

Percona Monitoring and Management Demonstration

Michael Coburn, Product Manager PMM
October 11th, 2017



Your presenter

- Michael Coburn - Product Manager PMM
 - *Working at Percona for almost 5 years*
 - *Consultant, Manager, TAM, now Product Manager*
 - *Canadian living in México*

Why does this talk exist?

- Troubleshooting performance issues can be a bit tricky, especially when you're given a broad statement that the database is slow.
- Learn to direct your attention to the correct moving pieces and fix what needs your attention.
- Learn how all this is done at Percona, what we monitor and track, and the tools we use.

Agenda

- What is PMM?
- Using Metrics Monitor to visualise events over time
- Using QAN to optimise MySQL queries
- What's new from our September 1.3 release

What is PMM

- Free, Open Source database troubleshooting and performance optimisation platform for MySQL and MongoDB
 - *We also support MariaDB and ProxySQL*
- Runs in **your** secure environment (not a SaaS) and on **your** equipment
- Secured with SSL encryption between pmm-client and pmm-server

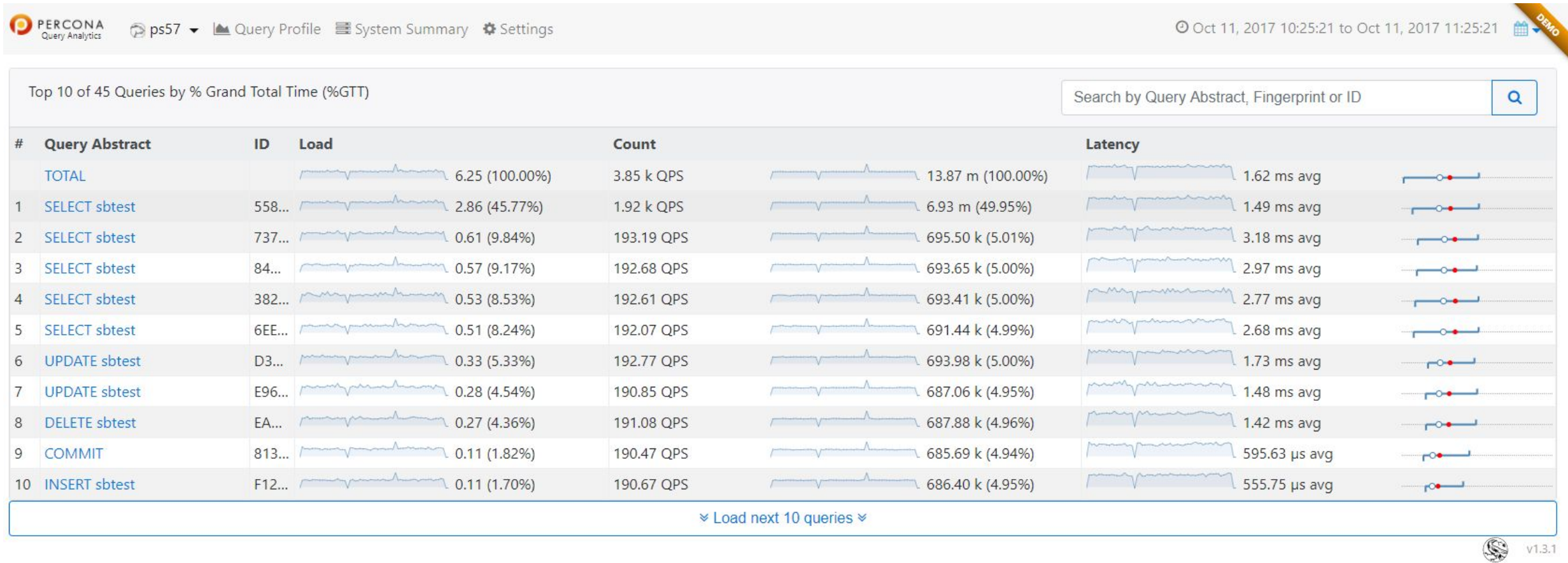
My database is slow !?!

- What is 'normal' ? Establishing a baseline
- Are other factors impacting your database performance?
 - *Hardware/virtualisation layer*
 - *Network*
 - *Application behaviour*
- But I already have `top` and `SHOW PROCESSLIST`, isn't that sufficient?

Query Analytics - QAN

MySQL query optimisation

QAN - Query Analytics - MySQL



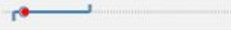
QAN overview

- Query Abstract
 - *Query pattern with placeholders*
- ID
 - *Unique fingerprint, used for query group by*
- Load
 - *Grand Total Time - percentage of time that the MySQL server spent executing a specific query*
- Count
 - *QPS, total count during window, % of total*
- Latency
 - *Min, Med, Avg, P95, Max*

PERFORMANCE_SCHEMA

SELECT sbtest

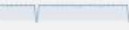
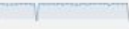
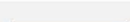
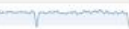
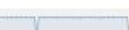
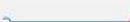
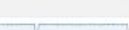
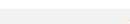
84D1DEE77FA8D4C3

Metrics			
Metrics	Rate/Sec	Sum	Per Query Stats
Query Count	1.75 (per sec) 	6.30 k 92.18% of total	
Query Time	18.82 load 	18:49:25 79.93% of total	10.76 sec avg 
Lock Time	<0.01 (avg load) 	325.80 ms 41.62% of total <0.01% of query time	52.38 µs avg 
Rows Sent	174.92 (per sec) 	629.70 k 92.21% of total	100.00 avg 
Rows Examined	1.75 m (per sec) 	6.30 b 92.14% of total 10.00 k per row sent	1.00 m avg 

Slow log - Percona Server enhanced

SELECT sbtest

558CAEF5F387E929

Metrics			
Metrics	Rate/Sec	Sum	Per Query Stats
Query Count	1.96 k (per sec) 	7.05 m 49.94% of total	
Query Time	2.92 load 	2:54:55 45.79% of total	1.49 ms avg 
Lock Time	0.27 (avg load) 	0:16:09 47.69% of total 9.23% of query time	137.39 µs avg 
Innodb IO Read Wait	2.03 (avg load) 	2:01:58 52.44% of total 70.83% of query time	1.05 ms avg 
Innodb Read Ops	386.54 (per sec) 	1.39 m 44.41% of total	0.00 avg 
Innodb Read Bytes	6.33 MB (per sec) 	22.80 GB 44.41% of total 16.38 KB avg io size	3.28 KB avg 
Innodb Distinct Pages	-	-	1.00 avg 
Query Cache Hits	30.52 (per sec) 	109.87 k 98.80% of total 1.56% QC hit ratio	-
Rows Sent	1.93 k (per sec) 	6.94 m 3.11% of total	0.00 avg 
Bytes Sent	381.91 KB (per sec) 	1.37 GB 4.89% of total 198.09 Bytes bytes/row	195.00 Bytes avg 
Rows Examined	1.93 k (per sec) 	6.94 m 1.36% of total 1.00 per row sent	0.00 avg 

Query fingerprint + EXPLAIN

QUERY

▼Fingerprint

```
SELECT c
FROM sbtest?
WHERE id=?
```

▼Example

```
SELECT c
FROM sbtest1
WHERE id=7865558
```

EXPLAIN

 innodb

EXPLAIN

▼CLASSIC

Id	SelectType	Table	Partitions	CreateTable	Type	PossibleKeys	Key	KeyLen	Ref	Rows	Extra
1	SIMPLE	sbtest1			const	PRIMARY	PRIMARY	4	const	1	

JSON EXPLAIN

JSON

```
{
  "query_block": {
    "select_id": 1,
    "cost_info": {
      "query_cost": "1.00"
    },
    "table": {
      "table_name": "sbtest1",
      "access_type": "const",
      "possible_keys": [
        "PRIMARY"
      ],
      "key": "PRIMARY",
      "used_key_parts": [
        "id"
      ],
      "key_length": "4",
      "ref": [
        "const"
      ],
      "rows_examined_per_scan": 1,
      "rows_produced_per_join": 1,
      "filtered": "100.00",
      "cost_info": {
        "read_cost": "0.00",
        "eval_cost": "0.20",
        "prefix_cost": "0.00",
        "data_read_per_join": "192"
      },
      "used_columns": [
        "id",
        "c"
      ]
    }
  }
}
```

CREATE TABLE

▼CREATE

```
CREATE TABLE `sbtest1` (  
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `k` int(10) unsigned NOT NULL DEFAULT '0',  
  `c` char(120) NOT NULL DEFAULT '',  
  `pad` char(60) NOT NULL DEFAULT '',  
  PRIMARY KEY (`id`),  
  KEY `k_1` (`k`)  
) ENGINE=InnoDB AUTO_INCREMENT=100000001 DEFAULT CHARSET=latin1 MAX_ROWS=1000000
```

TABLE STATUS

▼STATUS	
Name	Value
Name	sbtest1
Engine	InnoDB
Version	10
RowFormat	Dynamic
Rows	98.63 m
AvgRowLength	227.00 Bytes
DataLength	22.46 GB
MaxDataLength	0.00 Bytes
IndexLength	2.37 GB
DataFree	14.68 MB
AutoIncrement	100000001
CreateTime	2016-07-29T19:33:39Z
UpdateTime	2017-10-11T16:38:05Z
CheckTime	0001-01-01T00:00:00Z
Collation	latin1_swedish_ci
Checksum	-
CreateOptions	max_rows=1000000
Comment	

INDEXES

▼INDEXES

KeyName	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id	98630200	A	No	
k_1	BTREE	No	No	k	30879286	A	No	

QAN - Query Analytics - MongoDB

Top 10 of 113 Queries by % Grand Total Time (%GTT)

Search by Query Abstract, Fingerprint or ID



#	Query Abstract	ID	Load	Count	Latency
	TOTAL		31.86 (100.00%)	90.87 QPS	327.12 k (100.00%)
1	FIND sbtest11_id,c	e1c...	0.77 (2.42%)	0.24 QPS	858.00 (0.26%)
2	FIND sbtest4_id,c	350...	0.76 (2.39%)	0.24 QPS	851.00 (0.26%)
3	FIND sbtest13_id,c	7be...	0.75 (2.37%)	0.23 QPS	842.00 (0.26%)
4	FIND sbtest10_id,c	985...	0.75 (2.34%)	0.23 QPS	824.00 (0.25%)
5	FIND sbtest8_id,c	f3b...	0.75 (2.34%)	0.23 QPS	825.00 (0.25%)
6	FIND sbtest5_id,c	e28...	0.74 (2.34%)	0.23 QPS	827.00 (0.25%)
7	FIND sbtest7_id,c	56a...	0.73 (2.30%)	0.23 QPS	813.00 (0.25%)
8	FIND sbtest1_id,c	9b6...	0.73 (2.28%)	0.22 QPS	794.00 (0.24%)
9	FIND sbtest16_id,c	b30...	0.73 (2.28%)	0.22 QPS	805.00 (0.25%)
10	FIND sbtest12_id,c	673...	0.72 (2.27%)	0.22 QPS	794.00 (0.24%)

Load next 10 queries



v1.3.1

Query fingerprint + Example

QUERY

▼Fingerprint

FIND sbtest3 _id,c

▼Example

```
{
  "ns": "sbtest.sbtest3",
  "op": "query",
  "query": {
    "find": "sbtest3",
    "filter": {
      "_id": {
        "$gte": 743733,
        "$lte": 743832
      }
    },
    "sort": {
      "c": 1
    }
  }
}
```

JSON EXPLAIN

▼JSON

```
{
  "queryPlanner": {
    "plannerVersion": 1,
    "namespace": "sbtest.sbtest3",
    "indexFilterSet": false,
    "parsedQuery": {
      "$and": [
        {
          "_id": {
            "$lte": 743832
          }
        },
        {
          "_id": {
            "$gte": 743733
          }
        }
      ]
    },
    "winningPlan": {
      "stage": "SORT",
      "sortPattern": {
        "c": 1
      },
      "inputStage": {
        "stage": "SORT_KEY_GENERATOR",
        "inputStage": {
          "stage": "FETCH",

```

Server summary info

- Click the icon: 
- pt-summary
 - *Instance-specific information*
- pt-mysql-summary
 - *Facts about the running MySQL instance*

Metrics Monitor

Prometheus + Grafana + Beautiful dashboards from Percona!

What is Grafana

- Open Source data visualisation tool
- Popular datasources
 - *Prometheus*
 - *CloudWatch*
 - *Graphite*
 - *Elasticsearch*
- Templated Variables
 - *Define your graph metrics, and let the hosts get filled in automatically*
 - GREAT for large, dynamic environments where hosts are considered ephemeral

How can I...

- At a glance MySQL + Storage Engine indepth
 - *MySQL Overview, InnoDB, MyRocks*
- MongoDB high level + ReplSet views
 - *MongoDB Overview, MongoDB ReplSet*
- Compare servers' to each other
 - *Cross Server graphs*
- Show behaviour now() compared to past period
 - *Trends Overview dashboard*

How can I... - Percona Server only

- Table statistics
 - *Largest tables by rows and size, total DB size, tables by rows read and changed, auto_increment usage (about to hit the limit?)*
- User statistics
 - *Top users by connection count, network usage, rows read/changed*
- Query Response Time
 - *Average, read vs write, distribution*

Distribution methods

- Docker
 - *Most common deployment method for PMM today*
- Virtual appliance
 - *OVF/OVA file format, compatible with VMware, VirtualBox, Microsoft, Red Hat*
- AWS AMI
 - *Clone the image from any of the 14 regions*
 - *Deployment moving to AWS Marketplace*

Talk to Percona Experts at AWS Re:Invent!

Database Performance for Cloud Deployments

- Percona Support and Managed Services
 - *Amazon RDS, Aurora, Roll Your Own*
 - *MySQL/MariaDB/MongoDB*
 - *Reduce costs and optimize performance*
- Percona Monitoring and Management Demos
 - *Point-in-time visibility and historical trending of database performance*
 - *Detailed query analytics*
- Booth #1138





Database Performance Matters