Optimisation de base de données

Release 1.0

F. Conil

January 25, 2012

CONTENTS

Contents:

CHAPTER

ONE

AMÉLIORER LES PERFORMANCES MYSQL

1.1 Documentation MySQL

Commencer par lire le chapitre Optimization, http://dev.mysql.com/doc/refman/5.1/en/optimization.html

Lire la présentation suivante, http://www.slideshare.net/freshdaz/optimisation-de-mysql (O. Dasini)

Jeter un oeil à celle-ci http://en.oreilly.com/mysql2011/public/schedule/detail/17111 ?

Le blog de référence semble être : http://www.mysqlperformanceblog.com/. C'est le blog de la société Percona dont l'expertise porterait sur **InnoDB** d'après un extrait d'article sur buildout qui évoque MySQL.

Il y a un site http://dba.stackexchange.com/

1.2 Views

http://stackoverflow.com/questions/4789332/is-a-mysql-view-faster-than-a-normal-query

1.2.1 Materialized Views

http://stackoverflow.com/questions/93539/what-is-the-difference-between-views-and-materialized-views-in-oracle

http://stackoverflow.com/questions/tagged/materialized-views+mysql

A.7.5: Does MySQL 5.1 have materialized views?

No.

1.3 Slow queries

Quand on tape la commande **status**, elle indique 38 requêtes lentes.

Il semble qu'on puisse activer le log des requêtes lentes en "dé-commentant" la ligne correspondant à **log_slow_queries** dans */etc/mysql/my.cnf*

Les requêtes et le temps d'exécution sont alors stockées dans */var/log/mysql/*.

source : http://dev.mysql.com/doc/refman/5.1/en/slow-query-log.html

Il va maintenant falloir analyser ces requêtes et voir si on peut les optimiser.

1.4 Permissions

J'ai noté "Try to reduce permission-checking" or on a un GRANT pour quelques tables qui sont remplies avec les données de laboratoires extraites de HAL.

http://dev.mysql.com/doc/refman/5.1/en/select-optimization.html

La table w_referentielliris est utilisée dans la requête publication pour récupérer l'affiliation des auteurs.

Est-ce que ça impacte les performances des requêtes ? Dans quelle proportion ? Les requêtes qui ne référencent pas les tables concernées sont-elles impactées ?

1.5 Finding the biggest tables

The biggest tables are often the most promising candidates for optimization. This recipe shows you how to get an overview of the largest tables in your installation.

```
use information_schema;
select TABLE_SCHEMA, TABLE_NAME, (INDEX_LENGTH + DATA_LENGTH)/(1024*1024) AS SIZE_MB, TABLE_ROWS fro
```

L'exécution de cette requête a montré que la table w_admin_publis atteignait 1Go !

source : *MySQL Admin Cookbook*

1.6 Analyser les requêtes avec EXPLAIN

http://www.slideshare.net/phpcodemonkey/mysql-explain-explained

http://blip.tv/phpnw/phpnw08-track-2-talk-1-adrian-hardy-mysql-explain-explained-1801203

http://www.mysqlperformanceblog.com/2006/07/24/mysql-explain-limits-and-errors/

http://www.xaprb.com/blog/2006/10/12/how-to-profile-a-query-in-mysql/

1.7 Query Cache

Le premier article est probablement une reprise du second mais il est plus simple d'abord :

1. http://rackerhacker.com/2007/08/08/mysqls-query-cache-explained/
2. http://www.mysqlperformanceblog.com/2006/07/27/mysql-query-cache/

1.7.1 Premiers tests sur le serveur LIRIS

Les résultats ci-après proviennent du serveur LIRIS.

Com_select semble dépendre de la session et du coup je ne sais pas comment récupérer cette information "globalement".

```
mysql> show status like "%qcache%";
+-------------------------+---------+
| Variable_name           | Value   |
+-------------------------+---------+
| Qcache_free_blocks      | 48      |
| Qcache_free_memory      | 6592888 |
| Qcache_hits             | 2926832 |
| Qcache_inserts          | 1094729 |
| Qcache_lowmem_prunes    | 935081  |
| Qcache_not_cached       | 848156  |
| Qcache_queries_in_cache | 1576    |
| Qcache_total_blocks     | 3470    |
+-------------------------+---------+

mysql> show status like "%com_sel%";
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| Com_select    | 1     |
+---------------+-------+

mysql> show variables like "%query%";
+------------------------------+--------------------------------+
| Variable_name                | Value                          |
+------------------------------+--------------------------------+
| ft_query_expansion_limit     | 20                             |
| have_query_cache             | YES                            |
| long_query_time              | 2.000000                       |
| query_alloc_block_size       | 8192                           |
| query_cache_limit            | 1048576                        |
| query_cache_min_res_unit     | 4096                           |
| query_cache_size             | 16777216                       |
| query_cache_type             | ON                             |
| query_cache_wlock_invalidate | OFF                            |
| query_prealloc_size          | 8192                           |
| slow_query_log               | ON                             |
| slow_query_log_file          | /var/log/mysql/mysql-slow.log  |
+------------------------------+--------------------------------+
```

1.7.2 Récupération correcte des valeurs de variables

La récupération des variables "courantes" se fait par **mysqladmin**. Tests sur lirisped qui n'a pas forcément exactement la même configuration que le serveur LIRIS.

```
$ mysqladmin -u dbuser -p extended-status | grep "Qcache_"
Enter password:
| Qcache_free_blocks                | 1         |
| Qcache_free_memory                | 14864992  |
| Qcache_hits                       | 1860      |
| Qcache_inserts                    | 9         |
| Qcache_lowmem_prunes              | 0         |
| Qcache_not_cached                 | 252       |
| Qcache_queries_in_cache           | 9         |
| Qcache_total_blocks               | 24        |

$ mysqladmin -u dbuser -p extended-status | grep "Com_"
Enter password:
```

```
| Com_admin_commands                | 9         |
| Com_assign_to_keycache            | 0         |
| Com_alter_db                      | 0         |
| Com_alter_db_upgrade              | 0         |
| Com_alter_event                   | 0         |
| Com_alter_function                | 0         |
| Com_alter_procedure               | 0         |
| Com_alter_server                  | 0         |
| Com_alter_table                   | 0         |
| Com_alter_tablespace              | 0         |
| Com_analyze                       | 4         |
| Com_backup_table                  | 0         |
| Com_begin                         | 0         |
| Com_binlog                        | 0         |
| Com_call_procedure                | 0         |
| Com_change_db                     | 6         |
| Com_change_master                 | 0         |
| Com_check                         | 0         |
| Com_checksum                      | 0         |
| Com_commit                        | 0         |
| Com_create_db                     | 2         |
| Com_create_event                  | 0         |
| Com_create_function               | 0         |
| Com_create_index                  | 0         |
| Com_create_procedure              | 0         |
| Com_create_server                 | 0         |
| Com_create_table                  | 2         |
| Com_create_trigger                | 0         |
| Com_create_udf                    | 0         |
| Com_create_user                   | 0         |
| Com_create_view                   | 0         |
| Com_dealloc_sql                   | 0         |
| Com_delete                        | 0         |
| Com_delete_multi                  | 0         |
| Com_do                            | 0         |
| Com_drop_db                       | 1         |
| Com_drop_event                    | 0         |
| Com_drop_function                 | 0         |
| Com_drop_index                    | 0         |
| Com_drop_procedure                | 0         |
| Com_drop_server                   | 0         |
| Com_drop_table                    | 0         |
| Com_drop_trigger                  | 0         |
| Com_drop_user                     | 0         |
| Com_drop_view                     | 0         |
| Com_empty_query                   | 0         |
| Com_execute_sql                   | 0         |
| Com_flush                         | 7         |
| Com_grant                         | 0         |
| Com_ha_close                      | 0         |
| Com_ha_open                       | 0         |
| Com_ha_read                       | 0         |
| Com_help                          | 0         |
| Com_insert                        | 0         |
| Com_insert_select                 | 2         |
| Com_install_plugin                | 0         |
| Com_kill                          | 0         |
| Com_load                          | 0         |
```

```
| Com_load_master_data              | 0         |
| Com_load_master_table             | 0         |
| Com_lock_tables                   | 0         |
| Com_optimize                      | 0         |
| Com_preload_keys                  | 0         |
| Com_prepare_sql                   | 0         |
| Com_purge                         | 0         |
| Com_purge_before_date             | 0         |
| Com_release_savepoint             | 0         |
| Com_rename_table                  | 0         |
| Com_rename_user                   | 0         |
| Com_repair                        | 0         |
| Com_replace                       | 0         |
| Com_replace_select                | 0         |
| Com_reset                         | 0         |
| Com_restore_table                 | 0         |
| Com_revoke                        | 0         |
| Com_revoke_all                    | 0         |
| Com_rollback                      | 0         |
| Com_rollback_to_savepoint         | 0         |
| Com_savepoint                     | 0         |
| Com_select                        | 235       |
| Com_set_option                    | 3         |
| Com_show_authors                  | 0         |
| Com_show_binlog_events            | 0         |
| Com_show_binlogs                  | 0         |
| Com_show_charsets                 | 0         |
| Com_show_collations               | 0         |
| Com_show_column_types             | 0         |
| Com_show_contributors             | 0         |
| Com_show_create_db                | 0         |
| Com_show_create_event             | 0         |
| Com_show_create_func              | 0         |
| Com_show_create_proc              | 0         |
| Com_show_create_table             | 10        |
| Com_show_create_trigger           | 0         |
| Com_show_databases                | 17        |
| Com_show_engine_logs              | 0         |
| Com_show_engine_mutex             | 0         |
| Com_show_engine_status            | 2         |
| Com_show_events                   | 0         |
| Com_show_errors                   | 0         |
| Com_show_fields                   | 2010      |
| Com_show_function_status          | 0         |
| Com_show_grants                   | 2         |
| Com_show_keys                     | 9         |
| Com_show_master_status            | 0         |
| Com_show_new_master               | 0         |
| Com_show_open_tables              | 0         |
| Com_show_plugins                  | 0         |
| Com_show_privileges               | 0         |
| Com_show_procedure_status         | 0         |
| Com_show_processlist              | 1         |
| Com_show_profile                  | 6         |
| Com_show_profiles                 | 6         |
| Com_show_slave_hosts              | 0         |
| Com_show_slave_status             | 0         |
| Com_show_status                   | 19        |
```

```
| Com_show_storage_engines          | 0         |
| Com_show_table_status             | 0         |
| Com_show_tables                   | 20        |
| Com_show_triggers                 | 0         |
| Com_show_variables                | 13        |
| Com_show_warnings                 | 3         |
| Com_slave_start                   | 0         |
| Com_slave_stop                    | 0         |
| Com_stmt_close                    | 0         |
| Com_stmt_execute                  | 0         |
| Com_stmt_fetch                    | 0         |
| Com_stmt_prepare                  | 0         |
| Com_stmt_reprepare                | 0         |
| Com_stmt_reset                    | 0         |
| Com_stmt_send_long_data           | 0         |
| Com_truncate                      | 0         |
| Com_uninstall_plugin              | 0         |
| Com_unlock_tables                 | 0         |
| Com_update                        | 0         |
| Com_update_multi                  | 0         |
| Com_xa_commit                     | 0         |
| Com_xa_end                        | 0         |
| Com_xa_prepare                    | 0         |
| Com_xa_recover                    | 0         |
| Com_xa_rollback                   | 0         |
| Com_xa_start                      | 0         |

$ mysqladmin -u dbuser -p extended-status | grep "query_"
Enter password:
| Last_query_cost                   | 0.000000  |
```

Exécution de la “requête publis” :

```
$ mysql -DLabinfo_TEST2 -udbuser -p -e"source getpublis.sql"
```

Examen des variables après exécution de la requête.

```
$ mysqladmin -udbuser -p extended-status | grep "Qcache_"
Enter password:
| Qcache_free_blocks                | 1         |
| Qcache_free_memory                | 14864992  |
| Qcache_hits                       | 1860      |
| Qcache_inserts                    | 9         |
| Qcache_lowmem_prunes              | 0         |
| Qcache_not_cached                 | 255       |
| Qcache_queries_in_cache           | 9         |
| Qcache_total_blocks               | 24        |

$ mysqladmin -udbuser -p extended-status | grep "Com_"
Enter password:
| Com_admin_commands                | 9         |
| Com_assign_to_keycache            | 0         |
| Com_alter_db                      | 0         |
| Com_alter_db_upgrade              | 0         |
| Com_alter_event                   | 0         |
| Com_alter_function                | 0         |
| Com_alter_procedure               | 0         |
| Com_alter_server                  | 0         |
| Com_alter_table                   | 0         |
```

```
| Com_alter_tablespace              | 0         |
| Com_analyze                       | 4         |
| Com_backup_table                  | 0         |
| Com_begin                         | 0         |
| Com_binlog                        | 0         |
| Com_call_procedure                | 0         |
| Com_change_db                     | 6         |
| Com_change_master                 | 0         |
| Com_check                         | 0         |
| Com_checksum                      | 0         |
| Com_commit                        | 0         |
| Com_create_db                     | 2         |
| Com_create_event                  | 0         |
| Com_create_function               | 0         |
| Com_create_index                  | 0         |
| Com_create_procedure              | 0         |
| Com_create_server                 | 0         |
| Com_create_table                  | 2         |
| Com_create_trigger                | 0         |
| Com_create_udf                    | 0         |
| Com_create_user                   | 0         |
| Com_create_view                   | 0         |
| Com_dealloc_sql                   | 0         |
| Com_delete                        | 0         |
| Com_delete_multi                  | 0         |
| Com_do                            | 0         |
| Com_drop_db                       | 1         |
| Com_drop_event                    | 0         |
| Com_drop_function                 | 0         |
| Com_drop_index                    | 0         |
| Com_drop_procedure                | 0         |
| Com_drop_server                   | 0         |
| Com_drop_table                    | 0         |
| Com_drop_trigger                  | 0         |
| Com_drop_user                     | 0         |
| Com_drop_view                     | 0         |
| Com_empty_query                   | 0         |
| Com_execute_sql                   | 0         |
| Com_flush                         | 7         |
| Com_grant                         | 0         |
| Com_ha_close                      | 0         |
| Com_ha_open                       | 0         |
| Com_ha_read                       | 0         |
| Com_help                          | 0         |
| Com_insert                        | 0         |
| Com_insert_select                 | 2         |
| Com_install_plugin                | 0         |
| Com_kill                          | 0         |
| Com_load                          | 0         |
| Com_load_master_data              | 0         |
| Com_load_master_table             | 0         |
| Com_lock_tables                   | 0         |
| Com_optimize                      | 0         |
| Com_preload_keys                  | 0         |
| Com_prepare_sql                   | 0         |
| Com_purge                         | 0         |
| Com_purge_before_date             | 0         |
| Com_release_savepoint             | 0         |
```

```
| Com_rename_table                | 0         |
| Com_rename_user                 | 0         |
| Com_repair                      | 0         |
| Com_replace                     | 0         |
| Com_replace_select              | 0         |
| Com_reset                       | 0         |
| Com_restore_table               | 0         |
| Com_revoke                      | 0         |
| Com_revoke_all                  | 0         |
| Com_rollback                    | 0         |
| Com_rollback_to_savepoint       | 0         |
| Com_savepoint                   | 0         |
| Com_select                      | 237       |
| Com_set_option                  | 3         |
| Com_show_authors                | 0         |
| Com_show_binlog_events          | 0         |
| Com_show_binlogs                | 0         |
| Com_show_charsets               | 0         |
| Com_show_collations             | 0         |
| Com_show_column_types           | 0         |
| Com_show_contributors           | 0         |
| Com_show_create_db              | 0         |
| Com_show_create_event           | 0         |
| Com_show_create_func            | 0         |
| Com_show_create_proc            | 0         |
| Com_show_create_table           | 10        |
| Com_show_create_trigger         | 0         |
| Com_show_databases              | 17        |
| Com_show_engine_logs            | 0         |
| Com_show_engine_mutex           | 0         |
| Com_show_engine_status          | 2         |
| Com_show_events                 | 0         |
| Com_show_errors                 | 0         |
| Com_show_fields                 | 2010      |
| Com_show_function_status        | 0         |
| Com_show_grants                 | 2         |
| Com_show_keys                   | 9         |
| Com_show_master_status          | 0         |
| Com_show_new_master             | 0         |
| Com_show_open_tables            | 0         |
| Com_show_plugins                | 0         |
| Com_show_privileges             | 0         |
| Com_show_procedure_status       | 0         |
| Com_show_processlist            | 1         |
| Com_show_profile                | 6         |
| Com_show_profiles               | 6         |
| Com_show_slave_hosts            | 0         |
| Com_show_slave_status           | 0         |
| Com_show_status                 | 23        |
| Com_show_storage_engines        | 0         |
| Com_show_table_status           | 0         |
| Com_show_tables                 | 20        |
| Com_show_triggers               | 0         |
| Com_show_variables              | 13        |
| Com_show_warnings               | 3         |
| Com_slave_start                 | 0         |
| Com_slave_stop                  | 0         |
| Com_stmt_close                  | 0         |
```

```
| Com_stmt_execute                  | 0         |
| Com_stmt_fetch                    | 0         |
| Com_stmt_prepare                  | 0         |
| Com_stmt_reprepare                | 0         |
| Com_stmt_reset                    | 0         |
| Com_stmt_send_long_data           | 0         |
| Com_truncate                      | 0         |
| Com_uninstall_plugin              | 0         |
| Com_unlock_tables                 | 0         |
| Com_update                        | 0         |
| Com_update_multi                  | 0         |
| Com_xa_commit                     | 0         |
| Com_xa_end                        | 0         |
| Com_xa_prepare                    | 0         |
| Com_xa_recover                    | 0         |
| Com_xa_rollback                   | 0         |

$ mysqladmin -udbuser -p extended-status | grep "query_"
Enter password:
| Last_query_cost                   | 0.000000  |
```

Com_select semble avoir été incrémenté de 2 (pourquoi 2 ?) et les variables Qcache ont peu bougé.

1.8 Utiliser le profiling

On active le profiling avec la variable de **session** *profiling*. C'est une variable de session donc on ne peut pas la positionner ni dans un fichier de commande, ni via la ligne de commande.

http://dev.mysql.com/doc/refman/5.1/en/profiling-table.html

```
mysql> SET profiling = 1;

mysql> source getpublis.sql

mysql> show profile;
+--------------------------------+----------+
| Status                         | Duration |
+--------------------------------+----------+
| starting                       | 0.000038 |
| checking query cache for query | 0.000375 |
| Opening tables                 | 0.000179 |
| System lock                    | 0.000010 |
| Table lock                     | 0.000166 |
| init                           | 0.000233 |
| optimizing                     | 0.000045 |
| statistics                     | 0.000609 |
| preparing                      | 0.000070 |
| Creating tmp table             | 0.000241 |
| executing                      | 0.000006 |
| Copying to tmp table           | 0.326255 |
| converting HEAP to MyISAM      | 0.096271 |
| Copying to tmp table on disk   | 2.053016 |
| Sorting result                 | 0.410069 |
| Sending data                   | 0.801377 |
| end                            | 0.000041 |
| removing tmp table             | 0.000227 |
```

```
| end                            | 0.000011 |
| query end                      | 0.000008 |
| freeing items                  | 0.000681 |
| logging slow query             | 0.000011 |
| cleaning up                    | 0.000013 |
+--------------------------------+----------+
23 rows in set (0.02 sec)
```

1.9 Examiner les statistiques et les variables

1.9.1 mysqladmin extended-status

http://www.slideshare.net/phpcodemonkey/mysql-explain-explained, slide 22

```
$ mysqladmin -u dbuser -p -r -i 10 extended-status
```

Figures are relative, updated every 10 seconds

- Slow_queries = number of slow queries in last period
- Select_Scan = full table scans
- Select_full_join = full scans to complete join operations
- Created_tmp_disk_tables = filesorts
- Key_read_requests/Key_write_requests

Determine write/read weighting of our application and alter your indexes accordingly

1.10 Tests de charge

En étudiant l'utilisation de Memcached avec mysql, je suis arrivée sur un article [HandlerSocker] décrivant des tests de charges menés à l'occasion à l'aide de l'utilitaire **mysqlslap**.

Des présentations de Yoshinori Matsunobu sur SlideShare : http://www.slideshare.net/search/slideshow?searchfrom=header&q=Yoshino

Dans la présentation suivante, http://www.slideshare.net/matsunobu/linux-performance-tuning-stabilization-tips-mysqlconf2010

MySQL Sorting Algorithm : MySQL has two sorting algorithms (old algorithm / new algorithm). Choosing either of the two, depending on column length, data types, etc.. Currently there is no MySQL status variable to check which algorithm is used. Sometimes performance difference is huge (especially when used with LIMIT).

Voir aussi l'article suivant Linux and H/W optimizations for MySQL,

- http://www.slideshare.net/morgo/the-role-of-io-as-a-bottleneck

1.10.1 mysqlslap

La page man de mysqlslap n'est pas claire, je me suis basée sur un Howto mysqlslap de www.techrepublic.com

```
$ mysqlslap --create-schema=Labinfo -q "DESCRIBE L_PERSONNEL" -u<login_admin> -p -vv
```

j'ai retenu les options suivantes pour les tests

```
-vv     pour des informations détaillées
-i      iterations, number of times
-c      concurrency, number of client to simulate for query to run
avec
--number-of-queries limit each client to this number of queries
        En fait, c'est le nombre total qui sera divisé par le nombre de clients
-q      on peut passer un fichier, par ex : getpublis.sql
--csv=/tmp/mysqloutput.csv
```

Ce qui est intéressant, c'est d'exécuter nos requêtes. J'ai fait le test dans une machine virtuelle donc les temps ne sont plus mauvais que sur le site (autour de 5 secondes sur lirisped).

```
$ mysqlslap --create-schema=Labinfo --query=getpublis.sql -i3 -u<login_admin> -p -vv

Parsing engines to use.
Enter password:
Starting Concurrency Test
Generating primary key list
Generating primary key list
Generating primary key list
Generating stats
Benchmark
    Average number of seconds to run all queries: 12.306 seconds
    Minimum number of seconds to run all queries: 11.960 seconds
    Maximum number of seconds to run all queries: 12.876 seconds
    Number of clients running queries: 1
    Average number of queries per client: 1
```

L'éventuelle erreur suivante est due à un manque de place disque (faire du ménage dans ce cas)

```
mysqlslap: Cannot run query select ...
ERROR : Incorrect key file for table '/tmp/#sql_94b_0.MYI'; try to repair it
```

Autre article en français sur cet utilitaire http://www.dbnewz.com/2008/07/07/mysqlslap-et-supersmack-deux-outils-de-benchmark-pour-mysql/

1.11 Index

Est-ce que l'ajout d'un index peut dégrader des performances ?

1.12 Utiliser Memcached avec mysql

Memcached faisait partie des élements installés pendant la formation Plone. Le formateur n'avait pas le temps de rentrer dans les détails, j'ai donc recherché des informations sur ce système de cache.

1.12.1 Qu'est-ce que Memcached ?

Memcached aurait été créé à l'origine pour accélérer les sites Web dynamiques liés à une base de données pour charger en mémoire les données les plus souvent accédées. Memcached permet de mettre en mémoire des éléments et fournit des API pour cela. Cela paraissait intéressant pour nous.

Utiliser Memcached avec MySQL permet de fournir des méthodes accès NoSQL à MySQL.

Mais l'article MySQL précise bien : the memcached implementations for InnoDB and MySQL Cluster are still in their **early phases of development** and so **neither is suitable for production deployment**. Il y a un module téléchargeable pour Mysql 5.6, on vient de passer à MySQL 5.1.49 (version Debian Squeeze).

De plus, la FAQ indique qu'il ne s'agit pas d'une couche transparente, l'application doit être développée pour en tirer profit :

```
So the responsibility lies with the application to populate and get records from the database as oppo
Yes. You load the data from the database and write it into the cache provided by memcached. Using mer
```

Voir l'exemple d'interfaçage avec Python pour utiliser memcached avec MySQL.

Voir aussi :

- http://en.wikipedia.org/wiki/Memcached
- http://dev.mysql.com/doc/refman/5.0/en/ha-memcached.html

1.13 GROUP BY and HAVING versus JOIN

Il y a une réponse étonnante très bien notée (70 au moment où je l'ai consultée) : http://stackoverflow.com/questions/477006/sql-statement-join-vs-group-by-and-having

---

Note: That's right. The join version I proposed is twenty times faster than the aggregate version.

Je n'ai pas trop d'instructions de ce type, c'est à savoir.

1.14 FOUND_ROWS()

In the absence of the *SQL_CALC_FOUND_ROWS* option in the most recent successful *SELECT* statement, *FOUND_ROWS()* returns the number of rows in the result set returned by that statement. If the statement includes a *LIMIT* clause, *FOUND_ROWS()* returns the number of rows up to the limit.

```
SELECT FOUND_ROWS();
+--------------+
| FOUND_ROWS() |
+--------------+
|         7588 |
+--------------+
1 row in set (0.00 sec)
```

Je pensais m'en servir pour ne pas faire une requête pour le comptage des publications. Mais pour une publication, je récupère un nombre variable de lignes (dépend des auteurs ...) :(

http://dev.mysql.com/doc/refman/5.1/en/information-functions.html#function_found-rows

1.15 Liens à étudier

- http://dev.mysql.com/tech-resources/presentations/presentation-oscon2000-20000719/
- http://20bits.com/articles/10-tips-for-optimizing-mysql-queries-that-dont-suck/
- http://stackoverflow.com/questions/tagged/query-optimization+mysql

- http://www.mysqlperformanceblog.com/2006/05/17/mysql-server-memory-usage/

1.16 Références

- Le site de MySQL et les liens pointés
- Le livre *MySQL Admin Cookbook*

1.17 Café ARAMIS

Il faut mesurer différents éléments. Oracle donne des informations sur le temps CPU / I/O / serve (?)

1.17.1 Visualiser les paramètres de configuration et les différentes variables MySQL

Utiliser les différentes variables pour analyser le comportement.

Cache de requêtes

Calculer le ratio suivant pour savoir si le cache est sollicité.

Note: Innodb_buffer_pool_reads (nb lectures disque) / Innodb_buffer_pool_read_requests

Innodb_buffer_pool_read_requests : The number of logical read requests InnoDB has done.

Innodb_buffer_pool_reads : The number of logical reads that InnoDB could not satisfy from the buffer pool, and had to read directly from the disk.

Server System Variables

The MySQL server maintains many system variables that indicate how it is configured..

```
mysql> SHOW VARIABLES;
```

```
mysql> SHOW SESSION VARIABLES;
```

```
mysql> SHOW GLOBAL VARIABLES;
```

```
$ mysqladmin variables
```

Server Status Variables

The server maintains many status variables that provide information about its operation.

```
mysql> SHOW STATUS;
```

```
mysql> SHOW SESSION STATUS;
```

```
mysql> SHOW GLOBAL STATUS;
```

```
$ mysqladmin extended-status
```

PhpMyAdmin

S'il est installé, permet de consulter aisément les différentes variables. Conseils ?

1.17.2 Impact des accès disque

Réorganiser les tables.

Ou tout exporter et tout ré-importer pour une bonne organisation sur disque.

Supprimer / recréer des index.

Exécuter la requête par CRON pour qu'elle soit en mémoire => ne marche pas s'il y a beaucoup de requêtes exécutées, elle sera remplacée dans le cache.

Vue matérialisée (il n'y en a pas dans MySQL 5.1), "lien ?"

Query Rewrite "?"

Une duplication de données c'est MAL mais on n'a plus de problème d'espace disque, donc on peut s'autoriser des duplications.

Oracle : parfois forcer le rafraichissement des statistiques améliore les performances

Mettre la base de données sur une autre machine ou sur un autre disque.

Si la BD tape sur le même disque qu'Apache, ça peut améliorer les performances.

Index stockés au même endroit que les tables ? Est-ce possible avec MySQL ?

1.17.3 Paramétrer le cache

Note: query_cache_limit = 1M => 16M query_cache_size = 16M => 128M

Faire une **vue de la requête massive** et faire des requêtes spécifiques (par personne, par équipes, ...). Comme ça la requête restera en cache.

1.17.4 Prepared statement

```
r = "SELECT * FROM t WHERE id=?"

pr = prepared_statement(r);

for {
    execute (pr, array(i));
    }
```

1. Il ne ré-exécute pas l'optimisation de la requête à chaque itération, il l'a déjà faite
2. Protège contre l'injection SQL

1.17.5 Conversions

Un “SELECT * FROM t WHERE id=‘24”’ rame : Si **id** est un entier, primary key, il ne sera pas utilisé parce qu’une chaine n’est pas forcément convertible en entier => ce sont les identifiants de la table qui seront convertis en chaîne de caractères et comparés.

La conversion implicite qui marche, c’est nombre => chaîne de caractères.

Même problème avec les DATE.

1.17.6 Cardinalités

Si la majorité de l’index est occupé par une seule valeur, alors autant faire un **full scan**.

Créer un index qui commence par la fin d’un champ de type chaîne si c’est la fin qui est discriminante (par ex chemin de fichiers).

Enlever les index qui ne servent à rien (faible cardinalité ...).

Oracla a un advisor qui peut conseiller de mettre un index sur une colonne.

1.17.7 Partitionnement

Très intéressant en analyse de données temporelles ou data mining ou datawarehouse.

Partitionnement par mois pour ne pas scanner toute la table par exemple.

1.17.8 Stored Procedures

C’est dommage, je n’ai pas de notes sur ce point de la discussion mais d’après ce dont je me souviens, il serait intéressant d’utiliser davantage les **stored procedure**. Cela permet d’avoir la logique dans la base, et donc d’avoir un accès sécurisé pour différents clients (qui n’ont pas besoin de coder l’accès aux tables, les contrôles), c’est plus sûr. Ce n’est pas très utilisé par les développeurs à tort.

1.17.9 Divers

Oracle vend des machines spécifiques qui garantissent un taux d’I/O par sec lorsque l’on ajoute des disques. Les contrôleurs de disques peuvent exécuter du SQL. Taux de compression qui vont de 20 à 50.

BD Opera 40 To.

CHAPTER

TWO

METTRE EN PLACE DU MONITORING

2.1 Métrologie

La métrologie est la « science des mesures et ses applications ». Elle « comprend tous les aspects théoriques et pratiques des mesurages, quels que soient l'incertitude de mesure et le domaine d'application. (*source http://fr.wikipedia.org/wiki/M%C3%A9trologie*)

2.2 Solutions libres de monitoring

Les solutions dont j'ai entendu parler sont Nagios, Munin et Monit.

Nagios semble le plus connu parmi les ASR. Il n'y a de fiche Plume que pour Nagios.

http://serverfault.com/questions/44/what-tool-do-you-use-to-monitor-your-servers : Les réponses ne m'ont pas franchement intéressées car elles ne détaillent pas avantages / incovénients, je n'ai regardé que les premières réponses (les mieux notées).

2.3 Munin

Voici des exemples de graphiques de serveurs monitorés par Munin.

Ce qui m'intéresserait, serait de pouvoir avoir aussi des statistiques par VirtualHost, est-ce envisageable ? pas trop perturbant ? Le lien suivant propose des plugins pour monitorer par VirtualHost : http://benlumley.co.uk/2010/02/02/munin-plugins-to-graph-apache-virtual-hosts

Monitorer Apache avec Munin : http://technique.arscenic.org/monitoring/munin/monitoring-specifiques/article/monitorer-apache

2.4 Nagios

2.5 Monit

2.6 Divers utilitaires plus ou moins standards

La trilogie classique : iostat, vmstat, netstat.

Un utilitaire préconisé par Yoshinori Matsunobu (cd performances MySQL) : http://oprofile.sourceforge.net/about/. Cependant, il faut le recompiler comme module du kernel (pour la PROD, je n'y tiens pas).

2.7 Café ARAMIS

On peut récupérer des informations provenant du SNMP afin d'avoir une vue autre que celles données par le Mysql et les commandes SQL.

http://www.masterzen.fr/software-contributions/mysql-snmp-monitor-mysql-with-snmp/

```
$ apt-get install snmpd-server (10 secondes)
$ vi /etc/snmpd/snmpd.conf (16 secondes pour lui dire d'écouter sur eth0)
```

Sur un autre serveur, par exemple

```
$ apt-get install cacti (2 mns), avec base mysql
```

Le tout, début de monitoring en 20 mns ;-)

CHAPTER

THREE

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*

BIBLIOGRAPHY

[HandlerSocker] http://yoshinorimatsunobu.blogspot.com/search/label/handlersocket