Innodb Performance Optimization

Most important practices

Peter Zaitsev
CEO
Percona Technical Webinars
December 20th, 2017

PERCONA

About this Presentation

Innodb Architecture and Performance Optimization

3h In-depth tutorial delivered at Percona Live Conferences

This is the most important highlights + some new material

Performance Optimizations

What are the Goals of "Performance Optimization" ?

Goals of Performance Optimization

Make Queries Run Faster

- Performance

Improve Efficiency

- And Save Costs

Achieve Scalability

- Run “Larger” applications on the same system

This Presentation Focus

Getting most of the Single Node

Not touching Architecture Optimizations

Optimization Areas

- Hardware/Infrastructure
- Operating Systems
- MySQL Version
- MySQL Settings
- Queries
- Schema

Hardware and Infrastructure

Hardware/Infrastructure Optimizations

CPU

Memory

Storage

Network

General and Amazon Cloud

Most Suggestions apply to Cloud and Non-Cloud Environments

Specific Recommendations for AWS Cloud

CPU for Innodb

Faster CPUs are Better

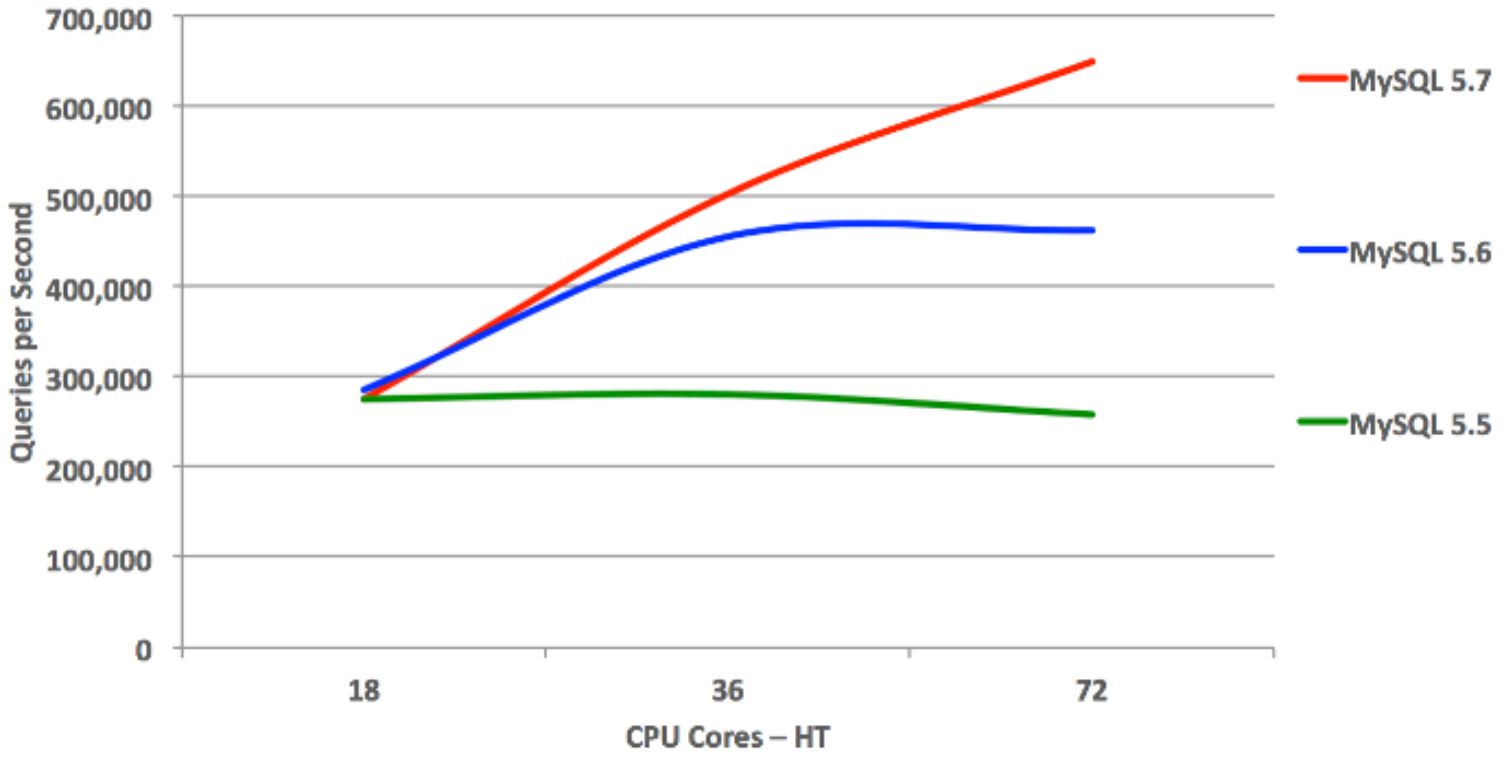
Mostly Single Threaded handling of a single Query

Multi-Core for Highly Concurrent Workload

MySQL now scales to 64+ Cores

MySQL/Innodb Multi Core Scalability

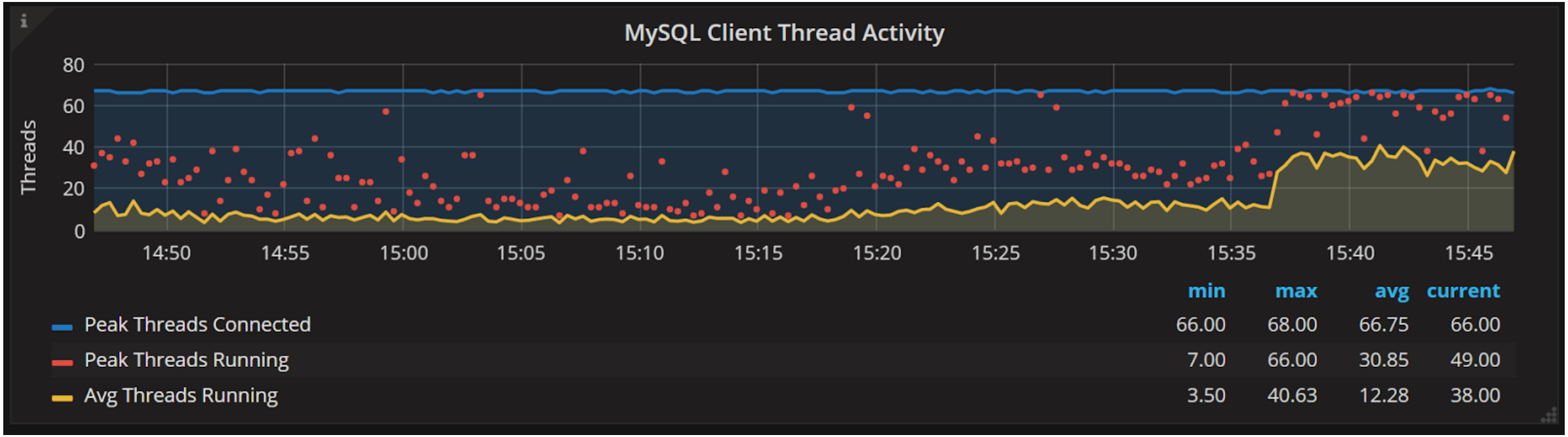
https://www.mysql.com/why-mysql/benchmarks/



PERCONA

How much Concurrency do you Have ?

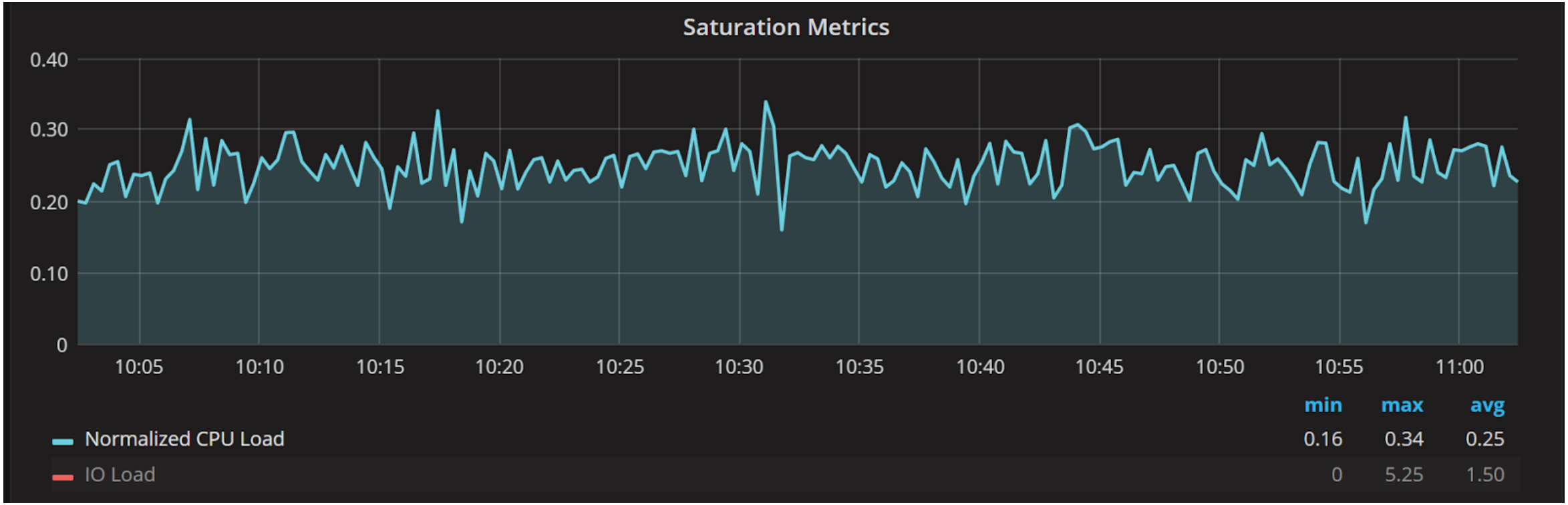
Status Variable ***threads_running*** is a great indicator



PERCONA

OK but what about CPU ?

Number of “runnable” processes in VMSTAT good proxy



PERCONA

CPU on AWS

M5 Instance Type	• General Purpose
C5 Instance Type	• Faster CPU but not as much memory
R4 Instance Type	• More Memory but less CPU
X1/X1e Instance Type	• Even More Memory per CPU
T2 Instance Type	• Burstable CPU for light workloads

Memory

Memory is mostly used for caching

Try to fit “Working Set” in memory

Caching Improves Reads and Writes Performance

Memory and Storage Capacity Planning

Think about Misses (Storage IO)	• Hit ratio is useless
Think about Reads and Writes separately	• Different Performance Profile
Think Throughput	• Do not push more IO than device can handle
Think Latency	• How many IO query can do and still be fast ?

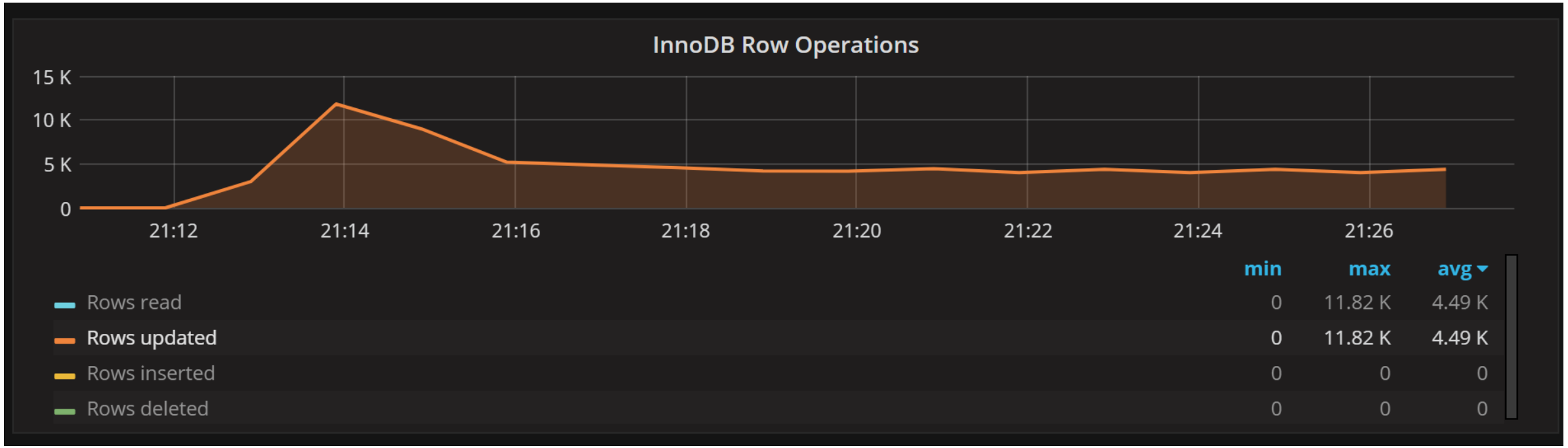
Writes

Delaying Writes Can’t go on forever

Behind Many Writes there is a Read

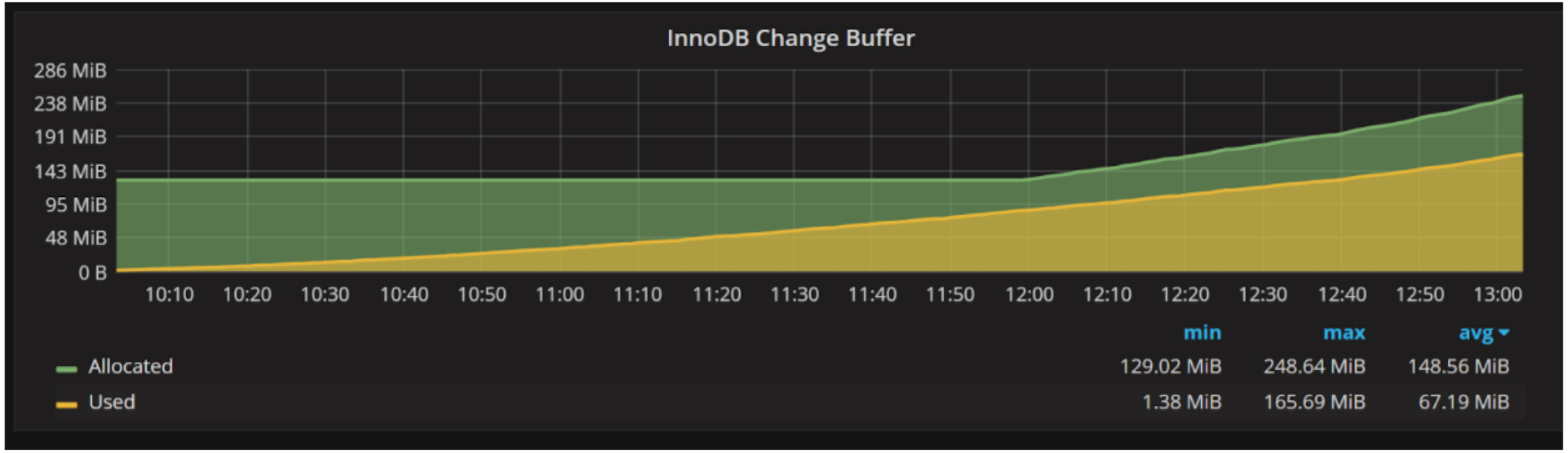
Flushing Impacts Performance

Faster Performance at First, Until Flushing needs to happen



PERCONA

Innodb Change Buffer – Long Warmup



PERCONA

Storage

Even Fastest Storage is a lot slower than Memory

Storage IO has to be done in 16K pages (by default)

Memory accesses can read/write few bytes

Type of Storage

Use SSD

Whenever Directly Attached or not

Local NVMe and Intel Optane are fastest

Storage

IO latency is critical

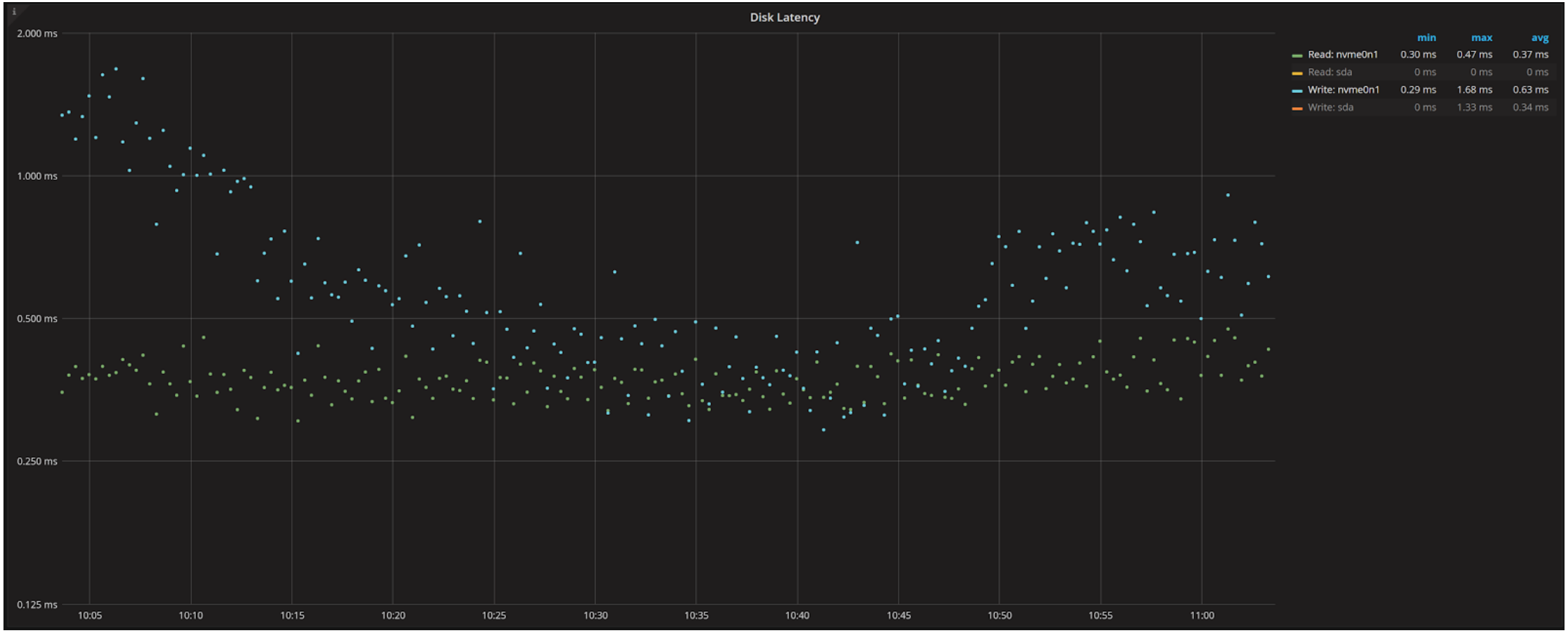
What is optimal concurrency (queue depth)

Reads vs Writes

Writes and Flushes

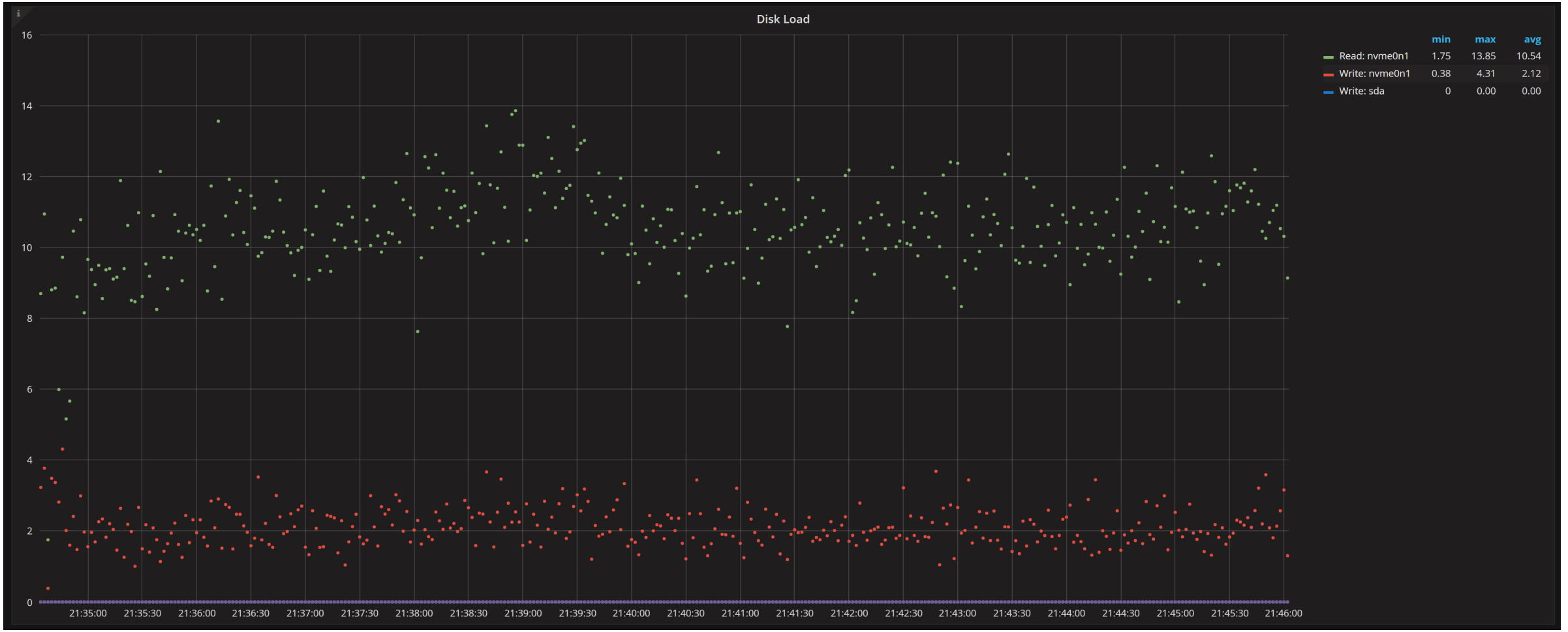
Write Cache, but Durable Write Cache

Read and Write Latency



PERCONA

Understanding IO Concurrency



PERCONA

AWS Storage Chocies

EBS io1 (Provisioned IOPS)

- Up to 500MB/sec and 32.000 IOPS per volume
- Scale IO capacity independently from the size
- Expensive

EBS gp2

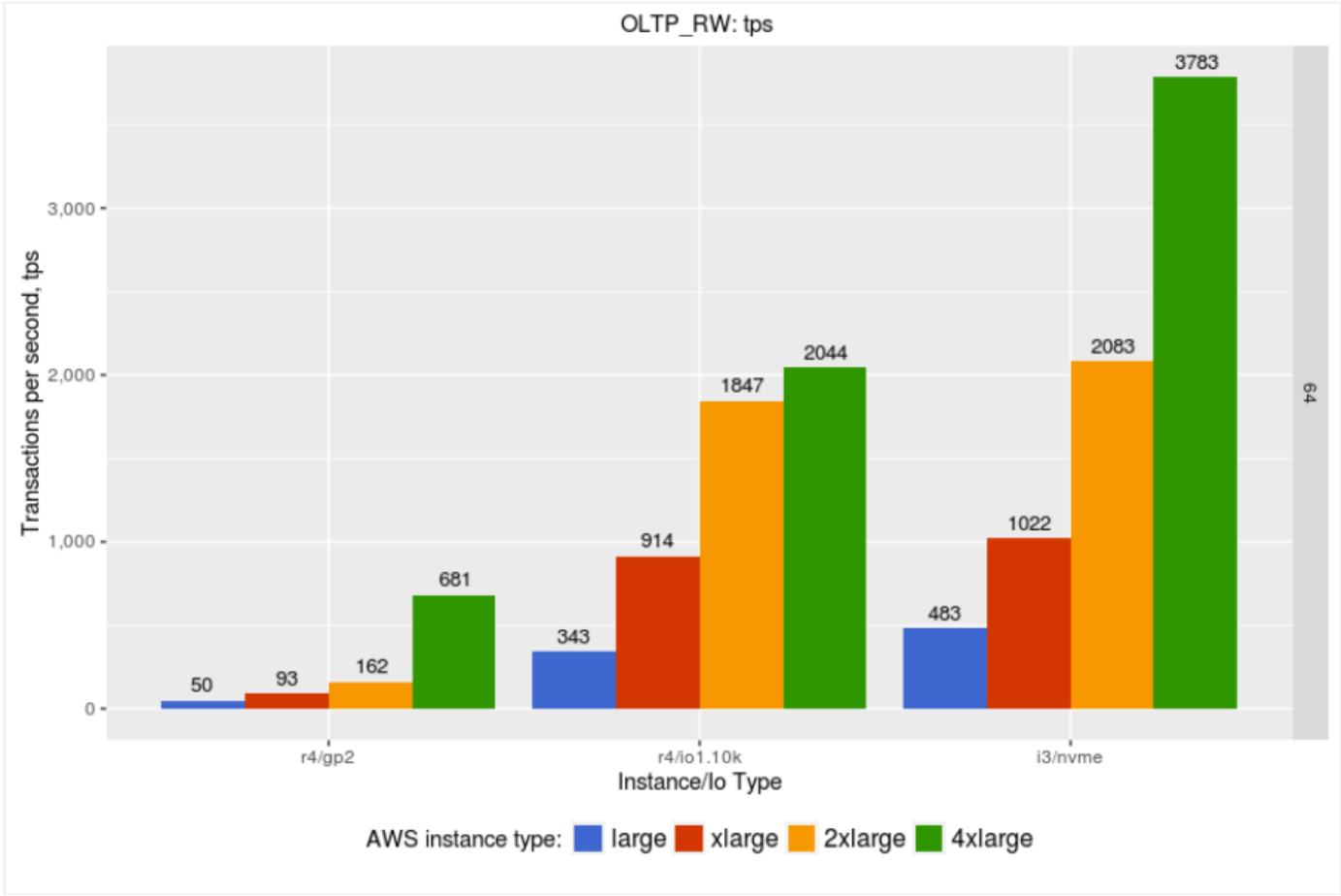
- Burstable for small volumes (less then 1TB)
- Performance scales with size – 3 IOPS per GB

Local NVMe (i3 instance type)

- Non Redundant
- Performance and Space Depends on Instance Size
- Up to 3.3M IOPS and 16GB/sec

PXC Best Practices on AWS

Extensive Benchmarks done http://bit.ly/2zeRpSZ



PERCONA

Network

"Front End" Network

- To the MySQL Client (Application)

"Back End"

- For Network Attached Storage

Network Performance

Latency

- The Most Critical

Bandwith

- Can be bottleneck for Large Data Transfer

Variance

- Network performance is not constant

Network in AWS

Single Availability Zone (AZ)	Single Region, Different AZ	Multiple Regions
• Lowest Latency (Fastest)	• Medium Latency	• Highest Latency (Slowest)

Hardware Related Innodb Settings

- **Innodb_buffer_pool_size**
  - How much memory to allocate for Caching
- **Innodb_buffer_pool_instances**
  - Increase up to 16 if you have large number of CPU cores
- **Innodb_flush_method=O_DIRECT**
  - Bypass OS cache. Good choice for most IO subsystems
- **Innodb_flush_log_at_trx_commit**
  - 1 – durable. 0 – not durable. 2 – in between
- **sync_binlog**
  - 1 – binlog durability but high performance cost
- **innodb_io_capacity**
  - Tell Innodb how much IO storage system can handle

Operating System

Linux

Modern Version

Cloud Optimized if Running in the Cloud

RHEL compatible and Debian/Ubuntu are most common

Operating System Settings

- **Number of Open Files for MySQL Process**
  - If using large number of connections or large number of tables
- **Filesystem**
  - XFS or EXT4
  - Mount –o noatime
- **Disk Scheduler**
  - Deadline or Noop
  - *cat /sys/block/sda/queue/scheduler*
  - NVMe Storage Does not have Scheduler

Operating System Settings

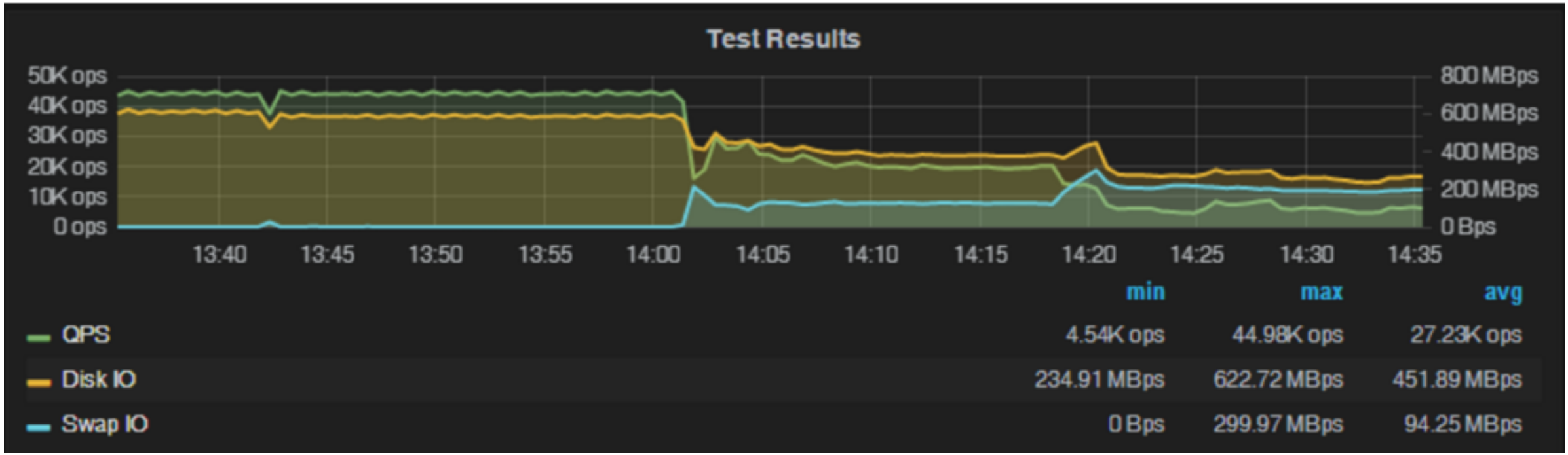
- **NUMA**
  - Only on NUMA (Multi-Socket Hardware)
  - MySQL 5.7 Option **innodb_numa_interleave=1**
  - Percona Server 5.5+ **numa_interleave=1**
- **CPU Governor**
  - Saving Power can cost you performance
  - **cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor**
  - **cpufreq-set -r -g performance**

Swapping

- **Should you Enable Swap ?**
  - Would you prefer MySQL to be killed (crash) due to lack of Memory ?
  - Would you prefer MySQL to slow down with lack of Memory ?
  - Would you want unused stuff to be swapped out from Memory ?
- **If using swap**
  - Set **vm.swappiness=1**
  - **echo 'vm.swappiness = 1' >> /etc/sysctl.conf**

Swap on SSD is bad but not that Bad

http://bit.ly/2BJBTAJ



PERCONA

MySQL Versions

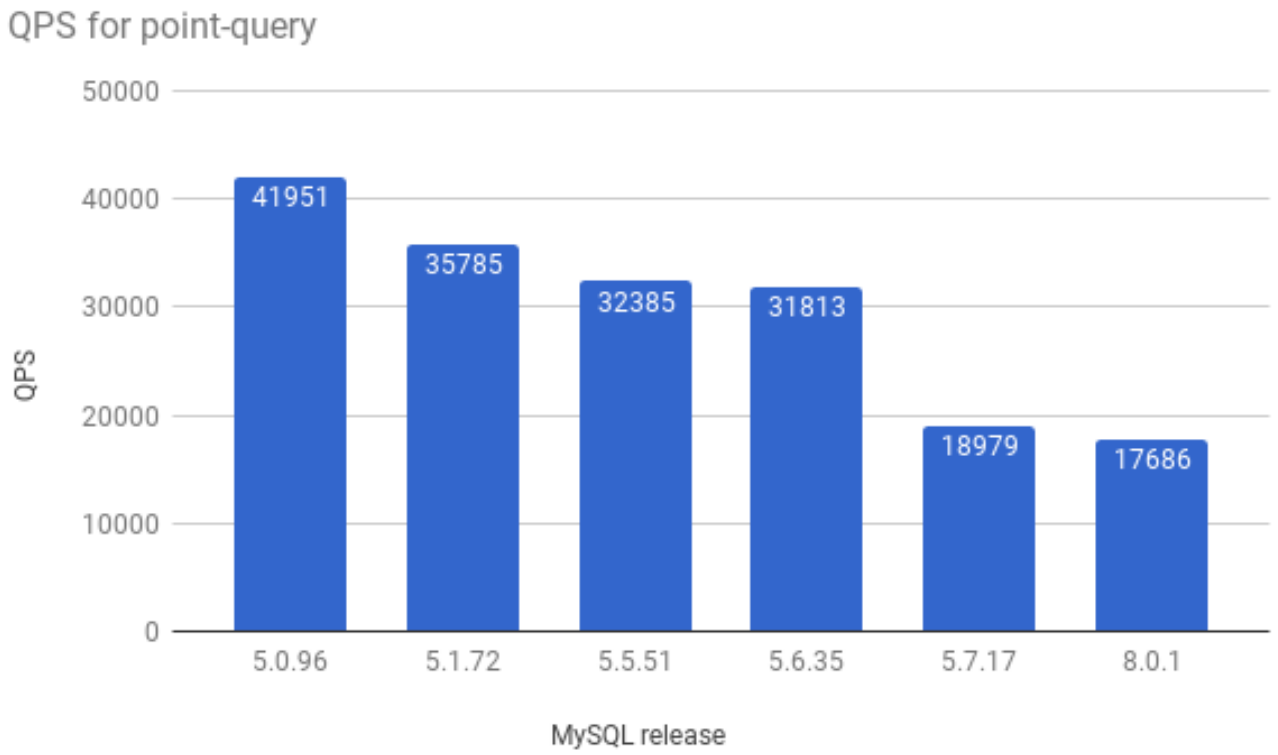
MySQL Version Best Practices

Use Modern Version (MySQL 5.7)

Consider Percona Server

Beware Single Thread Performance Regressions

http://bit.ly/2oMvu2a



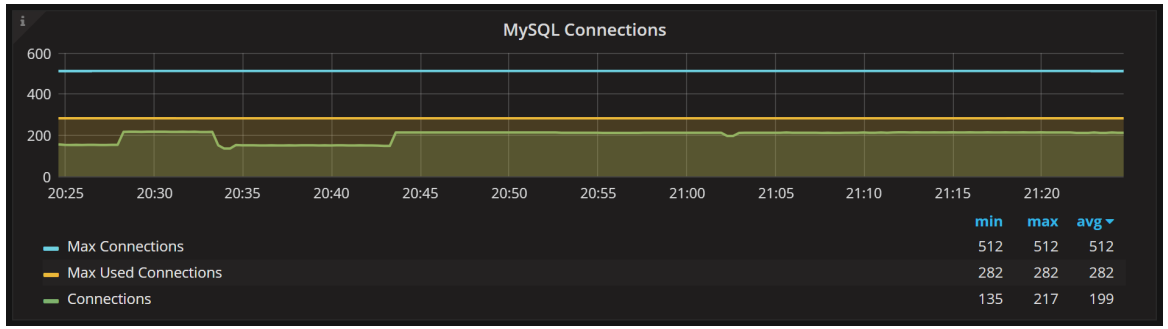
PERCONA

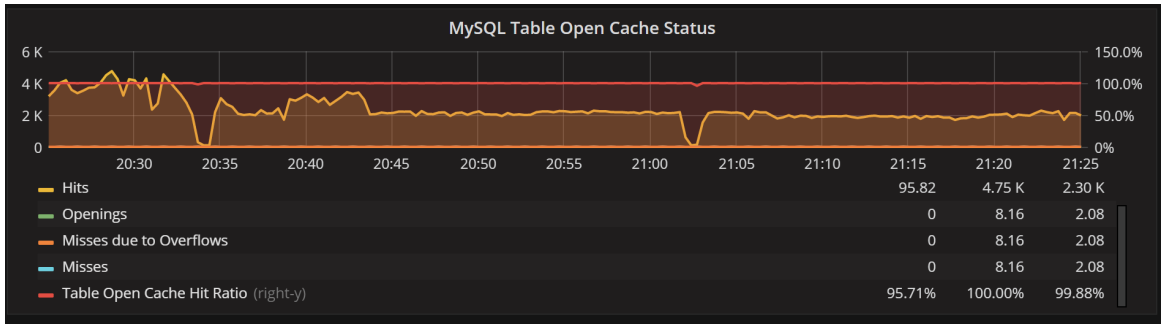
MySQL Configuration

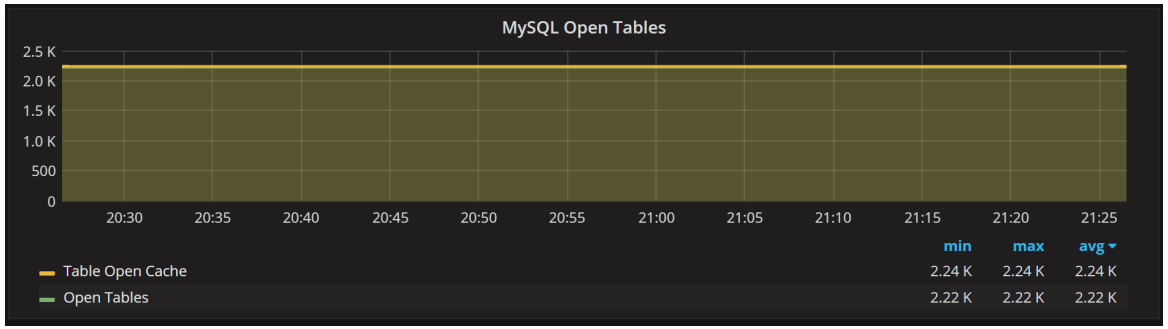
Settings to Think About

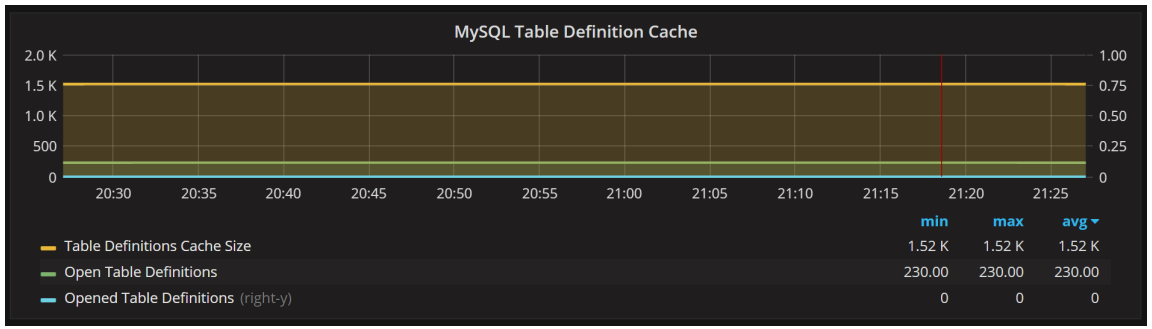
- **How Many Connections do you need ?**
  - *max_connections=N*
- **How Many Tables do you have ?**
  - *table_open_cache=X*
  - *table_definition_cache=Y*
- **What Character Set do you use ?**
  - *character-set-server=utf8*
  - *collation-server=utf8_general_ci*
- **How Long to Keep Binary Log**
  - *log_bin*
  - *expire_log_days=N*

Connection and Table Status







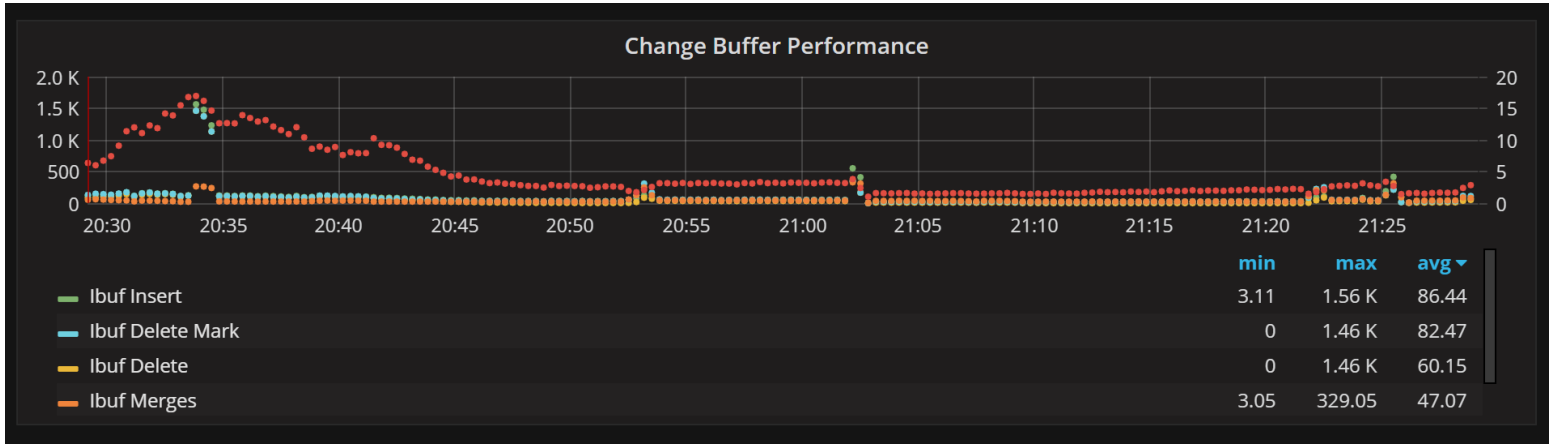


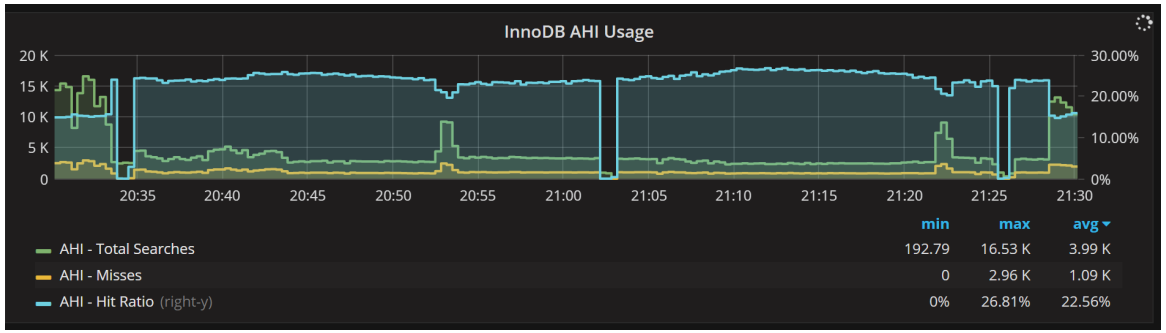
PERCONA

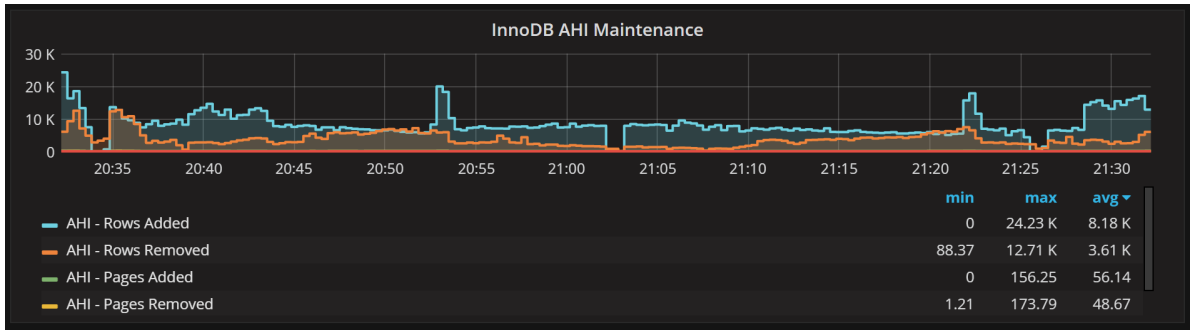
Settings for Innodb

- **Is Adapting Index Helping you or Hurting you ?**
  - innodb_adaptive_hash_index=1/0
  - innodb_adaptive_hash_index_parts=N
- **How much space to allow for Change Buffer**
  - innodb_change_buffer_max_size=5
- **How much of Buffer Pool to Reload ?**
  - innodb_buffer_pool_dump_pct=80
- **Double Write Can be disabled on ZFS**
  - innodb_doublewrite
- **Disable Innodb Flush Neighbors if running on SSD**
  - Innodb_flush_neighbors=0

Change Buffer and Adaptive Hash Index





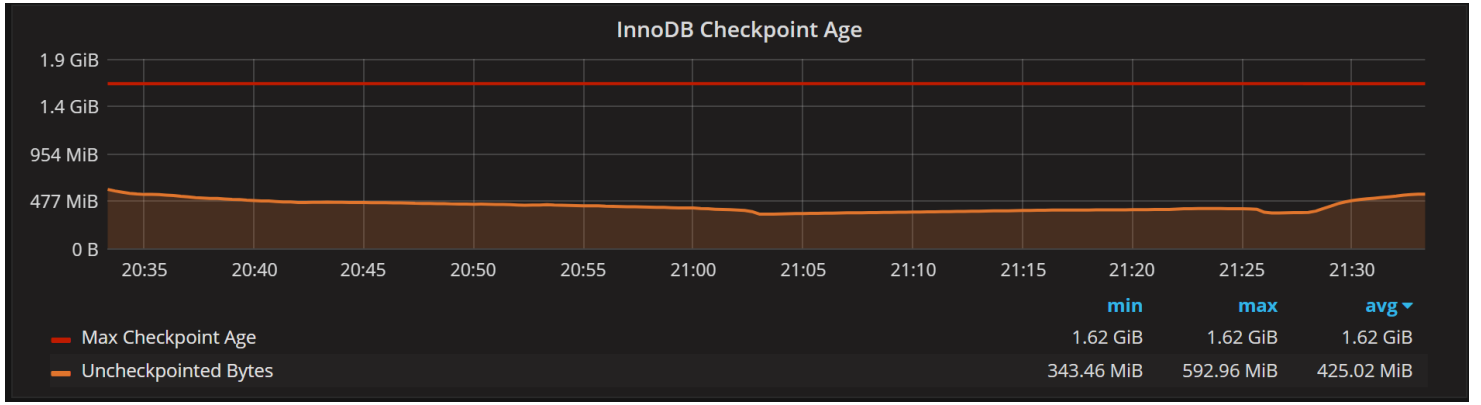


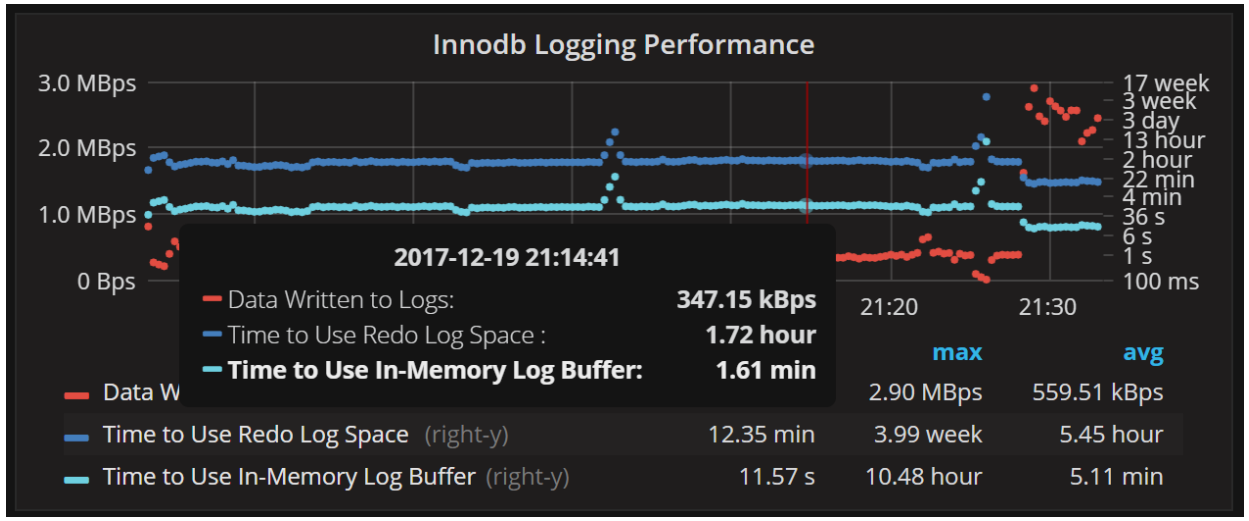
PERCONA

Settings for Innodb

- **How Much IO to Give Innodb**
  - Innodb_io_capacity
  - Innodb_io_capacity_max
- **How quickly to give up waiting for lock and return error ?**
  - innodb_lock_wait_timeout
- **Log Buffer For caching Writes**
  - innodb_log_buffer_size=32M
- **Tradeoff of Write Performance and Recover Time**
  - Innodb_log_file_size=1G
- **Protection from Purge Lag**
  - innodb_max_purge_lag=10000000
  - innodb_max_purge_lag_delay=10000

Log file and Log Buffer





PERCONA

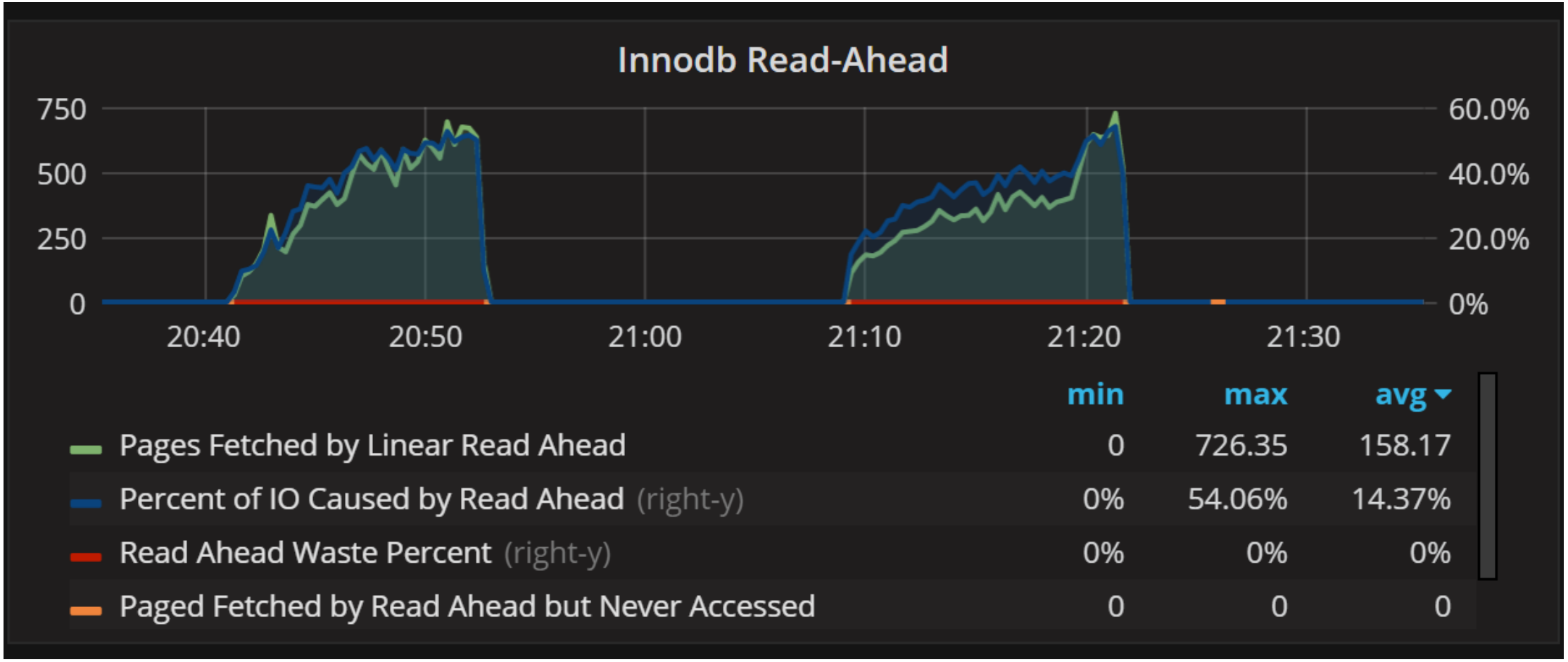
Settings for Innodb

- **Innodb Buffer Pool Scan Resistance**
  - innodb_old_blocks_pct=70
  - Innodb_old_blocks_time=5000
- **How many files Innodb can keep open**
  - innodb_open_files=N
- **Smaller Page sizes better for some workloads**
  - innodb_page_size=4096
- **On lock time out roll back transaction or statement ?**
  - innodb_rollback_on_timeout
- **Speed up Innodb Index Builds if you have memory**
  - innodb_sort_buffer_size

Settings for Innodb

- **Longer Spins can reduce Context Switching**
  - *innodb_spin_wait_delay*
- **How Innodb Index Statistics works**
  - *innodb_stats_persistent*
  - *innodb_stats_persistent_sample_pages*
  - *innodb_stats_transient_sample_pages*
- **Excessive Contention Protection**
  - *innodb_thread_concurrency*
  - Or use Thread Pool
- **Innodb Random Read Ahead can be helpful**
  - *innodb_random_read_ahead*
- **Innodb IO Configuration**
  - Ensure *innodb_use_native_aio* is on
  - innodb_read_io_threads
  - innodb_write_io_threads

Is Read-Ahead Helping or Hurting



PERCONA

Queries and Schema

Data is Clustered by Primary Key

Use Primary Keys

Use Short Primary Keys

Use Sequential Primary Keys (ie auto_increment)

Primary Key Lookups are faster (especially ranges)

Secondary Keys Refer to row by PK

Secondary Key Lookups are slower

Long primary keys = bloated secondary keys

Primary key essentially appended to primary keys

KEY(A) mean KEY(A,ID) for Queries

Multi Version Concurrency Control (MVCC)

- Readers do not block writers
- Writers do not block readers
- Readers read “stale” data
- Locking Reads are available when current state is needed
- Old versions of rows are kept in separate Undo Space
- Old index entries are kept in the indexes
- Purge Threads are constantly cleaning old garbage

Optimizing for MVCC

Avoid Long running Read Transactions concurrent with Heavy Writes

Avoid Transactions which modify large amount of rows

Avoid very Hot Rows as they can generate long Undo Chains

Repeatable-Read vs Read-Committed

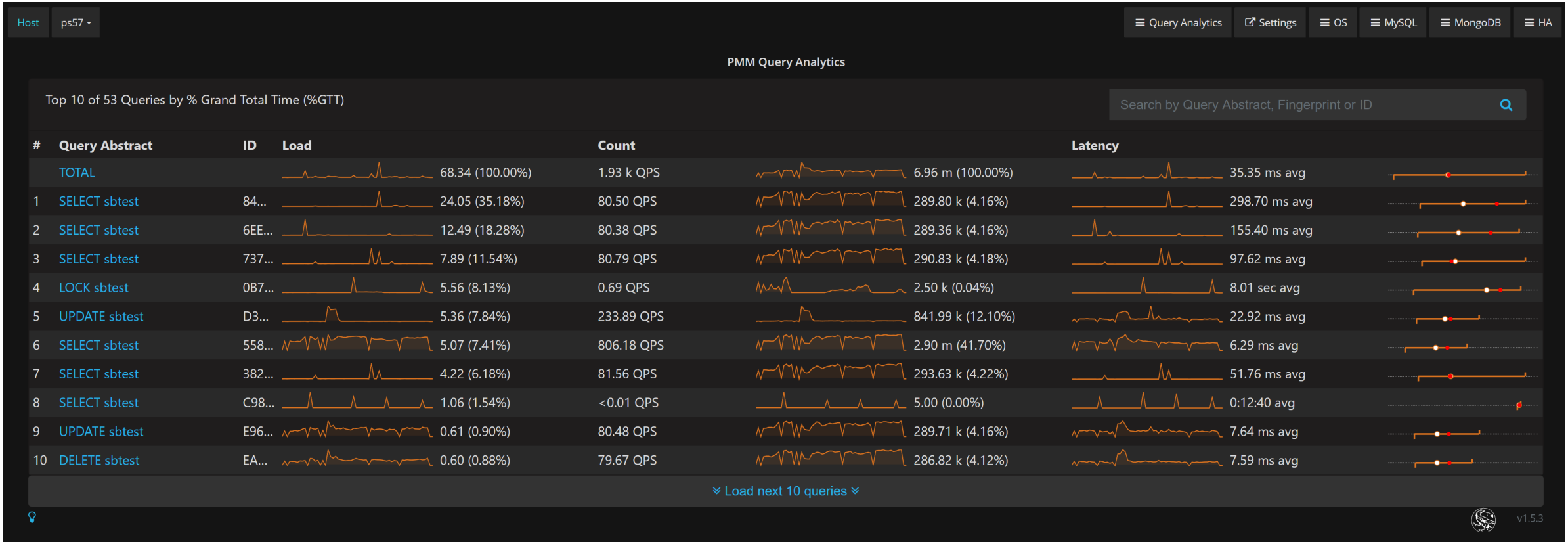
REPEATABLE-READ

- Default Isolation Mode
- View the data at the point of first select in transaction
- Can additionally delay Purge Process

READ-COMMITTED

- View data at the point of start of the statement
- Can help to reduce unpurged bloat
- Can increase “transaction allocation overhead” at high concurrency

Understanding Top Queries



PERCONA

Understanding what makes them slow

SELECT sbtest C98EE1E7A86B107A

Metrics — Query first seen: Aug 2, 2017 4:57 PM ··· Last seen: Today at 10:06 PM

Metrics	Rate/Sec	Sum	Per Query Stats
Query Count	<0.01 (per sec)	5.00 <0.01% of total	
Query Time	1.06 load	1:03:18 1.54% of total	0:13:02 avg
Lock Time	<0.01 (avg load)	1.52 ms <0.01% of total <0.01% of query time	307.12 µs avg
Innodb IO Read Wait	0.28 (avg load)	0:16:31 5.06% of total 63.34% of query time	0:08:15 avg
Innodb Read Ops	201.52 (per sec)	725.46 k 33.76% of total	362.73 k avg
Innodb Read Bytes	3.30 MB (per sec)	11.89 GB 33.76% of total 16.38 KB avg io size	5.94 GB avg
Innodb Distinct Pages	-	-	65.77 k avg
Rows Sent	<0.01 (per sec)	5.00 <0.01% of total	1.00 avg
Bytes Sent	0.10 (per sec)	345.00 Bytes <0.01% of total 69.00 Bytes bytes/row	69.00 Bytes avg
Rows Examined	80.56 k (per sec)	290.00 m 2.16% of total 58.00 m per row sent	65.00 m avg
Full Table Scans	<0.01 (per sec)	2.00 0.02% of total 40.00% of queries	-

PERCONA

Where all the Graphs Come From ?

Percona Monitoring and Management (PMM)

http://bit.ly/GetPMM

http://pmmdemo.percona.com

SAVE THE DATE!

April 23-25, 2018
Santa Clara Convention Center

PERCONA
LIVE

CALL FOR PAPERS IS NOW OPEN!

www.perconalive.com

Database Performance Matters