MySQL : maintaining (too) big tables

Frédéric Descamps

Frédéric Descamps

- lefred on IRC
- Senior Linux and Open Source Consultant @inuits.be
- Certified MySQL DBA since 2007
- First GNU/Linux distro (as far as I remember): Slackware 3.0 (kernel 1.2.13)
- Blog: http://www.lefred.be

How come we need to maintain tables ?

- A bit of history first:

once upon a time, there was a developer who found a great idea

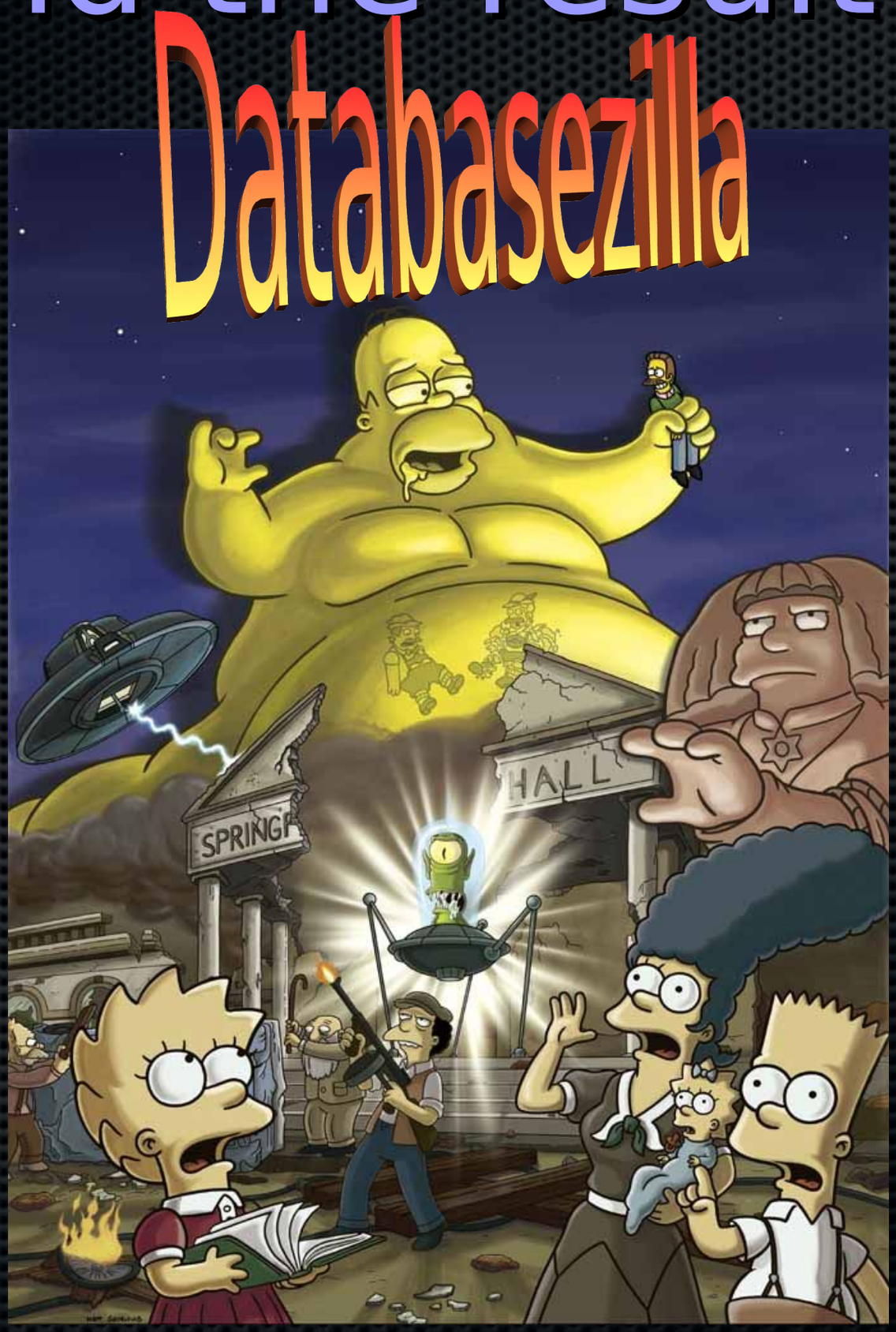
Then he decided to create a proof of concept and he coded, coded, coded a lot...

During the period he unfortunately didn't discuss with the DBA's and the OPS

But he discussed with many other people like marketing manager, sales manager, regional manager, product manager and all were very happy and asked to code even more !

Then he coded, coded, coded a lot...

And the result is



Why ?

- Impossible to stop de database, it's too important !
- There is too much data, tables are huge !!
  - Tables > 400GB with 500M rows
  - Primary keys are varchar(32)
- Impossible to optimize, add an index to a table (>4h needed to add an index)
- How do we alter the schema ?
- How do we migrate to another version of MySQL ?

Analyze and optimize

File Edit View Terminal Help

```
Server version: 5.1.37-community-log MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use asp
Database changed
mysql> analyze table dayview;
+-------------+---------+----------+----------+
| Table       | Op      | Msg_type | Msg_text |
+-------------+---------+----------+----------+
| asp.dayview | analyze | status   | OK       |
+-------------+---------+----------+----------+
1 row in set (0.48 sec)

mysql> optimize table dayview;
+-------------+----------+----------+-------------------------------------------------------------------+
| Table       | Op       | Msg_type | Msg_text                                                          |
+-------------+----------+----------+-------------------------------------------------------------------+
| asp.dayview | optimize | note     | Table does not support optimize, doing recreate + analyze instead |
| asp.dayview | optimize | status   | OK                                                                |
+-------------+----------+----------+-------------------------------------------------------------------+
2 rows in set (8 hours 2.16 sec)

mysql>
```

How big ?

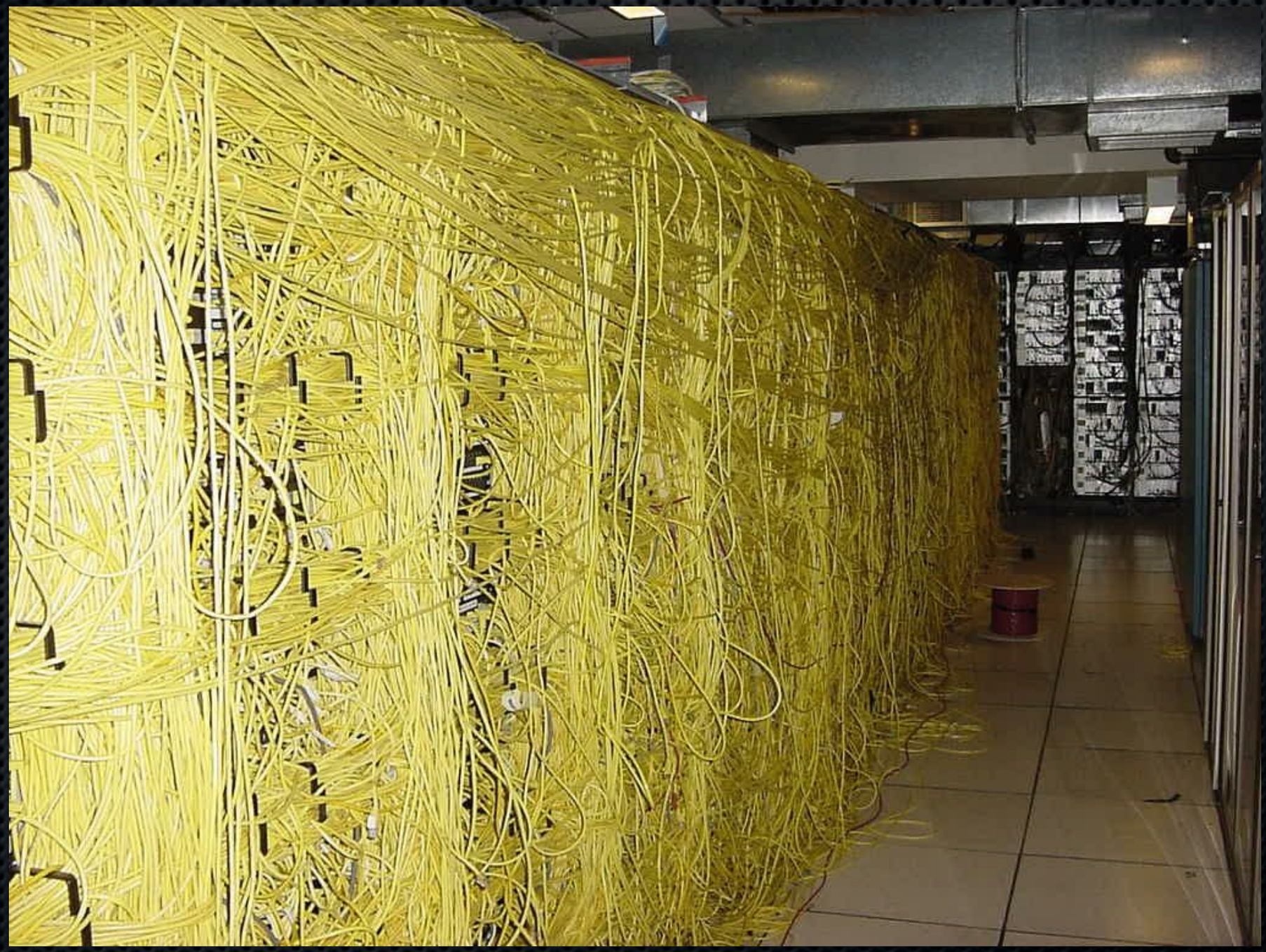
```
# ls -lh dayview.ibd

-rw-rw---- 1 mysql mysql 29G Feb  3 22:20 dayview.ibd
```

dba's feeling ?

Two very important points to never forget as developer :

- Don't underestimate the complexity of production environments



Don't !

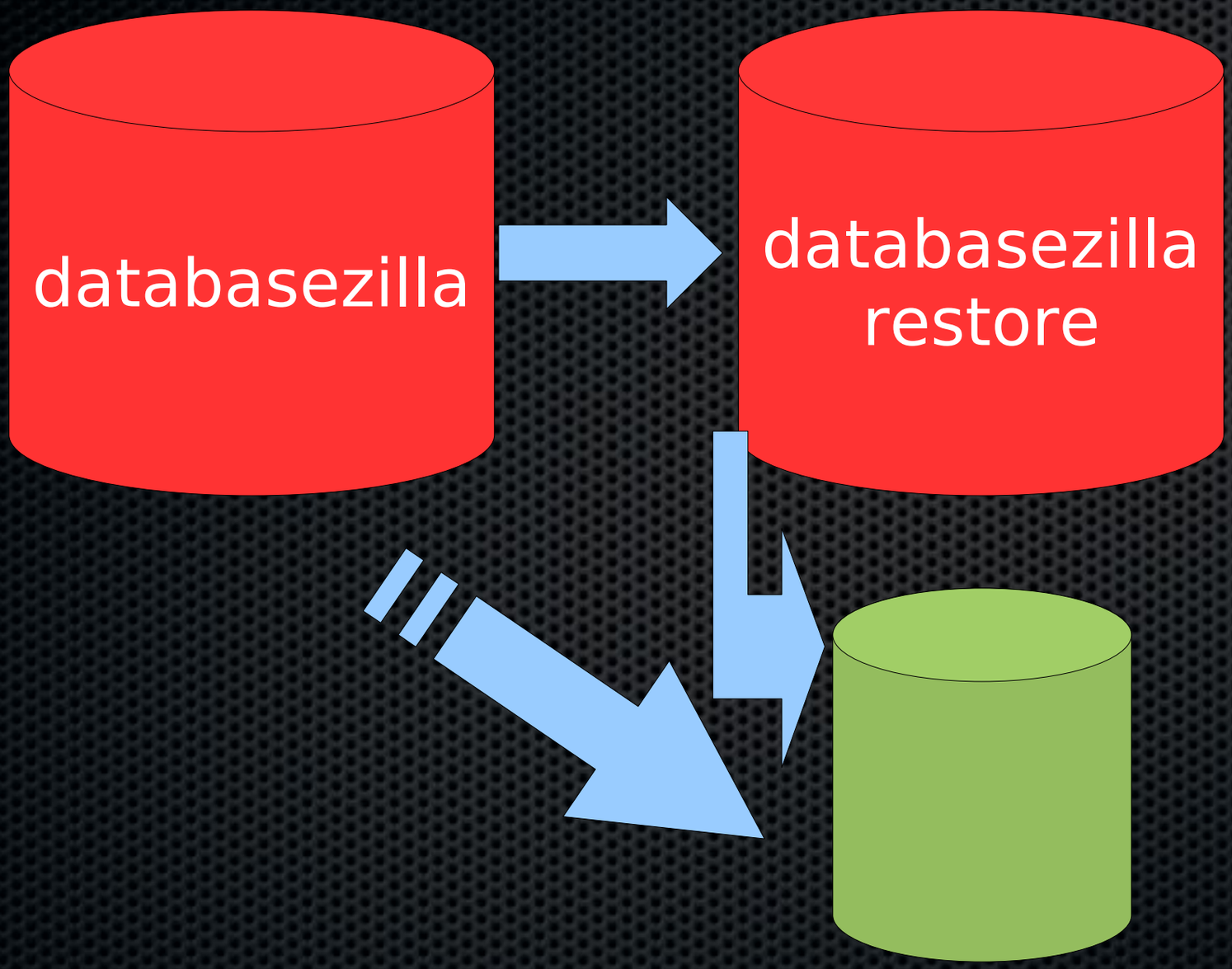
- Never say: “but on my laptop the query is very fast”

What now ?

- The plan is to create a new server more manageable with minimal downtime of the production
  - Reduce size if possible
  - Make queries faster
- The server will be a replica of the current production server with some different settings

The battle !

Ze Plan



- Restore last backup
- Prepare the new database
- Dump & restore data using maatkit
- Restart the replication
- Replace the production db by the new one (only downtime here)

Database size

- Deleting historical data sometimes doesn't help really to save some space... ibdata won't shrink
- Data could be fragmented
  - To avoid the use of that huge ibdata we add this to the configuration of the new server:

```
innodb_file_per_table
```

Anything else to change ?

- Maybe other settings aren't optimal for your production environment. To have an overview of them, I'd recommand the use of this two tools :
  - MySQLTuner
  - tuning-primer

Example 1

-------- General Statistics --
[--] Skipped version check for MySQLTuner script
[OK] Currently running supported MySQL version 5.1.37-community-log
[OK] Operating on 64-bit architecture

-------- Storage Engine Statistics ---
[--] Status: -Archive -BDB -Federated +InnoDB -ISAM -NDBCluster
[--] Data in InnoDB tables: 899G (Tables: 149)
[!!] Total fragmented tables: 82

-------- Performance Metrics ---
[--] Up for: 23h 46m 1s (112M q [1K qps], 70K conn, TX: 44B, RX: 22B)
[--] Reads / Writes: 51% / 49%
[--] Total buffers: 12.4G global + 152.2M per thread (100 max threads)
[!!] Maximum possible memory usage: 27.3G (173% of installed RAM)
[OK] Slow queries: 0% (963/112M)
[OK] Highest usage of available connections: 52% (52/100)
[OK] Key buffer size / total MyISAM indexes: 256.0M/93.0K
[OK] Key buffer hit rate: 100.0% (215K cached / 3 reads)
[OK] Query cache efficiency: 43.3% (23M cached / 54M selects)
[!!] Query cache prunes per day: 3049150
[OK] Sorts requiring temporary tables: 0% (0 temp sorts / 754K sorts)
[!!] Joins performed without indexes: 19547
[OK] Temporary tables created on disk: 0% (1K on disk / 545K total)
[OK] Thread cache hit rate: 99% (53 created / 70K connections)
[OK] Table cache hit rate: 99% (802 open / 809 opened)
[OK] Open file limit used: 1% (59/4K)
[OK] Table locks acquired immediately: 100% (72M immediate / 72M locks)
[!!] InnoDB data size / buffer pool: 899.3G/12.0G

Example 2

MySQL Version 5.1.37-community-log x86_64

Uptime = 1 days 0 hrs 2 min 31 sec
Avg. qps = 1324
Total Questions = 114605776
Threads Connected = 47

SLOW QUERIES
The slow query log is enabled.
Current long_query_time = 10.000000 sec.
You have 972 out of 114605948 that take longer than 10.000000 sec. to complete
Your long_query_time seems to be fine

BINARY UPDATE LOG
The binary update log is enabled
The expire_logs_days is not set.
The mysqld will retain the entire binary log until RESET MASTER or PURGE MASTER LOGS commands are run manually
Setting expire_logs_days will allow you to remove old binary logs automatically
See http://dev.mysql.com/doc/refman/5.1/en/purge-master-logs.html
Binlog sync is not enabled, you could loose binlog records during a server crash

WORKER THREADS
Current thread_cache_size = 8
Current threads_cached = 4
Current threads_per_sec = 0
Historic threads_per_sec = 0
Your thread_cache_size is fine

Example 2 (b)

MAX CONNECTIONS
Current max_connections = 100
Current threads_connected = 47
Historic max_used_connections = 52
The number of used connections is 52% of the configured maximum.
Your max_connections variable seems to be fine.

INNODB STATUS
Current InnoDB index space = 0 bytes
Current InnoDB data space = 0 bytes
Current InnoDB buffer pool free = 0 %
Current innodb_buffer_pool_size = 12.00 G
Depending on how much space your innodb indexes take up it may be safe
to increase this value to up to 2 / 3 of total system memory

MEMORY USAGE
Max Memory Ever Allocated : 20.06 G
Configured Max Per-thread Buffers : 14.86 G
Configured Max Global Buffers : 12.33 G
Configured Max Memory Limit : 27.20 G
Physical Memory : 15.67 G

nMax memory limit exceeds 90% of physical memory

Example 2 (c)

KEY BUFFER
Current MyISAM index space = 8 K
Current key_buffer_size = 256 M
Key cache miss rate is 1 : 74223
Key buffer free ratio = 81 %
Your key_buffer_size seems to be too high.
Perhaps you can use these resources elsewhere

QUERY CACHE
Query cache is enabled
Current query_cache_size = 64 M
Current query_cache_used = 32 M
Current query_cache_limit = 2 M
Current Query cache Memory fill ratio = 51.30 %
Current query_cache_min_res_unit = 4 K
MySQL won't cache query results that are larger than query_cache_limit in size

SORT OPERATIONS
Current sort_buffer_size = 64 M
Current read_rnd_buffer_size = 16 M
Sort buffer seems to be fine

JOINS
Current join_buffer_size = 64.00 M
You have had 20057 queries where a join could not use an index properly
join_buffer_size >= 4 M
This is not advised
You should enable "log-queries-not-using-indexes"
Then look for non indexed joins in the slow query log.

Example 2 (d)

OPEN FILES LIMIT
Current open_files_limit = 4096 files
The open_files_limit should typically be set to at least 2x-3x
that of table_cache if you have heavy MyISAM usage.
Your open_files_limit value seems to be fine

TABLE CACHE
Current table_open_cache = 1024 tables
Current table_definition_cache = 256 tables
You have a total of 0 tables
You have 843 open tables.
The table_cache value seems to be fine

TEMP TABLES
Current max_heap_table_size = 64 M
Current tmp_table_size = 256 M
Of 554783 temp tables, 0% were created on disk
Effective in-memory tmp_table_size is limited to max_heap_table_size.
Created disk tmp tables ratio seems fine

TABLE SCANS
Current read_buffer_size = 8 M
Current table scan ratio = 162 : 1
read_buffer_size seems to be fine

TABLE LOCKING
Current Lock Wait ratio = 0 : 114608692
Your table locking seems to be fine

Filesystem

- xfs is much faster to store databases and especially to store binlogs

Start the replication

- Change to the right binlog and the right position before the backup

Monitor your production

- While you will do any work on the database, I recommand you to use `innotop`, you will see directly if something is going wrong
- Use cacti to see obviously strange behavior

Restore the backup

- I won't explain you how to restore a backup, but on huge tables usually you don't have a dump of the data but only the files and the binlogs
- So restore the last production backup you have on a new server (most of the time we use flash copies)

Dump & restore the data

- Use maatkit parallel to dump and restore the data
- While this operation, be carreful to not delete the binlogs on te master that you will need to recover !

Remove historical data

- Now that you have a more performant database (as the indexes are rewritten) it's time to archive the old data
- As the tables are still huges I don't recommand to do a massive delete of many rows matching a query
- Use (again) maatkit to archive

ptxArchiver

- I wrote a small script to use mk-archiver that follows a defined strucure to archive all the records linked together (foreignkeys)

```
$ ptxArchiver.pl -t message -w "timestamp <
unix_timestamp(date_sub(now(),interval 17
MONTH))*1000"


<tables>
   <message>
      <message>
        field=id
        p_field=origin
      </message>
   </message>
</tables>
```

Analyse your MySQL

Analyze your tables

- MySQL also relies on using statistics for keeping track of data distribution in tables and for optimizing join statements.

```
File  Edit  View  Terminal  Help
mysql> use asp;
Database changed
mysql> analyze table dayview;
+-------------+---------+----------+----------+
| Table       | Op      | Msg_type | Msg_text |
+-------------+---------+----------+----------+
| asp.dayview | analyze | status   | OK       |
+-------------+---------+----------+----------+
1 row in set (0.45 sec)

mysql> analyze table trace;
+-----------+---------+----------+----------+
| Table     | Op      | Msg_type | Msg_text |
+-----------+---------+----------+----------+
| asp.trace | analyze | status   | OK       |
+-----------+---------+----------+----------+
1 row in set (1.57 sec)

mysql>
[1]+  Stopped                 mysql -u root -p
# ls -lh dayview.ibd trace.ibd
-rw-rw---- 1 mysql mysql  29G Feb  6 01:58 dayview.ibd
-rw-rw---- 1 mysql mysql 291G Dec  4 22:23 trace.ibd
```

Analyse your MySQL

- Use mt-query-digest on slow query log, binlogs or processlist
  - mk-query-digest --processlist 127.0.0.1,
- Don't forget innotop
- mysqlresources
- mysqlsla
- mysqlidxchk
- mysqlreport

innotop

File Edit View Terminal Help

```
[RO] Query List (? for help)                    localhost, 6+16:22:25, 2.06k QPS, 48/22/3 con/run/cac thds, 5.1.37-community-log

When    Load  QPS      Slow    QCacheHit  KCacheHit  BpsIn    BpsOut
Now     0.00    2.06k  1         34.47%    100.00%  529.30k    1.47M
Total   0.00  858.21   11.81k    40.96%    100.00%  188.65k  464.25k

Cmd    ID      State         User   Host          DB      Time      Query
Query   86750  Sending data  asp    garlic        asp     01:20:23  SELECT terminal.name tname, fmssession.starttime day, DATEDIFF(CURDA
Query   86782  update        asp    garlic        asp        01:50  insert into fmssession (starttime, duration, endtime, terminal_id, d
Query   86741  update        asp    garlic        asp        01:31  insert into fmssession (starttime, duration, endtime, terminal_id, d
Query   86738  update        asp    garlic        asp        01:27  insert into fmssession (starttime, duration, endtime, terminal_id, d
Query   86753  update        asp    garlic        asp        01:15  insert into fmssession (starttime, duration, endtime, terminal_id, d
Query   86779  update        asp    garlic        asp        01:04  insert into fmssession (starttime, duration, endtime, terminal_id, d
Query   86743  update        asp    garlic        asp        00:38  insert into fmssession (starttime, duration, endtime, terminal_id, d
Query   86730  update        asp    garlic        asp        00:28  insert into fmssession (starttime, duration, endtime, terminal_id, d
Query   86755  Sending data  asp    garlic        asp        00:25  SELECT count(*) FROM task LEFT JOIN terminal AS stateterminal ON tas
Query   86754  update        asp    garlic        asp        00:22  insert into fmssession (starttime, duration, endtime, terminal_id, d
Query   86778  Sending data  asp    garlic        asp        00:05  SELECT count(*) FROM task LEFT JOIN terminal AS stateterminal ON tas
Query   86737  Sending data  asp    garlic        asp        00:04  SELECT count(*) FROM message WHERE message.sourceid in ('5DA88080808
Query   86747  Sending data  asp    garlic        asp        00:03  SELECT count(*) FROM message WHERE message.sourceid in ('FB508080808
Query   86756  Sending data  asp    garlic        asp        00:03  SELECT count(*) FROM message WHERE message.sourceid in ('5DA88080808
Query   86748  Sending data  asp    garlic        asp        00:02  SELECT count(*) FROM task LEFT JOIN terminal AS stateterminal ON tas
Query   86780  Sending data  asp    garlic        asp        00:02  SELECT count(*) FROM message WHERE message.sourceid in ('170E8080808
Query   86704  Sending data  asp    garlic        asp        00:00  select trace0_.id as id49_1_, trace0_.timestamp as timestamp49_1_, t
```


THINK
Do it Right
The First
Time!
Plan Ahea
d

Links

- MySQLTuner - http://blog.mysqltuner.com/
- tuning-primer - http://www.day32.com/MySQL/
- ptxArchiver - http://www.lefred.be/?q=node/105
- maatkit - http://www.maatkit.org
- mysqlreport, mysqlsla, mysqlidxchx - http://hackmysql.com
- mysqlresources - http://datacharmer.org/downloads/mysqlresources.zip
- innotop - http://www.xaprb.com/blog/2006/07/02/innotop-mysql-innodb-monitor/

Thank you :-)