SDC
STORAGE DEVELOPER CONFERENCE
SNIA SANTA CLARA, 2016

Accelerating OLTP performance with NVMe SSDs

Veronica Lagrange
Changho Choi
Vijay Balakrishnan

SAMSUNG

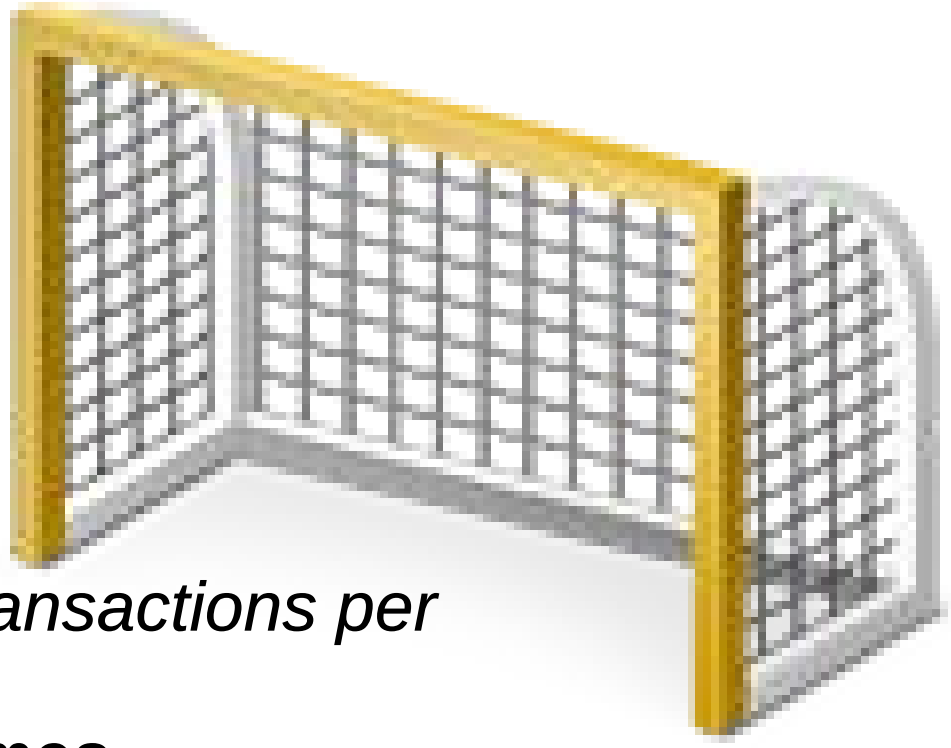
Agenda

- OLTP status quo
- Goal
- System environments
- Tuning and optimization
- MySQL Server results
- Percona Server results
- Summary

OLTP status quo

- On Line Transaction Processing is typically I/O bound
- ACID properties -> transaction must be durable
- Capacity planning: needed IOPS -> lots of storage devices and idle CPUs

Goals



- **Maximize throughput**
  - (**tpmP**= *New Order transactions per minute*)
- **Minimize Response Times**

Side line benefit: Increase Server Capacity

MySQL Server and Percona Server

- MySQL Server:
  - Open-source relational database management system (RDBMS)
  - The world's most used open-source/client-server RDBMS
  - Optimized for On Line **Transaction** Processing (OLTP)

- Percona Server:
  - A free, fully compatible, open source enhancement for MySQL Server
  - Developed and distributed by Percona
  - Especially optimized for the **I/O** subsystem

TPC-C and tpcc-mysql

TPC-C

- A 23-year old OLTP Benchmark.
- 5 types of well-defined transactions:
  1. **New Order (Read & Write)**
  2. Payment **(Read & Write)**
  3. Delivery **(Read & Write)**
  4. Order Status (Read Only)
  5. Stock Level (Read Only)
- Throughput is **New Order** Transactions per minute (tpmC)
- Relational schema

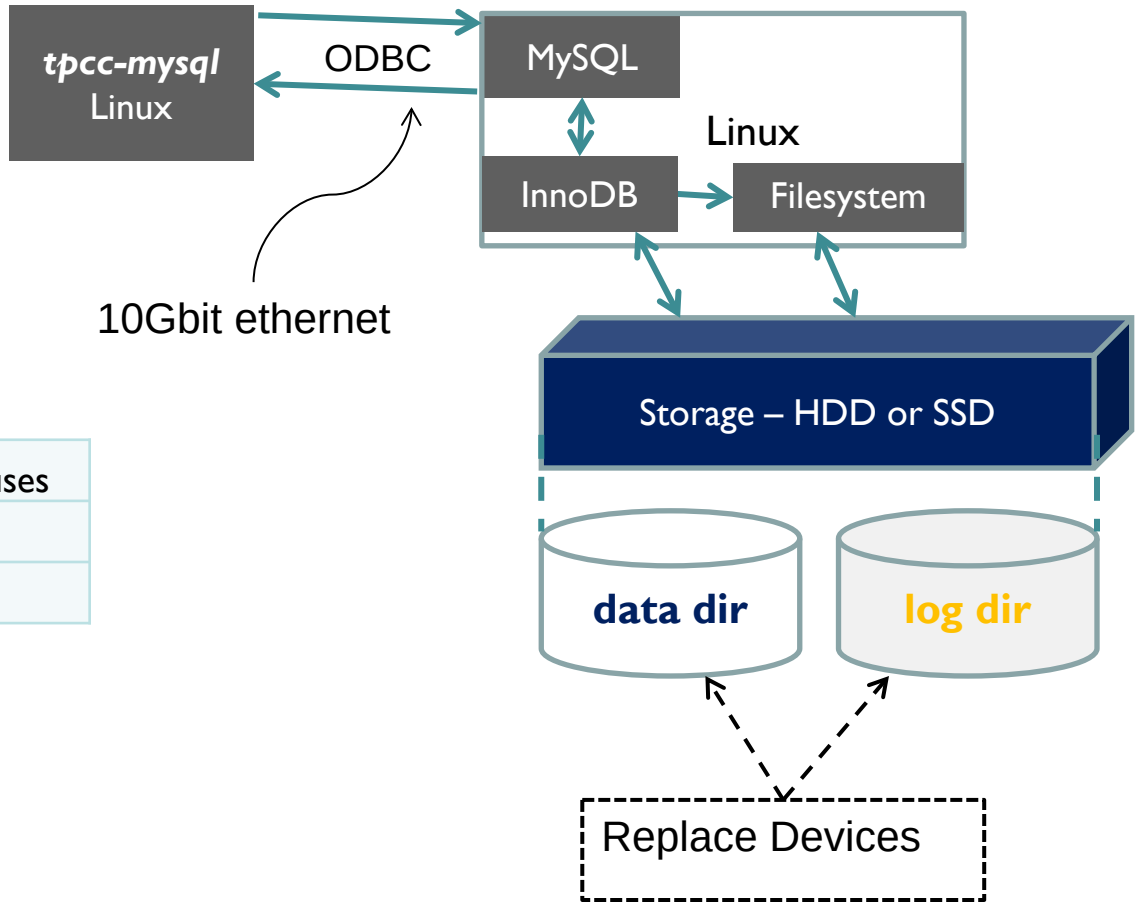
tpcc-mysql:

- The best open source implementation available
- Developed by Percona
- Not 100% compatible with the standard

Methodology

- Establish baseline configuration
- Characterize performance of system and software
- Identify key parameters for SSD NVMe
- Optimize system and software to achieve highest throughput

Performance Measurement Environment



Database size:	500 warehouses
initial size	43 GB
after 2 hour run	79 GB

Workload:
50 connections
100 connections
150 connections
200 connections
…

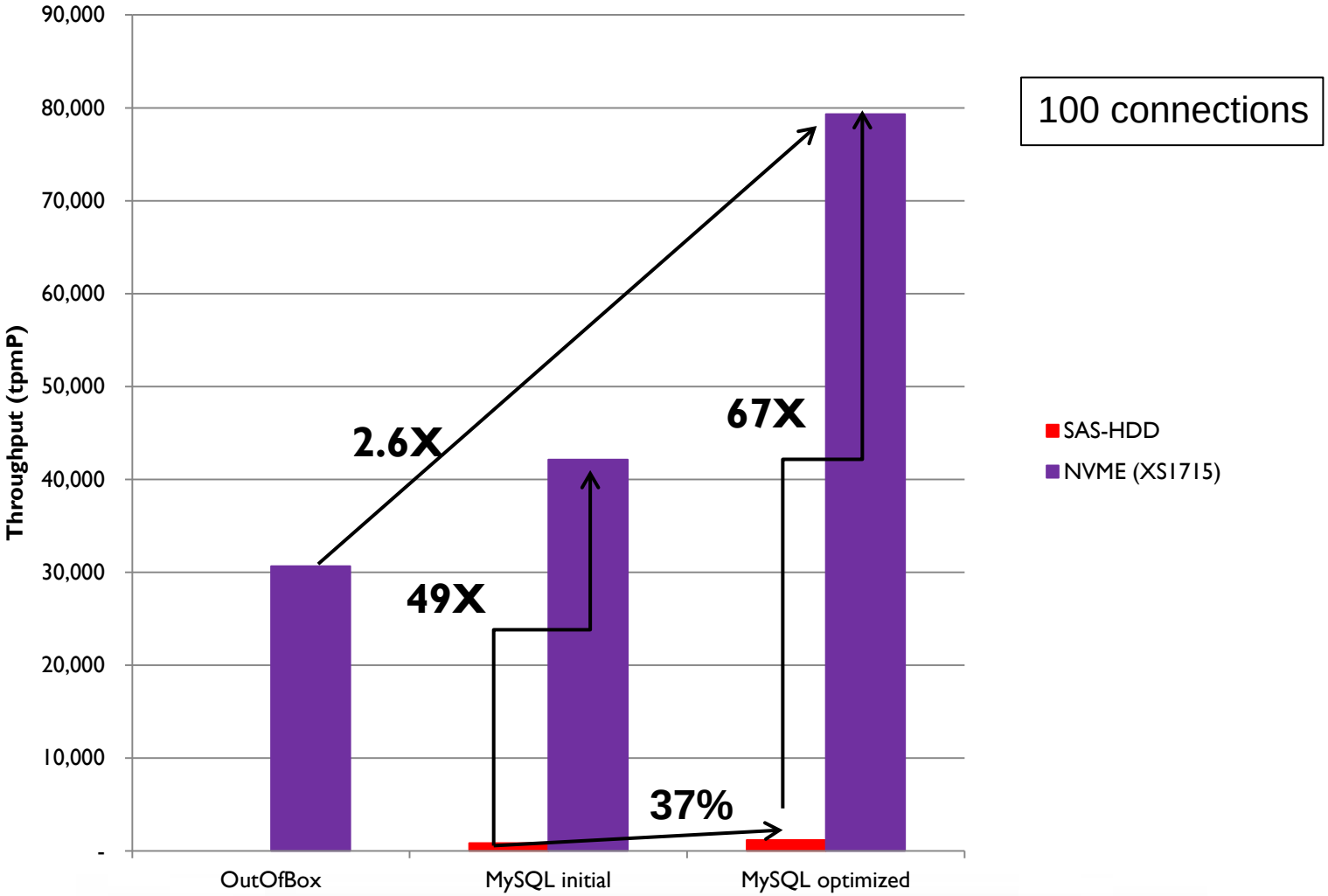
Dual Socket Server Environment

	Server
Model name	DELL 730xd Intel(R) Xeon(R) CPU E5-2670 v3 @ 2.30GHz
Memory	64GB
OS version	Linux 4.4.0-040400-generic
MySQL Server	5.7.11
Percona Server	5.7.11-4

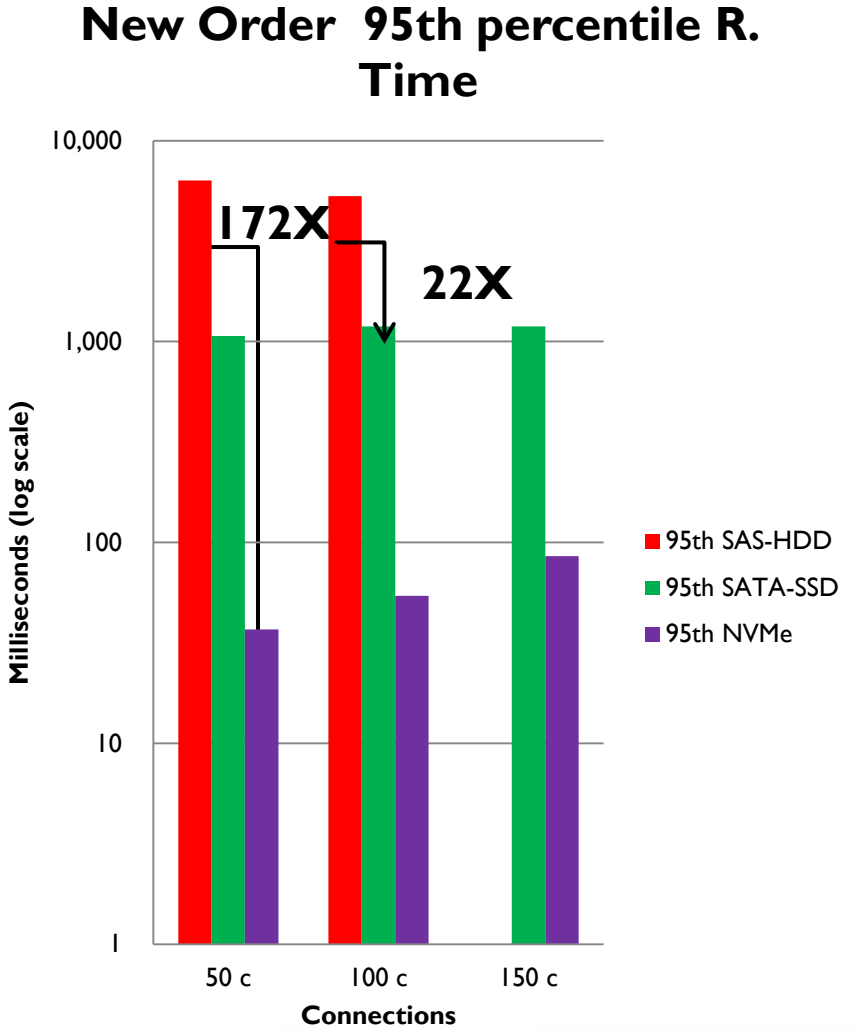
		Sequential Reads (MB/s)	Sequential Writes (MB/s)	Random Reads (IOPS)	Random Writes (IOPS)	Capacity
NVMe	XS1715*	3,000	1,400	750M	115K	1.6T
SATA	850Pro	550	520	100 K	90K	512GB
SAS	PM1633	1,350	750	190 K	30K	960GB

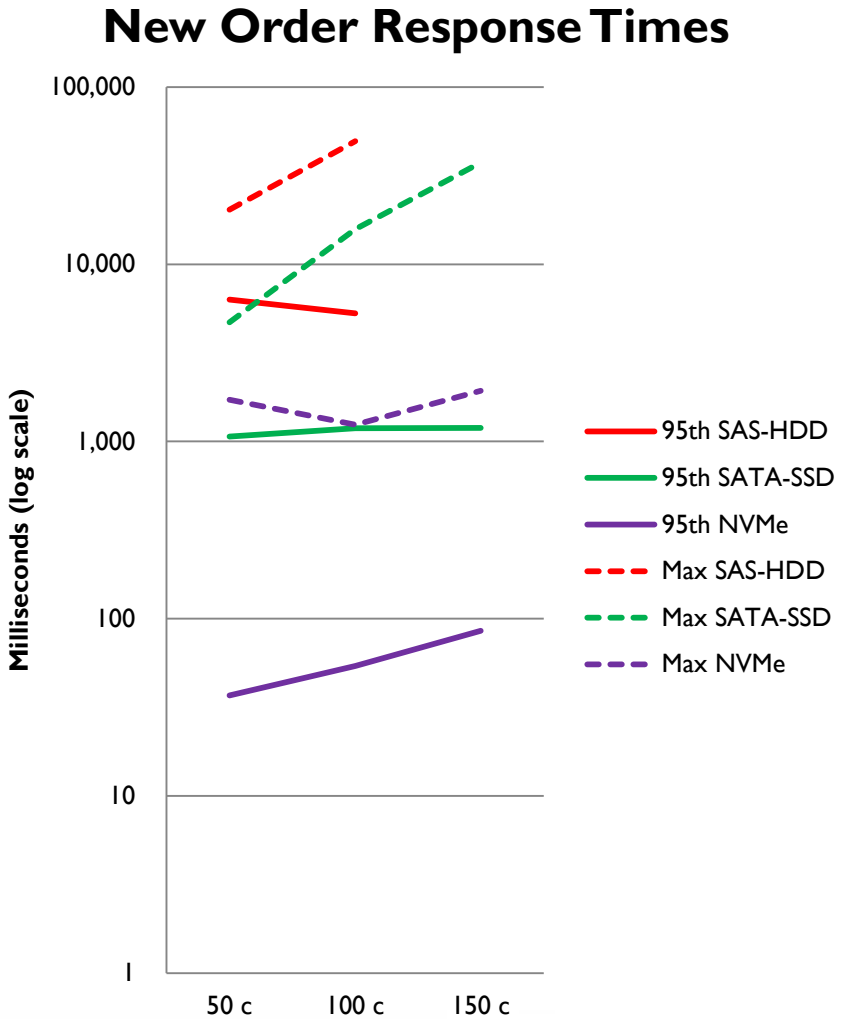
- HDD is a Seagate 15Krpm SAS HDD.
- XS1715 is a discontinued drive. Results from updated drive later in the presentation

MySQL Server Throughput



MySQL Server Optimized Response Times





MySQL Key Parameters

Parameter Name	**MySQL out-of-box**	**MySQL initial**	**MySQL optimized**	**Percona optimized**
innodb_flush_method	NULL	<empty>	O_DIRECT	
innodb_buffer_pool_size	128MB	3GB	12GB	
innodb_io_capacity	200	300,000		15,000
innodb_io_capacity_max	2,000	600,000		25,000
innodb_adaptive_hash_index	ON	OFF		0
innodb_fill_factor	100	50		100
innodb_page_cleaners	4	32		8
innodb_buffer_pool_instances	8	32		8
innodb_flush_neighbors	1	1		0
innodb_log_file_size	48MB	48MB	1G	10G
innodb_lru_scan_depth	1024	4000		8192
innodb_write_io_threads	4	16		
innodb_read_io_threads	4	16		
innodb_log_files_in_group	2	3		
innodb_max_dirty_pages_pct	75	90		
innodb_max_dirty_pages_pct_lwm	0	10		
join_buffer_size	256KB	32K		
sort_buffer_size	256KB	32K		
innodb_spin_wait_delay	6	96		6
innodb_max_purge_lag_delay	0	30000000		0
performance_schema	ON	OFF		

MySQL Server Top Tunables

The following parameters are especially important for NVMe:

- **innodb_io_capacity**
  - Sets the upper limit on the I/O activity
  - Default is 200 – a clear resource throttle for fast storage
  - “should be set to approximately the number of IOPS the system is capable of”
- **innodb_flush_method**
  - To use or not to use the filesystem cache
  - O_DIRECT will use the innodb_buffer_pool instead
- **innodb_buffer_pool_size**
  - The memory area where Innodb caches table and index data
  - Typical Goldilocks trade offs
  - Need more buffer pool space when using O_DIRECT

MySQL Server Top Tunables

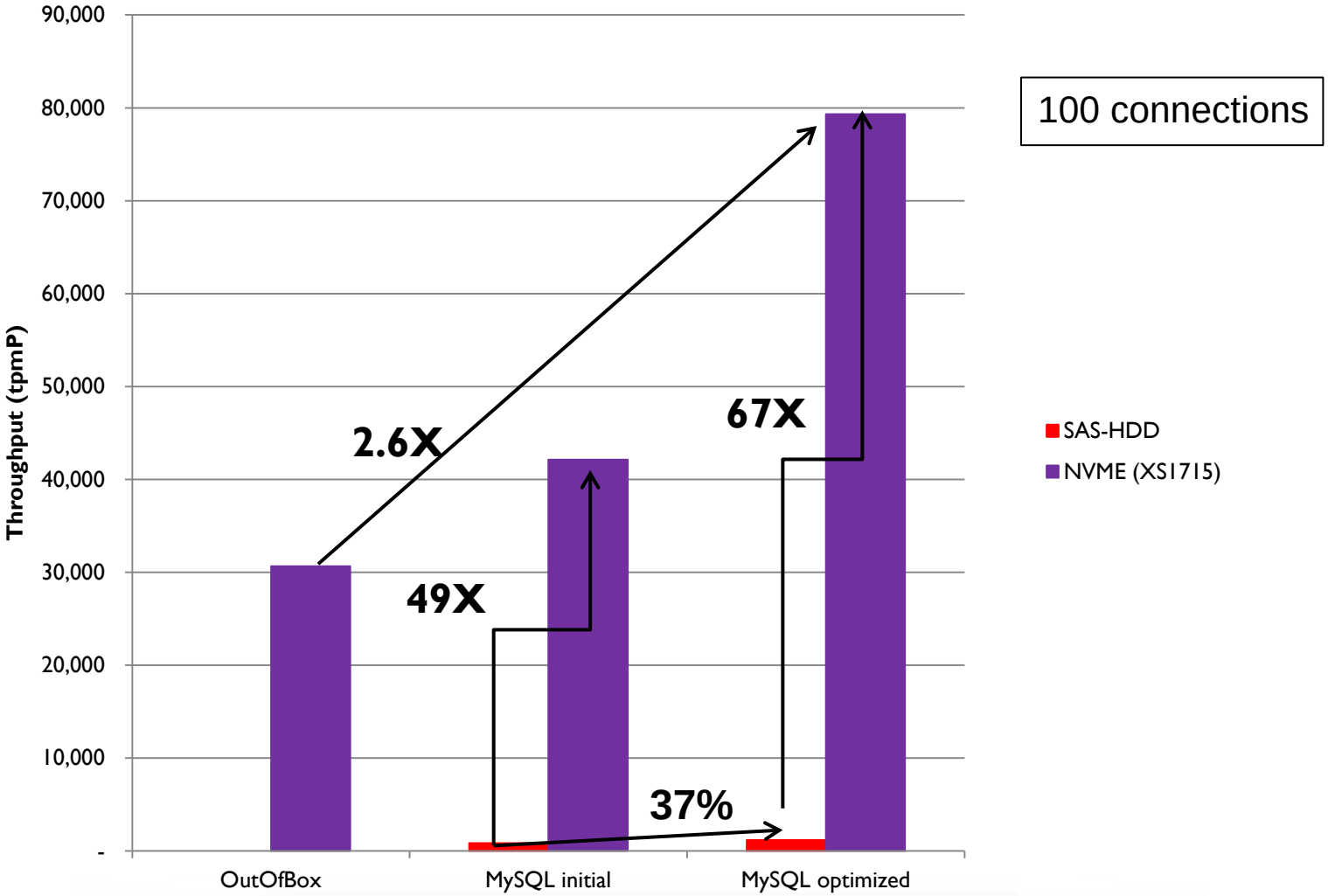
The following parameters are especially important for TPCC:

- **innodb_thread_concurrency**
  - Experimental results: For OLTP, default (0 = unlimited) yields better throughput. And, less System CPU utilization (no gatekeeping on thread creation count).
- **innodb_adaptive_hash_index**
  - Extra work to monitor index lookups and maintain the hash index structure
  - May become a source of contention
  - Experimental result: brought latency from 3+ minutes to less than 50 seconds for OrderStatus/Payment transaction types.
- **innodb_fill_factor.**
  - Percentage of each B-tree page that is filled during a sorted index build. A hint, not a hard limit
  - A smaller number may benefit transactions with INSERTs because it will decrease the number of page splits.

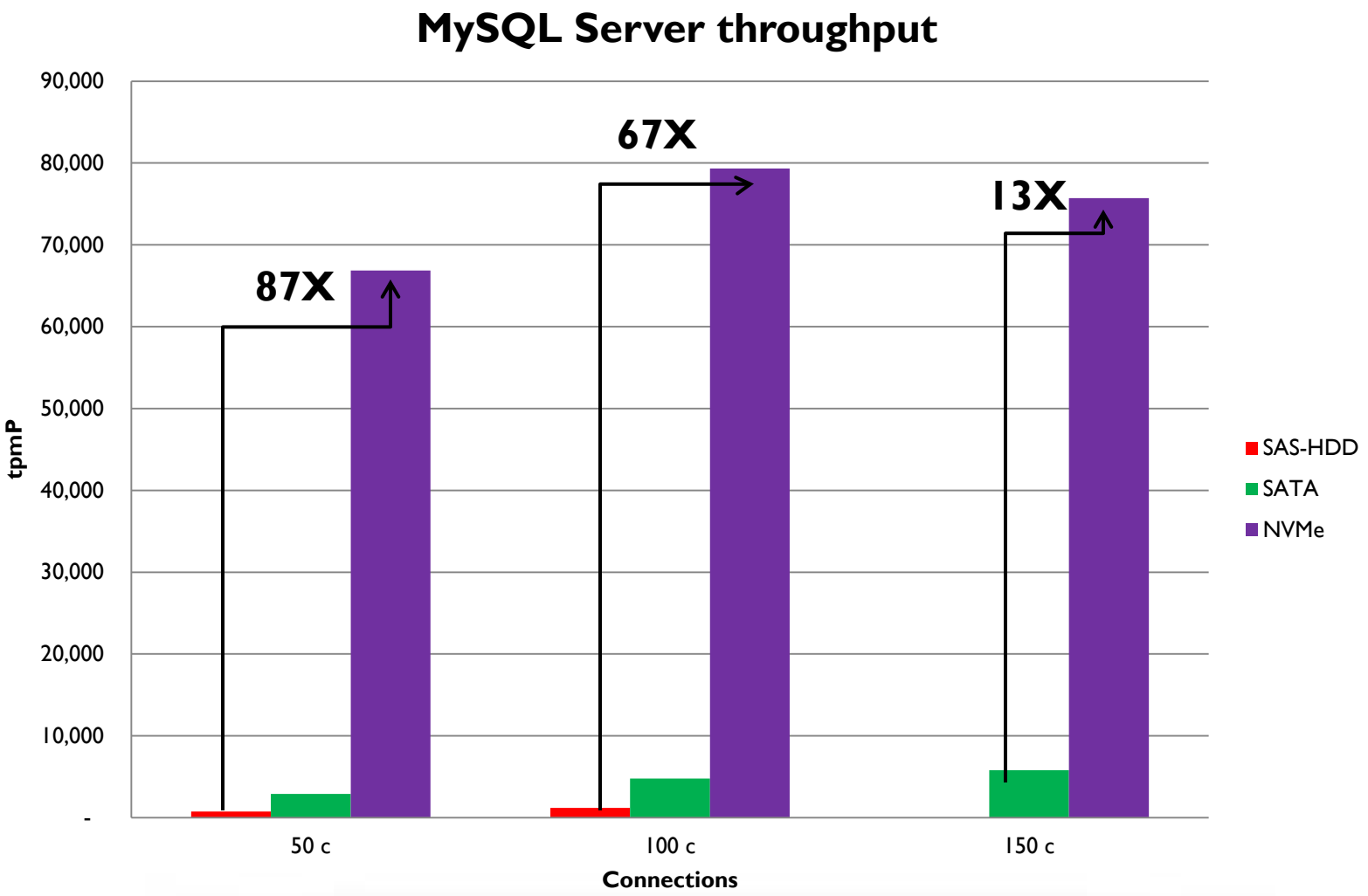
MySQL Server Optimization Learning

- MySQL Server is build for STABILITY
- Lots of latches to prevent overwhelming any subcomponent
- Important to tune I/O specific parameters
- Important to tune OLTP specific parameters
- Achieved more than 2x throughput with the optimization process

MySQL Server Optimized Throughput



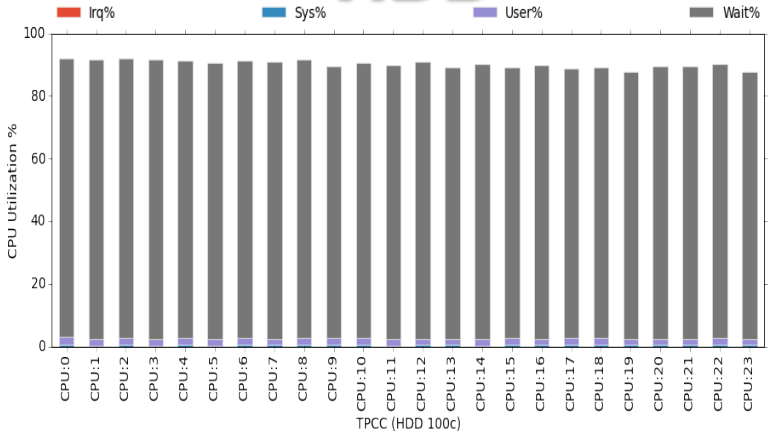
MySQL Server Optimized Throughput



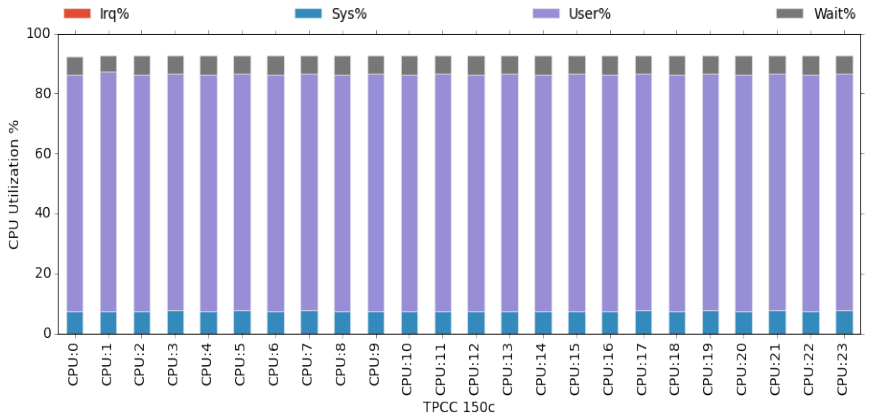
MySQL Server System Metrics

tpcc-mysql running on Dual-core:
SAS-HDD is I/O bound NVMe is CPU bound

HDD



NVMe



Mean CPU utilization (User%+Sys%)

	sas-hdd	nvme
50 c	1.81	
100 c	2.45	80.93
150 c		86.44

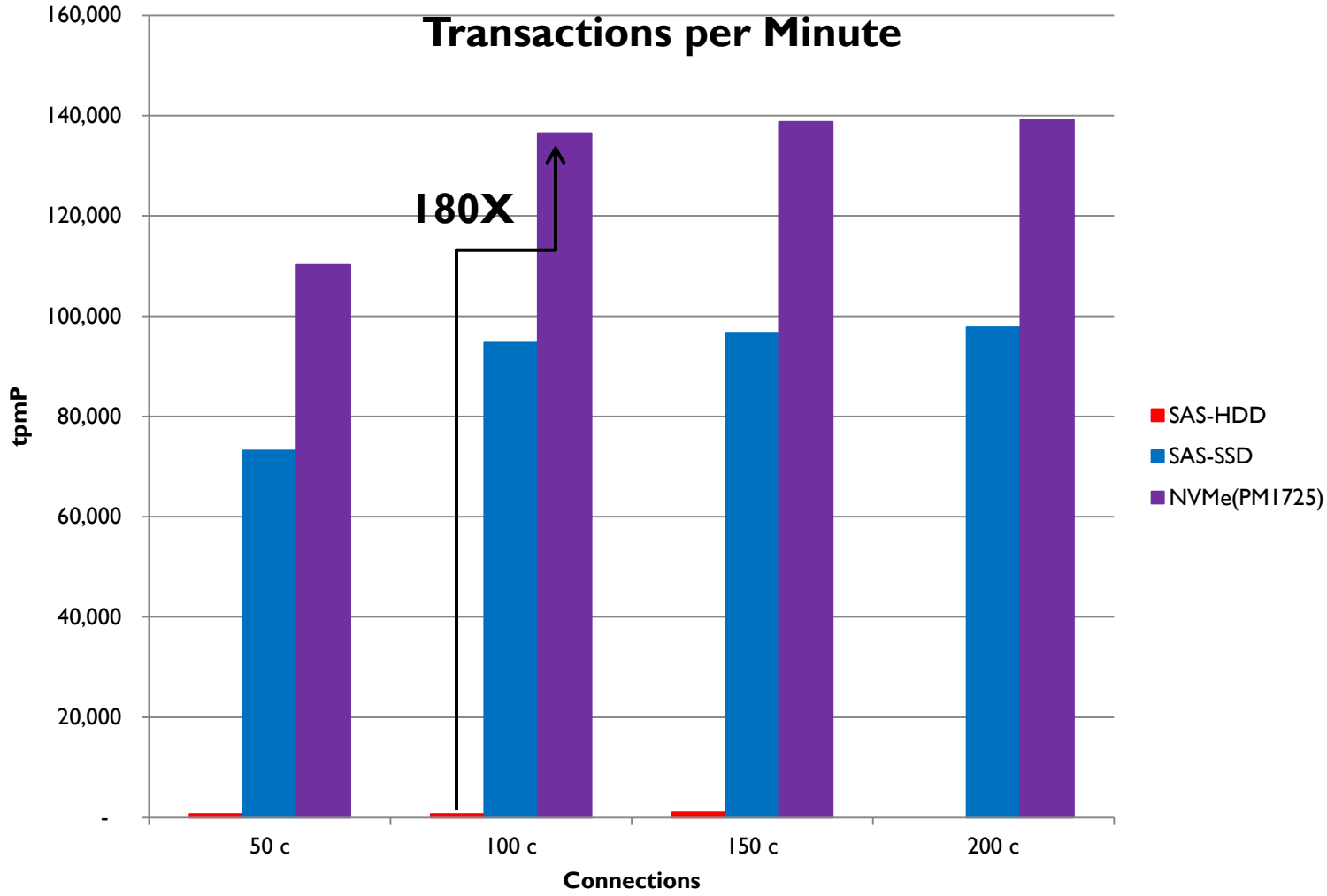
Percona Server on a Quad Socket Server

- ❑ Apply and tune optimization to Percona's Distribution
- ❑ Change to Quad-Socket Server
- ❑ NVMe is now PM1725

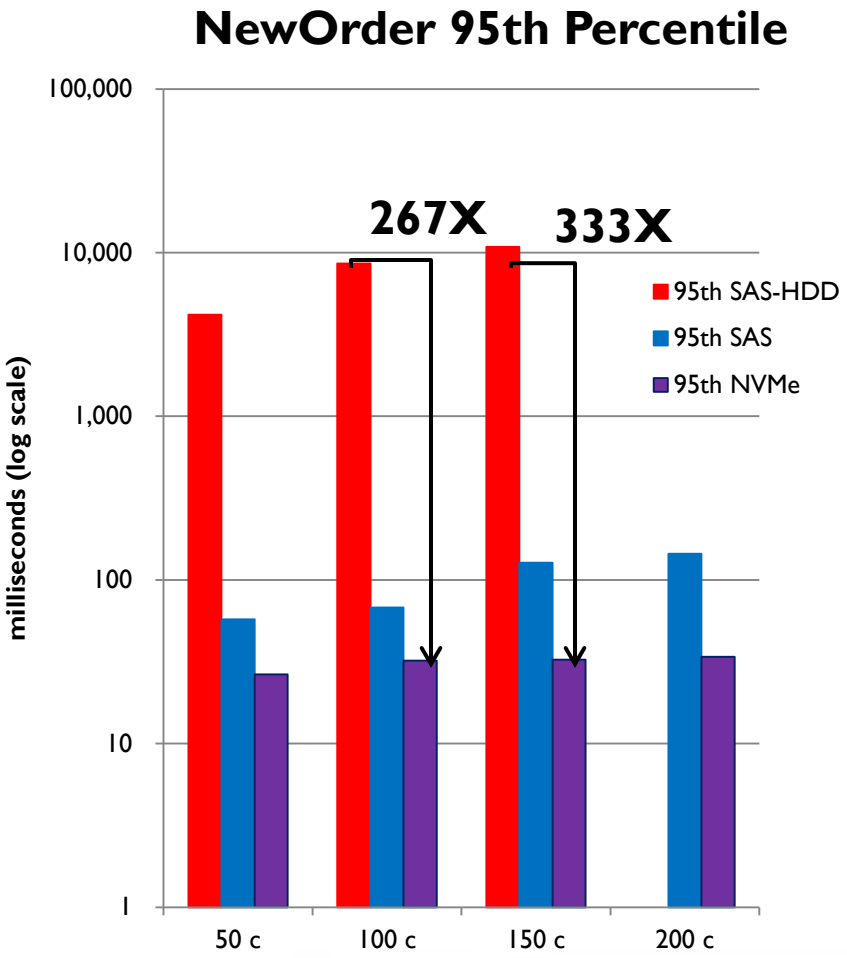
Dell PowerEdge R930 (Server)	
Model Name	Intel(R) Xeon(R) CPU E7-4850 v3 @ 2.20GHz
Memory	124GB
OS version	Linux 4.4.0-040400-generic
Storage	
SAS HDD	SEAGATE ST600MP0005 15K rpm
SATA SSD **SAS SSD**	Samsung 850 Pro Samsung PM1633
NVMe	**Samsung PM1725**

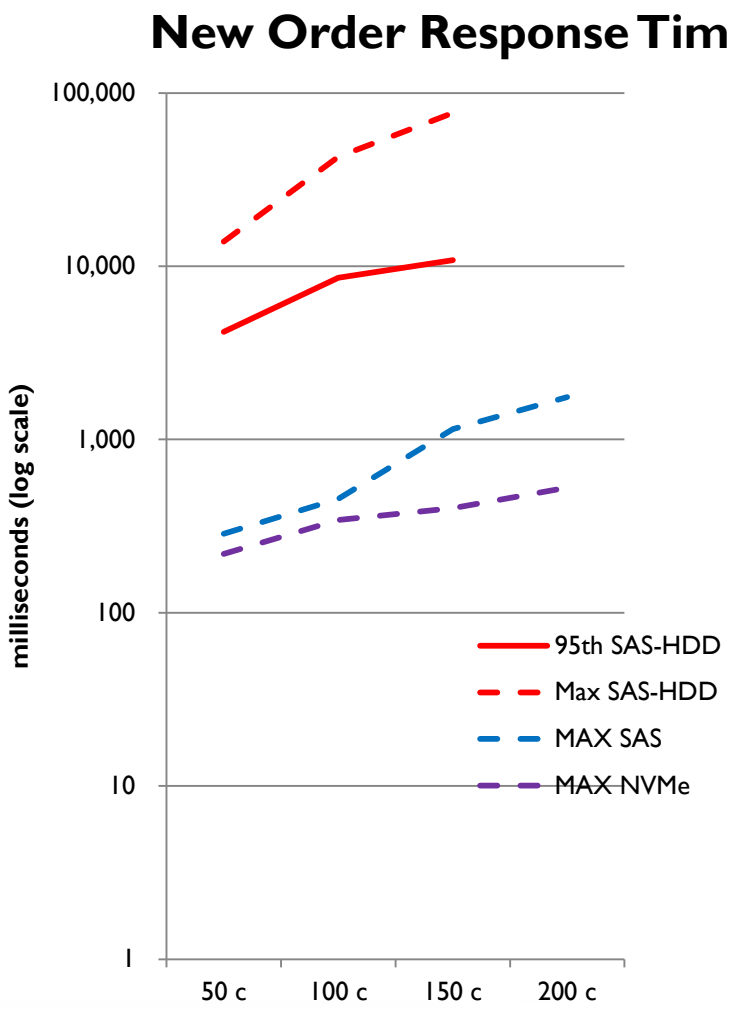
		Sequential Reads (MB/s)	Sequential Writes (MB/s)	Random Reads (IOPS)	Random Writes (IOPS)	Capacity
NVMe	PM1725	3,000	1,900	750M	130K	1.5T
SATA	850Pro	550	520	100 K	90K	512GB
SAS	PM1633	1,350	750	190 K	30K	960GB

Percona Server on a Quad Socket Server



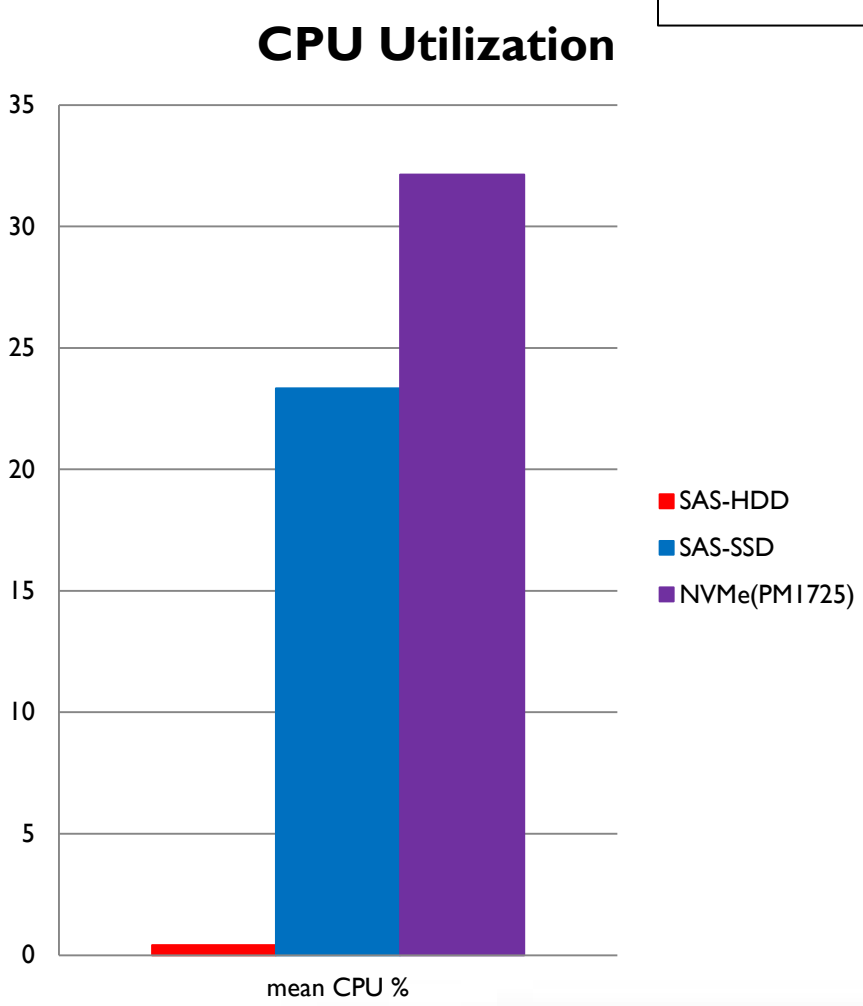
Percona Server Optimized – Response Time

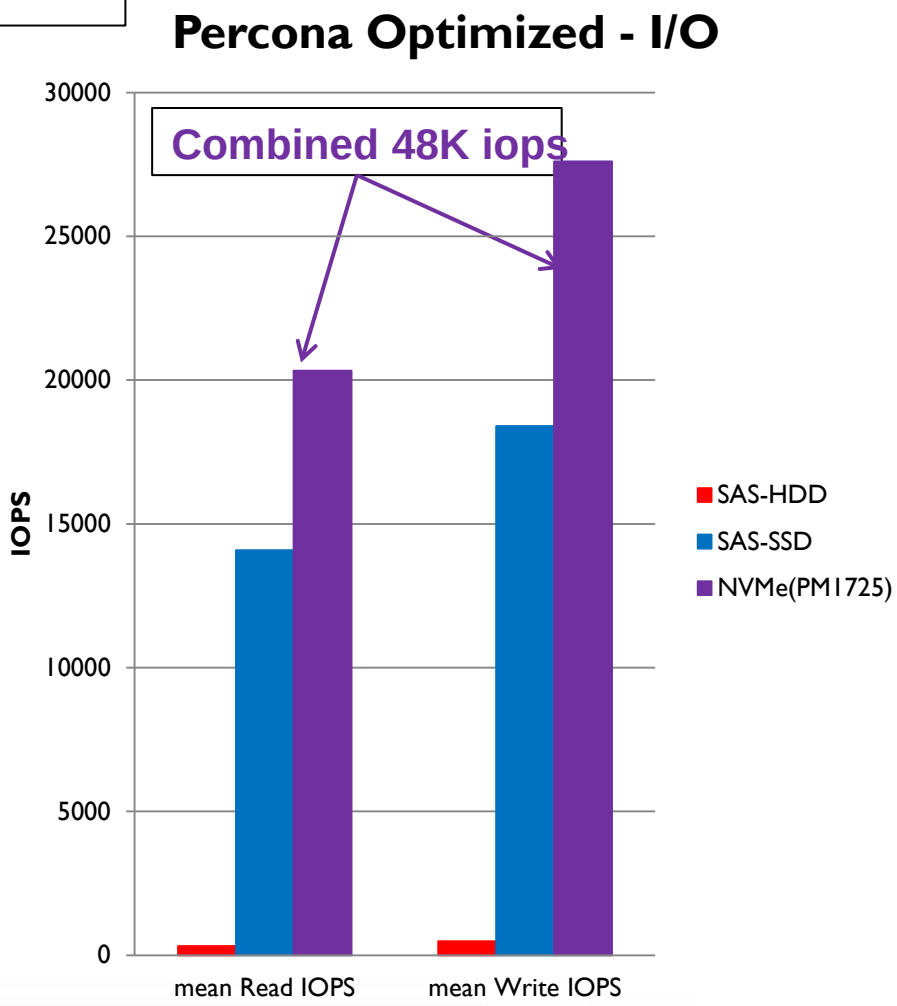




Percona Server: System Resources

100 connections



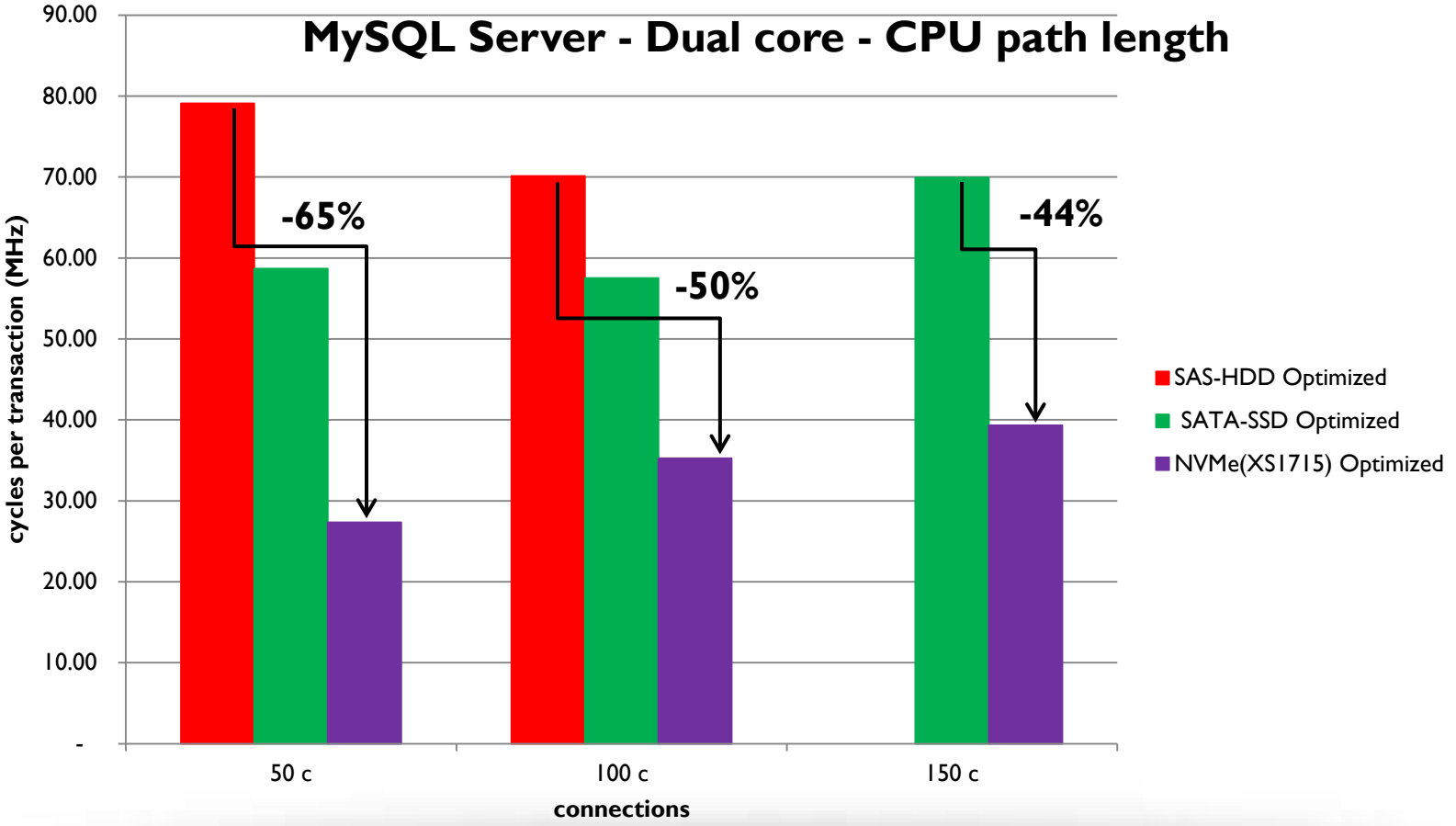


How about Server Capacity?

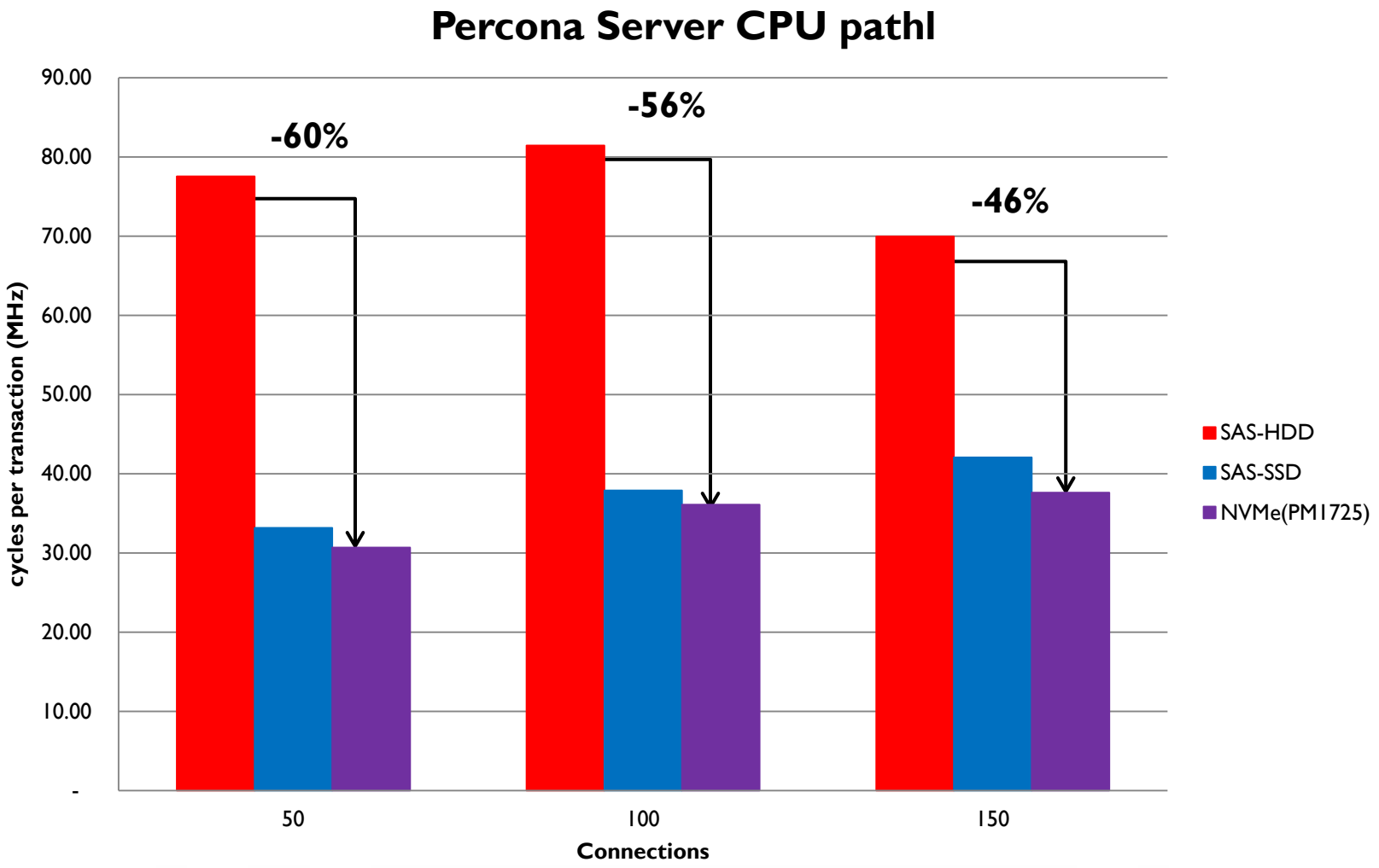
- ✓ Introducing CPU PATH LENGTH:
  - ❖ The average number of CPU cycles it takes to complete one transaction
  - ❖ **CPU PATHL = (CPU frequency * cores * total average CPU utilization) / (transactions per second)**
  - ❖ Where CPU frequency is the one reported under “Model Name”
- ✓ It is a measure of **how much work CPUs need** to do **to execute one transaction**
- ✓ Because the workload is always the same, CPU PATHL variations indicate the extra, book-keeping work that needs to be done by the Server to manage queues, buffers, context switches, etc.
- ✓ We notice that faster devices require less book-keeping by the CPUs, therefore freeing resources, therefore increasing the Server Capacity.

Server Capacity: Dual Socket MySQL Server

Decrease CPU path length by at least 50% when replacing storage from HDD to NVMe.



Server Capacity: Quad Socket Percona Server



Summary

- NVMe throughput can be 100x better than HDD
- All SSD maximum latencies are much smaller than 95th percentile HDD response times
- OLTP paradigm change from I/O bound on HDD to healthier CPU utilization when using NVMe
- Tuning is critical

Next steps

- Multiple instances using same NVMe
- Leverage fast storage:
  - Optimize software by removing “latency workarounds” added over time to minimize HDD latencies
  - Or, do less caching and buffering, do more I/O.

Questions?

veronica.l@samsung.com

Backup Slides

Better I/O balance between DATA and LOG disks with Percona Servers

Example using NVMe, 100-connection test, both on Dual Socket Server.

	MySQL Server	Percona Server
Mean CPU (User+Sys) %	81	65
Mean CPU Wait %	10	15
Mean Data Disk Reads IOPS	19,919	30,291
Mean Data Disk Writes IOPS	79,274	31,933
Mean Log Disk Writes IOPS	3,541	32,634

Percona Server Throughput