действие добавит индекс в таблицу `tagent_data`, что позволит запросам, использующим систему отчетности, запрос или модуль данных или пользовательский график, возвращать результаты за значительно меньшее время, чем раньше.

## Специальные токены MySQL

Существуют некоторые очень «специальные» токены MySQL, которые могут помочь или помешать производительности:

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

Этот параметр (`innodb_thread_concurrency`) в версии 5.1.12 и выше, если он равен 0, означает отсутствие ограничения на параллельность, в версиях ниже, если он равен 20 или более, он означает то же самое: отсутствие ограничения на параллельность.

```
innodb_flush_method = O_DIRECT
```

Этот параметр влияет на способ записи на диск.

```
innodb_lock_wait_timeout = 90
```

Не дает MySQL «упасть» (MySQL has gone away) и остановиться при обычном сбое работы. Если остановка превышает 90 секунд, это большая проблема.

## Ссылки

Ссылки:

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

## MySQL Percona XTraDB

Percona - это улучшенная версия MySQL, особенно в отношении масштабируемости. Она позволяет лучше использовать многопроцессорные системы и ускоряет доступ к дискам.

Чтобы настроить MySQL percona, воспользуйтесь этим отличным онлайн-мастером, который сгенерирует файл `/etc/my.cnf`: [Percona Wizard Configurator](#)

## Определение размеров Pandora FMS для высокой производительности

В этом разделе описаны различные методы конфигурирования Pandora FMS в среде с высокой пропускной способностью. В нем также описаны различные инструменты для тестирования нагрузки, полезные для настройки среды на максимально возможную вычислительную мощность.

Pandora FMS была настроена для поддержки нагрузки около 2500 агентов в системах, где база данных, консоль и сервер находятся на одной машине. Рекомендуемое число составляет около 2500 агентов на систему, но эта цифра сильно варьируется в зависимости от того, являются ли они XML-агентами, более удаленными модулями, с высокими или низкими интервалами, с большими или маленькими системами памяти, все это сильно изменяет количество агентов, которыми система может эффективно управлять. В ходе лабораторных испытаний 10 000 агентов были успешно запущены на одном сервере с базовым, но высоко оптимизированным оборудованием.

## Пример конфигурации сервера с высокой пропускной способностью

Предположим, что машина CentOS с 16 ГБ оперативной памяти и 8 процессорами, которую мы хотим оптимизировать для максимальной пропускной способности сервера данных (XML).

### my.cnf

(Показаны только наиболее значимые параметры)

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
user=mysql
character-set-server=utf8
skip-character-set-client-handshake
# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0
# Mysql optimizations for Pandora FMS
# Please check the documentation in http://pandorafms.com for better results
max_allowed_packet = 64M
innodb_buffer_pool_size = 800M
innodb_lock_wait_timeout = 90
innodb_file_per_table
```

```
innodb_flush_log_at_trx_commit = 0
innodb_flush_method = 0_DIRECT
innodb_log_file_size = 64M
innodb_log_buffer_size = 16M
innodb_io_capacity = 100
thread_cache_size = 8
thread_stack      = 256K
max_connections = 100
wait_timeout = 900
key_buffer_size=4M
read_buffer_size=128K
read_rnd_buffer_size=128K
sort_buffer_size=128K
join_buffer_size=4M
query_cache_type = 1
query_cache_size = 64M
query_cache_min_res_unit = 2k
query_cache_limit = 256K
sql_mode=""
[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
```

## **pandora\_server.conf**

(Отображаются только важные параметры)

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

Следует иметь в виду:

- Число *verbose* относится к количеству информации, записываемой в журналы, и рекомендуется не превышать 10. Чем больше это число, тем ниже будет производительность Pandora FMS из-за большого объема информации, которую необходимо записывать в журналы.
- Очень большое количество потоков (+5) приносит пользу только процессам с длительным ожиданием ввода-вывода, таким как сеть или сервер плагинов, в случае с датасервером, который все время обрабатывает данные, это может даже снизить производительность, вот почему мы используем 5. На системах с медленной БД мы используем еще меньше потоков), попробуйте различные комбинации от 1 до 10. В случае оптимизации системы для сетевого сервера это число будет гораздо больше - от 10 до 30.
- При высоком пороге сервера (15) БД страдает меньше, а увеличение максимального количества обрабатываемых файлов означает, что каждый раз, когда сервер «ищет файлы», он заполняет буферы. Эти два элемента конфигурации тесно связаны между собой. В случае оптимизации сетевого сервера целесообразно снизить порог сервера до 5 или 10.
- Некоторые параметры конфигурации могут влиять на производительность Pandora FMS,

например, параметр `agent_access` (настраивается из консоли).

## Инструменты анализа мощности (Capacity)

Pandora FMS имеет несколько инструментов, которые помогут вам правильно определить размеры вашего оборудования и программного обеспечения для того объема данных, который вы ожидаете получить. Один из них используется для «атаки» непосредственно на базу данных с фиктивными данными (`dbstress`), а другой генерирует фиктивные XML-файлы (`xml_stress`).

### Pandora FMS XML Stress

Это небольшой скрипт, который генерирует файлы данных XML, подобные тем, которые отправляют агенты Pandora FMS. Он находится в `/usr/share/pandora_server/util/pandora_xml_stress.pl`

Сценарии считывают имена агентов из текстового файла и генерируют файлы данных XML для каждого агента в соответствии с файлом конфигурации, где модули определены как шаблоны.

Модули заполняются данными в случайном порядке. Можно задать начальное значение и вероятность изменения данных модуля.

Запустите сценарий следующим образом:

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

Пример конфигурации файла:

```
# 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
```

## Отправка и получение локальных настроек агента

Установив в вашем «pandora\_xml\_stress.conf» значение конфигурации «get\_and\_send\_agent\_conf» равным 1, вы можете заставить агентов тестирования нагрузки работать как обычные агенты, поскольку они отправляют свой файл конфигурации и md5. А из Pandora Console Enterprise вы можете изменить удаленную конфигурацию так, чтобы при последующих выполнениях pandora\_xml\_stress использовалась пользовательская конфигурация из Pandora Console Enterprise, а не через определение «pandora\_xml\_stress.conf».

Кроме того, вы можете настроить, где локально сохранять conf ваших тестовых агентов с помощью конфигурационного маркера «directory\_confs» в файле «pandora\_xml\_stress.conf».

## Конфигурационный файл

- max\_threads: Количество потоков, на которых выполняется скрипт, это улучшает ввод/вывод.
- agent\_file: Путь к файлу со списком имен, разделенных новой строкой.
- temporal: Путь к каталогу, в котором будут созданы фиктивные файлы данных XML.
- log\_file: Путь к файлу журнала, в котором скрипт будет сообщать о своем выполнении.
- xml\_version: Версия файла данных XML (по умолчанию 1.0).
- encoding Кодировка файлов данных XML (по умолчанию ISO-8859-1).
- os\_name: Имя операционной системы фиктивных агентов (по умолчанию Linux).
- os\_version: Версия операционной системы фиктивных агентов (по умолчанию 2.6).
- agent\_interval: Интервал между фиктивными агентами в секундах (по умолчанию 300).
- time\_from: Начальная дата для создания фиктивных файлов данных XML, в формате «YYYY-MONTH-MONTH-DATE-HOUR:MIN:SEC»
- time\_to: Конечная дата создания фиктивных файлов данных XML, в формате «YYYY-MONTH-MONTH-DATE-HOUR:MIN:SEC»
- get\_and\_send\_agent\_conf: Булево значение 0 или 1, когда это значение активно, фиктивные агенты будут пытаться загрузить с помощью удаленной конфигурации более актуальную версию стандартного конфигурационного файла агента. А с консоли предприятия Pandora FMS

вы можете их редактировать.

- `startup_delay`: Числовое значение времени в секундах до того, как каждый агент начнет генерировать файлы, используется для предотвращения соревнования выполнения агентов.
- `timezone_offset`: Числовое значение смещения часового пояса.
- `timezone_offset_range`: Числовое значение, используемое для случайной генерации часовых поясов в этом диапазоне.
- `latitude_base`: Числовое значение - широта, на которой будут появляться фиктивные агенты.
- `longitude_base`: Числовое значение, это долгота, на которой будут появляться фиктивные агенты.
- `altitude_base`: Числовое значение, это высота, на которой будут появляться фиктивные агенты.
- `position_radius`: Числовое значение, представляет собой диапазон вокруг, окружность с данным радиусом, в котором фиктивный агент появляется случайным образом.

## Определение модулей

Определение модуля в сценарии конфигурационного файла и при включенной удаленной конфигурации также будет одинаковым.:

```
module_begin
module_name <name of the module>
module_type <type, i.e: generic_data>
module_description <description>
module_exec type =<type_generation_xml_stress>;<other options separated by ;>
module_unit <units>
module_min_critical <value>
module_max_critical <value>
module_min_warning <value>
module_max_warning <value>
module_end
```

И каждый может настроить его по своему усмотрению:

- `type_generation_xml_stress`: Может принимать значения RANDOM, SCATTER, CURVE.
- `module_attenuation <value>`: Сгенерированное значение умножается на заданное значение, обычно от 0,1 до 0,9.
- `module_attenuation_wdays <value> <value> ... <value>`: Значение модуля затухает только в указанные дни, начиная с воскресенья (0) и заканчивая субботой (6). Например, следующий модуль моделирует снижение сетевого трафика на 50% по субботам и воскресеньям:

```
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>`: Если установлено в единицу, предыдущее значение модуля

всегда прибавляется к новому значению, в результате чего получается возрастающая функция.

- Другие: Смотрите ниже, какие у вас есть варианты в зависимости от типа генерации модуля.

Обратите внимание, что конфигурационные токены *min/max\_critical* и *min/max\_warning* доступны только в версии 5.0 или выше.

### Случайные (RANDOM)

У них есть следующие возможности:

- variation: Вероятность в %, что оно отличается от предыдущего значения.
- min: Минимальное возможное значение.
- max: Максимальное возможное значение..

### Числовые

Генерируются случайные числовые значения между min b max значениями диапазона.

### Логические

Генерируемые значения от 0 до 1.

### Цепь

Генерируется цепь длиной от min значения до max значения, символы случайные от A до Z, включая верхний и нижний регистр и цифровые значения.

### Внешний источник данных (SOURCE).

Позволяет выбрать обычный текстовый файл в качестве источника данных. Опции:

- src: исходный файл для данных.

Файл содержит одни данные на строку, ограничений по строкам нет. Например:

```
4
5
6
10
```

Поддерживаются все типы значений (числовые и текстовые строки). Этот вид модулей использует каждый из данных файла для генерации данных модулей в Pandora, данные берутся последовательно. Например, данные из предыдущего модуля выглядели бы в Pandora следующим образом:

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

## Рассеивание (SCATTER)

Они действительны только для числовых данных, а создаваемые графики похожи на графики сердцебиения, т.е. нормальное значение и время от времени «удар».

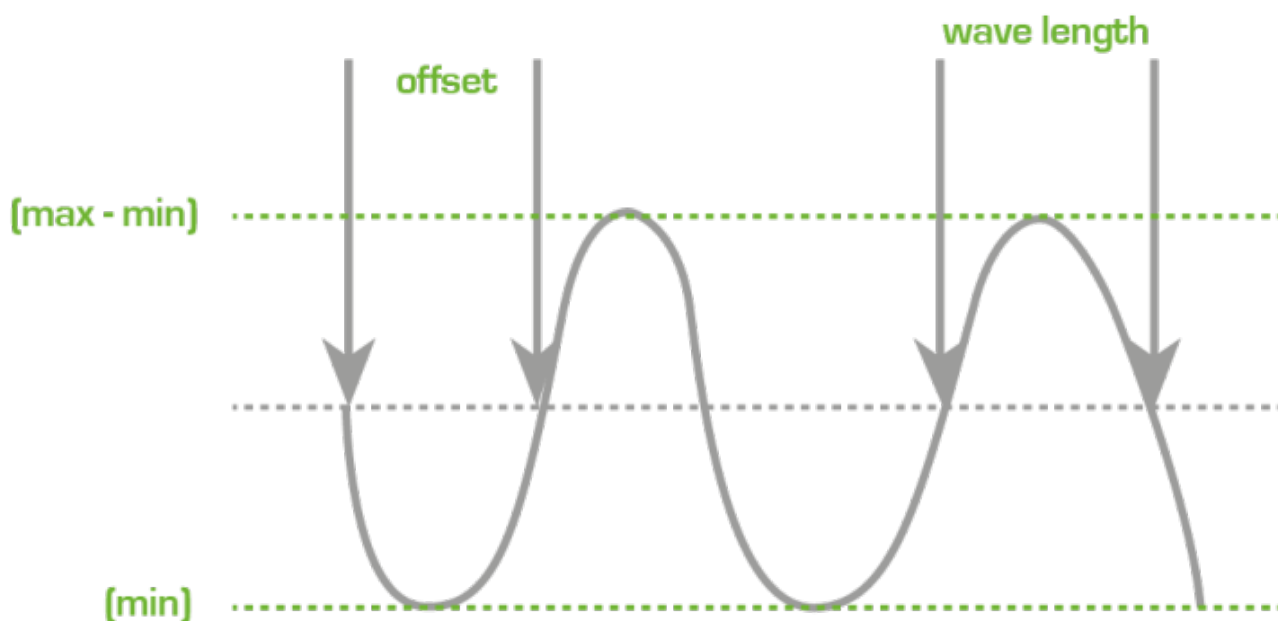
Он имеет следующие опции:

- min: Минимальное значение, которое может иметь значение.
- max: Максимальное значение, которое может иметь значение.
- prob: Вероятность в % возникновения «сердцебиения».
- avg: Среднее значение, отображаемое по умолчанию, если нет «сердцебиения».

## Кривая (CURVE)

Генерирует данные модуля по тригонометрической кривой. У них есть следующие варианты:

- min: Минимальное значение, которое может иметь значение.
- max: Максимальное значение, которое может иметь значение.
- time\_wave\_length: Числовое значение в секундах длительности *пиковой продолжительности ВОЛНЫ*.
- time\_offset: Численное значение в секундах начала волны от времени ноль с нулевым значением модуля (аналогично графику синуса).



Интересные заметки

- Этот инструмент предварительно настроен на поиск всех агентов для модулей с именами «random», «curve» или «boolean» и с интервалом от 300 секунд до 30 дней.

## Как измерить вычислительную мощность датасервера

Существует небольшой скрипт под названием «pandora\_count.sh», который находится в журнале /tool в каталоге сервера Pandora FMS. Этот скрипт используется для измерения скорости обработки XML файлов сервером данных и использует в качестве ссылки общее количество ожидающих обработки файлов в /var/spool/pandora/data\_in, поэтому для его использования вам необходимо иметь несколько тысяч пакетов для обработки (или сгенерировать их с помощью упомянутого выше инструмента). Этот сценарий просто подсчитывает пакеты, которые существуют сейчас, и вычитает их из пакетов, которые были 10 секунд назад, затем делит результат на 10, и это количество пакетов было обработано за последние 10 секунд, показывая скорость в секунду. Это грубый показатель, но он полезен для настройки конфигурации сервера.

## Pandora FMS DB Stress

Это небольшой инструмент для тестирования производительности вашей базы данных. Его также можно использовать для «предварительного создания» периодических или случайных данных (с использованием тригонометрических функций) и заполнения фиктивных модулей. Для автоматического ввода данных с помощью этого инструмента необходимо создать агента и назначить ему модули. Имена должны называться со следующей нотацией:

- *random*: чтобы генерировать случайные данные.
- *curve*: для построения кривой совпадений с использованием тригонометрических функций. Полезно для просмотра работы по интерполяции с различными интервалами и т.д.
- *boolean*: генерировать случайные булевы данные.

Поэтому вы можете использовать любое имя, содержащее слова «*random*», «*curve*» и/или «*boolean*», например:

- random\_1
- curve\_other

Может быть выбран только тип модуля «data\_server».

*Тонкая настройка инструмента Pandora FMS DB Stress.*

Этот инструмент предварительно настроен на поиск во всех агентах модулей с именами «*random*», «*curve*» или «*boolean*» и с интервалом от 300 секунд до 30 дней.

Если вы хотите изменить это поведение, вы должны отредактировать *script*

*pandora\_dbstress* и изменить некоторые переменные в начале файла:

```
# 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;
```

Первая строка переменной, соответствующая *target\_module*, должна быть установлена на фиксированный модуль или -1 для обработки всех совпадающих целей. Вторая строка переменной соответствует *target\_agent*, для конкретного агента. Третья строка соответствует *target\_interval*, определенному в секундах и представляющему собой интервал периодичности модуля по умолчанию. Четвертая строка - *target\_days* и представляет собой количество дней в прошлом с текущей даты *timestamp*.

## Инструменты для поиска неисправностей и диагностики Pandora FMS

Иногда у пользователя возникают проблемы, и разработчики Pandora FMS не могут помочь ему, не имея дополнительной информации о его системах. В версии 3.0 мы разработали несколько инструментов, чтобы помочь вам решить некоторые проблемы:

### Диагностическая информация

В более современных версиях Pandora FMS есть возможность получить диагностическую информацию о нашей установке Pandora FMS.

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

The screenshot shows the Pandora FMS web interface. The left sidebar contains navigation options: Operation, Management, Discovery, Resources, Profiles, Configuration, Alerts, Servers, Settings, Admin tools (selected), 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 connections
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

В этом окне мы можем увидеть информацию о конфигурации Pandora FMS и MySQL, а также графики системы само мониторинга.

## pandora\_diagnostic.sh

```

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

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

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

```

Это инструмент, расположенный в `/usr/share/pandora_server/util` и предоставляющий много информации о системе:

- Информация о процессоре
- Время работы и средняя загрузка процессора
- Информация о памяти
- Информация о ядре/релизе
- Файл дампа конфигурации mysql
- Файл дампа (с фильтрацией паролей) конфигурации сервера Pandora FMS.
- Информационные журналы Pandora FMS (но не полный журнал!)
- Информация о диске
- Информация о процессах Pandora FMS
- Полная информация журнала ядра (dmesg).

Вся информация генерируется в файле .txt, поэтому пользователи могут отправить эту информацию всем, кто захочет им помочь, например, на форуме пользователей Pandora FMS или в публичных списках рассылки Pandora FMS. Эта информация не должна содержать никаких конфиденциальных сведений. Обратите внимание, что вам может понадобиться запуск с привилегиями root, если вы хотите, чтобы файлы `pandora_server.conf` и `my.cnf` были разобраны.

Это один из примеров выполнения:

```
$ ./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'
```

А это некоторые части вывода файла

```
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

[Вернуться в оглавление Документации Pandora FMS](#)

1)

[http://dev.mysql.com/doc/refman/5.0/en/innodb-parameters.html#sysvar\\_innodb\\_thread\\_concurrency](http://dev.mysql.com/doc/refman/5.0/en/innodb-parameters.html#sysvar_innodb_thread_concurrency)

2)

<http://optimmysql.blogspot.com/2008/04/variable-day-out-5-innodbthreadconcurr.html>

3)

<http://mysqlha.blogspot.com/2010/03/do-we-still-need-innodbthreadconcurr.html>