If you can't clearly read this text, move closer. You'll thank me later.

Find Query Problems Proactively With Query Reviews

Presented by: Sheeri K. Cabral

Twitter: @sheeri

Pythian

love your data

About Pythian

- Recognized Leader:
  - Global industry-leader in database infrastructure services for Oracle, Oracle Applications, MySQL and SQL Server
  - 150 current multinational companies such as Forbes.com, Fox Sports and Western Union to help manage their complex IT deployments
- Expertise:
  - One of the world's largest concentrations of dedicated, full-time DBA expertise.
- Global Reach & Scalability:
  - 24/7/365 global remote support for DBA and consulting, systems administration, special projects or emergency response

Query Review

- What is it?
  - Systematic review of all queries
- Why do it?
  - Find queries before they become a problem
  - Often a sample query is non-trivial to find

Query Review

- Who should do it?
  - Optimization knowledge
- When and where should it be done?
  - dev → test,load test,staging → production

Main tool

- mk-query-digest
  - “query fingerprint”
- Can be used on:
  - Slow query logs
  - Binary logs
  - General query logs

More mk-query-digest sources

- Direct database querying
  - Uses SHOW FULL PROCESSLIST
- pglog (Postgres)
- Parsing tcpdump for traffic:
  - MySQL
  - memcached
  - HTTP

Getting mk-query-digest

- wget maatkit.org/get/mk-query-digest
  - Easiest
  - Not always up-to-date!
- http://code.google.com/p/maatkit/
  - More work
  - You get all the maatkit tools, not just one
  - Most up to date

Last week, wget got rev 6067, download is 6070!

What is reported on

- Default setup uses --limit 95%:20
  - To see all queries, --limit 100%
- No --filter by default
- --filter
  - Any attribute at http://code.google.com/p/maatkit/wiki/EventAttributes
  - User, host, database, process id, lock_time, Memc_miss, Rows_sent, Rows_examined, Rows_affected, Rows_read, Query_time, insert_id

Other filters

- If using Percona's patches, you can filter on queries that cause:
  - Filesorts, disk filesorts
  - Temp tables, Temp disk tables
  - Full table scan, full join
  - Query cache hit
  - and more...

Output

- Overall summary
- Detailed report of matching queries
- Query Analysis Summary

- Commands run for examples:

```
perl mk-query-digest --limit 100% \
--review h=127.0.0.1,P=3307,D=maatkit,t=query_review,u=user,p=pass \
--create-review-table --type genlog genlog127.sql > genlogoutput.txt

perl mk-query-digest --limit 100% \
--review h=127.0.0.1,P=3307,D=maatkit,t=query_review,u=user,p=pass \
--type binlog binlog325.sql > binlogoutput.txt
```

Overall summary (genlog)

```
# 229.7s user time, 860ms system time, 94.79M rss, 145.48M vsz
# Overall: 906.22k total, 720 unique, 143.84 QPS, 0x concurrency_________
#                   total     min     max     avg     95%  stddev median
# Exec time             0       0       0       0       0       0      0
# Time range        2010-03-12 10:45:01 to 2010-03-12 12:30:01
# bytes           242.78M       5  69.06k  280.91  563.87  819.66 112.70
```

Overall summary (binlog)

```
# 390.2s user time, 1.8s system time, 62.70M rss, 113.45M vsz
# Overall: 1.07M total, 252 unique, 245.71 QPS, 5.69Gx concurrency_______
#                    total min       max       avg    95% stddev median
# Exec time  2478625699859 8s   0 4294967295s 23168970s 992ms 302909074s 0
# Time range        2010-04-10 07:14:17 to 2010-04-10 08:26:51
# @@session             86      0       1    0.50    0.99    0.50    0.99
# @@session            585      1       4    3.42    3.89    0.68    3.89
# @@session          3.44k      8      33   20.57   31.70   12.00   31.70
# @@session          1.34k      8       8       8       8       0       8
# @@session              1      1       1       1       1       0       1
# @@session        837.08k 837.08k 837.08k 837.08k 837.08k       0837.08k
# @@session             85      0       1    0.50    0.99    0.50       0
# bytes            415.05M      5   1.02M  349.05  563.87   1.34k 537.02
# error cod              0      0       0       0       0       0       0
```

Query analysis part 1 (genlog)

```
# Query 9: 1.69 QPS, 0x concurrency, ID 0x188B27831A9DE05B at byte
268215186

# This item is included in the report because it matches --limit.

#              pct   total     min     max     avg     95%  stddev  median

# Count          1   10647

# Exec time      0       0       0       0       0       0       0       0

# Databases                1 proddb

# Time range 2010-03-12 10:45:02 to 2010-03-12 12:30:01

# bytes          0 613.45k      59      59      59      59       0      59
```

Query analysis part 1 (binlog)

```
# Query 5: 3.90 QPS, 297.34Mx concurrency, ID 0x188B27831A9DE05B at byte
596881917

# This item is included in the report because it matches --limit.

#              pct   total     min     max     avg     95%  stddev median

# Count          1      16829

# Exec time    5 1284195222560s 0 4294967295s 76308469s 992ms 546294873s 0

# Databases                1 proddb

# Time range 2010-04-10 07:14:52 to 2010-04-10 08:26:51

# bytes          0 969.38k      58      59   58.98   56.92        0   56.92

# error cod      0       0       0       0       0       0       0       0
```

Query analysis part 2 (genlog)

```
# Query_time distribution
#   1us
#  10us
# 100us
#   1ms
#  10ms
# 100ms
#    1s
#  10s+
# Review information
#    first_seen: 2010-03-12 10:45:02
#     last_seen: 2010-03-12 12:30:01
#   reviewed_by:
#   reviewed_on:
#      comments:
```

Query analysis part 2 (binlog)

```
# Query_time distribution
#   1us
#  10us
# 100us
#   1ms
#  10ms
# 100ms
#    1s  ################################################################
#  10s+  ################
# Review information
#    first_seen: 2010-03-12 10:45:02
#     last_seen: 2010-04-10 08:26:51
#   reviewed_by:
#   reviewed_on:
#      comments:
```

Query analysis part 3 (genlog)

```
# Tables
#    SHOW TABLE STATUS FROM `proddb` LIKE 'colors'\G
#    SHOW CREATE TABLE `proddb`.`colors`\G
update colors set publishable_flag = true where id = 267354\G
# Converted for EXPLAIN
# EXPLAIN
select  publishable_flag = true from colors where  id = 267354\G
```

Query analysis part 3 (binlog)

```
# Tables

#    SHOW TABLE STATUS FROM `proddb` LIKE 'colors'\G

#    SHOW CREATE TABLE `proddb`.`colors`\G

update colors set publishable_flag = true where id = 284297\G

# Converted for EXPLAIN

# EXPLAIN

select  publishable_flag = true from shopping_events where  id =
284297\G
```

Query analysis part 1 (binlog)

```
# Query 5: 3.90 QPS, 297.34Mx concurrency, ID 0x188B27831A9DE05B at byte
596881917

# This item is included in the report because it matches --limit.

#              pct   total     min     max     avg     95%  stddev median

# Count          1      16829

# Exec time    5 1284195222560s 0 4294967295s 76308469s 992ms 546294873s 0

# Databases                1 proddb

# Time range 2010-04-10 07:14:52 to 2010-04-10 08:26:51

# bytes          0 969.38k      58      59   58.98   56.92        0   56.92

# error cod      0       0       0       0       0       0       0       0
```

```
update colors set publishable_flag = true where id = 284297\G
```

Query Analysis Summary

```
# Profile

# Rank Query ID           Response time              Calls  R/Call
Item

# ==== ================== ========================== ====== ==============

#    1 0x85FFF5AA78E5FF6A 985694996247 1.0000 39.8% 177057   55671054.8720
                                                                BEGIN

#    2 0x8F345B7550CA9147 466433474976 3.0000 18.8% 686030    6799024.4592
                                                  INSERT user_events_live

#    3 0xCACEE7C0CF15B39B 261993005782 1.0000 10.6%  63756   41093074.5000
                                                              UPDATE skus

#    4 0x308A3C4E761F5834 137868450337 5.0000  5.6%  17845   77258868.2194
                                                   UPDATE shopping_events

#    5 0x188B27831A9DE05B 128419522256 0.0000  5.2%  16829   76308468.8668
                                                        UPDATE colors

#    6 0xD8F78067CE3F07AB 127990025536 0.0000  5.2%  18180   70401554.2002
                                                              UPDATE offers

#    7 0x3C70600B502E3A08 121547574585 5.0000  4.9%  16829   72225072.5447
                                                           UPDATE products
```

The query_review table

- Remember, we did the command:

```
perl mk-query-digest --limit 100% \
--review h=127.0.0.1,P=3307,D=maatkit,t=query_review,u=user,p=pass \
--create-review-table --type binlog binlog325.sql > binlogoutput.txt
```

- What does the query review table look like?

```
mysql> select * from query_review where checksum=0x188B27831A9DE05B\G
*************************** 1. row ***************************
   checksum: 17685507227138048 91
fingerprint: update colors set publishable_flag = true where id = ?
     sample: update colors set publishable_flag = true where id =
100563
 first_seen: 2010-03-12 10:45:02
  last_seen: 2010-04-10 08:26:51
reviewed_by: NULL
reviewed_on: NULL
   comments: NULL
1 row in set (0.00 sec)
```

How do we review a query?

- EXPLAIN, SHOW CREATE TABLE, etc.
- Now what?

```
mysql> update query_review set reviewed_by='Sheeri',
reviewed_on=now(), comments='This query is OK, it uses the primary key
to search on.' where checksum=17685507227138048 91;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

- One query down.....

```
mysql> select count(*) from query_review where reviewed_on is null;
+----------+
| count(*) |
+----------+
|      769 |
+----------+
1 row in set (0.00 sec)
```

- 769 to go!

Systematic approach

- You can look at a few queries per day

- Reviewed queries do not appear in subsequent reports of mk-query-digest
  - If you have something in reviewed_by
  - Unless you specify --report-all

Query review

- --no-report to just parse a log to the database:

```
perl mk-query-digest --limit 100% --no-report -review \
h=127.0.0.1,P=3307,D=maatkit,t=query_review,u=user,p=pass \
--type binlog mybinlog.txt
```

- Can save counts, etc to an historical table

```
perl mk-query-digest --limit 100% --no-report -review \
h=127.0.0.1,P=3307,D=maatkit,t=query_review,u=user,p=pass \
--create-review-history-table -review-history \
h=127.0.0.1,P=3307,D=maatkit,t=qr_history,u=user,p=pass \
--type genlog mygenlog.txt
```

Query review history

```
mysql> select * from qr_history where checksum=0x188B27831A9DE05B\G

*************************** 1. row ***************************
            checksum: 17685507227138048 91
              sample: update colors set publishable_flag = true where id
= 284297
              ts_min: 2010-04-10 07:14:52
              ts_max: 2010-04-10 08:26:51
              ts_cnt: 16829
      Query_time_sum: 1.2842e+12
      Query_time_min: 0
      Query_time_max: 4.29497e+09
   Query_time_pct_95: 0.992137
   Query_time_stddev: 5.46295e+08
   Query_time_median: 0
       Lock_time_sum: NULL
       Lock_time_min: NULL
       Lock_time_max: NULL
    Lock_time_pct_95: NULL
    Lock_time_stddev: NULL
    Lock_time_median: NULL
                                          Rows_sent_sum: NULL
                                          Rows_sent_min: NULL
                                          Rows_sent_max: NULL
                                       Rows_sent_pct_95: NULL
                                       Rows_sent_stddev: NULL
                                       Rows_sent_median: NULL
                                     Rows_examined_sum: NULL
                                     Rows_examined_min: NULL
                                     Rows_examined_max: NULL
                                  Rows_examined_pct_95: NULL
                                  Rows_examined_stddev: NULL
                                  Rows_examined_median: NULL
```

Query review history

```
mysql> select * from qr_history where checksum=0x188B27831A9DE05B\G

*************************** 1. row ***************************
            checksum: 17685507227138048 91
              sample: update colors set publishable_flag = true where id
= 284297
              ts_min: 2010-04-10 07:14:52
              ts_max: 2010-04-10 08:26:51
              ts_cnt: 16829
     Query_time_sum: 1.2842e+12
     Query_time_min: 0
     Query_time_max: 4.29497e+09
   Query_time_pct_95: 0.992137
   Query_time_stddev: 5.46295e+08
   Query_time_median: 0
```

```
************* 2. row *************
checksum: 17685507227138048 91
sample: update colors set
publishable_flag = true where id =
279850
ts_min: 2010-03-24 10:45:01
ts_max: 2010-03-24 12:30:00
ts_cnt: 7109
Query_time_sum: 0
Query_time_min: 0
Query_time_max: 0
Query_time_pct_95: 0
Query_time_stddev: 0
Query_time_median: 0
```

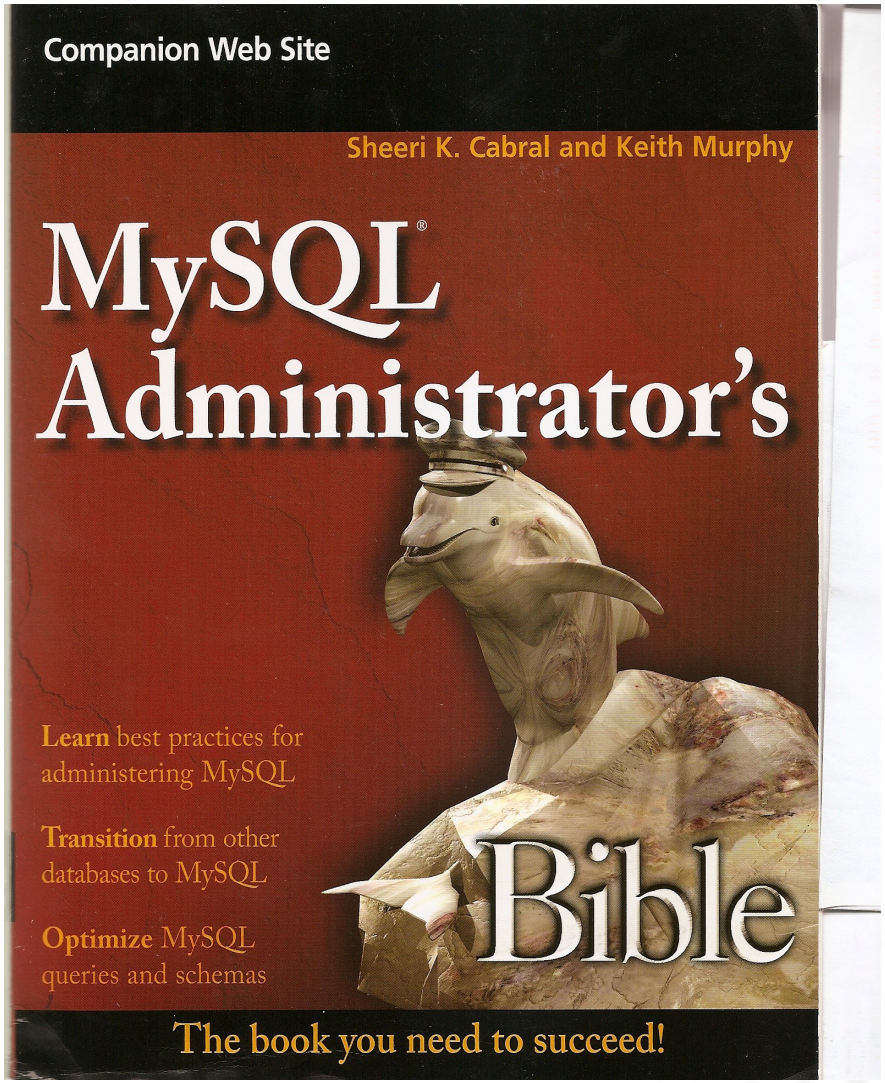
What I'd like to see

- Besides query reviews being common practice...
- More fields in the query_review table
  - what index(es) are used – fields, index type
  - Tables involved and their approx row count
  - Approx rows examined from EXPLAIN
- More fields in the query_review_history table
  - Source (genlog, binlog, etc)
  - When the review was done.

Start Today!

- Grab a log
- Find a test machine with a database
- Start EXPLAINing all your queries
- mk-query-digest has tons of other great features other than query reviews.....

Thank You.



Win a signed copy of Sheeri's book.

Leave your business card and you could win a book. We'll invite you to read our blog posts, follow us on twitter, and join our next webinars.

Drawing will be immediately after the talk once all cards are collected.

Thank You

Questions, Comments, Feedback?

Sheeri Cabral

cabral@pythian.com

Blog: www.pythian.com/news/author/sheeri

Twitter: @sheeri

Ask me about saving 15% on our Adoption Accelerator for MySQL while at MySQL Conference 2010!