RED HAT SUMMIT

MySQL performance on OpenShift with Container-Native Storage

Red Hat Storage Architecture Team

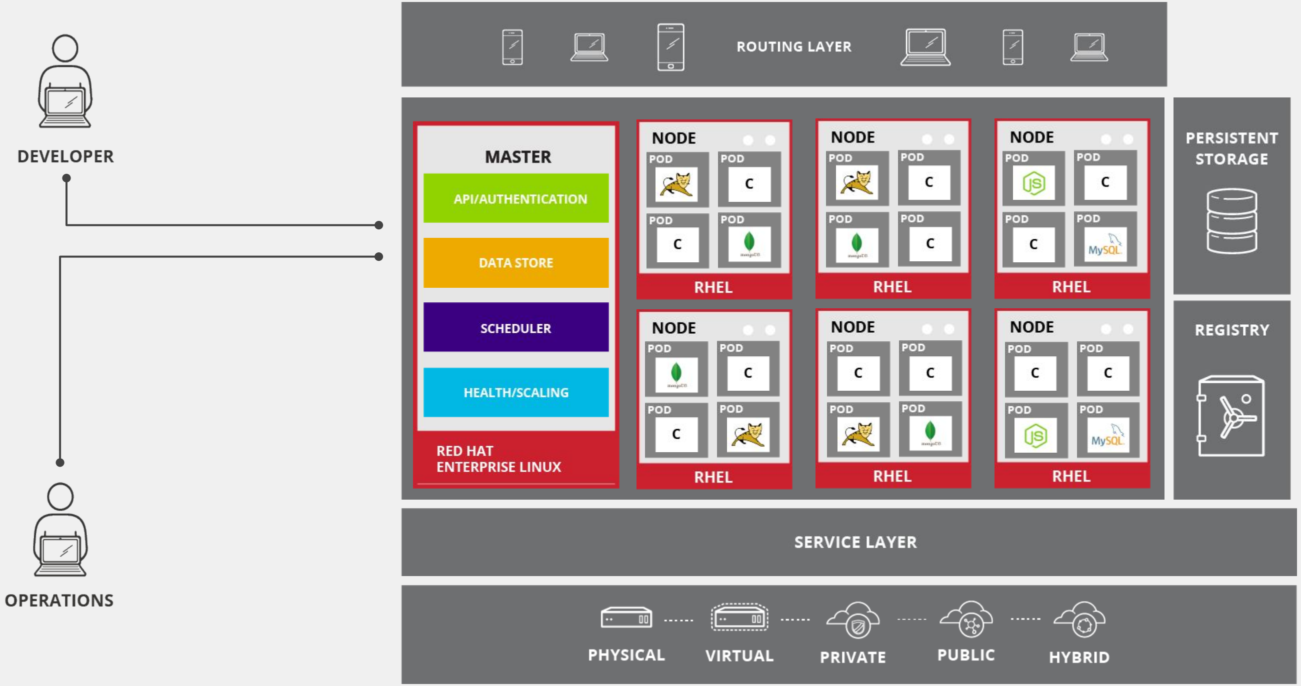
Daniel Messer
Technical Marketing Manager
05/08/2018

A primer on Container-Native Storage

In short: GlusterFS in pods, orchestrated by OpenShift + REST API

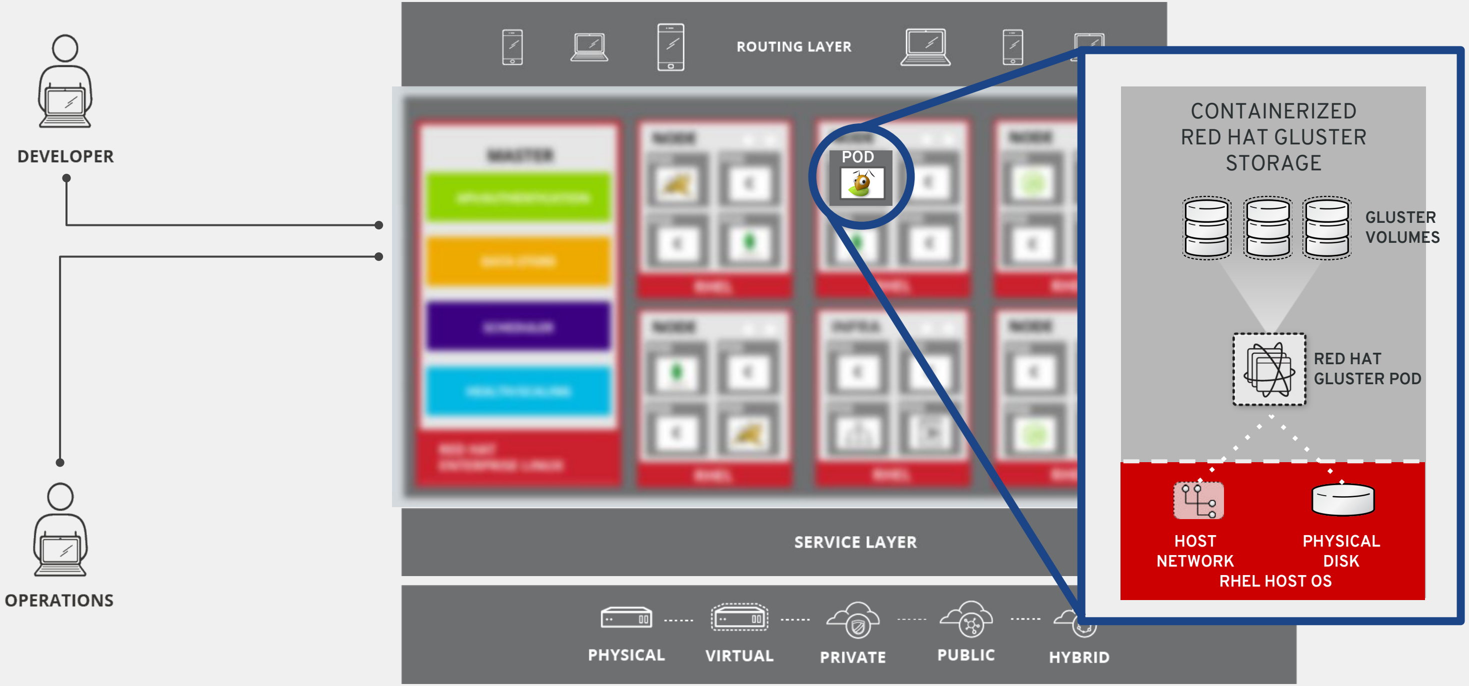
A primer on Container-Native Storage

In short: GlusterFS in pods, orchestrated by OpenShift + REST API



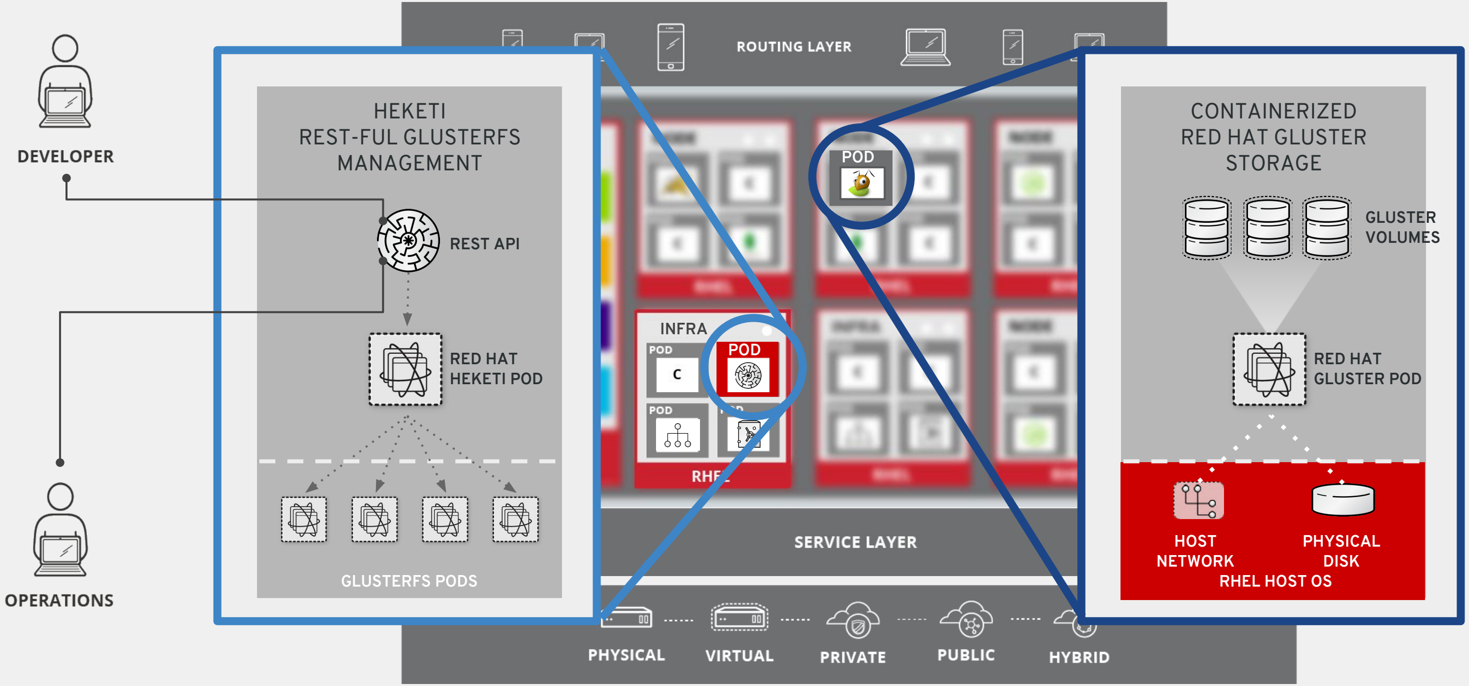
A primer on Container-Native Storage

In short: GlusterFS in pods, orchestrated by OpenShift + REST API



A primer on Container-Native Storage

In short: GlusterFS in pods, orchestrated by OpenShift + REST API



What do you get?

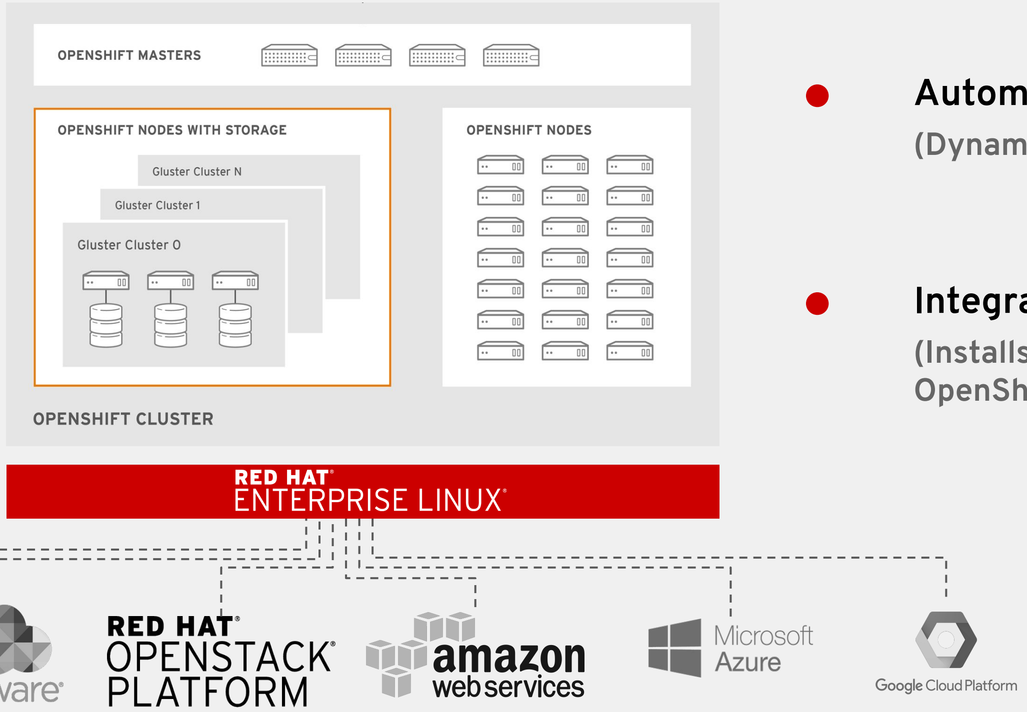
In short: a transparently replicating FUSE mount inside a pod leveraging the GlusterFS protocol

```
sh-4.2$ df -h
Filesystem                                     Size  Used Avail Use% Mounted on
overlay                                         20G  9.2G   11G  46% /
tmpfs                                          1.9G     0  1.9G   0% /dev
tmpfs                                          1.9G     0  1.9G   0% /sys/fs/cgroup
/dev/xvdc                                       10G   41M   10G   1% /etc/hosts
/dev/mapper/docker_vol-dockerlv                 20G  9.2G   11G  46% /run/secrets
shm                                             64M     0   64M   0% /dev/shm
10.0.1.80:vol_4adefe9a8096eb80596ff561aff273d6  10G  223M  9.8G   3% /var/lib/mysql/data
tmpfs                                          1.9G   16K  1.9G   1% /run/secrets/kubernetes.io/serviceaccount
tmpfs                                          1.9G     0  1.9G   0% /proc/scsi
tmpfs                                          1.9G     0  1.9G   0% /sys/firmware
```

```
sh-4.2$ mount | grep mysql
10.0.1.80:vol_4adefe9a8096eb80596ff561aff273d6 on /var/lib/mysql/data type fuse.glusterfs   ↲
 (rw,relatime,user_id=0,group_id=0,default_permissions,allow_other,max_read=131072)
```

Why is this a good idea?

- **Scalable**
  (1000 PVs/cluster)
- **Highly-available**
  (across cloud AZs)



- **Automated**
  (Dynamic Provisioning)
- **Integrated**
  (Installs with/runs on OpenShift)

BUT WILL THIS RUN MY **DATABASE?**

MOST POPULAR DATABASE ON OPENSHIFT ONLINE:

MYSQL

sysbench?

Well-known artificial benchmark for MySQL

Stresses the MySQL database engine using a set of predefined SELECT, UPDATE, INSERT and DELETE operations in a single table.

- Stresses IO and query processing subsystem
- Does not involve application logic
- Operates on a single table only, insert millions of rows

Verdict:

- **Is not using a real world data model**
- **Does not tell you anything about user/application-perceived performance**

DVDStore!

A full stack database-centric benchmark: https://github.com/dvdstore

An open source benchmark tool simulating and e-commerce platform.

- Simulates users logging in, browsing product catalog, reading/writing reviews and place orders including think time
- Has application (PHP) code driving the database and rendering user interfaces
- Fully normalized schema with several tables

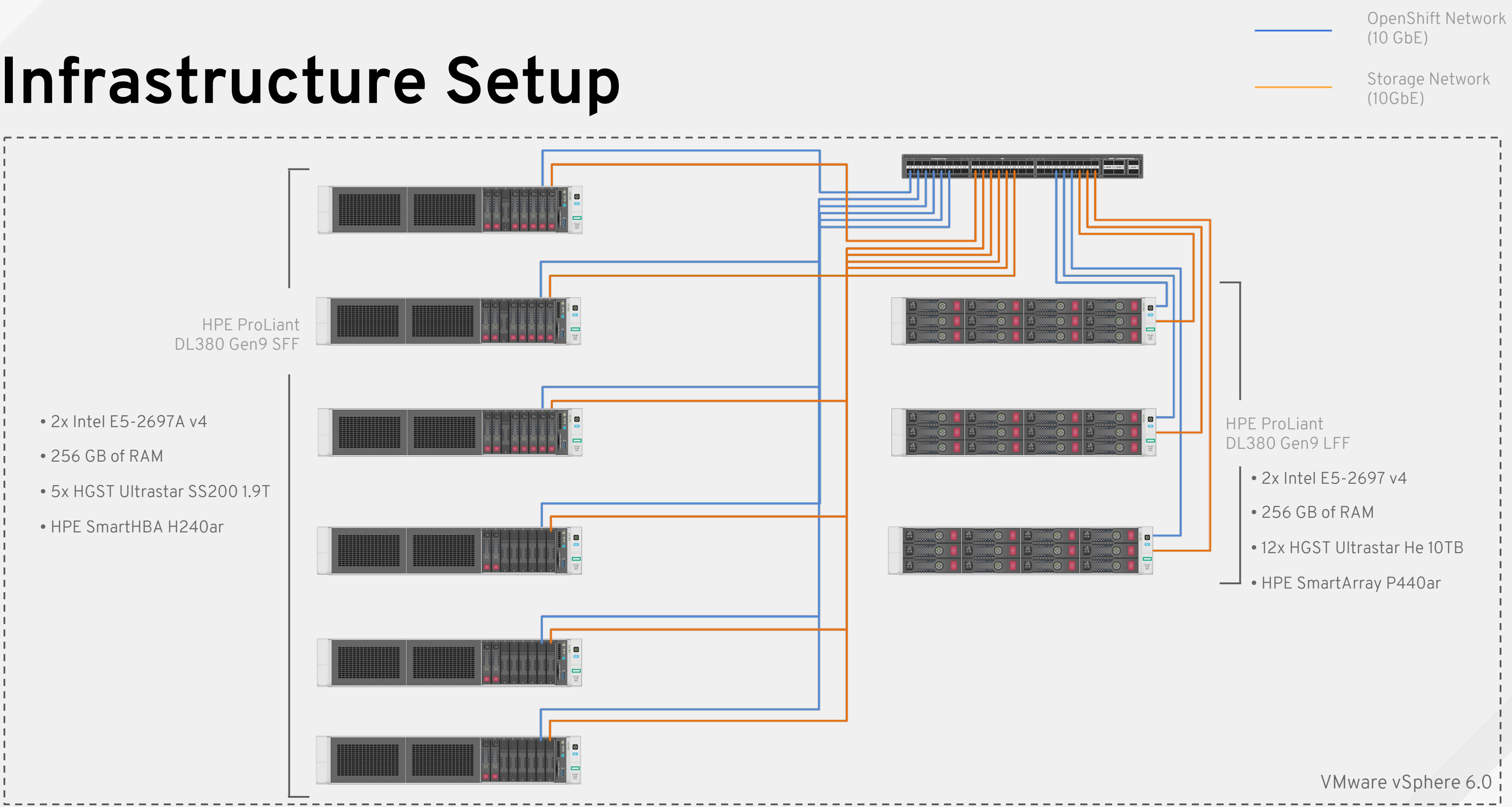
Verdict:

- **Represents a typical web application stack as they are found with many customers**
- **Reports application/user perceived performance in Orders Per Minute (OPM)**

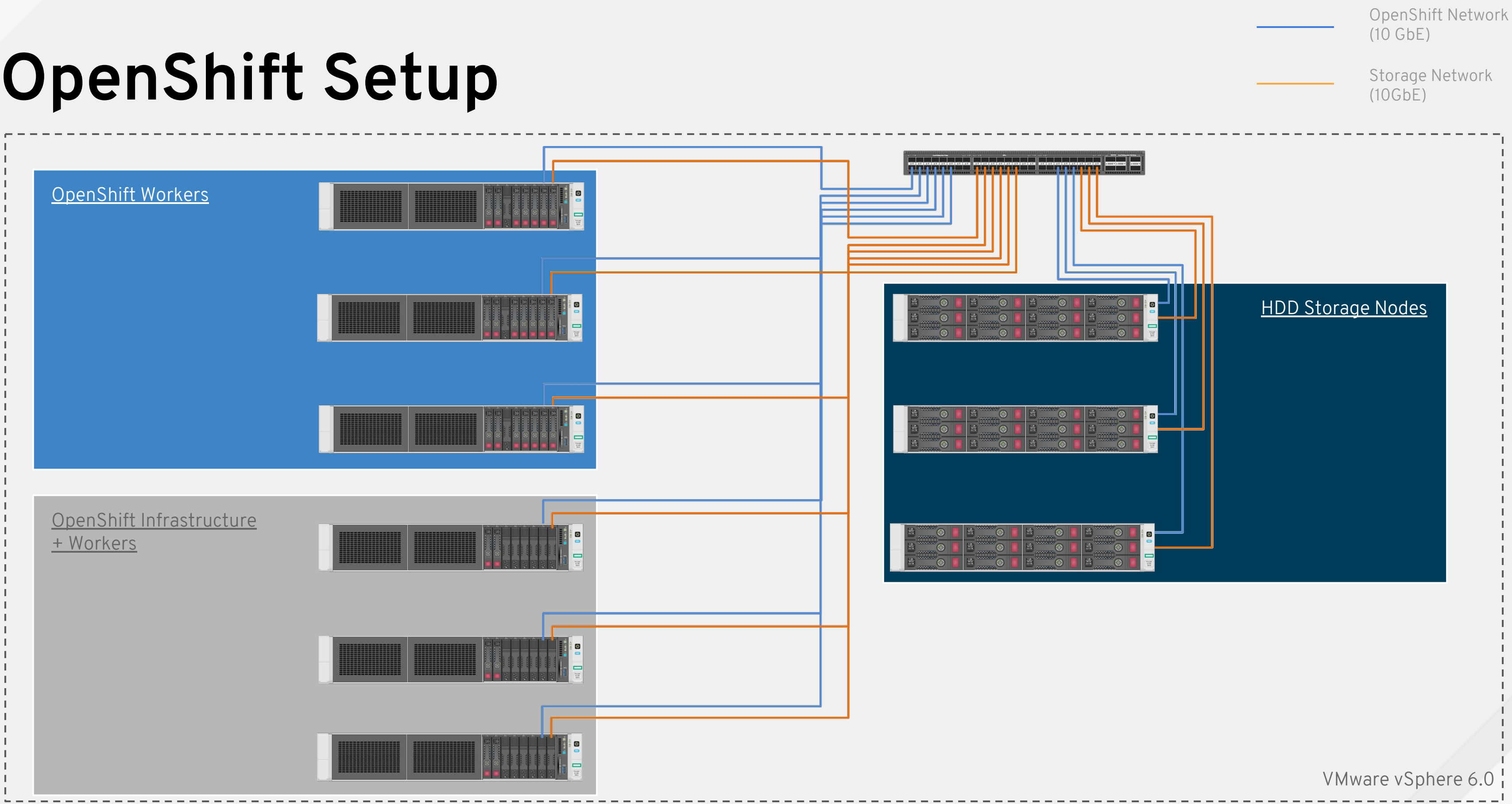
LEADING QUESTIONS:

- ❖ What is the recommended setup to achieve good performance?
- ❖ How many of these web application stacks can we run?
  - ➢ within a target latency envelope (<700 ms)?
- ❖ How does workload and cluster performance scale as we increase load?
- ❖ Will this kind of workload need SSDs or HDDs?

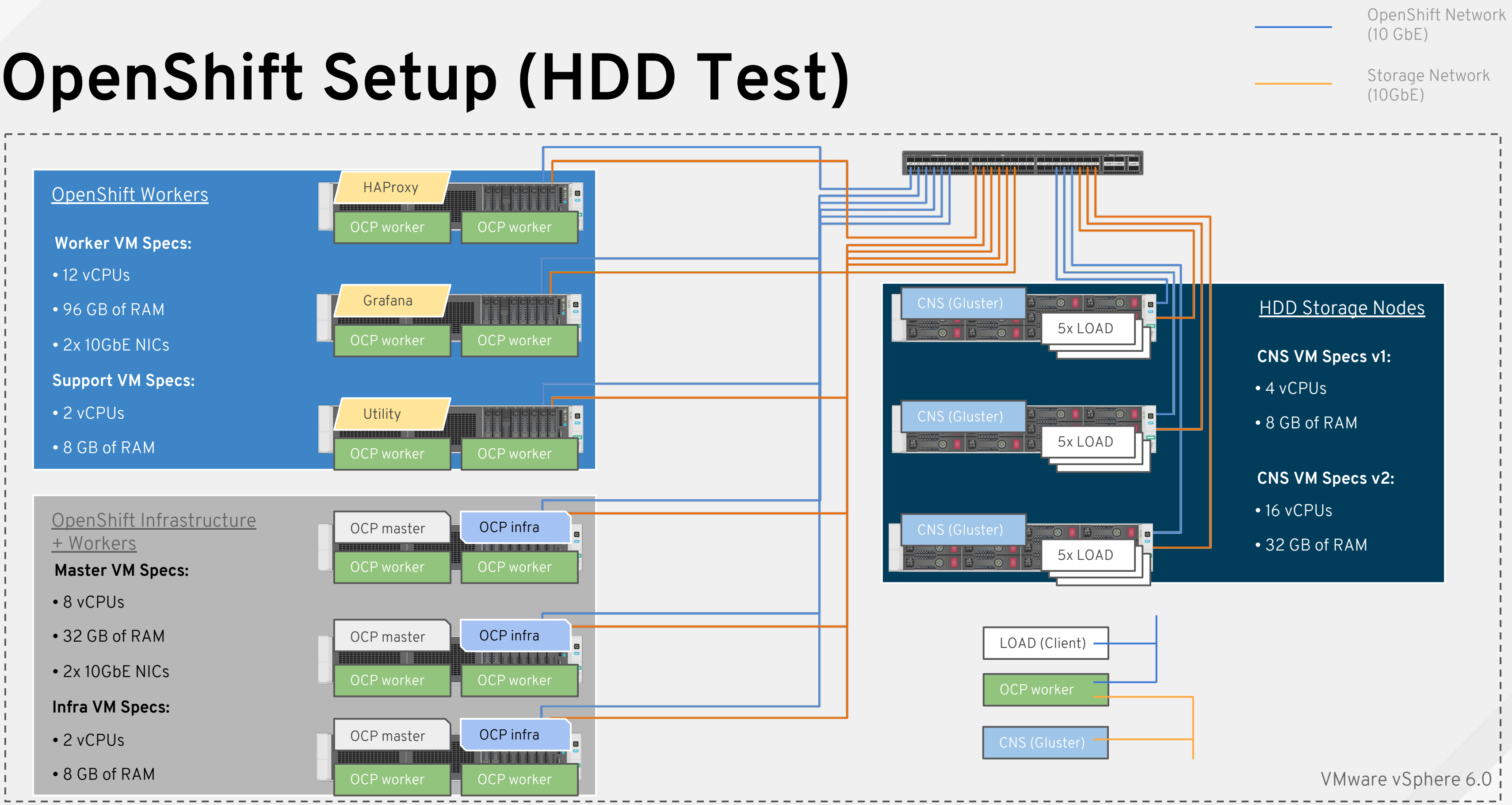
Infrastructure Setup



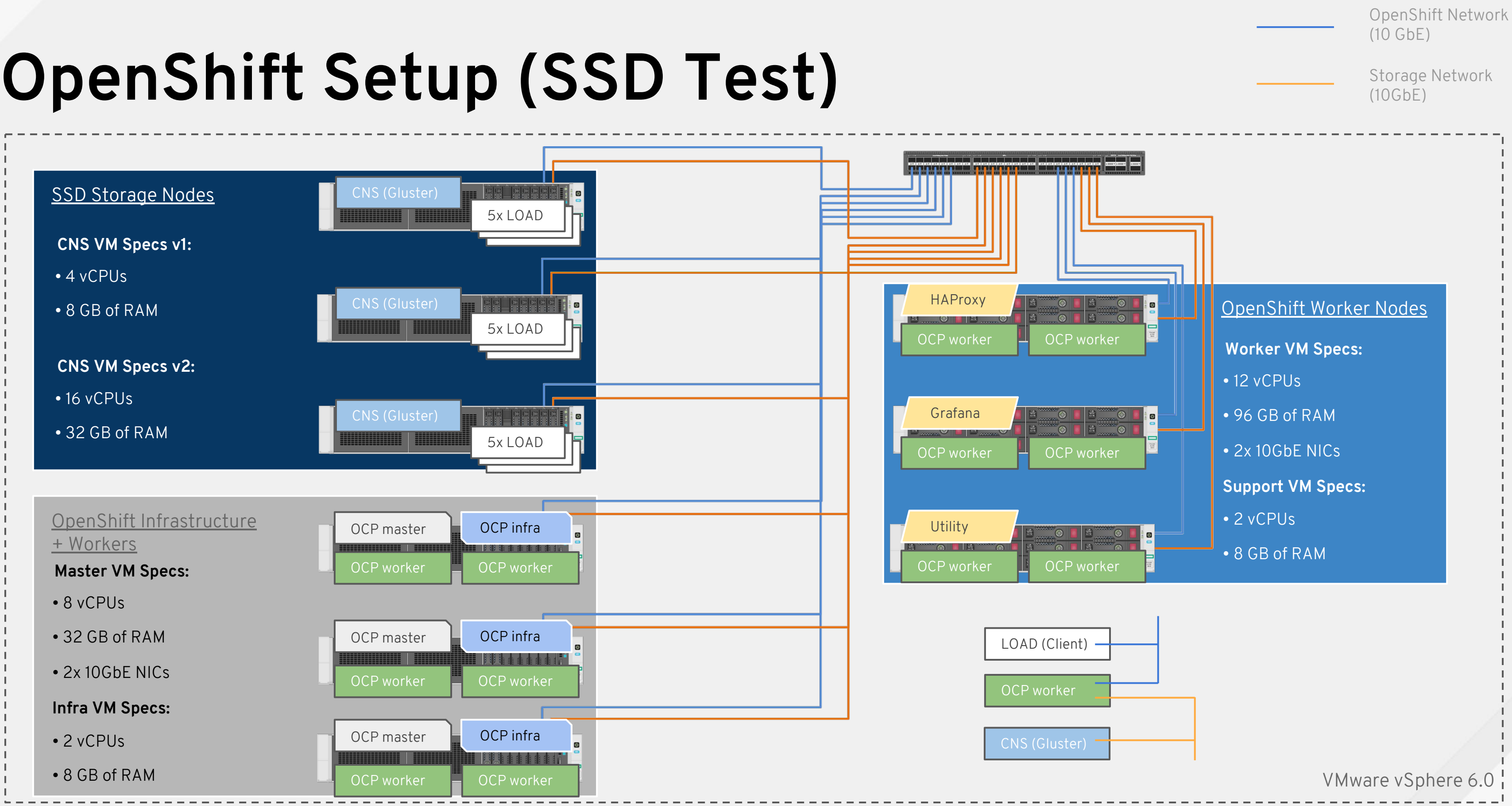
OpenShift Setup



OpenShift Setup (HDD Test)



OpenShift Setup (SSD Test)



Container-Native Storage Setup

Standard 3-node cluster design with 5 SSDs vs. 12 HDDs per node

Coupled

- GlusterFS runs in Pods on app nodes
- GlusterFS uses host networking (same NIC as OpenShift SDN)
- Heketi pod runs inside OpenShift, orchestrating GlusterFS via kubeexec

De-Coupled

- GlusterFS runs outside of OpenShift
- GlusterFS can use a different network than OpenShift SDN
- Heketi pod runs inside OpenShift, orchestrating GlusterFS via SSH

Load Driver Setup - two versions of DVDStore

OPM numbers are not comparable!

DS2

- stresses the database with lots of independent queries
- known to be IO centric
- “headless” test mode: load driver queries database directly
- No product reviews
- 20 GB database
- Think time: 10ms
- 8 concurrent users / database
- OCP stock mysql:5.7 images

DS3

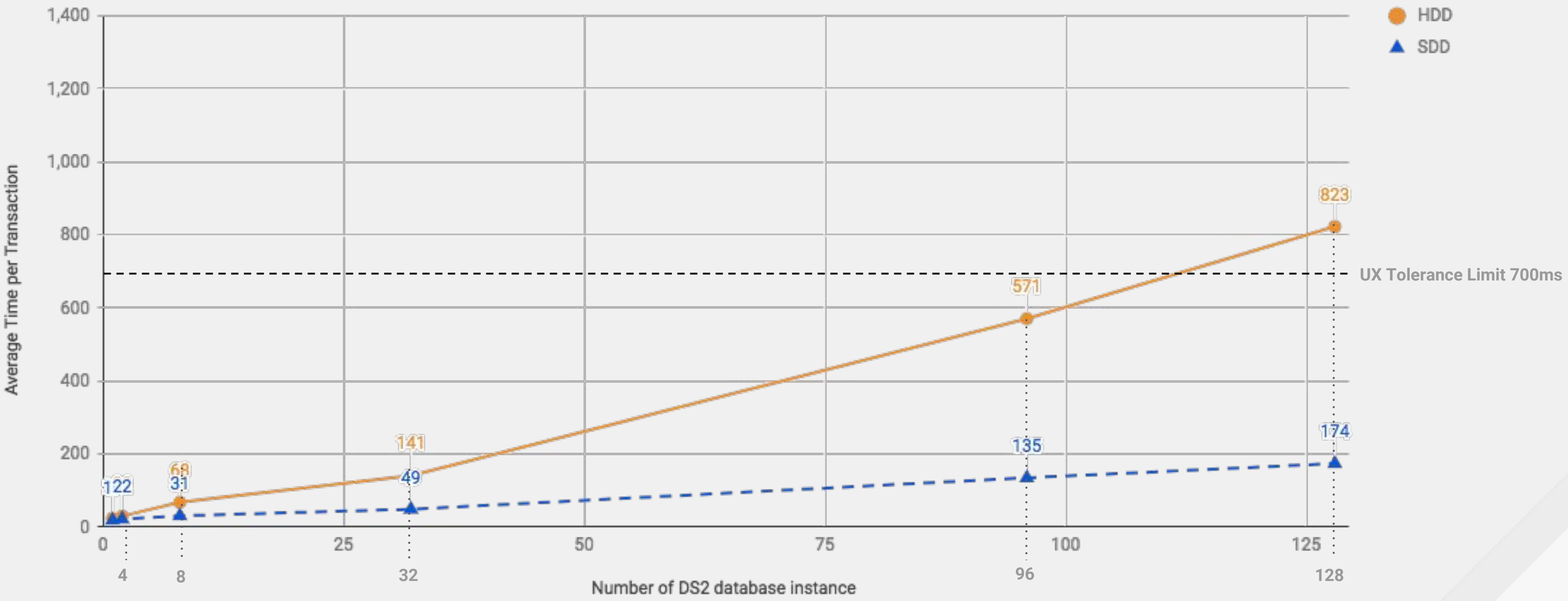
- Optimized database access with newer client libraries, stored procedures
- Known to be less demanding on the database
- Load driver accesses application via API (simulating end users)
- 5 GB database
- Think time: 300 ms
- 6 concurrent users / application stack
- OCP stock mysql:5.7/php:5.6 images

BENCHMARK RESULTS

DVDStore2 - Driving the RDBMS

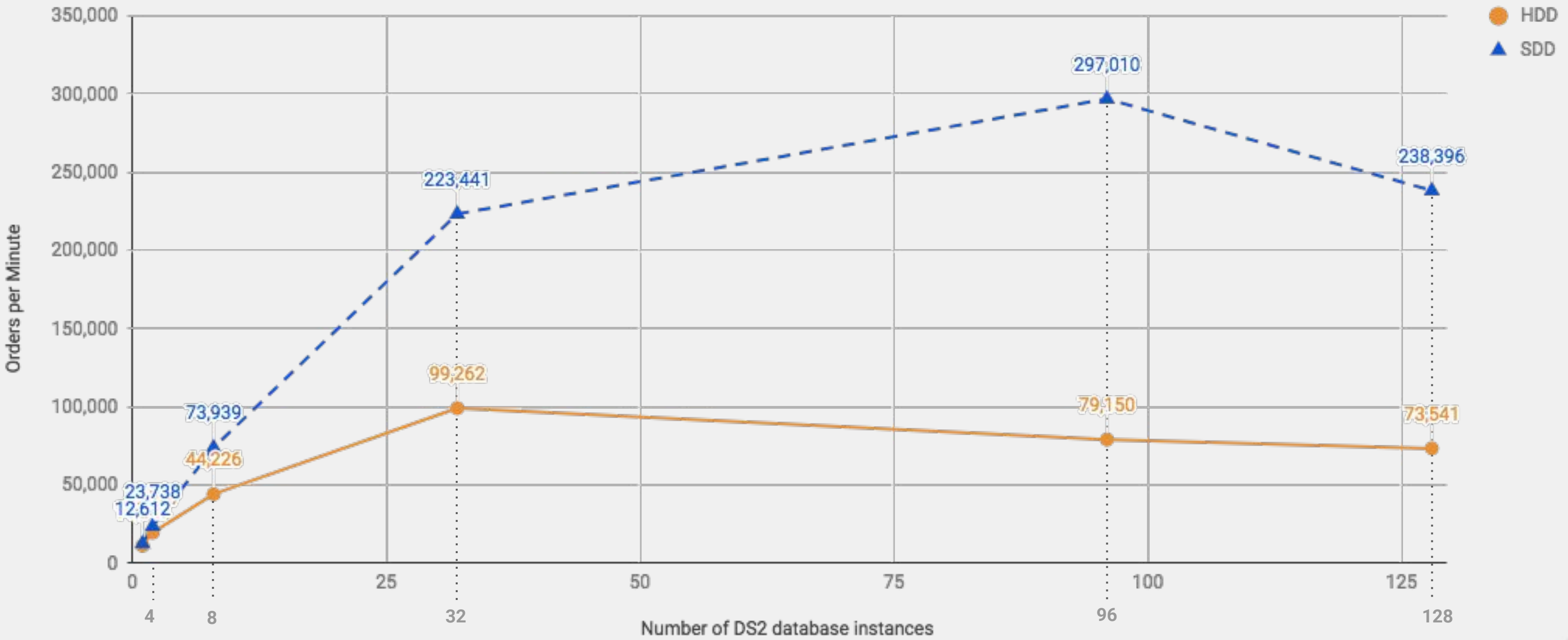
DVDStore2 - Database only - CNS Spec #1

Average Transaction Response Time



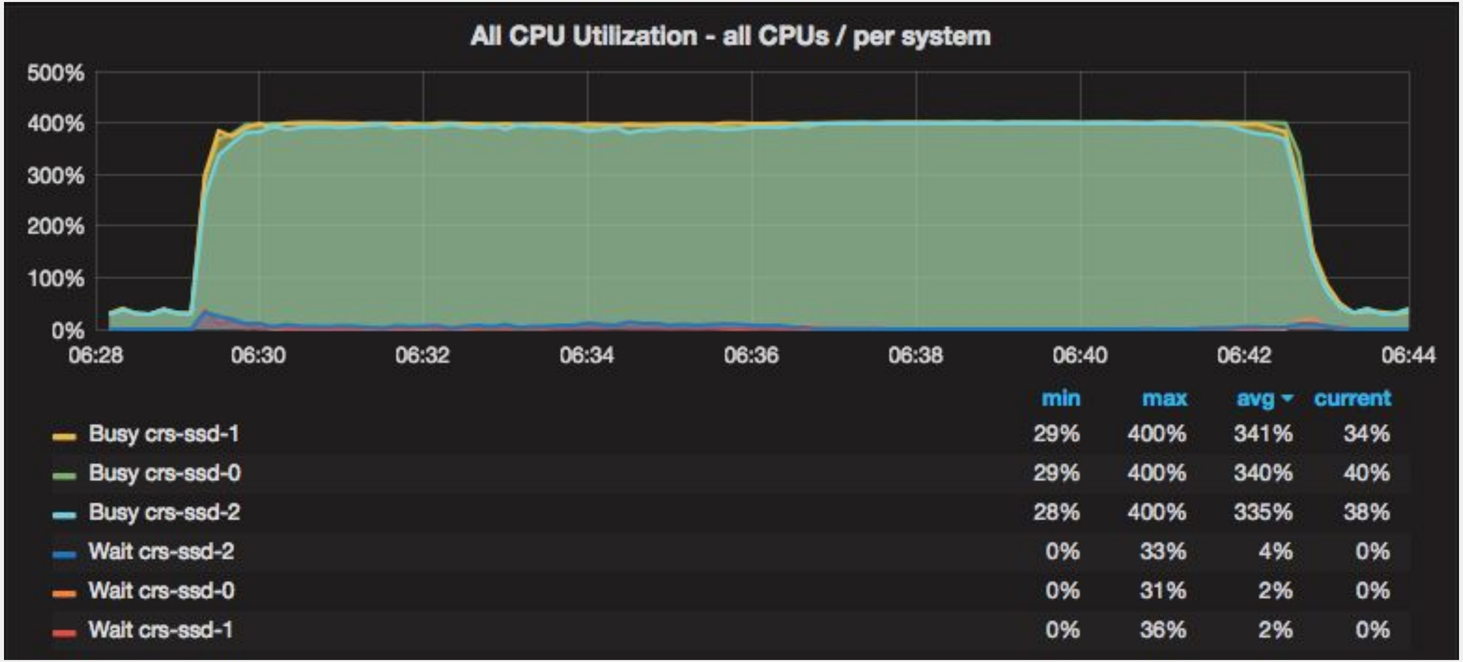
DVDStore2 - Database only - CNS Spec #1

Aggregate Order Throughput results



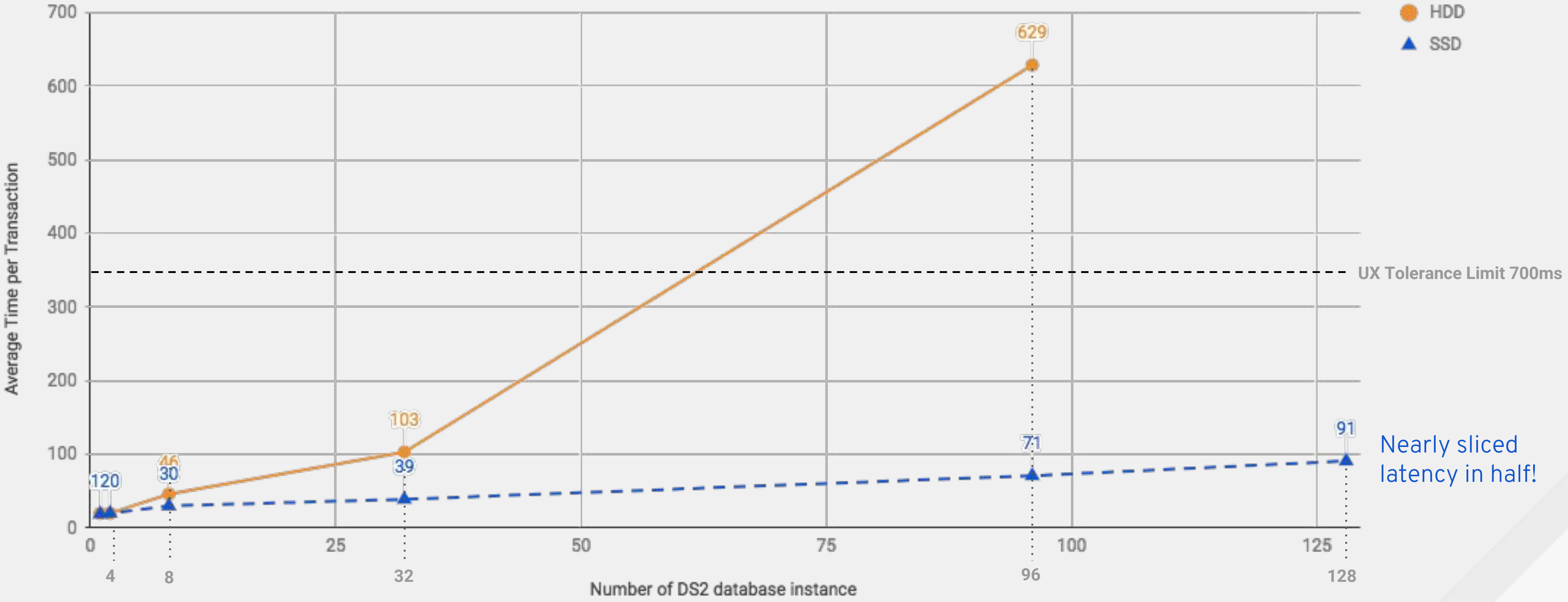
DVDStore2 - Database only - CNS Spec #1

Investigating performance bottleneck - CPU utilization CNS nodes



DVDStore2 - Database only - CNS Spec #2

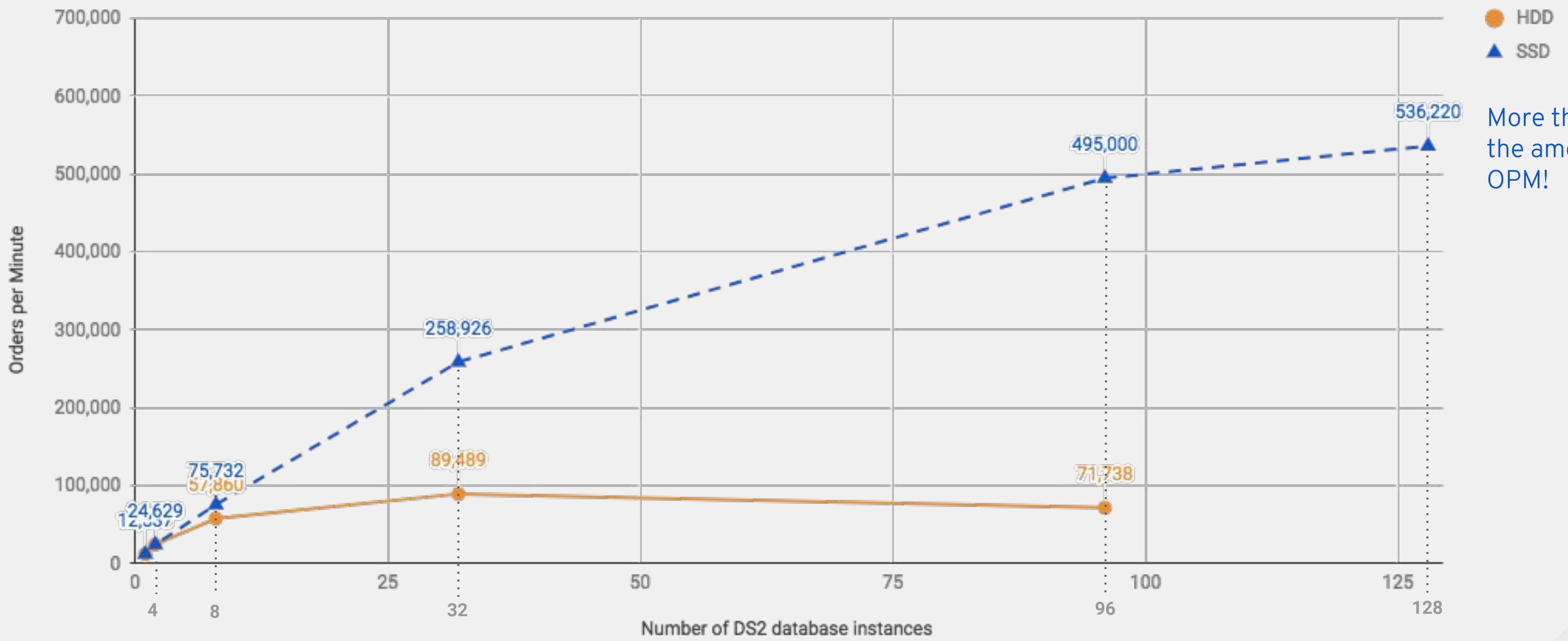
Average Transaction Response Time with increased resources for CNS: 16 vCPU / 32GB vRAM



Nearly sliced latency in half!

DVDStore2 - Database only - CNS Spec #2

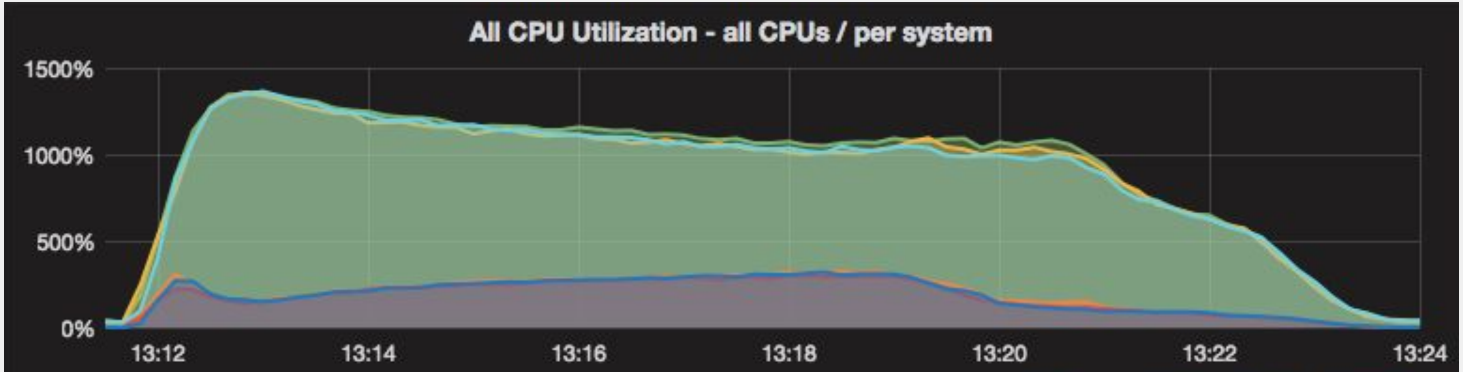
Aggregate Order Throughput results with increased resources for CNS: 16 vCPU / 32GB vRAM



More than twice the amount of OPM!

DVDStore2 - Database only - CNS Spec #2

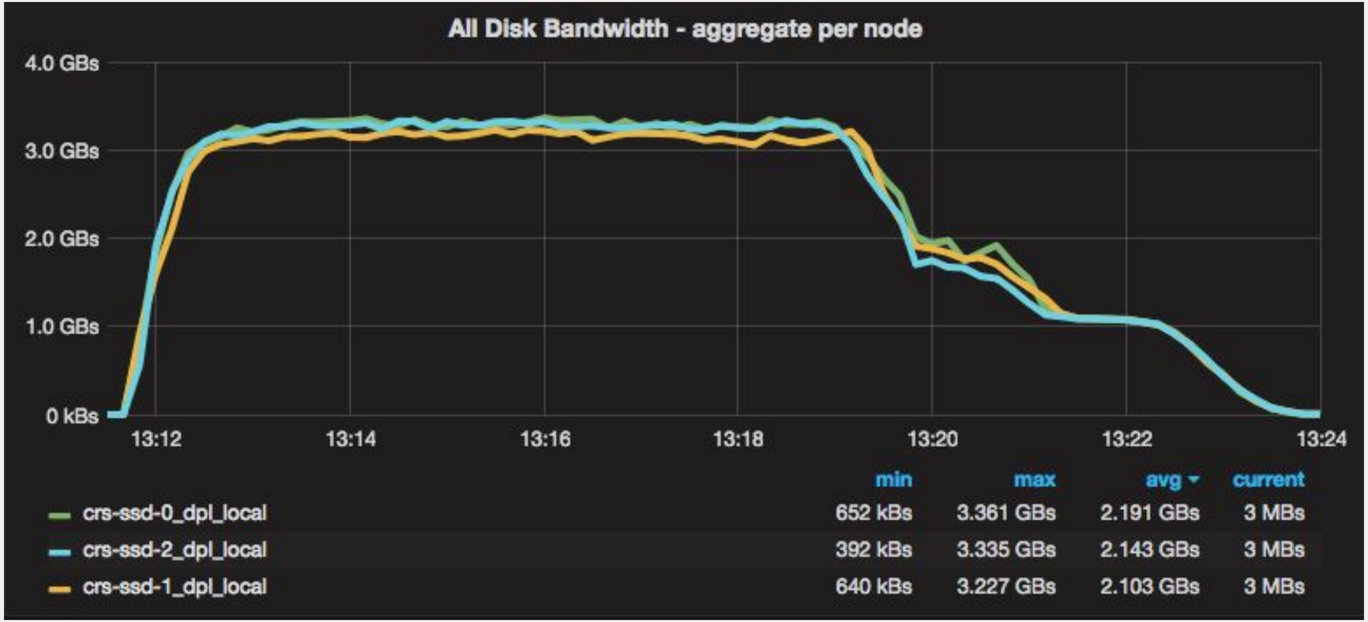
Removing the bottleneck on CNS Nodes on SSD after going to 16 vCPUs



	min	max	avg	current
Busy crs-ssd-0	28%	1365%	862%	39%
Busy crs-ssd-1	30%	1360%	837%	32%
Busy crs-ssd-2	28%	1363%	835%	42%
Wait crs-ssd-0	0%	324%	177%	0%
Wait crs-ssd-2	0%	321%	171%	0%
Wait crs-ssd-1	0%	300%	165%	0%

DVDStore2 - Database only - CNS Spec #2

Checking other resources for bottlenecks: SSD utilization on CNS nodes

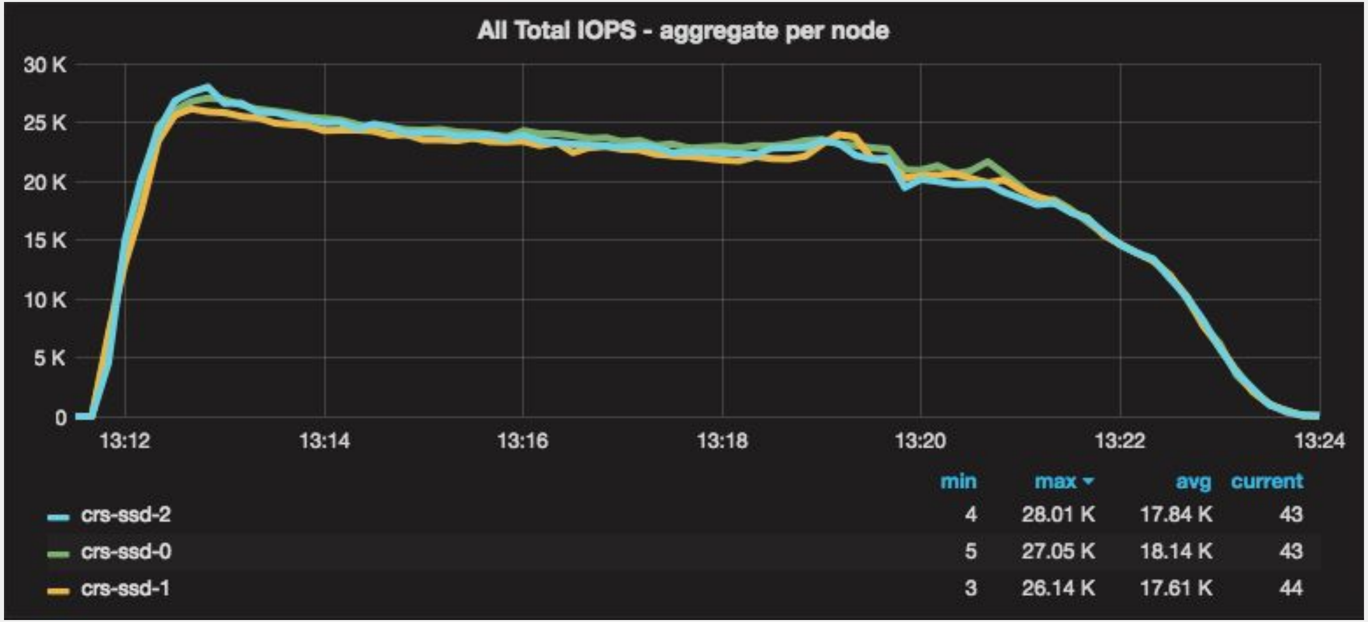


5x HGST Ultrastar SS200 SSD

- SAS 12Gb/s
- 1.8GB/s read
- 1.0GB/s write
- 250K random read (4K)
- 37K random write (4K)
- 100 µs access latency

DVDStore2 - Database only - CNS Spec #2

Checking other resources for bottlenecks: SSD utilization on CNS nodes

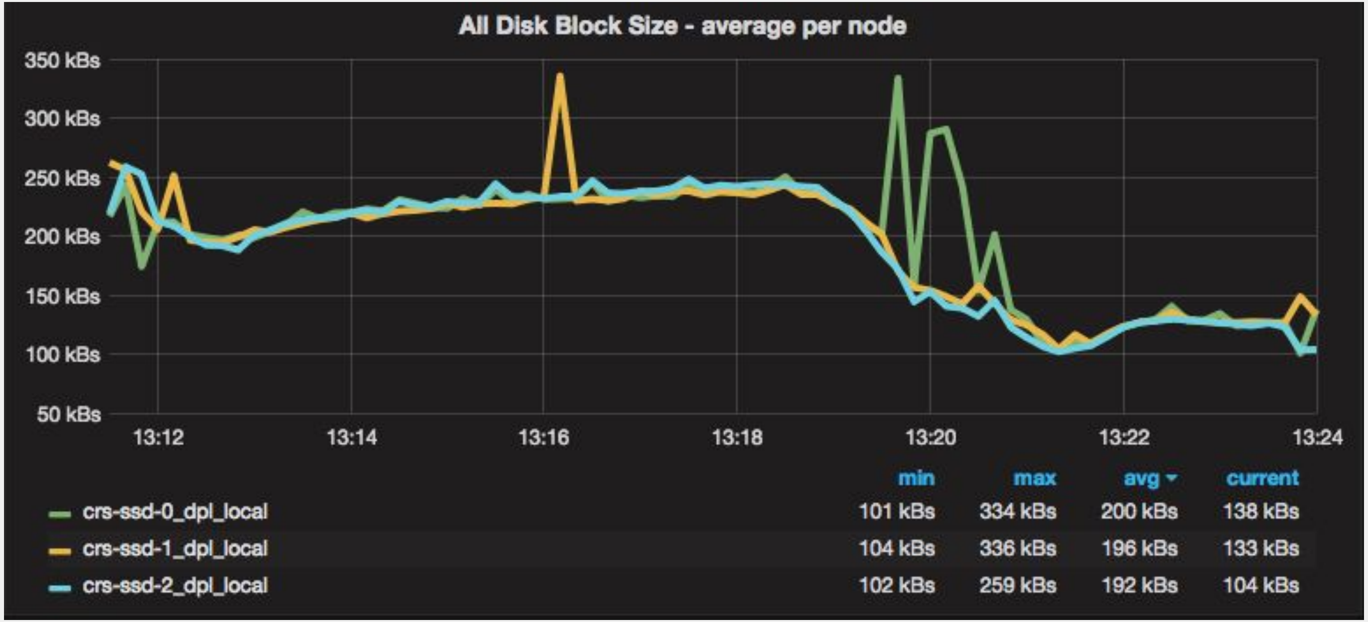


5x HGST Ultrastar SS200 SSD

- SAS 12Gb/s
- 1.8GB/s read
- 1.0GB/s write
- 250K random read (4K)
- 37K random write (4K)
- 100 µs access latency

DVDStore2 - Database only - CNS Spec #2

Checking other resources for bottlenecks: SSD utilization on CNS nodes

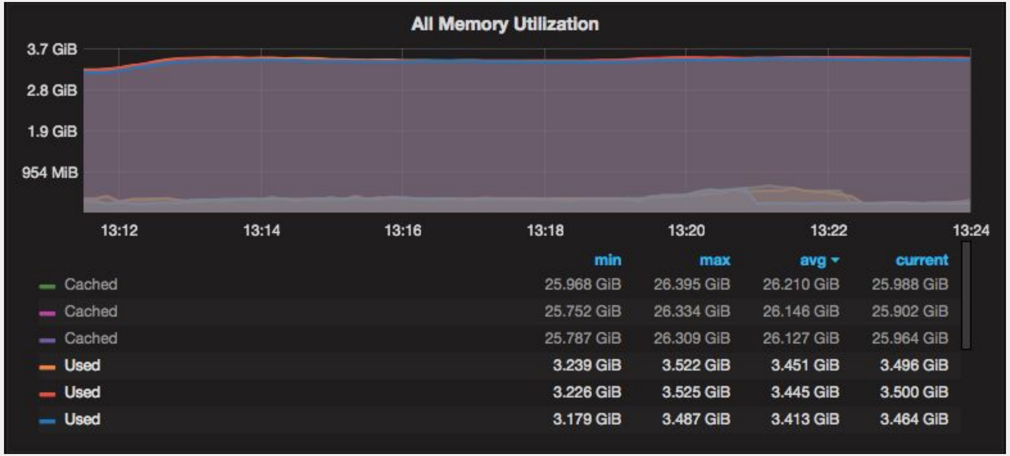


5x HGST Ultrastar SS200 SSD

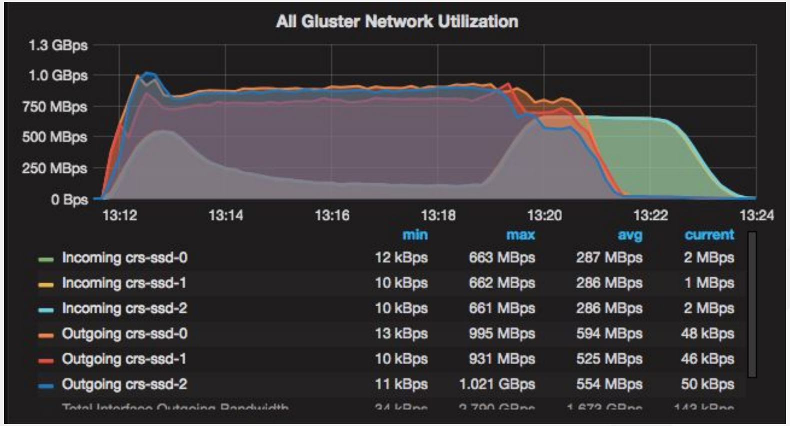
- SAS 12Gb/s
- 1.8GB/s read
- 1.0GB/s write
- 250K random read (4K)
- 37K random write (4K)
- 100 µs access latency

DVDStore2 - Database only - CNS Spec #2

Checking other resources for bottlenecks: memory and network utilization on CNS nodes with SSDs



10GbE starts to become a bottleneck



Adding memory to CNS nodes helps with caching, less relevant with SSDs

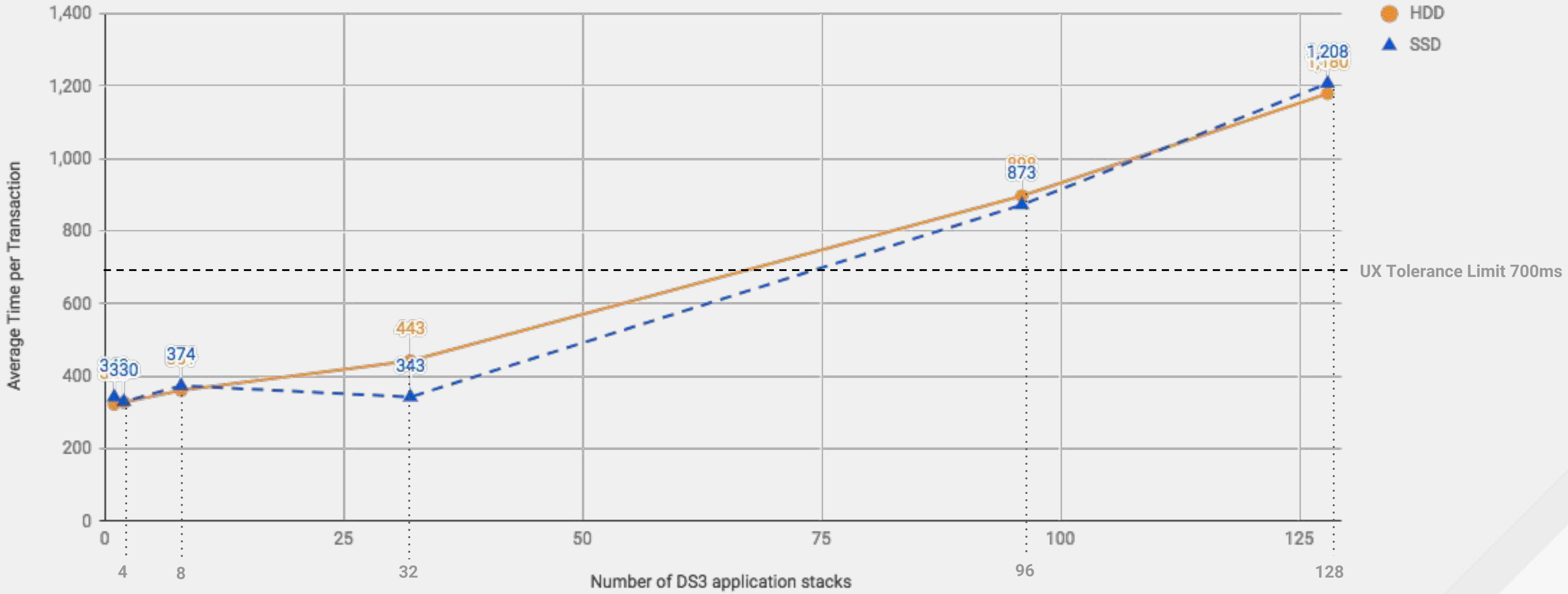
VERDICT

- A 3 node CNS cluster runs 128 busy MySQL database instances very well!
- SSDs outperform HDDs for database-heavy loads, HDD not within UX tolerance
- Watch resource utilization, adding CPU to CNS nodes will have the biggest impact
- Transactional MySQL workload actually becomes more sequential on the backend

DVDStore3 - Driving the whole stack

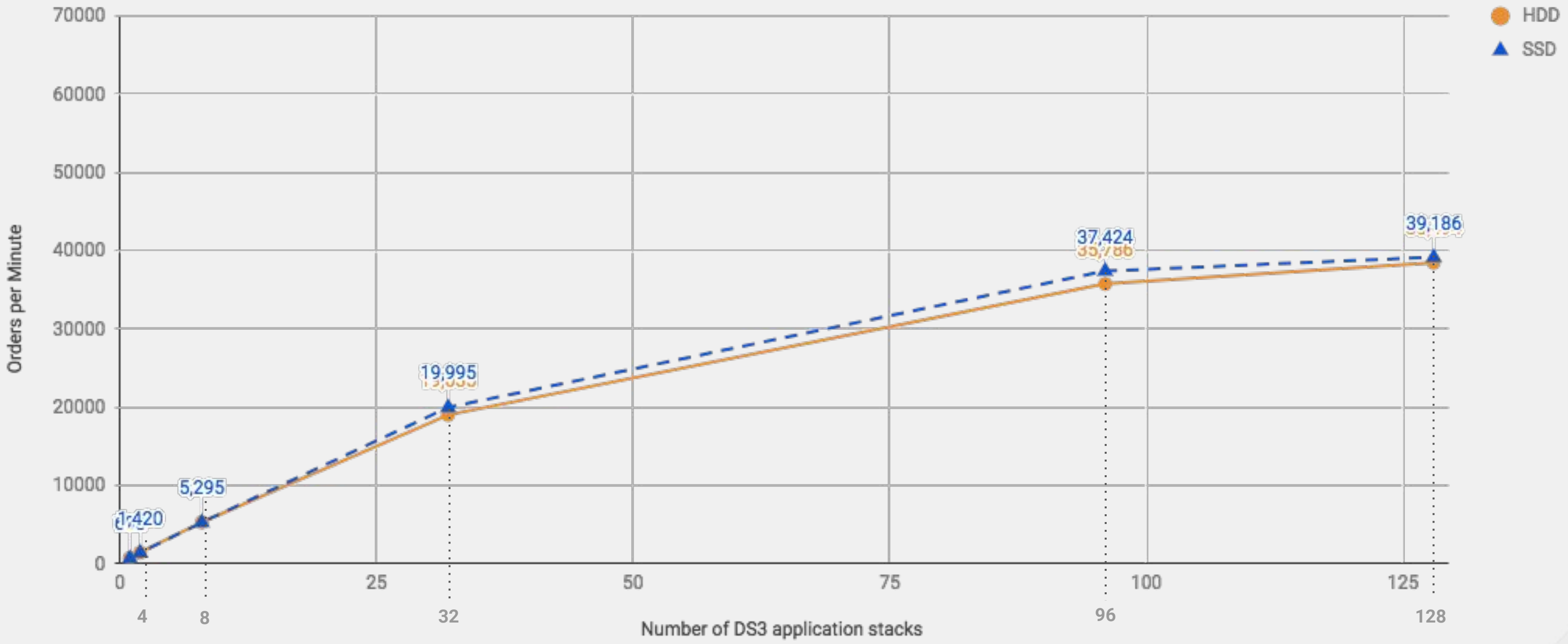
DVDStore3 - LAMP Stack - CNS Spec #2

Average Transaction Response Time



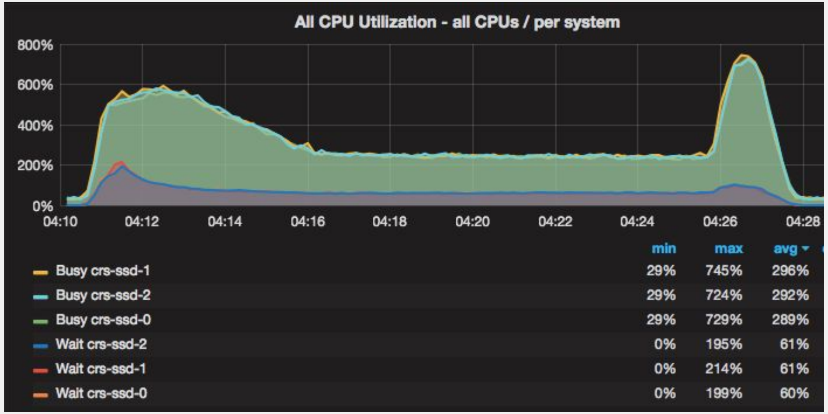
DVDStore3 - LAMP Stack - CNS Spec #2

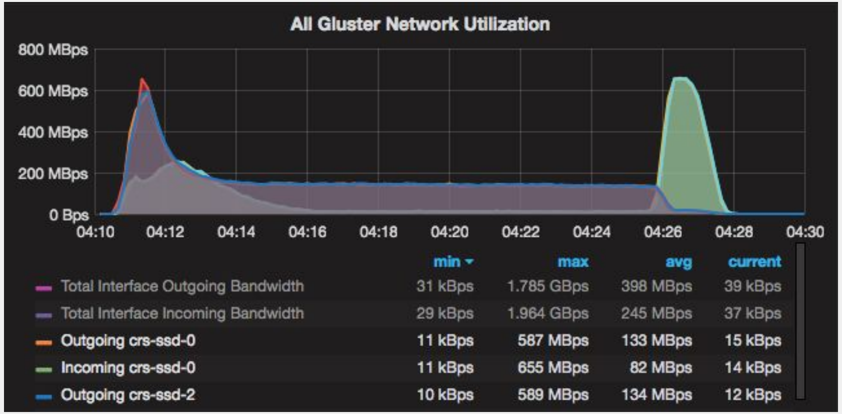
Aggregate Order Throughput results

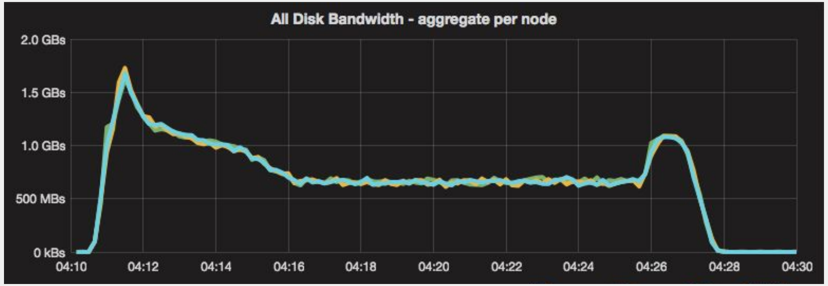


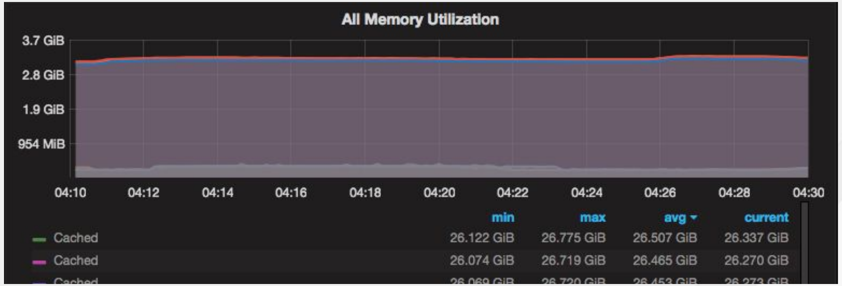
DVDStore2 - Database only - CNS Spec #2

No obvious bottlenecks on CNS nodes



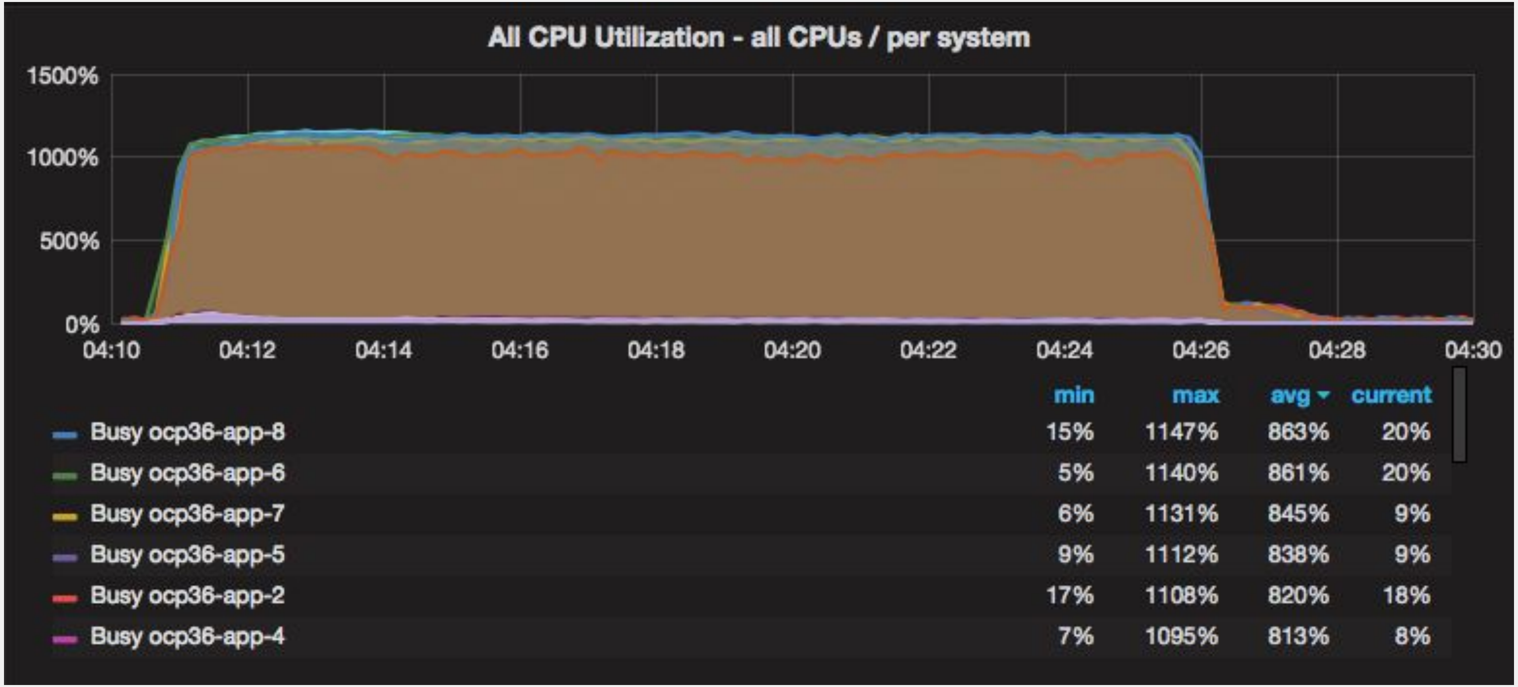






DVDStore3 - LAMP Stack - CNS Spec #2

CPU bottlenecks on Application nodes



VERDICT

- ➔ Database-access and query optimization lead to shift in the bottleneck from DB disk to App CPU
- ➔ SSDs do not provide significant performance gain versus HDD
- ➔ Watch resource utilization, adding CPU to application nodes will have the biggest impact
- ➔ CNS resource entitlement could be scaled down to 6-8 vCPUs and 8GB RAM

SUMMARY

LEADING QUESTIONS:

- **What is the recommended setup to achieve good performance?**
  - Use SSDs and separate networks for IO-heavy/DB-centric workloads
  - Know your workload!
- **How many of these web application stacks can we run?**
  - 128 MySQL instances under an OLTP workload on SSDs and 16 vCPU CNS nodes
- **How does workload and cluster performance scale as we increase load?**
  - CPU and Network utilization increases on DB heavy workloads
- **Will this kind of workload need SSDs or HDDs?**
  - Yes for DS2, no for DS3

SO WILL THIS RUN MY DATABASE?

MOST LIKELY YES!

GET THE FULL WHITEPAPER:

https://red.ht/cns-mysql-performance-paper

RED HAT SUMMIT

THANK YOU

plus.google.com/+RedHat

linkedin.com/company/red-hat

youtube.com/user/RedHatVideos

facebook.com/redhatinc

twitter.com/RedHat

WHAT’S NEXT?

TESTING BLOCK-STORAGE FROM CNS

a.k.a gluster-block