

**FOSDEM 2011**

# **A practical overview of Maatkit**

**Stéphane Combaudon**

[stephane.combaudon@gmail.com](mailto:stephane.combaudon@gmail.com)

# Maatkit in brief

- Set of Perl scripts (30+) for MySQL DBAs (& PostgreSQL to some extent)
- GPL
- Created by Baron Schwartz, maintained mainly by Percona
- In active development
- See <http://www.maatkit.org>

# A strange slow query - 1

```
CREATE TABLE user (  
  user_id int NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  login varchar(30) NOT NULL DEFAULT '',  
  ...  
  UNIQUE KEY (login)) ENGINE=InnoDB;
```

- Output from mysqldumpslow:

```
Count: 195 Time=33.05s (6445s) Lock=0.00s (0s) Rows=129903.9 (25331256)  
SELECT user_id FROM user WHERE login = N
```

- Seems like the optimizer thinks the unique index lookup isn't the best execution plan?!

# A strange slow query - 2

- However tests with real logins always take 0s
- We need more information to investigate

```
mk-query-digest --limit 10 mysql-slow.log
```

```
# Query 2: 0.00 QPS, 0.07x concurrency, ID 0xEB7F16321805BBED at byte 9841263
# Scores: Apdex = 0.00 [1.0], V/M = 0.08
# Query_time sparkline: |      ^|
# Time range: 2010-12-21 06:06:08 to 2010-12-22 05:20:36
# Attribute      pct   total      min      max      avg      95%   stddev   median
# =====
# Count          1     178
# Exec time      3    5879s     31s     35s     33s     35s     2s      32s
# Lock time      0         0         0         0         0         0         0         0
# Rows sent     99 625.36M      0  26.49M   3.51M  21.31M   7.09M      0
# Rows examine  15   1.71G 775.10k  27.14M   9.81M  22.38M   5.99M   8.43M
# Query size    0   27.42k      84     363  157.72  346.17   66.96  118.34
[...]
```

```
# EXPLAIN /*!50100 PARTITIONS*/
SELECT user_id FROM user WHERE login = 0\G
```

# A strange slow query - 3

- Can you see the reason of the problem now?
  - login is a VARCHAR, but the query uses an INT
  - MySQL converts login to an INT
  - The index cannot be used anymore
  - A much more expensive exec. plan is chosen
- Easy fix: the application should ensure the value is always a string
- Aside note: there is much more to say on this query!!

# Catching queries live - 1

- DBA: Do you know the server runs lots of admin commands that have an important overhead?
- Colleague: Impossible, these queries are very cheap, even if they are run tons of times
- mk-query-digest can parse tcpdump output
- So you can have a report of the most expensive queries running at the moment

# Catching live queries - 2

```
tcpdump -i eth0 port 3306 -s 65535 -c 10000 -x\  
-n -q -tttt | mk-query-digest --type tcpdump \  
--report-format=profile --limit=10
```

```
# Profile
```

# Rank	Query ID	Response time	Calls	R/Call	Apdx	V/M	Item
# 1	0x04FE01C5B31FD305	1.4855 40.9%	906	0.0016	1.00	0.20	ADMIN PING
# 2	0x348032401B50B8DE	0.4378 12.1%	38	0.0115	1.00	0.03	ADMIN STMT_PREPARE
# 3	0x23923E0C446133EF	0.3690 10.2%	61	0.0060	1.00	0.19	SELECT
# 4	0x5D51E5F01B88B79E	0.2904 8.0%	7	0.0415	1.00	0.23	ADMIN CONNECT
# 5	0xD236026DAE106F23	0.2481 6.8%	115	0.0022	1.00	0.03	SELECT
# 6	0x62832C3BC7102748	0.2053 5.7%	29	0.0071	1.00	0.06	SELECT
# 7	0xAA6A8F46BCBA8C34	0.0915 2.5%	22	0.0042	1.00	0.02	SELECT
# 8	0xCAD2DA760976EA94	0.0693 1.9%	22	0.0032	1.00	0.03	SELECT
# 9	0x36B36079433E6973	0.0625 1.7%	23	0.0027	1.00	0.01	SELECT
# 10	0x4E4572ED44FF00A3	0.0613 1.7%	15	0.0041	1.00	0.04	SELECT
# MISC	0xMISC	0.3110 8.6%	1286	0.0002	NS	0.0	<66 ITEMS>

```
e0
```

- More than 60% of the time is spent to perform useless queries...

# Bad backups - 1

- Colleague: A dedicated slave provides backups
- DBA: Great! Are you sure the slave has the same data as its master?
- Cllg: ??? We have a Nagios alert if replication breaks
- DBA: And what if replication goes out-of-sync without any error?
- Cllg: ?????? Isn't replication supposed to work?

# Bad backups - 2

- mk-table-checksum provides consistency checks between a master and its slaves
- Concept of the tool
  - Run checksums on master and let them flow through replication to the slaves
  - When replication has caught up, you can compare the checksums

# Bad backups - 3

- On the master, running the checksums:

```
mk-table-checksum --chunk-size=100000 \  
--replicate=mydb.checksum masterhost
```

- When replication has caught up:

```
mk-table-checksum --replicate=mydb.checksum \  
--replicate-check=1 masterhost
```

Differences on P=3306,h=10.8.3.4

DB	TBL	CHUNK	CNT_DIFF	CRC_DIFF	BOUNDARIES
mydb	search	1	-42270	1	`search_query_id` > 0 AND `search_query_id` < '2975138'
mydb	search	2	-2417	1	`search_query_id` >= '2975138' AND `search_query_id` < '5950259'
mydb	search	3	-3236	1	`search_query_id` >= '5950259' AND `search_query_id` < '8925380'
mydb	search	4	-1869	1	`search_query_id` >= '8925380' AND `search_query_id` < '11900501'
mydb	search	5	-877	1	`search_query_id` >= '11900501' AND `search_query_id` < '14875622'
mydb	search	6	-718	1	`search_query_id` >= '14875622' AND `search_query_id` < '17850743'
mydb	search	7	-611	1	`search_query_id` >= '17850743' AND `search_query_id` < '20825864'
mydb	search	8	-551	1	`search_query_id` >= '20825864' AND `search_query_id` < '23800985'

- Next step is to resync the backup slave!

# Resyncing tables - 1

- Colleague: I had to skip events on a table, how can I resync the table?
- mk-table-sync can use information from mk-table-checksum
- It generates queries to resolve differences
- For safety, the queries should always be run on the master

# Resyncing tables - 2

- `mk-table-sync --replicate=mydb.checksum --print --sync-to-master slavehost`

```
REPLACE INTO `search_query` (`search_query_id`, `search_query`, `language`, `channel_id`, `hit_count`, `result_count`, `first_video_id`, `last_seen`) VALUES ('6009', 'hikaru no go special', 0x6A70, '8', '15957', '2440', '6689889', '2011-02-03') /*m
aatkit src_db:
ktablesync dst_db:
ablesync lock:1 transaction:1 changing_src:dba_checks.checksum replicate:dba_checks.checksum bidirectional:0 pid:31925 user:stc host:logstore-04*/;
REPLACE INTO `search_query` (`search_query_id`, `search_query`, `language`, `channel_id`, `hit_count`, `result_count`, `first_video_id`, `last_seen`) VALUES ('6018', 'clannad after story 21', 0x6A70, '8', '16796', '547', '15032133', '2011-02-03') /*m
aatkit src_db:
ktablesync dst_db:
ktablesync lock:1 transaction:1 changing_src:dba_checks.checksum replicate:dba_checks.cksum bidirectional:0 pid:31925 user:stc host:logstore-04*/;
```

- Use `--execute` instead of `--print` to run the queries
- This tool is **very** powerful and must be used with **extra** care!

# Updating big tables - 1

- Colleague : Last time we wanted to perform massive update in this big table, the script ran for 5 days. Is there a better way?
- mk-archiver is optimized to scan efficiently big tables for purging or archiving purposes
- You can extend it with plugins to make it perform whatever task needing to scan lots of data

# Updating big tables - 2

```
mk-archiver --source D=mydb,t=video \  
--dest D=mydb,t=dest,m=my_plugin --where "... " \  
--no-delete --commit-each --limit=100000 \  
--progress=100000 --bulk-insert
```

TIME	ELAPSED	COUNT
2011-02-03T15:35:05	0	0
2011-02-03T15:35:10	5	100000
2011-02-03T15:35:30	25	200000
2011-02-03T15:35:53	48	300000
2011-02-03T15:36:14	69	400000
2011-02-03T15:36:35	90	500000
2011-02-03T15:36:54	109	600000
2011-02-03T15:37:15	130	700000
2011-02-03T15:37:35	149	800000
2011-02-03T15:37:54	169	900000
2011-02-03T15:38:13	188	1000000
2011-02-03T15:38:32	207	1100000
2011-02-03T15:38:51	226	1200000
2011-02-03T15:39:10	245	1300000
2011-02-03T15:39:29	264	1400000
2011-02-03T15:39:49	284	1500000

- 20s to update each chunk, stable over time
- The table has 20M rows
- Job done in 4,000s, a little more than 1h!

# If you don't like it, kill it

- Colleague: Could we kill long-running queries automatically (and have a log)?

```
mk-kill --busy-time=10 --kill --heartbeat \  
--iterations 0 --print
```

```
# 2011-02-02T13:57:34 Checking processlist, iteration 0  
# 2011-02-02T13:57:34 Matched 0 queries  
# 2011-02-02T13:57:39 Checking processlist, iteration 0  
# 2011-02-02T13:57:39 Matched 0 queries  
# 2011-02-02T13:57:44 Checking processlist, iteration 0  
# 2011-02-02T13:57:44 Matched 0 queries  
# 2011-02-02T13:57:49 Checking processlist, iteration 0  
# 2011-02-02T13:57:49 Matched 1 queries  
# 2011-02-02T13:57:49 KILL 88 (Query 13 sec) select ...  
# 2011-02-02T13:57:49 Killed 88  
# 2011-02-02T13:57:54 Checking processlist, iteration 0  
# 2011-02-02T13:57:54 Matched 0 queries
```

- Options changed recently, check the man page!

Questions?