Pandora FMS の最適化と問題解決

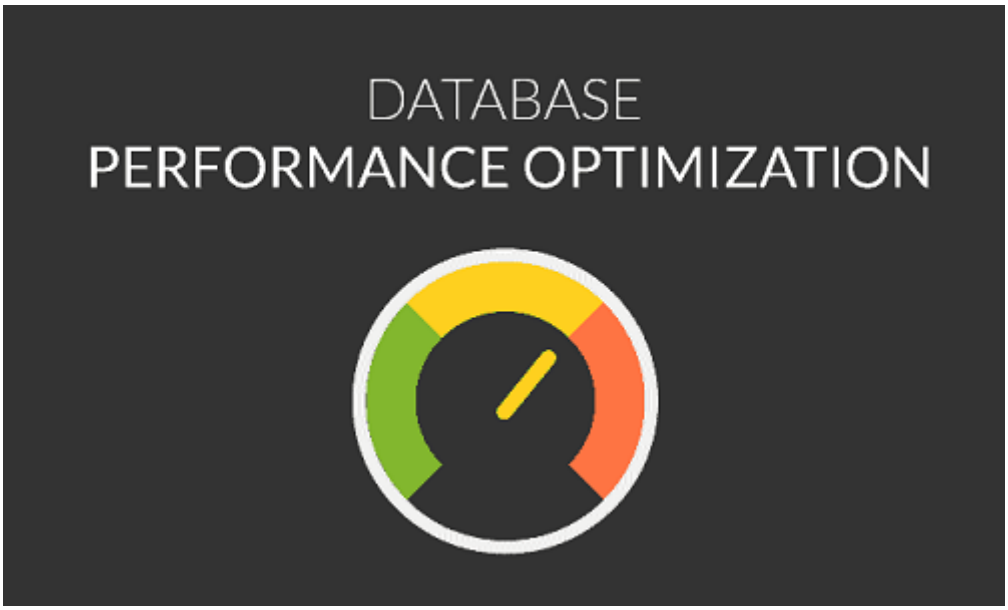
Pandora FMS の最適化と問題解決

Pandora FMS ドキュメント一覧に戻る

Pandora FMS の最適化と問題解決

概要

Pandora FMS サーバは、約 2000のデバイス(5 から 8万モジュール、ハードウエアに依存します)のモニタリングが可能です。ただし、そのためには、データベースの設定を調整する必要があります。



この章ではまたPandora FMS の問題発見および解決のためのテクニックについて説明します。

Enterprise 版のための MySQL 最適化

“Pandora FMS におけるデータバックアップとリカバリ ” についての詳細は、こちらを参照してください。

一般的に推奨する設定

- 特に明示されていない限り、ここでの説明は MySQL バージョン 5.7 を前提としています。
- “MySQL 5.7 から MySQL 8 へのアップグレード ” も合わせて参照してください。

2GiBを超えるテーブルを持つ巨大なシステムを必要とするのであれば、次のガイドラインに従う必要があります。

- MySQL® は 64Bit システムの利用を推奨します。32Bit システムでは、2038年以降重大な問題があります。
- よりよいパフォーマンスを得るために十分なメモリとCPUを用意します。われわれの経験では、CPUよりもメモリのほうが重要です。エンタープライズレベルであれば最低 4GiB 必要です。巨大なシステムであれば、16GiB 割り当てるというのもよいでしょう。メモリが十分あれば、利用されるインデックスの大半をメモリ上に置くことでインデックスの更新が高速化されるということを忘れてはいけません。
- 障害が発生した場合にシステムを切り離すようにできるようにするのが良いでしょう。特定のサーバにデータベースがあるシステムの場合は、ギガビットイーサを利用すべきです。待ち時間はパフォーマンスにとって重要です。
- ディスクの最適化は、とても大きなデータベースにとっては大変重要です。データベースおよびテーブルを異なるディスクに分割すべきです。MySQL では、シンボリックリンクを使うことができます。システムおよび、データベースで異なるディスクを使い、一つのハードディスクのアクセスを減らすようにすることが重要です。

システムの応答時間を早めるために、SSD を利用することを
お勧めします。

- 可能であれば、`-skip-locking` (いくつかのシステムではデフォルトで有効化されています) を使ってください。これにより外部ロックを行わず、パフォーマンスが向上します。
- もし、MySQLサーバとクライアントを同じマシン上で起動するなら、TCP/IPのかわりにsocketを使用してMySQLサーバへ接続しましょう(これにより7.5%性能が改善されます)。MySQLサーバに接続する際、ホスト名を指定しないか*localhost*と指定することでsocket経由で接続することができます。MySQLサーバを1台しか起動しないのであれば、binary logの出力とレプリケーションを無効にしましょう。
- MySQL® の最近のバージョン(5.5以降)を利用することにより、古いバージョン(5.0.x)よりも 20% ほどパフォーマンスが向上します。
- よりパフォーマンスの高い、修正版の MySQL (Percona Server for MySQL®) を使うことを強くお勧めします。デフォルトでは、プラグインは Percona 用に作成しています。

以下の点においてパフォーマンスは大きく影響を受けることに注意してください。

- MySQL® でレプリケーション設定を行う時のみ binary log を利用してください。
- クエリのトレースログや slowquery ログは利用しないでください。

MySQL の設定をチェックするためのツール

MySQL サーバの設定を“ 最適化 ”するためのツールはたくさんあります。

Mattew Montgomery の MySQL Tuning Primer は、MySQL のパフォーマンスをチェックする(コマンドライン)ツールです。そして、それを改善するためのちょっとした提案をしてくれます。https://bugs.launchpad.net/mysql-tuning-primer を確認してみてください。

バイナリログ出力の停止

Pandora FMS HA システムを設定している場合は、バイナリ

ログは必須です。ここに記載している内容は、Pandora FMS
サーバが 1台の場合にのみ有効です。

多くの Linux ディストリビューションにおいてデフォルトで有効になっています。無効にするには、my.cnf ファイル (通常、/etc/my.cnf にあります) を編集し、次の行をコメントアウトします。

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

両方の行をコメントアウトし、MySQL サーバを再起動します

ディスク I/O パフォーマンス

“Pandora FMS におけるデータバックアップとリカバリ ” に
関する詳細は、こちらを参照してください。

ディスク I/O に直結し、考慮すべきとても重要な 3つの設定があります MySQL において、不適切な I/O アクセスは、通常最も重要なボトルネックになります。

innodb_log_file_size

```
innodb_log_file_size = 64M
```

デフォルトではこの値が設定されていますが、問題が発生した場合の回復とディスク占有率が増えることを除いては、事前検証なしに高くすることができます(512Mでも) MySQL のデフォルト値は 5M です。これは、トランザクション量が多い本番環境では非常に低い値です。 稼働中のシステムで値を変更するには、次のようにします。

1. データベースのフルバックアップ(SQL ダンプ)を行います。
2. Innodb バイナリインデックスファイル(通常は /var/lib/mysql/ib* です)を削除します。
3. my.cnf を編集します。
4. MySQL を再起動します。
5. ダンプを読み込みます。

サーバは、新たなトランザクションログファイルを新たなサイズで再生成し、通常通り動きだします。この処理は innodb_file_per_table トークン(後述)を有効化するのと同じであるため、処理全体を同時に行うことをお勧めします。

innodb_io_capacity

```
innodb_io_capacity = 100
```

デフォルトでは、このパラメータの値は 100 です。ただし、事前にシステムのディスクの IOPS を知る必要があります。正確な IOPS およびハードディスクのモデルを(smartctlを使って)知ることができます。ここでのお勧めの値は、7500RPM → 100 IOPS, 15000 RPM → 190 IOPS, SSD → 1500 IOPS です。

smartctl をインストールするには、次のように
smartmontools パッケージをインストールする必要があります。

```
yum install smartmontools,

apt install smartmontools, etc.
```

innodb_file_per_table

テーブルごとにファイルを作成します。

MySQL 5.0 では、各 InnoDB テーブルとそのインデックスを独自のファイルに保存することができます。この機能は、各テーブルに独自のテーブルスペースがあるため、“ 複数のテーブルスペース ” と呼ばれます。

複数のスペーステーブルの使用は、特定のテーブルを別々の物理ディスクに移動したい場合や、残りの InnoDB テーブルの使用を中断せずにテーブルのバックアップを復元したい場合に役立ちます。

my.cnf の mysqld セクションに以下の設定を追加することにより、複数のテーブルスペースを有効化できます。

```
[mysqld]
innodb_file_per_table
```

サーバーを再起動した後、InnoDBは、新しく作成された各テーブルを、テーブルが属するデータベースディレクトリ内の独自の name_tabla.ibd ファイルに保存します。これは MyISAM が行う動作と似ていますが、MyISAM はテーブルを tbl_name.MYD データファイルと tbl_name.MYI インデックスファイルに分割します。

InnoDB の場合、データとインデックスは .ibd ファイルにまとめて保持されます。tbl_name.frm ファイルは通常どおり作成する必要があります。innodb_file_per_table 行が my.cnf から削除され、サーバが再起動された場合、InnoDB は共有表スペース・ファイルに表を再作成します。

innodb_file_per_table は、テーブルの作成にのみ影響します。 このオプションを使用してサーバを起動すると、.ibd ファイルを使用して新しいテーブルが作成されますが、共有テーブルスペース内の既存のテーブルに引き続きアクセスできます。このオプションを削除すると、共有スペースに新しいテーブルが作成されますが、複数のテーブルスペースに作成されたテーブルにアクセスするこ

とは引き続き可能です。

トランザクションごとのディスク書き込みの回避

デフォルトでは、MySQLは各接続開始時の autocommit の値を `autocommit =1` としています。MyISAMの場合、更新結果がディスクに保存されることを保証していないため、この設定はそんなに悪い設定ではありません。しかしInnoDBの場合、InnoDBテーブルへのあらゆる　insert / update / delete によって、ディスクへの書き込みが発生するということを意味します。

さて, なぜディスクへの書き込みがよくないのでしょうか?　そんなことはありません。ディスクへの書き込みは何らかのcommitment後、データベースが障害後再起動したときもデータが存在することを保障します。問題は、DBのパフォーマンスがディスクの物理的な速度によって制限を受けることです。書き込みを確認する前にデータをディスクに書き込まなければならないことを考えると、少し時間がかかるでしょう。

ディスクの書き込みに平均9msかかるとすると、おおよそ1秒当たり67コミットに制限されます。これは非常に遅いです。そして、ディスクのあるセクタに書き込み中は、そのセクタを読み込むことはできません。InnoDBはこれらの制限のいくつかを複数の書き込みを同時に実行することで回避しています。しかしそれでも制約が存在します。

システムによる“ 自動 ” 書き込み(およそ1秒ごとに書き込みを行う)を利用することで、各トランザクションが完了するたびにディスクへ書き込むことを回避することができます。問題が発生した場合、最近1秒のデータが失われますが、パフォーマンス向上を得ようとしていることを考慮すれば、許容できる範囲でしょう。デフォルトでは、`innodb_flush_log_at_trx_commit = 0` となっています。

KeyBuffer の大きさ

システムに搭載された物理メモリ量に依存しますが、非常に重要なグローバル変数であり、これを設定することでDELETEおよびSELECTの実行速度が改善されます。

```
key_buffer_size = 4M
```

上記は、デフォルトの設定値です。

他の重要なバッファ

いくつかのディストリビューションでは、デフォルトで設定されていないいくつかのバッファがあります。これらのパラメータを設定することによりデフォルトより高いパフォーマンスを得ることができます。これらのトークンが MySQL の設定ファイルに存在するかどうかを確認することが重要です。

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

MySQL バージョン 8 以降では、*query cache* のサポートが廃止されました。
詳細については、https://dev.mysql.com/blog-archive/mysql-8-0-retiring-support-for-the-query-cache/ を参照してください。

InnoDB の同時実行スレッド

Pandora の MySQL サーバパフォーマンスを向上に有効なパラメータがあります。そのパラメータは、'innodb_thread_concurrency です。このパラメータは、MySQL で “ 同時実行スレッド ” をいくつにするのかを設定するのに利用します。このパラメータがうまく設定されていない場合、デフォルトよりも遅くなることがあります。そのため、次のような情報に注意を払うことが特に重要です。

- MySQL バージョン。異なるバージョンの MySQL では、このパラメータはとても異なる動作をします。
- 物理プロセッサ数。

MySQL の公式ドキュメントは、こちら です。

推奨値は、CPU(物理)の数に 2を掛けたものに InnoDB が配置されているディスクの数を加えたものです。

お勧めの値は、(物理的な)CPU数の2倍に、InnoDB が置かれているディスクの数を足した数です MySQL の最近のバージョン (> 5.0.21) では、デフォルトは 8 です。0 に設定すると、“ 可能な限り多くのスレッドを開きます ”。

```
innodb_thread_concurrency = 0
```

他に [1) 2)] によると、古いバージョンの MySQL でテストを行ったところ、大きい値にしたときに複数の物理 CPU のあるサーバでパフォーマンスの問題があったとのことです。(2008年の情報です)

MySQL フラグメンテーション

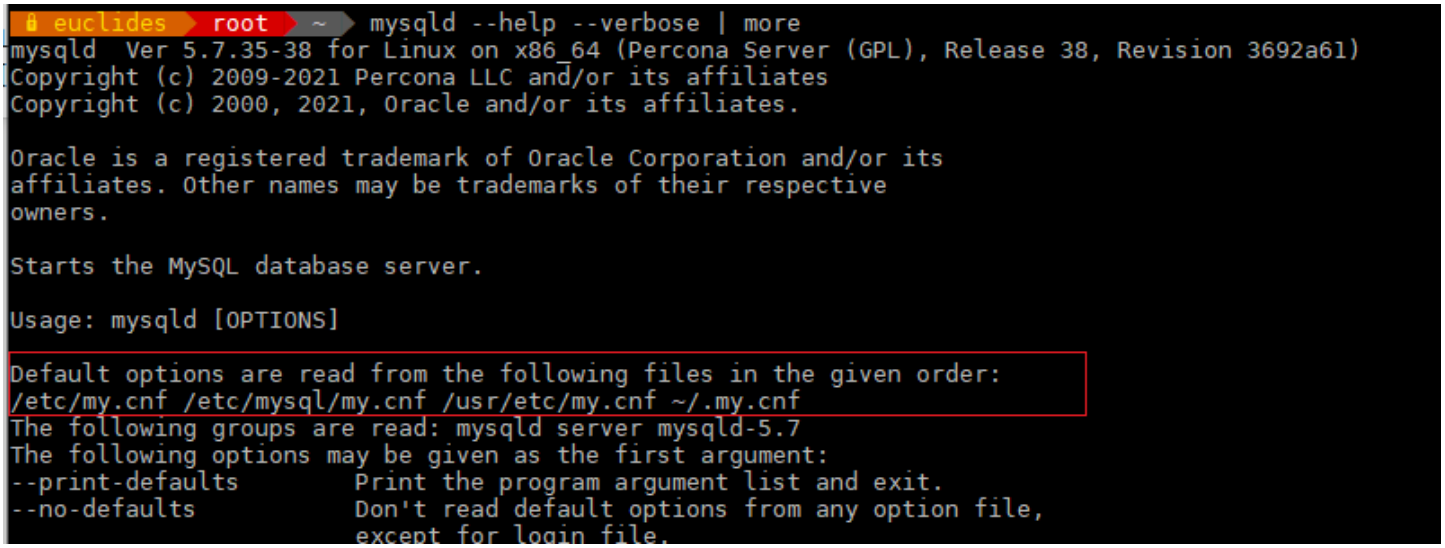
ファイルシステムのように、データベースもフラグメンテーションが発生しシステム全体の速度が低下します Pandora のように高いパフォーマンスを必要とするシステムでは、高速で信頼性の高いデータベースが必要です。高負荷なシステムでは、監視システムが停止する可能性があります。

Pandora FMS で良いパフォーマンスを得るには MySQL の設定がとても重要です。パフォーマンスの問題がある場合は、MySQL の設定やデータベースに関する問題である可能性があります。

my.cnf 設定の確認

MySQL サーバの“基本設定”である my.cnf から確認します。設定ファイルは、INI フォーマットで書かれており、次のコマンドで場所を確認できます。

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



このファイルは以下のような設定ファイルになっています(この例は、メモリ 4GB の 2013 年の古い平均的なサーバハードウエアです)。[mysqld] セクション内のトークンを確認します。

```
[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
```

MySQL バージョン 8 以降では、*query cache* のサポートが廃止されました。
詳細については、https://dev.mysql.com/blog-archive/mysql-8-0-retiring-support-for-the-query-cache/
を参照してください。

MySQL 8 を利用しており HA 環境が無い場合 は、[mysqld] セクションの次のコマンドでバイナリログを無効化します。

```
skip-log-bin
```

my.cnf ファイルを変更したら、MySQL サービスを再起動
します。

- systemctl status mysqld.service にてサービスの状態を確認します。
- エラーの確認には /var/log/mysqld.log の最後を見てください。
- より詳細は、MySQL サイトの こちらを確認してください。

データベースのリストア

サーバ管理のより詳細は、こちらを確認してください。

my.cnf ファイルを変更する場合の共通の問題としては、トランザクションログの値があります。 ログに次のようなエラーが出た場合は、データベースをバックアップし以前の設定をリストアします(root 権限を利用)

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

1. バックアップを作成したらMySQL サービスを停止します。

```
systemctl stop mysql
```

2. MySQL データファイルがあるフォルダ(デフォルトは /var/lib/mysql)へ行きます。

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

3. フォルダを他の場所へ移動します(/var/lib/mysql → /var/lib/mysql_backup)

```
mv mysql mysql_old
```

4. 新たなフォルダを作成します(/var/lib/mysql)

```
mkdir mysql
```

5. フォルダのオーナーを設定します。

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

6. MySQL データでフォルダを初期化します。

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

7. サービスを開始します。

```
systemctl start mysql
```

次のようなエラーとなった場合、

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

SELinux が動作している可能性があります。次のコマンドを実行すると、
拒否されたかどうかを確認できます。

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

- SELinux を無効化するには、こちらを確認してください。
- SELinux を有効化した状態で利用する場合は, こちらを確認してください。

8. 設定プログラムを起動し、ウィザードに従います。

```
mysql_secure_installation
```

9. MySQL へコマンドラインからログインし、データベースを再構築します。

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

MySQL の root ユーザのパスワードを設定する必要があるかもしれません。
それには以下のコマンドを実行します。pandora の部分は任意のパスワードに置き換えてください。

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

10. ユーザに権限を割り当てます。

```
 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. バックアップを読み込みます。

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

時々、MySQL/Percona システムは my.cnf の設定を正しく
読み込みません。(通常、[mysqld] セクションの外にトークンがある場合です)

my.cnf を編集後、mysql を再起動したのち、設定が反映されて動作しているかを確認する必要があります。それには、SHOW VARIABLES コマンドを用います。(結果には 100を超える要素が含まれる可能性があり、次の結果とは異なる場合があります)

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

以下の変数もひとつずつ確認します。

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

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

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

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

mysql>
```

各テーブルごとに分割されたデータファイルがアクティブであるか確認

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

“innodb_file_per_table” トークンを有効化した場合、それぞれのテーブルに対応した “.ibd” ファイルが 100以上(Pandoraのバージョンに依存)存在します。このようなファイルが無い場合は、全データが大きなファイルに記録されます。これは、テーブルのフラグメンテーションが全テーブル共通して発生し、毎週パフォーマンスが劣化することを意味します。

すでに単一のデータベースで実行している場合は、設定を変更し MySQL を再起動したあとにデータベースを再作成する必要があります。

テーブルごとのフラグメンテーションを確認

MySQL の CLI を使って、以下のクエリを実行します。

```
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;
```

フラグメンテーションのあるテーブルが次のように表示されます。

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

10% 以上フラグメンテーションがあるテーブルに対してのみ対応します。

10% 以上のフラグメンテーションがあるテーブルのみを対象とします。大きなテーブル(tagente_datos など)は、フラグメンテーションが大きいと最適化に長時間かかります。本番システムでは大きなインパクトとなりますので、注意してください。

“tagent_module_inventory” テーブルを最適化するには次のようにします。

```
optimize table  tagent_module_inventory;
```

次のような警告メッセージが表示されます。

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

再度確認すると、フラグメンテーションが無くなっていることがわかります。

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

最適化を実行するには、操作を実行できるだけのディスクの空き容量が必要です。そうでないとエラーとなり処理が実行されません。

システム負荷

より一般的な事として、システムの(ディスク) I/O がボトルネックになっていないことを確認する必要があります。システムの状態を取得するのにvmstat コマンドを実行します。

```
vmstat 1 10
```

最後のカラム (CPU WA) を見て、10より大きい値であればディスク I/O の問題があり、解決する必要があります。

CPU-US が高いのは通常です。しかし、CPU-SY は 10～15 を超えないようにすべきです。

SWAP-SI および SWAP-SO はゼロであるべきです。そうでなければシステムがスワップメモリを使っていることを意味し、パフォーマンスを落とします。メモリを増やすか、アプリケーションが使うメモリを減らす(Pandora サーバのスレッドMySQL のバッファの調整など)必要があります。

以下は、“ 正常 ” なシステムの出力サンプルです。

```
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
```

テーブルパーティショニングの利用

MySQL のテーブルパーティショニングを使う場合、上述の複数のテーブルスペース (innodb_file_per_table) も使用して下さい。

MySQL 5.1 では、巨大なテーブルを小さい複数の論理的なテーブルに分割することができるテーブルパーティショニングを行えるようになりました。 (詳細は MySQL のマニュアルを参照して下さい: http://dev.mysql.com/doc/refman/5.1/ja/partitioning-overview.html)

もし Pandora FMS (メインおよびヒストリの両方の)データベースに大量のデータがあり、グラフ描画のようなデータを参照する処理がとても遅いと感じるなら、テーブルパーティショニングを行うことでパフォーマンスを改善できるでしょう。

最初に、innodb_file_per_table が有効であり、データベースがそれを使用していることを確認してください /var/lib/mysql/pandora_history/*.ibd に多くのファイルが表示されるはずです。 そうでない場合は、データベースをダンプし、my.ini を変更し、mysql を再起動し、現在のデータベースを削除して、ダンプから再作成します。

innodb_file_per_table があることを確認したら、固定日付に基づいて 2つのメインデータテーブルを異なるパーティションに分割します。 この例では、2015年以降のデータを分割していますが、自分のニーズに合わせて変更できます。

これには十分なディスク容量が必要です tagente_datos.ibd の大きさを確認してください 10G の場合、操作を開始するには少なくとも 15GB の空きが必要です。

テーブルサイズによっては、この操作に時間がかかる場合があります。例えば、100日間(50,000,000行を超える)約 7500 モジュールのデータを含むテーブルを分割する場合、約 1.5時間かかりました。

MySQL の CLI 以下のクエリを実行します。

```
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)
);
```

パーティション分割を再編成するには、毎月以下のクエリを実行します。

```
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);
```

“Feb16” を当月に変更します。

tagente_datos をモジュールIDに応じて100のパーティションに分割する場合は、次のクエリを実行します:

```
  ALTER TABLE tagente_datos PARTITION BY HASH(id_agente_modulo) PARTITIONS 100;
```

“tagente_datos” テーブルの大きさによっては、この操作には数時間かかる場合があることに注意してください。 実行中のパーティションファイルのサイズを見ると、進行状況を確認できます。

```
[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
```

データベースの再構成

部分的再構成

MySQLは他のデータベースシステム、たとえばOracle ™と同様、時間がたつにつれ、性能が劣化していきます。これは大きなテーブルに対してデータの削除と追加を続けることによって発生するデータのフラグメンテーションによるものです。大量のトラフィックが発生する大きな環境において、性能の改善および劣化を防ぐ非常に簡単な方法があります。それは定期的にデータベースの再構築を実施することです。

そのために、1時間程度のサービス停止を計画すべきです。

計画的なサービス停止時に、Pandora FMS Webコンソール
とサーバも停止すべきです (注意 tentacle サーバはデータを
受け取れるようにしておき、サーバが復旧次第データを処
理できるようにします)

サービス停止後、データベースのダンプを取得(エクスポート)します。

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

データベースを削除します。

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

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

データベースを作成し、先ほどエクスポートしたデータをインポートします。

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

巨大でハードウエア性能があまり高くないシステムでは、この処理全体で数分かかります。たとえば、1500エージェントおよび 100,000モジュールのシステムなどです。この処理は自動化可能ですが、細心の注意が必要なため、毎月もしくは半月に一度、手動で実行することをお勧めします。

この処理を自動化することは可能ですが、非常にデリケートなため、手動で実行するのが最善のオプションです。

全体の再構築

この節の説明は、Innodb データベースのみ該当しますPandora FMS は、Innodb データベースを利用しています。

時間ともに MySQL のパフォーマンスが落ちると、システム全体のパフォーマンスに影響します。この場合は、すべてのデータベーススキーマをゼロから再構築する以外に解決策はありませんMySQL がデータの保存に利用しているバイナリファイルを利用し、またそれを再構築します。

/var/lib/mysql ディレクトリを見ると、常に同じ名前で状態に応じて以下のような 3つのファイルがあります。

```
-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
```

ibdata1 は、すべての Innodb データを保存する先です。非常に断片化されたシステムでは、再構築やインストールをした効率の良いシステムと比べると非常に処理時間を要します。前に言及した innodb_file_per_table パラメータは、このパフォーマンスを調整します。

それぞれのデータベースは /var/lib/mysql ディレクトリにあり、一つのディレクトリで構造が定義されています。それを削除します。

手順はとても簡単です。

1. (mysqldump にて)全スキーマをディスクにダンプします。

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

2. MySQL を停止します。
3. ibdata1, ib_logfile0, ib_logfile1 および InnoDB データベースディレクトリを削除します。
4. MySQL を再起動します。
5. バックアップファイル(all.sql)をインポートします。

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

システムが高速化します。

インデックスオプション

他のシステムリソースを犠牲にしてMySQL パフォーマンスを最適化できるいくつかの場合があります。

以下のインデックスの最適化はグラフ生成を(とても)高速化しますが、多くのディスクスペースを必要とします。また、インデックスのオーバーヘッドにより、若干 INSERT/DELETE 処理が遅くなります。

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

現在MySQL の Pandora FMS の最も重いテーブルでは、この最適化がデフォルトになっていますMySQL テーブルを最適化する前に専門家に相談するのが良いです。

スロウクエリ

いくつかのシステムでは保持している情報によって、通常よりもシステムのパフォーマンスが悪いスロウクエリが見られることがあります。テーブルを最適化するために、(システムのパフォーマンスに影響する)一定時間を超えたこのタイプのクエリをログに記録するようにできます。これを有効にするには次のようにします。

- my.cnf を編集し、次の設定を加えます。

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

- 利用できるように次のように設定します。

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

- mysql を再起動します。
- スロウクエリの分析が終了したら、ファイル my.cnf をリセットして、集約された行にコメントを付け、MySQL サービスを再起動することを忘れないでください。

特定のテーブルのオプティマイズ

他のあまり“過激”ではないフラグメンテーションの問題を解決する方策として、Pandora FMSのテーブルのいくつかに対してMYSQL OPTIMIZEツールを使用するというのがあります。

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

この作業によって、パフォーマンスが改善されるでしょう。この作業は1週間に複数回実行する必要はありません。システムの稼働中にさっと完了するでしょう。巨大な環境においては、OPTIMIZEコマンドがブロックされるかもしれません。このような場合、一番よい代替案はDBの再構築です。

これらの作業を実施した後、以下のコマンドを実行すべきです。

```
FLUSH TABLES;
```

MySQLのマニュアルから、

InnoDB のテーブルに対しては OPTIMIZE TABLE は ALTER TABLE にマップされており、インデックス統計の更新とクラスタされたインデックス内の使用されていないスペースを解放するためにテーブルが再構築されます。

7.0 OUM715 以降のインストールでは、デフォルトで以下の
設定が適用されます。

注意: Pandora FMS バージョン 7.0 OUM 715 より前では、データベース(通常の DB とヒストリ DB 共)に以下の対応をすることをお勧めします。

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

この操作は、*tagente_data* テーブルにインデックスを追加します。以前よりも、レポートシステム、データのクエリ、モジュール、カスタムグラフで大幅に処理が短縮されます。

MySQL の特別なトークン

MySQL には、いくつかの“特別な”トークンがあります。これらは、パフォーマンスを良くしたり悪くしたりします。

- innodb_thread_concurrency:

```
# Set to 0 in mysql 5.1.12 or higher
```

```
innodb_thread_concurrency              = 20
```

この *innodb_thread_concurrency* というパラメータは、5.1.12 以降のバージョンに存在し、0 の場合、平行処理の制限がなくなります。しかし、以前のバージョンでは、20を設定すると同様の意味になります。

- innodb_flush_method:

```
innodb_flush_method = O_DIRECT
```

この重要なパラメータは、ディスクにデータをどう書くかに影響します。

- innodb_lock_wait_timeout:

```
innodb_lock_wait_timeout = 90
```

これは、データベースが長いトランザクションによりロックし“スタック”状態になったとき(mysqlはメッセージを出力しません)に有効です。ただし、もし、90以上のロックがあるのであれば、それは本当に何らかの問題がある場合です。

参考情報

参考:

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

MySQL Percona XTraDB

Percona は、MySQL の“ハイパフォーマンス”版で、スケーラビリティ改善およびシステムの全 CPU を利用した高速化、また、ディスクのアクセス改善がされています。

percona サーバを設定するには、*/etc/my.cnf* を生成する、それ用のよくできた次の設定ウィザードを利用することができます。Percona 設定ウィザード

Pandora FMS のキャパシティ計測

この章では、高いキャパシティが必要な環境での Pandora FMS を設定するための異なる手法を説明します。また、処理を実行する環境を調整するのに便利な負荷テストを行うためのツールについても説明します。

Pandora FMS は、1つのサーバでデータベース、コンソール、サーバを動かした場合、2500エージェントに対応できるように設定されています。推奨するエージェント数は、1システムあたり 2500 で

す。しかし、この数は、XML エージェントの割合、リモートモジュールの割合、監視間隔、システムのメモリ量に応じて変化します。

これらの全ての要素が、1つのシステムで管理できるエージェント数に関わります。テスト環境では、通常のハードウエアの 1台のサーバで 10000 エージェントを実行できましたが、高度な最適化をしています。

高キャパシティサーバの設定例

例えば16GB の RAM および 4 CPU のマシンで、データサーバが最大のパフォーマンス (XML 処理) を出すように最適化したいと思います。

my.cnf

重要なパラメータのみを示しています。

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

```
query_cache_limit = 256K
sql_mode=""
[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
```

MySQL バージョン 8 以降では、*query cache* のサポートが廃止されました。パフォーマンスの向上を目的としていますが、スケーラビリティに深刻な問題があり、簡単に深刻なボトルネックになる可能性があります。

pandora_server.conf

重要なパラメータのみを示しています。

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

以下の点を認識しておく必要があります。

- verbose の数は、どの程度の情報をログに書くかで、10を超える数字にはしないでください。数字を大きくすると、ログに多くの情報を書くため Pandora FMS のパフォーマンスが下がります。
- パラメータ server_threshold の数が大きい(15)と、データベースへの影響が少なくなり、処理されるファイルの最大数を大きくすると、サーバはファイルを検索してバッファをいっぱいにします。 これら 2つの要素の設定は密接に関連しています。 ネットワークサーバを最適化する場合は、server_threshold を 5 または 10 より下げることをお勧めします。
- dataserver_threads に設定された非常に多数のスレッド(5以上)は、ネットワークサーバやプラグインサーバなど E/S 待機時間が長い処理がある場合にのみメリットがあります。継続的に処理されているデータサーバの場合は、パフォーマンスに影響を与える可能性さえあります。 データベースの速度が遅いシステムでは、使用するスレッドをさらに少なくします。1～10 のさまざまな組み合わせを試してください。ネットワークサーバ用にシステムを最適化する場合、その数は 10～30の間の大きな値にします。
- agent_access (コンソールから設定できます) のようないくつかの設定パラメータは、Pandora FMS のパフォーマンスに多くの影響を与えます。

キャパシティ分析ツール (Capacity)

Pandora FMS には、ハードウエアおよびソフトウエアで、取得可能なデータ量を適切に計測できるいくつかのツールがあります。一つは、ダミーデータで直接データベースへアクセスするもの (dbstress) もう一つは、ダミーの XML ファイルを生成するもの (xml_stress) です。

Pandora FMS の XML 負荷

Pandora FMS エージェントから送られるような XML データファイルを生成する小さなスクリプトが以下にあります。

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

スクリプトは、テキストファイルからエージェント名を読み取り、設定ファイルに従って各エージェントの XML データファイルを生成します。ここで、モジュールはテンプレートとして定義されています。

モジュールの値はランダムな値になります。モジュールデータの初期値および変化率は設定可能です。

スクリプトは次のように実行します。

```
./pandora_xml_stress.pl <設定ファイル>
```

pandora_xml_stress.conf 設定ファイルの例を示します。

```
# Maximum number of threads, 10 by default.
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, /tmp by default.
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, 1.0 by default.
xml_version 1.0

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

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

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

# Agent interval, 300 by default.
agent_interval 300

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

```
# Data file generation end date, now by default.
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, 2 by default.
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, 41121 by default
# 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_3
module_type generic_proc
module_descripcion Module 3 description.
# Initial data.
module_data 1
module_end
```

エージェントのローカル設定の送受信

pandora_xml_stress.conf で、get_and_send_agent_conf を 1 に設定して実行した場合、テスト負荷エージェントで、通常のエージェントのように設定ファイルおよび md5 を送ることができます。

Enterprise 版の Pandora コンソールでは、pandora_xml_stress で実行する内容をリモートでの設定変更で行うことができます。pandora_xml_stress.conf ではなく Enterprise 版の Pandora コンソールから行った設定を利用するようになります。

ほかにも、pandora_xml_stress.conf 内の directory_confis にて、テストエージェントの設定ファイル .conf をどこに保存するかを設定できます。

設定ファイル

- max_threads スクリプトの実行スレッド数。処理率を改善します。
- agent_file 行ごとに名前を書いたファイルのパス。
- temporal 架空の XML データファイルを生成するディレクトリのパス。
- log_file スクリプトの実行について情報を出力するログのパス。
- xml_version XML データファイルのバージョン。(デフォルトは 1.0)
- encoding XML データファイルのエンコーディング。(デフォルトは ISO-8859-1)
- os_name 仮想エージェントの OS 名。(デフォルトは Linux)
- os_version 仮想エージェントの OS バージョン。(デフォルトは 2.6)
- agent_interval 仮想エージェントの実行秒間隔。(デフォルトは 300)
- time_from 仮想 XML データの開始時間。“ 年-月-日 時間:分:秒 ” というフォーマットにて。
- time_to 仮想 XML データの終了時間。“ 年-月-日 時間:分:秒 ” というフォーマットにて。
- get_and_send_agent_conf 0 または 1 の値です。有効な場合、仮想エージェントは、リモート設定により、より新しいエージェントの設定ファイルをダウンロードしようとします Pandora FMS Enterprise のコンソールから編集可能になります。
- startup_delay それぞれのエージェントがファイル生成を開始するまでの時間を秒で指定します。競合を回避するために利用します。
- timezone_offset タイムゾーンのオフセット値です。
- timezone_offset_range ランダムに指定した範囲でタイムゾーンを生成します。
- latitude_base 数値です。仮想エージェントを表示する緯度です。
- longitude_base 数値です。仮想エージェントを表示する経度です。
- altitude_base 数値です。仮想エージェントを表示する高度です。
- position_radius 数値です。指定した半径内の円に仮想エージェントがランダムに表示されます。

モジュール定義

スクリプト設定の中に一つのモジュール定義があり、リモート設定を有効にした場合も一緒で、次の通りです。

```
module_begin
module_name <モジュール名>
module_type <タイプ, 例: generic_data>
module_description <説明>
module_exec type=<type_generation_xml_stress>;<; 区切りの他のオプション>
module_unit <ユニット>
module_min_critical <値>
module_max_critical <値>
module_min_warning <値>
module_max_warning <値>
module_end
```

それぞれの項目は次のように設定可能です。

- type_generation_xml_stress: RANDOM,SCATTER,CURVEのいずれかを設定できます。
- module_attenuation <値>: モジュールの値を指定した値で掛け合わせます。通常 0.1 と 0.9 の間です。
- module_attenuation_wdays <値> <値> … <値>: 指定した日にのみモジュールの値を計算します。日曜(0) から土曜(6) を指定できます。例えば、以下のモジュールでは、土曜と日曜にネットワークトラフィックを 50% にします。

```
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 <値>: 1に設定すると、モジュールの以前の値を常に新たな値に加えます。値が増え続ける機能です。
- その他: 実効タイプに依存して、どのようなオプションがあるかは以下を確認してください。

min_critical, max_critical, min_warning および
max_warning は、バージョン 5.0 以上にのみであることに
注意してください。

RANDOM

次のオプションがあります。

- variation 前回の値から変化が発生する可能性を % で指定します。
- min 値の最小値を指定します。
- max 値の最大値を指定します。

Numeric

min と max の間の数値をランダムに生成します。

Boleans

0 または 1 の値を生成します。

String

min と max の間の長さの文字列を生成します。文字は、A から Z の間のランダムで、大文字、小文字を含み、また数字や記号を含みます。

外部データソース (SOURCE)

データのソースとしてプレーンテキストファイルを利用することができます。次のオプションがあります。

- src: ソースデータファイル

ファイルは、1行に1データを含む形式で、行数に制限はありません。例えば次の通りです。

```
4
5
6
10
```

二種類のデータ(数値と文字列)を扱うことができます。これらのモジュールは、ファイルのデータ順番に読み込んでPandora でのモジュールデータを生成します。上記のデータの例では、次のように表示されます。

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

SCATTER

数値データの場合のみ有用で、ハートビートのようなグラフを生成します。これは通常の値で、ある時間に *“beat”* を出します。

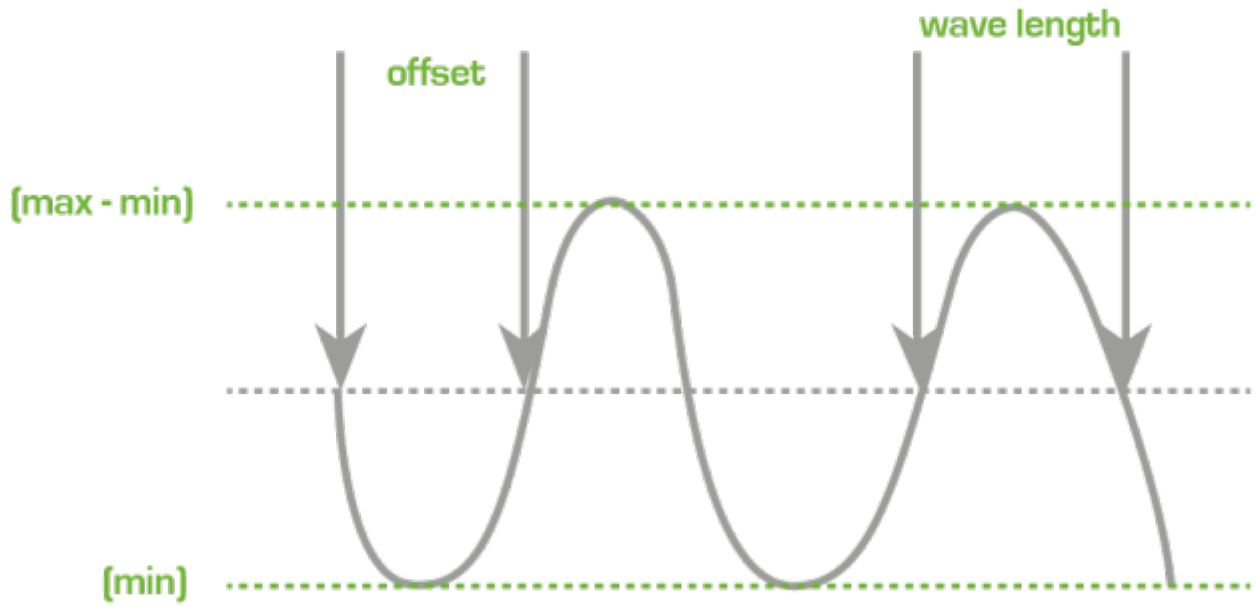
次のオプションがあります。

- min とりうる最小の値。
- max とりうる最大の値。
- prob *“beat”* を生成する頻度を % で指定します。
- avg *“beat”* が無い場合に、デフォルトで表示する平均値。

CURVE

三角関数を使った曲線でモジュールデータを生成します。 次のオプションがあります。

- min とりうる最小の値。
- max とりうる最大の値。
- time_wave_length 山が出現する時間。
- time_offset モジュールの値が 0 の時の波形の開始タイミングを秒で指定します。(正弦波に似ています)



注意事項

- このツールは、すべてのエージェントで、300秒から 30日の間隔を使用する “random”“curve”またはブール名のモジュールを検索するように事前設定されています。

データサーバの処理能力の計測方法

pandora_count.sh というスクリプトが Pandora FMS サーバパッケージの /usr/share/pandora_server/util/ ディレクトリにあります。このスクリプトは、データサーバの XML ファイル処理率を計測するのに利用します。このツールは、/var/spool/pandora/data_in に残っているファイルを利用します。そのため、未処理の数千のファイルを用意しておく (もしくは前述のツールで生成しておく) 必要があります。実行後は、CTRL+C で停止できます。

このスクリプトは、現在存在するファイルをカウントし、10秒前に処理したものを除外し、結果を 10 で割ることによって、1秒間の処理率を求めています。これは初歩的なソリューションですが、サーバの設定を修正するための情報を提供します。

Pandora FMS の DB 負荷

データベースパフォーマンスを確認するためのツールがあります。これはまた、架空のデータを定

期的もしくは不定期に生成するためにも利用できます。エージェントを作成し、このツールを使って自動的にデータを挿入するためのモジュールを作成しておく必要があります。

- *random*: 不定期データを生成します。
- *curve*: 三角関数を利用して曲線データを生成します。異なる間隔で補完を見るのに便利です。
- *boolean*: ランダムな boolean データを生成します。

random, *curve* および *boolean* という語を含む任意の名前を使うことができます。

- random_1
- curve_other

data_server モジュールのみ選択することができます。

Pandora FMS の DB 負荷ツールの調整

このツールは、すべてのエージェントから、*random*、*curve* または *boolean* という名前がついかモジュールを検索するようにあらかじめ設定されています。また、実行間隔は 300秒から 30日の間を指定できます。

もしこの設定を変更したい場合は、*pandora_dbstress* スクリプトを編集し、ファイルの先頭にあるいくつかの変数を変更する必要があります。

```
# 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. 最初の行の target_module の設定では、対象のモジュールを指定するか -1 を指定すると全てが対象になります。
2. 2行目の target_agent は、エージェントの指定です。
3. 3行目の target_interval は、秒単位で実行間隔を指定します。
4. 4行目の target_days は、現在日時から何日後までかを表します。

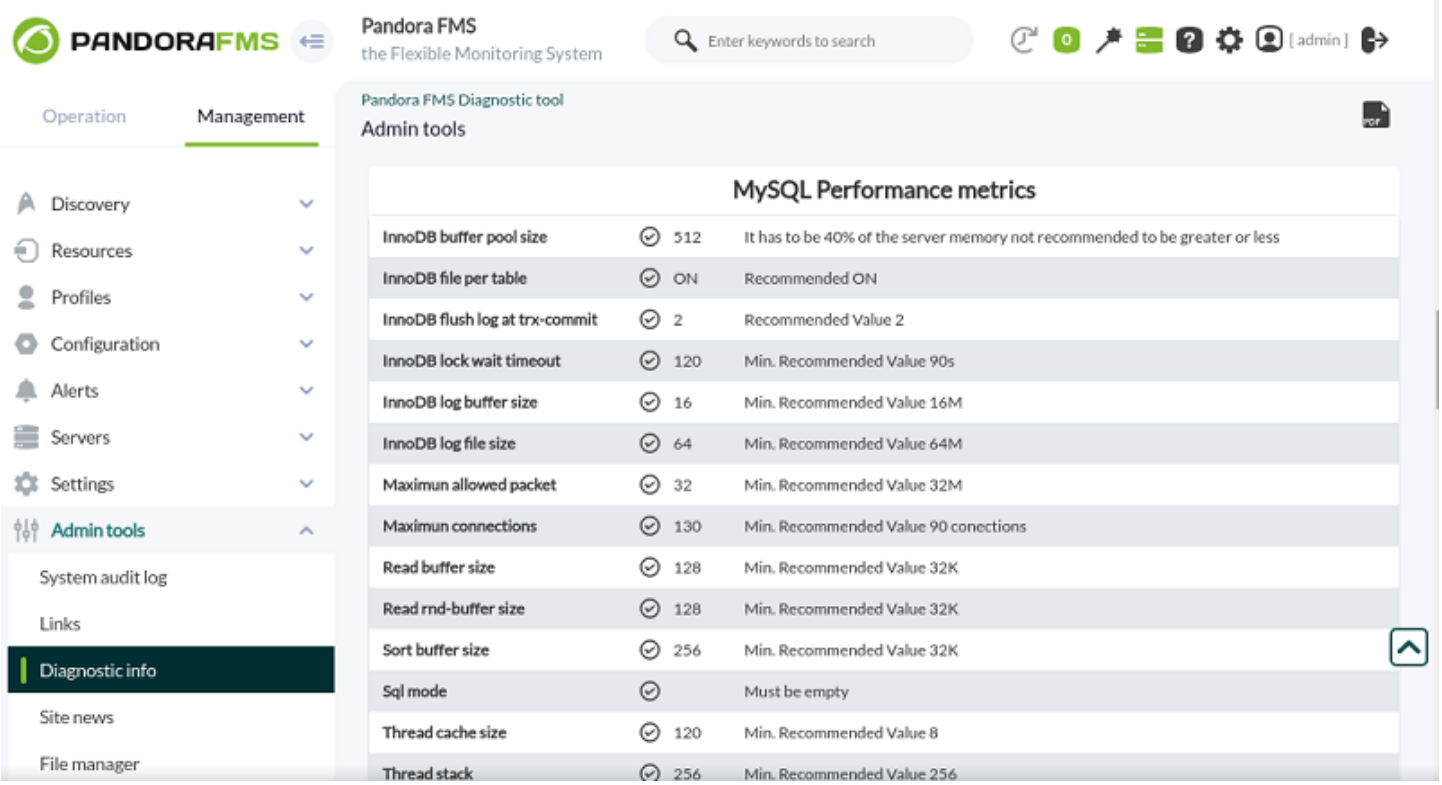
Pandora FMS の診断ツール

時々、ユーザが問題に遭遇し、Pandora の開発者もユーザのシステムについての詳細情報が無くて手助けできないことがあります。バージョン 3.0 では、ユーザの問題を解決する手助けとなる 2つの小さなツールを作成しました。

診断情報

Pandora FMS の最新バージョンには、Pandora FMS のインストールに関する診断情報を取得する機能があります。

それは、管理ツール(Admin tools) → 診断情報(Diagnostic Info) にあります。



このウインドウでは、Pandora FMS と MySQL の設定情報を見ることができるのに加えて、自己監視システムのグラフを見ることができます。

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'
```

/usr/share/pandora_server/util にあるツールで、システムの多くの情報を取得します。

- CPU 情報
- Uptime および CPU ロードアベレージ
- Memory 情報
- Kernel/Release 情報
- mysql 設定ファイルの内容
- PandoraFMS サーバ設定ファイルの内容 (パスワードはフィルタリングされます)
- Pandora FMS ログ情報 (ただし、全ログではありません)
- Disk 情報
- Pandora FMS プロセス情報

- kernel ログ情報 (dmesg)

すべての情報は、.txt ファイル内に生成されるので、この情報を手助けしてくれる人に送信することができます。たとえば Pandora FMS ユーザフォーラムや Pandora FMS public メーリングリストなどです。この情報には、機密情報は含まれません。*pandora_server.conf* および *my.cnf* ファイルの内容を取得したい場合は、root 権限で実行する必要があることに注意してください。

以下に実行例を示します。

```
$ ./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'
```

また、出力ファイルの例を示します。

```
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
```

1)
http://optimmysql.blogspot.com/2008/04/variable-day-out-5-innodbthreadconcurre.html
2)
[http://mysqlha.blogspot.com/2010/03/do-we-still-need-innodbthreadconcurrenc.html]