PANDORAFMS
ENTERPRISE

Pandora FMS
Administrator Manual
MySQL Server Monitoring in Unix

Administrator Manual Monitorización SQL Server

Indice

1Changelog....3
2Introduction....4
3Requirements....5
3.1.Documentation the Area that requires the monitoring should report....5
4Compatibility Matrix....6
5Agent modules clasification....7
6Instalation....13
7Monitoring....14
8Plugin parametrization....15
8.1. MySQL Access Configuration and General data....15
8.1.1. MySQL user, password and host....15
8.1.2. General Data....15
8.2.System Parameters....16
8.2.1.System Checking Block....16
8.2.2.Checking the connectivity with the database....16
8.2.3.Check if the MySQL process is active....16
8.2.4.Server memory Check (process)....16
8.2.5. Number of TIME_WAIT connections in the system....17
8.2.6.Server disk space check....17
8.2.7.Size of the ibdata1 File....17
8.2.8.Searching errors in the error logs of the database....17
8.3.Performance parameters....18
8.3.1. Block of performance check....18
8.3.2. Number of MySQL active connections....18
8.3.3. Number of aborted connections due to the client did not close correctly the connection....18
8.3.4. Number of bytes get by the clients....18
8.3.5. Number of bytes sent by the clients....19
8.3.6. Information of the server status (SHOW GLOBAL STATUS ó SHOW STATUS)....19
8.3.7. Information of the InnoDB status (SHOW INNODB STATUS)....19
8.3.8. Number of insertions in the data base....19
8.3.9.Number of locks on tables in the database when doing a transaction....20
8.3.10.Active locks on tables and registers for each active sesion....20
8.3.11.I/0 Requests pending....20
8.3.12. Total size of data in GB....20
8.4.Monitoring via SQL....21
8.5.Execution of commands depending on conditions....21
8.6.Parameterization of the command associated to a check....22
8.7.Module status....22
8.8.Kind of data....22
8.9.Data management in checks of kind of performance....23

1 CHANGELOG

Date	Author	Change	Version
12/03/12	Tomas	First Version	v1r1
01/07/13	Mario P.	Changes in code	V1r2
25/04/14	Mario P.	Changes in code	V1r3

2 INTRODUCTION

This document has as main aim the description of the MySQL database monitoring on Unix. A series of " base" modules has been chosen on the basis of our experience in the system monitoring and on the requirements of some of our clients. Besides, all the specifications collected in different real production environments have been added, taking real specifications of database administrators.

For the information extraction we use:

- **An external configuration file** where all the plugin parameterization is defined. This configuration file could do calls (includes) to other files.
- **The software already installed in the system** is used (MySQL, system commands, MySQL alerts files, ect) for the monitoring done by the plugin without having to install libraries or third party utilities.
- **An existing log parser is used** (the one from Pandora) to process the MySQL alert logs.This parser should be "automatic" and should be based in the report of all the critical error messages with the "ERROR"* form.
- **A serie of basic checks "by default"** are done, but they could be deleted or customized.
- **An "open" interface is available to specify SQL free queries**, allowing to do all kind of SQL queries that are done with other tools or in a manual way by the administrators.
- The system is integrated with the Unix agent and with the capacity of distributing file collections, so the plugin could be distributed by one side, and the file colections by individual way-by agent-and/or by policy.

It is important to mention, that same as with the rest of the montoring with PandoraFMS, the MySQL monitoring plugin could be used to collect information kind "text string" (to manage it as events) or kind numeric (to do performance management).

3 REQUIREMENTS

The requirements for doing that this monitoring works properly are the following:

- To install the Pandora FMS agent.
- To install Perl
- To have one MySQL database installed in the machine where it is going to monitor, with connection to this database.
- To specify the name, user, password and host of the MySQL database.
- It is necessary that the user whith which the Pandora FMS agent is executed, that is, the user that will execute the plugin, will have access to the following MySQL resources:
  - MySQL homedir Directory(directory with the MySQL , usually /var/lib/mysql).
  - MySQL log file (usually /var/lib/mysql/mysql.log).
  - Access and writting permissions to a directory for temporary files.

3.1. Documentation the Area that requires the monitoring should report.

For the right MySQL monitoring, it should be necessary that the Technical Area sends some specific information that will be included in the configuration files. These information is the following:

- User, password, host with access to the MySQL database. This user should have permission to read all the tables from which he want to extract information. This user should have pemissions to read all the tables from which he want to extract information.
- Log files and MySQL base directory.
- If you want to monitor some item that does not come defined “by default”, then it will be necessary that you provide that SQL code to do this monitoring, and also an exit data example, specifying if it is numeric, kind string, etc).

4 Compatibility Matrix

The agent compatibility matrix is the following:

Systems where it has been tested	• Ubuntu 10.04 with MySQL 5.0 • OpenSuse 11.2 with MySQL 5.5

Systems where it should work	• Systems “Unix like” with MySQL 5.0 or newer

5 Agent modules clasification

Most of these parameters must be processed as "delta" or using the data type "generic_data_inc" as they are counters.

- Aborted_connects
- Binlog_cache_disk_use
- Binlog_cache_use
- Binlog_stmt_cache_disk_use
- Binlog_stmt_cache_use
- Bytes_received
- Bytes_sent
- Com_admin_commands
- Com_assign_to_keycache
- Com_alter_db
- Com_alter_db_upgrade
- Com_alter_event
- Com_alter_function
- Com_alter_procedure
- Com_alter_server
- Com_alter_table
- Com_alter_tablespace
- Com_analyze
- Com_begin
- Com_binlog
- Com_call_procedure
- Com_change_db
- Com_change_master
- Com_check
- Com_checksum
- Com_commit
- Com_create_db
- Com_create_event
- Com_create_function
- Com_create_index
- Com_create_procedure
- Com_create_server
- Com_create_table
- Com_create_trigger
- Com_create_udf
- Com_create_user
- Com_create_view
- Com_dealloc_sql
- Com_delete
- Com_delete_multi
- Com_do
- Com_drop_db
- Com_drop_event
- Com_drop_function
- Com_drop_index
- Com_drop_procedure

- Com_drop_server
- Com_drop_table
- Com_drop_trigger
- Com_drop_user
- Com_drop_view
- Com_empty_query
- Com_execute_sql
- Com_flush
- Com_grant
- Com_ha_close
- Com_ha_open
- Com_ha_read
- Com_help
- Com_insert
- Com_insert_select
- Com_install_plugin
- Com_kill
- Com_load
- Com_lock_tables
- Com_optimize
- Com_preload_keys
- Com_prepare_sql
- Com_purge
- Com_purge_before_date
- Com_release_savepoint
- Com_rename_table
- Com_rename_user
- Com_repair
- Com_replace
- Com_replace_select
- Com_reset
- Com_resignal
- Com_revoke
- Com_revoke_all
- Com_rollback
- Com_rollback_to_savepoint
- Com_savepoint
- Com_select
- Com_set_option
- Com_signal
- Com_show_authors
- Com_show_binlog_events
- Com_show_binlogs
- Com_show_charsets
- Com_show_collations
- Com_show_contributors
- Com_show_create_db
- Com_show_create_event
- Com_show_create_func
- Com_show_create_proc
- Com_show_create_table
- Com_show_create_trigger

- Com_show_databases
- Com_show_engine_logs
- Com_show_engine_mutex
- Com_show_engine_status
- Com_show_events
- Com_show_errors
- Com_show_fields
- Com_show_function_status
- Com_show_grants
- Com_show_keys
- Com_show_master_status
- Com_show_open_tables
- Com_show_plugins
- Com_show_privileges
- Com_show_procedure_status
- Com_show_processlist
- Com_show_profile
- Com_show_profiles
- Com_show_relaylog_events
- Com_show_slave_hosts
- Com_show_slave_status
- Com_show_status
- Com_show_storage_engines
- Com_show_table_status
- Com_show_tables
- Com_show_triggers
- Com_show_variables
- Com_show_warnings
- Com_slave_start
- Com_slave_stop
- Com_stmt_close
- Com_stmt_execute
- Com_stmt_fetch
- Com_stmt_prepare
- Com_stmt_reprepare
- Com_stmt_reset
- Com_stmt_send_long_data
- Com_truncate
- Com_uninstall_plugin
- Com_unlock_tables
- Com_update
- Com_update_multi
- Com_xa_commit
- Com_xa_end
- Com_xa_prepare
- Com_xa_recover
- Com_xa_rollback
- Com_xa_start
- Compression
- Connections
- Created_tmp_disk_tables
- Created_tmp_files

- Created_tmp_tables
- Delayed_errors
- Delayed_insert_threads
- Delayed_writes
- Flush_commands
- Handler_commit
- Handler_delete
- Handler_discover
- Handler_prepare
- Handler_read_first
- Handler_read_key
- Handler_read_last
- Handler_read_next
- Handler_read_prev
- Handler_read_rnd
- Handler_read_rnd_next
- Handler_rollback
- Handler_savepoint
- Handler_savepoint_rollback
- Handler_update
- Handler_write
- Innodb_buffer_pool_pages_data
- Innodb_buffer_pool_pages_dirty
- Innodb_buffer_pool_pages_flushed
- Innodb_buffer_pool_pages_free
- Innodb_buffer_pool_pages_misc
- Innodb_buffer_pool_pages_total
- Innodb_buffer_pool_read_ahead_rnd
- Innodb_buffer_pool_read_ahead
- Innodb_buffer_pool_read_ahead_evicted
- Innodb_buffer_pool_read_requests
- Innodb_buffer_pool_reads
- Innodb_buffer_pool_wait_free
- Innodb_buffer_pool_write_requests
- Innodb_data_fsyncs
- Innodb_data_pending_fsyncs
- Innodb_data_pending_reads
- Innodb_data_pending_writes
- Innodb_data_read
- Innodb_data_reads
- Innodb_data_writes
- Innodb_data_written
- Innodb_dblwr_pages_written
- Innodb_dblwr_writes
- Innodb_have_atomic_builtins
- Innodb_log_waits
- Innodb_log_write_requests
- Innodb_log_writes
- Innodb_os_log_fsyncs
- Innodb_os_log_pending_fsyncs
- Innodb_os_log_pending_writes
- Innodb_os_log_written

- Innodb_page_size
- Innodb_pages_created
- Innodb_pages_read
- Innodb_pages_written
- Innodb_row_lock_current_waits
- Innodb_row_lock_time
- Innodb_row_lock_time_avg
- Innodb_row_lock_time_max
- Innodb_row_lock_waits
- Innodb_rows_deleted
- Innodb_rows_inserted
- Innodb_rows_read
- Innodb_rows_updated
- Innodb_truncated_status_writes
- Key_blocks_not_flushed
- Key_blocks_unused
- Key_blocks_used
- Key_read_requests
- Key_reads
- Key_write_requests
- Key_writes
- Last_query_cost
- Max_used_connections
- Not_flushed_delayed_rows
- Open_files
- Open_streams
- Open_table_definitions
- Open_tables
- Opened_files
- Opened_table_definitions
- Opened_tables
- Performance_schema_cond_classes_lost
- Performance_schema_cond_instances_lost
- Performance_schema_file_classes_lost
- Performance_schema_file_handles_lost
- Performance_schema_file_instances_lost
- Performance_schema_locker_lost
- Performance_schema_mutex_classes_lost
- Performance_schema_mutex_instances_lost
- Performance_schema_rwlock_classes_lost
- Performance_schema_rwlock_instances_lost
- Performance_schema_table_handles_lost
- Performance_schema_table_instances_lost
- Performance_schema_thread_classes_lost
- Performance_schema_thread_instances_lost
- Prepared_stmt_count
- Qcache_free_blocks
- Qcache_free_memory
- Qcache_hits
- Qcache_inserts
- Qcache_lowmem_prunes
- Qcache_not_cached

- Qcache_queries_in_cache
- Qcache_total_blocks
- Queries
- Questions
- Rpl_status
- Select_full_join
- Select_full_range_join
- Select_range
- Select_range_check
- Select_scan
- Slave_heartbeat_period
- Slave_open_temp_tables
- Slave_received_heartbeats
- Slave_retried_transactions
- Slave_running
- Slow_launch_threads
- Slow_queries
- Sort_merge_passes
- Sort_range
- Sort_rows
- Sort_scan
- Ssl_accept_renegotiates
- Ssl_accepts
- Ssl_callback_cache_hits
- Ssl_cipher
- Ssl_cipher_list
- Ssl_client_connects
- Ssl_connect_renegotiates
- Ssl_ctx_verify_depth
- Ssl_ctx_verify_mode
- Ssl_default_timeout
- Ssl_finished_accepts
- Ssl_finished_connects
- Ssl_session_cache_hits
- Ssl_session_cache_misses
- Ssl_session_cache_mode
- Ssl_session_cache_overflows
- Ssl_session_cache_size
- Ssl_session_cache_timeouts
- Ssl_sessions_reused
- Ssl_used_session_cache_entries
- Ssl_verify_depth
- Ssl_verify_mode
- Ssl_version
- Table_locks_immediate
- Table_locks_waited
- Tc_log_max_pages_used
- Tc_log_page_size
- Tc_log_page_waits
- Threads_cached
- Threads_connected
- Threads_created

- Threads_running
- Uptime
- Uptime_since_flush_status

6 Instalation

Copy the plugin to the agent plugin directory, or distribute it with file collections. Do the same with the conf.file The call from the agent will be similar to this, but using the paths where the plugin and the conf would be installed.

```
module_plugin perl /etc/pandora/plugins/Pandora_Plugin_mysqlserver_v1r1.pl
/etc/pandora/plugins/mysql.conf
```

7 MONITORING

The plugin monitors “by default” the following things:

- Checking of connectivity with the database.
- Checks if the MySql process is active.
- Checks the server memory (process) .
- Number of connections TIME_WAIT in the system .
- Checks the space in the server disk (usually/var/lib/mysql).
- Size of the file ibdata1 .
- Search of errors in the database error logs (usually /var/lib/mysql/mysql.log).

Besides, it also monitors the following performance parameters:

- Number of MySQL active connections.
- Time of server activity (uptime).
- Number of aborted connections because of which the client did not close correctly the connection.
- Number of bytes got by the clients.
- Information about the server status (SHOW GLOBAL STATUS or SHOW STATUS).
- Information about the status of InnoDB (SHOW INNODB STATUS) .
- Number of bytes sent by clients.
- Number of database entries.
- Number of locks on tables on the database when doing a transaction.
- Active locks on tables and registers for each active session.
- Not answered queries from I/0
- Total data size in GB.

All these modules comes parametrized in the “mysql.conf” file that comes with the plugin package. These modules could be deleted or extended by a MySQL administrator.

8 PLUGIN PARAMETRIZATION

The plugin is used after having previously configured the configuration external file.

> ***NOTE:*** *It is extremely important to consider that the configuration files that are though for the plugin in UNIX should be edited and stored with carriage return kind "UNIX" and that carriage return kind "WINDOWS" are used, then the plugin will be not work correctly.*

There are three functional blocks in the configuration file:

8.1. MySQL Access Configuration and General data

In order that the plugin could monitor the database, you should give the access credentials and some general data that after will be used in the checkings:

8.1.1. MySQL user, password and host

The access credentials should be fulfill in the following way:

```
conf_mysql_user mysql

conf_mysql_pass 1234

conf_mysql_host 127.0.0.1
```

There is the possibility that we have a blank password mysql user. When mysql pass is empty comment the line (#conf_mysql_pass).

8.1.2. General Data

Here should be specifyed the MySQL home directory (usually /var/lib/mysql), the log file, one directory for temporary files and the complete path to the plugin of Pandora logs parse:

```
conf_mysql_homedir /var/lib/mysql
conf_mysql_basedir /var/lib/mysql
conf_mysql_logfile /var/log/mysql.log
conf_temp /tmp
conf_logparser /etc/pandora/plugins/grep_log
```

8.2. System Parameters

Next are described the system specific checking modules

8.2.1. System Checking Block

A block of system checking in the configuration file is like this:

```
check_begin
# Linea de comentario.
<token of system check>
check_end
```

Next are described the plugin checks:

8.2.2. Checking the connectivity with the database

With this check, it is posible to check if the database has connectivity with other software elements. If this check is not satifactory, then **the monitoring will be aborted.**

```
check_begin
check_mysql_service
check_end
```

8.2.3. Check if the MySQL process is active

This check verifies that the MySQL process is active in the system:

```
check_begin
check_mysql_service
check_end
```

8.2.4. Server memory Check (process)

Throug this check we verify the MySQL server memory usage.

```
check_begin
check_mysql_memory
check_end
```

8.2.5. Number of TIME_WAIT connections in the system

It shows the number of connections in TIME_WAIT status in the system:

```
check_begin
check_system_timewait
check_end
```

8.2.6. Server disk space check

Checks the space in the disk (in KB) by the database. It will be check the size of the homedir MySQL directory specifying in the General data section:

```
check_begin
check_system_diskusage
check_end
```

8.2.7. Size of the ibdata1 File

Checks the size (in KB) of the ibdata 1 file that is at the homedir MySQLdirectory:

```
check_begin
check_mysql_ibdata1
check_end
```

8.2.8. Searching errors in the error logs of the database

It will search the "ERROR" string in the log file specified in the General data section. This module will always return the data kind async_string, and in case that the check get return data, the associated command will be executed (explained in the section 8.5. Executing comands on condition):

```
check_begin
check_mysql_logs
check_end
```

8.3. Performance parameters

Next are described the modules of performance check.

8.3.1. Block of performance check

Similar to the system checks as we show next:

```
check_begin
# Linea de comentario.
<token of commentary check>
check_end
```

Next are show the performance check blocks:

8.3.2. Number of MySQL active connections

It returns the number of active connections in the database:

```
check_begin
mysql_status Full processlist
check_end
```

8.3.3. Number of aborted connections due to the client did not close correctly the connection.

```
check_begin
mysql_status Aborted_connects
check_end
```

8.3.4. Number of bytes get by the clients.

```
check_begin
mysql_status Bytes_received
check_end
```

8.3.5. Number of bytes sent by the clients

```
check_begin
mysql_status Bytes_sent
check_end
```

8.3.6. Information of the server status (SHOW GLOBAL STATUS ó SHOW STATUS)

These queries are based on the MySQL command SHOW GLOBAL STATUS. It is posible to do searchs of some token and its associated value inside the data returned by this command.

```
check_begin
mysql_status <token to search>
check_end
```

8.3.7. Information of the InnoDB status (SHOW INNODB STATUS)

These queries are based on the MySQL command SHOW INNODB STATUS. Between the data returned by this command it is posible to search some token and its associated value:

```
check_begin
check_name <check_name>
mysql_innodb <token to search in innodb status>
check_end
```

8.3.8. Number of insertions in the data base

Number of transactions kind INSERT done in the database:

```
check_begin
mysql_status Com_insert
check_end
```

8.3.9. Number of locks on tables in the database when doing a transaction

```
check_begin
mysql_status Com_lock_tables
check_end
```

8.3.10. Active locks on tables and registers for each active sesion

```
check_begin
# Number of locks over DB tables
mysql_status Table_locks_waited
check_end

check_begin
# Number of row locks
mysql_status Innodb_row_lock_waits
check_end
```

8.3.11. I/0 Requests pending

```
check_begin
mysql_status Pending_io
check_end
```

8.3.12. Total size of data in GB

```
check_begin
mysql_status Total_size
check_end
```

8.4. Monitoring via SQL

One of the most powerful features of the plugin is the posibility of specify its own SQL order to get the value. Lets see some example:

```
check_begin
check_name num_tables
check_schema information_schema
check_sql SELECT COUNT(*) FROM tables
check_end
```

Check_name The name of the check that will be shown on the Pandora interface.

Check_sql The query that should return a simple data (number or string).

Check_schema The MySQL schema to which the plugin will be connectedin order to do the query.

8.5. Execution of commands depending on conditions

In all modules it would be posible to specify the execution of one command if one condition is fulfilled. These conditions could be:

- == (equal to one value)
- != (Different from one value)
- < (lower than a given value)
- > (Bigger than a given value).

If the condition is fullfilled, then the configured command will be executed with the token *post_execution*. In case of the *check_mysql_logs* check, the command will be executed if this check returns some data.

One example of this configuration is this:

```
check_begin
check_mysql_cpu
post_condition > 95
post_execution snmptrap -v 1 -c public x.x.x.x 1.2.4.4.65.6.4.3 6 128
check_end
```

8.6. Parameterization of the command associated to a check

As we have mentioned before, in each check one command to execute could be configured if the condition is fulfilled. Besides, it is posible to use one token in the command in order that if this is being executed, then it would be remplaced by the check value. For example, through the _DATA_ macro in this example, the content of the check is saved on */tmp/mysql_cpu_result* the file:

```
check_begin
check_mysql_cpu
post_condition > 95
post_execution echo _DATA_ >> /tmp/mysql_cpu_result
check_end
```

8.7. Module status

Besides, the modules will return a status if it is required with the configuration token:

```
post_status WARNING
```

O also:

```
post_status CRITICAL
```

8.8. Kind of data

By default, all monitors will return *generic_data*, unless that the following configuration token would be shown.

```
module_type generic_data_inc
module_type async_string
(or other valid)
```

The only exception is the monitor **check_mysql_service** with automatically one module kind *generic_proc*.

8.9. Data management in checks of kind of performance

In checks of kind of performance and SQL queries, it is posible to configure it in order that the absolute value would be returned, as MySQL *data_absolute*) returns it (or the difference between the actual data and the previous one (*data_delta*).**In the first execution of the check** configured with *data_delta,* **no data will be returned.** From the second execution on , the increase will be returned.

If, for example, should be the case that **the difference between the current data an the previous one would be negative**, then **the value of the check will be reseted** and the check will not return any data.

By default the checks of kind of performance will be configured as *data_absolute*.. For example, this check will return the increase of the active ssesions in MySQL:

```
check_begin
mysql_status Full processlist
data_delta
check_end
```