The Five Minute DBA

MySQL User Conference 2010

Matt Yonkovit

Percona Inc
MySQLPerformanceBlog.com
About Us...

- http://www.percona.com
- http://MysqlPerformanceBlog.com
- http://www.bigdbaahead.com

Matt Yonkovit
Principal Architect
Who is a five minute dba

• A five minute dba is someone who is not a database administrator by trade, but rather is drafted into the role out of necessity
• Typically they have development skills or General system administration skills
• These are the jacks of all trades
When do you become a DBA

• When it hits the fan of course!
Disclaimer!

It's important to distinguish this up front:

- I am giving you the easy answers, which are not always the correct answers.
- If you only have 5 minutes to spend on one of these database topics, you lose a lot of flexibility.
- I could spend lots of time talking about everything here, instead I am focusing on maximizing your benefit.
Common Mistakes

• Running old outdated buggy versions
• Default Configuration
• Inadequate resources (shared or dedicated)
• Not Enough Memory
• Over-allocation of Memory
• Treating the database like a black box
• Poorly written & optimized queries
• Too many queries
• Too many Connections
Performance Analysis

• Define what is slow (as low level as possible): single page, entire app, writing data, reading data etc.
  – Understand your workload
  – Response time

• You want to target your specific problem, however reviewing your entire stack is a good thing

• Don't tune via google (each problem is unique)

• Use Tools available to you
Tuning walk through

• Define what is slow? If its a single page, the process is drastically different then if its general slowness.
  – Assume General Slowness
  – DO NOT ASSUME ITS THE DATABASE!
• I do a quick review of the obvious:
  –
Versions?

- MySQL 4.0
- MySQL 5.0
- MySQL 5.1
- MySQL 5.4/5.5
- Percona Server
- MariaDB
- OurDelta
Storage Engine Selection

- Innodb
- Innodb Plugin
- XtraDB
- MyISAM
- Maria
- Many Many more....
Hardware & OS

• Old Kernel?
• Multi-core servers, scalability getting better (~16 cores + )
• Fit your hot data into memory!
  – Typically more memory is better
• IO is a Major bottleneck
  – RAID For your databases is a must
  – Battery Backed Controller
  – RAID 10 typically better then RAID 5
  – SSD's for optimal performance
• Network should not be forgotten
Tuning Walk through

• Next I start by looking at the server (Sar, vmstat, top, iostat -x)
  – Too much io, maybe not enough memory, missing indexes, or bad config
  – Cpu maxed out could be lots of users, lots of concurrent queries, or something else.
  – Is the problem even MySQL? Often times we see issues with non-database software
  – top process is something else (i.e. apache, php, cron job, etc)
  – Network is often overlooked
Operating System Tools

- Sysstat
  - iostat
  - Sar
- Vmstat
- Top
SYSSTAT

• Sar is your friend, every unix/linux server should setup sar to collect stats in the background.
  – Sar Can give you historical reports on: CPU, Disk, Memory, Network, and a lot more.
  – Scheduled to collect data via cron

• Iostat is valuable in gauging disk performance
  – avwait+svctim = response time (iostat -x)
**Sar Network Stats**

```
09:35:40 PM IFACE  rxpck/s  txpck/s  rxB/s  txB/s  rxcmp/s  txcmp/s  rxmcst/s
09:35:41 PM  lo   0.00   0.00   0.00   0.00   0.00   0.00   0.00
09:35:41 PM  eth1  38472.00  38467.00  2960.72  27532.34   0.00   0.00   0.00
09:35:41 PM  eth0   11.00    3.00   2.35   0.41   0.00   0.00   0.00
09:35:42 PM IFACE  rxpck/s  txpck/s  rxB/s  txB/s  rxcmp/s  txcmp/s  rxmcst/s
09:35:42 PM  lo   0.00   0.00   0.00   0.00   0.00   0.00   0.00
09:35:42 PM  eth1  38441.58  38443.56  2969.10  27494.13   0.00   0.00   0.00
09:35:42 PM  eth0    2.00    1.00   0.13   0.47   0.00   0.00   0.00
09:35:43 PM IFACE  rxpck/s  txpck/s  rxB/s  txB/s  rxcmp/s  txcmp/s  rxmcst/s
09:35:43 PM  lo   0.00   0.00   0.00   0.00   0.00   0.00   0.00
09:35:43 PM  eth1  39780.00  39777.00  3070.57  28451.08   0.00   0.00   0.00
09:35:43 PM  eth0   2.00    1.00   0.13   0.47   0.00   0.00   0.00
```
matt@matt-desktop:~$ sar 1
Linux 2.6.31-15-generic (matt-desktop)  04/07/2010    _i686_    (2 CPU)

<table>
<thead>
<tr>
<th>Time</th>
<th>CPU</th>
<th>%user</th>
<th>%nice</th>
<th>%system</th>
<th>%iowait</th>
<th>%steal</th>
<th>%idle</th>
</tr>
</thead>
<tbody>
<tr>
<td>09:37:20 PM</td>
<td>all</td>
<td>0.51</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>99.49</td>
</tr>
<tr>
<td>09:37:21 PM</td>
<td>all</td>
<td>0.00</td>
<td>0.00</td>
<td>0.50</td>
<td>0.00</td>
<td>0.00</td>
<td>99.50</td>
</tr>
<tr>
<td>09:37:22 PM</td>
<td>all</td>
<td>0.00</td>
<td>0.00</td>
<td>0.50</td>
<td>0.00</td>
<td>0.00</td>
<td>99.50</td>
</tr>
<tr>
<td>09:37:23 PM</td>
<td>all</td>
<td>0.00</td>
<td>0.00</td>
<td>0.50</td>
<td>0.00</td>
<td>0.00</td>
<td>99.50</td>
</tr>
<tr>
<td>09:37:24 PM</td>
<td>all</td>
<td>0.00</td>
<td>0.00</td>
<td>0.50</td>
<td>0.00</td>
<td>0.00</td>
<td>100.00</td>
</tr>
<tr>
<td>09:37:25 PM</td>
<td>all</td>
<td>0.00</td>
<td>0.00</td>
<td>0.50</td>
<td>0.00</td>
<td>0.00</td>
<td>99.50</td>
</tr>
<tr>
<td>09:37:26 PM</td>
<td>all</td>
<td>0.00</td>
<td>0.00</td>
<td>0.50</td>
<td>0.00</td>
<td>0.00</td>
<td>100.00</td>
</tr>
<tr>
<td>09:37:27 PM</td>
<td>all</td>
<td>0.00</td>
<td>0.00</td>
<td>0.50</td>
<td>0.00</td>
<td>0.00</td>
<td>99.50</td>
</tr>
</tbody>
</table>
### Sar Memory

```bash
matt@matt-desktop:~$ sar -r 1
Linux 2.6.31-15-generic (matt-desktop)  04/07/2010    _i686_     (2 CPU)

<table>
<thead>
<tr>
<th>Time</th>
<th>kbmemfree</th>
<th>kbmemused</th>
<th>%memused</th>
<th>kbuffers</th>
<th>kbcached</th>
<th>kbcommit</th>
<th>%commit</th>
</tr>
</thead>
<tbody>
<tr>
<td>09:38:53 PM</td>
<td>665232</td>
<td>1395448</td>
<td>67.72</td>
<td>864</td>
<td>193856</td>
<td>1523272</td>
<td>44.77</td>
</tr>
<tr>
<td>09:38:54 PM</td>
<td>665232</td>
<td>1395448</td>
<td>67.72</td>
<td>864</td>
<td>193856</td>
<td>1523272</td>
<td>44.77</td>
</tr>
<tr>
<td>09:38:55 PM</td>
<td>665232</td>
<td>1395448</td>
<td>67.72</td>
<td>864</td>
<td>193856</td>
<td>1523272</td>
<td>44.77</td>
</tr>
<tr>
<td>09:38:56 PM</td>
<td>665232</td>
<td>1395448</td>
<td>67.72</td>
<td>864</td>
<td>193856</td>
<td>1523272</td>
<td>44.77</td>
</tr>
<tr>
<td>09:38:57 PM</td>
<td>665232</td>
<td>1395448</td>
<td>67.72</td>
<td>864</td>
<td>193856</td>
<td>1523272</td>
<td>44.77</td>
</tr>
<tr>
<td>09:38:58 PM</td>
<td>665232</td>
<td>1395448</td>
<td>67.72</td>
<td>864</td>
<td>193856</td>
<td>1523272</td>
<td>44.77</td>
</tr>
<tr>
<td>09:38:59 PM</td>
<td>665232</td>
<td>1395448</td>
<td>67.72</td>
<td>864</td>
<td>193856</td>
<td>1523272</td>
<td>44.77</td>
</tr>
<tr>
<td>09:39:00 PM</td>
<td>665232</td>
<td>1395448</td>
<td>67.72</td>
<td>864</td>
<td>193856</td>
<td>1523272</td>
<td>44.77</td>
</tr>
<tr>
<td>09:39:01 PM</td>
<td>664108</td>
<td>1396572</td>
<td>67.77</td>
<td>896</td>
<td>194588</td>
<td>1523360</td>
<td>44.78</td>
</tr>
<tr>
<td>09:39:02 PM</td>
<td>664108</td>
<td>1396572</td>
<td>67.77</td>
<td>896</td>
<td>194588</td>
<td>1523360</td>
<td>44.78</td>
</tr>
<tr>
<td>09:39:03 PM</td>
<td>664108</td>
<td>1396572</td>
<td>67.77</td>
<td>896</td>
<td>194588</td>
<td>1523360</td>
<td>44.78</td>
</tr>
</tbody>
</table>
```
matt@matt-desktop:~$ iostat -x 1
Linux 2.6.31-15-generic (matt-desktop) 04/07/2010 _i686_ (2 CPU)

avg-cpu: %user  %nice  %system %iowait  %steal  %idle
          0.38    0.04    0.48    0.02   0.00  99.08

Device:          rrqm/s  wrqm/s  r/s  w/s  rsec/s  wsec/s  avgrq-sz  avgqu-sz  await  svctm  %util
sda              0.10   30.82  0.17  0.44  11.77   250.03   428.22    0.02  28.10   2.17  0.13

avg-cpu: %user  %nice  %system %iowait  %steal  %idle
          56.00    0.00    37.50    6.50   0.00    0.00

Device:          rrqm/s  wrqm/s  r/s  w/s  rsec/s  wsec/s  avgrq-sz  avgqu-sz  await  svctm  %util
sda             30.00  12497.00  11.00  102.00   424.00  86640.00  770.48    7.11  58.55  3.68  41.60

avg-cpu: %user  %nice  %system %iowait  %steal  %idle
          38.31    0.00   29.35  19.90   0.00  12.44

Device:          rrqm/s  wrqm/s  r/s  w/s  rsec/s  wsec/s  avgrq-sz  avgqu-sz  await  svctm  %util
sda             19.00  12004.00  10.00  130.00  232.00  111208.00   796.00  10.98  81.94  3.94  55.20
VMSTAT

• Can give you a realtime view of changes on the system to CPU/Memory

```plaintext
# vmstat 1
procs -----------memory---------- ---swap-- -----io------
   r   b  swpd free   buff  cache si  so   bi   bo  in cs us sy id wa
 0 0  96 10856 91496  625748  0  0   3   8  1  2  0  0 100  0
 0 0  96 10856 91496  625772  0  0   0   0  327 22  0  0 100  0
 0 0  96 10856 91496  625772  0  0   0   0  295 18  0  0 100  0
 0 0  96 10856 91496  625772  0  0   0   0  320 19  0  0 100  0
 0 0  96 10856 91496  625772  0  0   0   0  302 16  0  0 100  0
 0 0  96   9116 91496  625772  0  0   0   0  304 47  0  0 100  0
 0 0  96   9092 91496  625772  0  0   0   0  349 34  0  0 100  0
 0 0  96   9100 91496  625772  0  0   0   0  302 16  0  0 100  0
```
• Get a current Snapshot of running processes

Tasks: 159 total, 2 running, 157 sleeping, 0 stopped, 0 zombie
Cpu(s): 15.3%us, 15.5%sy, 0.0%ni, 54.6%id, 0.8%wa, 3.3%hi, 10.6%si, 0.0%st
Mem: 2060680k total, 1303952k used, 756728k free, 664k buffers
Swap: 1341388k total, 64k used, 1341324k free, 102512k cached

<table>
<thead>
<tr>
<th>PID</th>
<th>USER</th>
<th>PR</th>
<th>NI</th>
<th>VIRT</th>
<th>RES</th>
<th>SHR</th>
<th>S</th>
<th>%CPU</th>
<th>%MEM</th>
<th>TIME+</th>
<th>COMMAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>9460</td>
<td>root</td>
<td>20</td>
<td>0</td>
<td>1028m</td>
<td>1.0g</td>
<td>524</td>
<td>R</td>
<td>90</td>
<td>51.0</td>
<td>2:10.80</td>
<td>redis-server</td>
</tr>
<tr>
<td>9544</td>
<td>matt</td>
<td>20</td>
<td>0</td>
<td>2468</td>
<td>1184</td>
<td>884</td>
<td>R</td>
<td>0</td>
<td>0.1</td>
<td>0:00.03</td>
<td>top</td>
</tr>
<tr>
<td>1</td>
<td>root</td>
<td>20</td>
<td>0</td>
<td>2528</td>
<td>1116</td>
<td>768</td>
<td>S</td>
<td>0</td>
<td>0.1</td>
<td>0:01.07</td>
<td>init</td>
</tr>
<tr>
<td>2</td>
<td>root</td>
<td>15</td>
<td>-5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>S</td>
<td>0</td>
<td>0.0</td>
<td>0:00.00</td>
<td>kthreadd</td>
</tr>
<tr>
<td>3</td>
<td>root</td>
<td>RT</td>
<td>-5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>S</td>
<td>0</td>
<td>0.0</td>
<td>0:00.01</td>
<td>migration/0</td>
</tr>
<tr>
<td>4</td>
<td>root</td>
<td>RT</td>
<td>-5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>S</td>
<td>0</td>
<td>0.0</td>
<td>0:00.04</td>
<td>ksoftirqd/0</td>
</tr>
<tr>
<td>5</td>
<td>root</td>
<td>RT</td>
<td>-5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>S</td>
<td>0</td>
<td>0.0</td>
<td>0:00.00</td>
<td>watchdog/0</td>
</tr>
<tr>
<td>6</td>
<td>root</td>
<td>RT</td>
<td>-5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>S</td>
<td>0</td>
<td>0.0</td>
<td>0:00.01</td>
<td>migration/1</td>
</tr>
</tbody>
</table>
cacti

• Great Visual Presentation of Historical Stats
  – MySQL, Apache, Sphinx, Memcached, ( Entire Stack)

• Baron Wrote and Maintains the MySQL Templates

• Can add a lot of value...

• Example:
  Mysterious Slowdown
Last night

- OMG The Server was sooo slow
Last Night Dig Deeper

![Graph showing load average from 01 to 07 with 1 Minute Average: 0.12, 5 Minute Average: 0.08, 15 Minute Average: 0.04, Total: 0.04]
And Even Deeper

Oops slow down was the nightly backup....
#2 Slowdown

```
db-master - MySQL Temporary Objects

Created Tmp Tables
- Cur: 88.0m  Avg: 342.8m  Max: 68.4

Created Tmp Disk Tables
- Cur: 81.3m  Avg: 151.5m  Max: 3.6

Created Tmp Files
- Cur: 0.0    Avg: 0.0    Max: 0.0
```
Tuning Walk Through

• Do a quick review of the configuration parameters, making sure that the values are set to something reasonable.
Top General ConfigParms

- `query_cache_size` (Disable in many versions)
- `log_slow_queries`
  - `long_query_time`
- `max_connections`
- `expire_logs_days`
- MyISAM
- `key_buffer_size`
Top InnoDB ConfigParms

- `innodb_buffer_pool_size`
- `innodb_flush_log_at_trx_commit`
- `innodb_log_file_size`

Plugin/MYSQL 5.5+/XtraDB Specific
- `innodb_io_capacity`
- `innodb_read_ahead`
Memory and Swapping

• Disk IO is bad, Avoiding swapping things that should be in memory to disk (do not disable swap however)
• read_buffer, read_rnd_buffer, sort_buffer, join_buffer are all allocated per thread so be careful!
• You want temp tables to be built in memory not on disk (TMPFS?):

| Created_tmp_disk_tables      | 48  |
| Created_tmp_files            |  5  |
| Created_tmp_tables           | 229 |
Tuning walk through

- Next I typically see what’s running in the data, with `show processlist` and `show innodb status`.
  - What is happening right now? Is there long running queries? Are there excessive locks?
  - If the slowdown was in the past, and not now I will analyze the slow query logs.
Database Specific Tools

- Mysql
  - “show full processlist”
  - “show global status”
  - “Show innodb status”
- Mysqladmin
- Slow Query Log
  - Mk-query-digest
  - Mysqldumpsslow
- Innotop
- Mytop
- mext
### Show Processlist

```sql
mysql> show processlist
_row_1_row
-------
<table>
<thead>
<tr>
<th>Id</th>
<th>User</th>
<th>Host</th>
<th>db</th>
</tr>
</thead>
<tbody>
<tr>
<td>52</td>
<td>bench</td>
<td>domU-12-31-39-0C-41-81.compute-1.internal:42315</td>
<td>simple_benchmark</td>
</tr>
<tr>
<td>65</td>
<td>bench</td>
<td>domU-12-31-39-0C-41-81.compute-1.internal:42328</td>
<td>simple_benchmark</td>
</tr>
<tr>
<td>66</td>
<td>bench</td>
<td>domU-12-31-39-0C-41-81.compute-1.internal:42329</td>
<td>simple_benchmark</td>
</tr>
</tbody>
</table>
```

1. **Id**: 52
   - **User**: bench
   - **Host**: domU-12-31-39-0C-41-81.compute-1.internal:42315
   - **db**: simple_benchmark
   - **Command**: Sleep
   - **Time**: 0
   - **State**: NULL

14. **Id**: 65
   - **User**: bench
   - **Host**: domU-12-31-39-0C-41-81.compute-1.internal:42328
   - **db**: simple_benchmark
   - **Command**: Query
   - **Time**: 0
   - **State**: init
   - **Info**: insert into simple_benchmark.simple_comments (lookup_id, comment, mydate) values ( '8519163', 'Comme

15. **Id**: 66
   - **User**: bench
   - **Host**: domU-12-31-39-0C-41-81.compute-1.internal:42329
   - **db**: simple_benchmark
   - **Command**: Sleep
   - **Time**: 0
   - **State**: NULL
MySQLadmin

- Mysqladmin extended -r -i1 will give you mysql stats at 1 second intervals, change the -i to increase decrease...

- A few stats of note:
  - Created_tmp_disk_tables
  - Created_tmp_tables
  - Innodb_rows_deleted
  - Innodb_rows_inserted
  - Innodb_rows_read
  - Innodb_rows_updated
  - Qcache_hits
  - Qcache_inserts
  - Questions
**mext**

- Mext can take the output from mysqladmin and put the status variables side by side, this makes it very convenient to look for differences

```plaintext
./mext -- mysqladmin ext -uroot -p -ri10 -c5
Enter password:
Aborted_clients       414  0  0  0  0  0
Aborted_connects      5567  0  0  1  0
Binlog_cache_disk_use 143  0  0  0  0
Binlog_cache_use      1684365 137 103 121 103
Bytes_received        6573889322 1019405 976357 1004167 1049549
[cut]
Com_alter_table      38510  0  3  2  0
Com_delete           8565850 211 181 170 170
[cut]
Com_drop_table       25934  0  2  1  0
[cut]
Com_insert           18209688 431 332 400 382
Com_insert_select    2746057  0  0  0  0
[cut]
Com_select           20053974 2304 2175 2257 2363
```
Slow Query Log

• The Slow Query log is your friend! As mentioned earlier queries that take over a certain amount of time are logged here. (Percona Patches for complete picture)
• Summarize the slow query log with Maatkit, mk-query-digest or mysqldumpsow if maatkit is not available
• Look for not only the longest running query, but also the one with lots of executions and cumulated time
Mk-Query-Digest Sample output

<table>
<thead>
<tr>
<th>#</th>
<th>Rank</th>
<th>Query ID</th>
<th>Response time</th>
<th>Calls</th>
<th>R/Call</th>
<th>Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0x8C7EB27BB63FF462</td>
<td>151160.7560</td>
<td>20.7%</td>
<td>71573</td>
<td>2.1120</td>
<td>SELECT inventory?_? items</td>
</tr>
<tr>
<td>2</td>
<td>0x63492E9334BB2FC6</td>
<td>138992.1057</td>
<td>19.0%</td>
<td>48642</td>
<td>2.8575</td>
<td>SELECT inventory?_? items</td>
</tr>
<tr>
<td>3</td>
<td>0x5944495FF54B5660</td>
<td>92821.6612</td>
<td>12.7%</td>
<td>21195</td>
<td>4.3794</td>
<td>SELECT inventory?_? items</td>
</tr>
<tr>
<td>4</td>
<td>0x91C4D337AA26ACC3</td>
<td>70416.1841</td>
<td>9.6%</td>
<td>11453</td>
<td>6.1483</td>
<td>SELECT friends</td>
</tr>
<tr>
<td>5</td>
<td>0x3D9076526CC5050F</td>
<td>51275.8723</td>
<td>7.0%</td>
<td>9777</td>
<td>5.2445</td>
<td>SELECT inventory?_? items</td>
</tr>
<tr>
<td>6</td>
<td>0x31BCEC4EC58F55DA</td>
<td>29466.2476</td>
<td>4.0%</td>
<td>9719</td>
<td>3.0318</td>
<td>SELECT inventory?_? outfit_items items</td>
</tr>
<tr>
<td>7</td>
<td>0x2FF2EA7DB37919C1</td>
<td>25728.2731</td>
<td>3.5%</td>
<td>11822</td>
<td>2.1763</td>
<td>SELECT markings</td>
</tr>
<tr>
<td>8</td>
<td>0xB8E818280406E88F</td>
<td>22495.9115</td>
<td>3.1%</td>
<td>9440</td>
<td>2.3830</td>
<td>SELECT quests quest_status</td>
</tr>
</tbody>
</table>
# Query 2: 1.24 QPS, 3.54x concurrency, ID 0x63492E9334BB2FC6 at byte 868762
# This item is included in the report because it matches --limit.
# Count 18 48642
# Exec time 18 138992s 1s 25s 3s 7s 2s 2s
# Lock time 29 40s 24us 219ms 815us 568us 8ms 103us
# Rows sent 16 1.18M 0 109 25.49 62.76 16.46 20.43
# Rows exam 0 2.36M 0 218 50.97 124.25 32.93 40.45
# Users 1 user1
# Hosts 10 db1... (7420), db1... (6991) ... 8 more
# Databases 2 userdata2 (26716), userdata4 (21926)
# bytes 13 7.37M 156 160 158.98 158.58 2 158.58

```sql
# Tables
# /* read */ SHOW TABLE STATUS FROM `user` LIKE 'inentory'\G
# /* read */ SHOW CREATE TABLE `user`.'inventory'\G
# /* read */ SHOW TABLE STATUS FROM `user` LIKE 'items'\G
# /* read */ SHOW CREATE TABLE `user`.'items'\G
# EXPLAIN
SELECT name, url, item_id, items.type, stype, level FROM inventory
INNER JOIN items USING (item_id) WHERE (owner_id = 323 AND status = 'p')\G
```
Other Tools

- Innotop and mytop: These give a nice top like interface to show you what's happening in the database at the current moment (i.e., running queries, current stats, etc.)
- New Relic is awesome for Rails...
Memcached?

- Review: More memory the better
- Caching frequently used data is an excellent way to boost performance
- This typically requires changes to your app
- Performance improvement can be dramatic
Tuning walk through

- Review the slow query log, parse the log with mk-query-digest.
- If I see something here, I will analyze it
- If you're still at a loss, try using tcpdump to dump SQL statements into mk-query-digest and look for repeated queries. (Easier than the general query log if you need sub second slow queries).
Query Tuning

- Use Explain Plan
- Add Indexes where needed, do not over index
  - mk-duplicate-key-checker
- Lots of small queries can sometimes be more impactful than 1 big one.
Query Tuning

mysql> explain SELECT `userid` AS `userid` FROM `user` WHERE `b_id`='301965' ;

+----+----------+---------+------|---------------|---------+--------|------|-----|-----------------+
<table>
<thead>
<tr>
<th>id</th>
<th>select_type</th>
<th>table</th>
<th>type</th>
<th>possible_keys</th>
<th>key</th>
<th>key_len</th>
<th>ref</th>
<th>rows</th>
<th>Extra</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SIMPLE</td>
<td>user</td>
<td>ALL</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>9774</td>
<td>Using where</td>
</tr>
</tbody>
</table>

1 row in set (0.00 sec)

• Notice Full scan
• 9K+ rows each run
Add Index

mysql> explain SELECT `userid` AS `userid` FROM `user` WHERE `b_id`='301965' ;

<table>
<thead>
<tr>
<th>id</th>
<th>select_type</th>
<th>table</th>
<th>type</th>
<th>possible_keys</th>
<th>key</th>
<th>key_len</th>
<th>ref</th>
<th>rows</th>
<th>Extra</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SIMPLE</td>
<td>user</td>
<td>ALL</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>9774</td>
<td>Using where</td>
</tr>
</tbody>
</table>

1 row in set (0.00 sec)

mysql> alter table user add key (b_id);

Query OK, 10062 rows affected (0.31 sec)
Records: 10062 Duplicates: 0 Warnings: 0

mysql> explain SELECT `blocked` AS `userid` FROM `user` WHERE `b_id`='301965' ;

<table>
<thead>
<tr>
<th>id</th>
<th>select_type</th>
<th>table</th>
<th>type</th>
<th>possible_keys</th>
<th>key</th>
<th>key_len</th>
<th>ref</th>
<th>rows</th>
<th>Extra</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SIMPLE</td>
<td>user</td>
<td>ref</td>
<td>b_id</td>
<td>b_id</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1 row in set (0.00 sec)
Questions?

- matt@percona.com