Analyze MySQL Query Performance
with Percona Cloud Tools

Vadim Tkachenko
Percona, Co-founder /CTO

www.percona.com
cloud.percona.com
www.MySQLPerformanceBlog.com
This webinar

Performance

Percona Cloud Tools

Queries metrics
Why do we care about performance

As usually it comes to money...
Why do we care about performance

As usually it comes to money…

We want to get what we paid for
In case with databases

We pay for servers

We expect to have queries done
Performance

Tasks

Time
Good Performance (response time)

Task executed in time

- Lunch order served within 5 mins
- Query executed in 3ms
Good Performance (throughput)

Expected amount of tasks executed in time period

• Support engineer handled 20 cases per day
• Database executed 3000 queries per second
What is more important?

Response time

Throughput?
Response time

If you care about users

• Web page loaded in less than 1 sec
Throughput

If you care about investments

• $10,000 server should handle 100,000 users per day
Performance optimization - Balancing act

Maximize throughput (handled users per time period) keeping reasonable response time
Percona Cloud Tools

Tools for performance tasks
Database job is

Queries

Let’s make sure it is done properly
Percona Cloud Tools

Performance Optimization
• Find query that runs too long

Troubleshooting
• Why this query runs too long
  • Access to a history of queries

Capacity planning
• Estimate how this query will run in future

Queries review
• Are there new queries in database?
Percona Cloud Tools are for:

- Database users
- Application developers who run queries
- DBAs
Percona Cloud Tools why?

To make performance tasks more efficient

• “Queries review is always in my todo, but I never do it because it is too complicated”
Percona Cloud Tools alternatives

**pt-query-digest**
- Command line tool

**BOX Anemometer**
- [https://github.com/box/Anemometer](https://github.com/box/Anemometer)
- DIY and host yourself

**MySQL Enterprise Query Analyzer**
- Mysql-proxy based with MySQL Enterprise subscription

**MONYog**
- Does not provide history

**New Relic**
- low resolution/lack of detail
Components

Slow log files

• Percona Server, Percona XtraDB Cluster, MySQL, MariaDB

Percona Toolkit

• pt-query-digest
• pt-agent

Percona Cloud
Components
Slow log file

# Time: 130601 8:01:06.058915
# User@Host: root[root] @ localhost [] Id: 42
# Schema: imdb Last_errno: 0 Killed: 0
# Query_time: 7.725616 Lock_time: 0.000328 Rows_sent: 4 Rows_examined: 1543720 Rows_affected: 0
# Bytes_sent: 272 Tmp_tables: 0 Tmp_disk_tables: 0 Tmp_table_sizes: 0
# QC_Hit: No Full_scan: Yes Full_join: No Tmp_table: No Tmp_table_on_disk: No
# Filesort: No Filesort_on_disk: No Merge_passes: 0
# InnoDB_IO_r_ops: 6415 InnoDB_IO_r_bytes: 105103360 InnoDB_IO_r_wait: 0.001279
# InnoDB_rec_lock_wait: 0.000000 InnoDB_queue_wait: 0.000000 # InnoDB_pages_distinct: 6430
SET timestamp=1370073666;
SELECT id,title,production_year FROM title WHERE title = 'Bambi';
What is transferred to Percona Cloud?

• Metrics

# Time: 130601 8:01:06.058915
# User@Host: root[root] @ localhost [] Id: 42
# Schema: imdb Last_errno: 0 Killed: 0
# Query_time: 7.725616 Lock_time: 0.000328 Rows_sent: 4 Rows_examined: 1543720
# Bytes_sent: 272 Tmp_tables: 0 Tmp_disk_tables: 0 Tmp_table_sizes: 0

• Query Fingerprint

SELECT id,title,production_year FROM title WHERE title = ?;

• Query Example (can be disabled)

INSERT INTO suspicious_users (username, email)
VALUES ("Tom Basil", "me@tombasil.com");
Installation

Account https://cloud.percona.com

Percona Toolkit on MySQL server

• http://www.percona.com/software/percona-toolkit
• http://www.percona.com/doc/percona-toolkit/2.2/installation.html

pt-agent initialization
Pt-agent API key

- [https://cloud.percona.com/api-key](https://cloud.percona.com/api-key)
Pt-agent install

- `pt-agent --install --user={mysql username} --password={password} --api-key={API Key copied from web site}`

Step 1 of 11: Verify the user is root: OK
Step 2 of 11: Check Perl module dependencies: OK
Step 3 of 11: Check for crontab: OK
Step 4 of 11: Verify the API key: OK
Step 5 of 11: Connect to MySQL: OK
Step 6 of 11: Check if MySQL is a slave: NO
Step 7 of 11: Create a MySQL user for the agent: OK
Step 8 of 11: Initialize `/etc/percona/agent/my.cnf`: OK
Step 10 of 11: Create the agent: OK
pt-agent has daemonized and is running as PID 13506:
  --lib /var/lib/pt-agent
  --log /var/log/pt-agent.log
  --pid /var/run/pt-agent.pid

These values can change if a different configuration is received.

OK

INSTALLATION COMPLETE
The agent has been installed and started, but it is not running any services yet.
Go to https://cloud.percona.com/agents#node1 to enable services for the agent.
Pt-agent enable Query Analytics

- https://cloud.percona.com/agents
Pt-agent config Query Analytics

Agent: node1

Query Analytics: On

This service will be enabled once the configuration is saved.

How often to report: Every 1 Minutes
Long query time: 0.000001
Max Slow Query Log Size: 16M
Remove old slow logs

Send your application's actual queries

Percona Server
Slow log verbosity: Full
Log slow admin statements
Log slow slave statements
Log slow rate limit: 1

Save
Pt-agent check status

- Status Log: Agents -> select agent -> Status Log
Pt-agent reports data

![Percona Cloud Tools](image)

**All servers (1)**

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last report</td>
<td>2 minutes ago</td>
</tr>
<tr>
<td>Reported QPS</td>
<td>15.00</td>
</tr>
<tr>
<td>Total time</td>
<td>4.76ms</td>
</tr>
<tr>
<td>Load</td>
<td>0.00</td>
</tr>
<tr>
<td>Response 95%</td>
<td>1.02ms</td>
</tr>
</tbody>
</table>
Pt-agent installation

• [http://www.mysqlperformanceblog.com/2014/02/03/quick-installation-guide-for-percona-cloud-tools-for-mysql/](http://www.mysqlperformanceblog.com/2014/02/03/quick-installation-guide-for-percona-cloud-tools-for-mysql/)

• Ask for help – hit “Support” button

Using Percona Cloud Tools

Performance Optimization  Troubleshooting  Query review  Capacity planning and trending
Using Percona Cloud Tools

1. Performance Optimization

• Find “bad” queries
Profile

Analyze MySQL Query Performance with Percona Cloud Tools
Profile

Total time

- Total time = \(\text{SUM (individual queries)}\)
- Queries with the biggest Total time – most load on the server

Max time

- “ran away” queries
- Worst user experience

95% time

- User experience in most cases

Load

- % of wallclock time the server is busy with queries
Profile

Examples from live server

• Mysqlperformanceblog.com
• bm-cloud-db02-01
Using Percona Cloud Tools

2. Troubleshooting

• Why this query runs too long
Troubleshooting

Why this query was slow

• Rows sent (too many rows sent ?)
• Rows examined vs Rows sent
• Temporary table
• Filesort
• IO
• JOIN more than 3 tables

* In this webinar I do NOT show techniques how to fix queries

• Too wide topic; we have a lot of materials
## Troubleshooting - examples

### Metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Percent</th>
<th>Total</th>
<th>Average</th>
<th>Minimum</th>
<th>Median</th>
<th>95%</th>
<th>Maximum</th>
<th>Stddev</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query count</td>
<td>0</td>
<td>80.36K</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Query_time</td>
<td>11.76%</td>
<td>1h04m</td>
<td>46.84ms</td>
<td>384.00μs</td>
<td>42.04ms</td>
<td>70.54ms</td>
<td>1.76ms</td>
<td>17.26ms</td>
</tr>
<tr>
<td>Lock_time</td>
<td>0.37%</td>
<td>3.86s</td>
<td>48.04μs</td>
<td>26.00μs</td>
<td>18.01μs</td>
<td>51.83μs</td>
<td>276.00μs</td>
<td>4.80μs</td>
</tr>
<tr>
<td>Rows_sent</td>
<td>71.53%</td>
<td>457.92M</td>
<td>5.69K</td>
<td>32.00</td>
<td>5.86K</td>
<td>5.86K</td>
<td>5.95K</td>
<td>752.71</td>
</tr>
<tr>
<td>Rows_examined</td>
<td>18.08%</td>
<td>1.72G</td>
<td>21.46K</td>
<td>128.00</td>
<td>22.22K</td>
<td>22.22K</td>
<td>23.82K</td>
<td>3.30K</td>
</tr>
<tr>
<td>Rows_read</td>
<td>18.08%</td>
<td>1.72G</td>
<td>21.46K</td>
<td>128.00</td>
<td>22.22K</td>
<td>22.99K</td>
<td>23.82K</td>
<td>3.30K</td>
</tr>
<tr>
<td>Merge_passes</td>
<td>75.77%</td>
<td>74.50K</td>
<td>0.35</td>
<td>0.00</td>
<td>0.35</td>
<td>0.35</td>
<td>1.00</td>
<td>0.00</td>
</tr>
<tr>
<td>InnoDB_IO_r_bytes</td>
<td>0.00%</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>InnoDB_IO_r_ops</td>
<td>0.00%</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>InnoDB_IO_r_wait</td>
<td>0.00%</td>
<td>0.00μs</td>
<td>0.00μs</td>
<td>0.00μs</td>
<td>0.00μs</td>
<td>0.00μs</td>
<td>0.00μs</td>
<td>0.00μs</td>
</tr>
<tr>
<td>InnoDB_pages_distinct</td>
<td>0.00%</td>
<td>20.51M</td>
<td>254.68</td>
<td>23.00</td>
<td>246.00</td>
<td>246.37</td>
<td>276.00</td>
<td>1.43</td>
</tr>
<tr>
<td>InnoDB_queue_wait</td>
<td>0.00%</td>
<td>0.00μs</td>
<td>0.00μs</td>
<td>0.00μs</td>
<td>0.00μs</td>
<td>0.00μs</td>
<td>0.00μs</td>
<td>0.00μs</td>
</tr>
<tr>
<td>InnoDB_rec_lock_wait</td>
<td>0.00%</td>
<td>0.00μs</td>
<td>0.00μs</td>
<td>0.00μs</td>
<td>0.00μs</td>
<td>0.00μs</td>
<td>0.00μs</td>
<td>0.00μs</td>
</tr>
<tr>
<td>Query_length</td>
<td>0.38%</td>
<td>36.52MiB</td>
<td>476.49</td>
<td>384.00</td>
<td>463.00</td>
<td>463.17</td>
<td>494.00</td>
<td>2.43</td>
</tr>
<tr>
<td>Bytes_sent</td>
<td>2.81%</td>
<td>3.77GiB</td>
<td>49.16KiB</td>
<td>353.00</td>
<td>48.95KiB</td>
<td>49.01KiB</td>
<td>51.45KiB</td>
<td>6.07KiB</td>
</tr>
<tr>
<td>Tmp_tables</td>
<td>2.29%</td>
<td>53.08K</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Tmp_disk_tables</td>
<td>0.00%</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Tmp_table_sizes</td>
<td>63.66%</td>
<td>219.63GiB</td>
<td>2.92MiB</td>
<td>0.00</td>
<td>3.86MiB</td>
<td>4.47MiB</td>
<td>4.53MiB</td>
<td>1.96MiB</td>
</tr>
</tbody>
</table>

### Query Plan

- **Filesor**: 29142
- **Filesor on disk**: 26806
- **Full join**: 0
- **Full scan**: 8630
- **Query cache hits**: 0
- **Temporary tables**: 20512
- **Temporary tables on disk**: 0
Using Percona Cloud Tools

3. Query review

• Are there new queries in database?
Query Review

Detect new queries
• “Never trust these developers”

Collaboration tool
• But communicate with them

Query properties
• New / Reviewed / Needs Attention
• Comments
• Tags
Query Review - examples
Using Percona Cloud Tools

4. Capacity planning and trending

• Estimate how this query will run in future
Trending – server

Server Summary

<table>
<thead>
<tr>
<th>Metric</th>
<th>Total</th>
<th>Average</th>
<th>Minimum</th>
<th>Median</th>
<th>95%</th>
<th>Maximum</th>
<th>Stddev</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query count</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Query time</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Total (seconds)

Average QPS

[Graphs showing query time and total time metrics over a month]
**Trending - query**

The graph shows the trend in query performance over time. It includes metrics such as query time, total query count, average query per second (QPS), and total rows examined. The data is displayed from November to February.
Roadmap

More metrics

- MySQL status
- Whole server metrics (CPU, Memory, IO, network)

Better agent

- Support Amazon RDS
- Not Perl based

Your feedback!
Conclusions

**Percona Cloud Tools**

- GUI tools for performance tasks
- No excuses now for not doing a query review

**Register and check out reports!**

**Still in Beta**

- We have a lot of ideas and work in progress
Advanced Rates End March 2\textsuperscript{nd} at 11:30pm PST, 2014