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](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

HPE ProLiant DL380 Gen9 LFF
- 2x Intel E5-2697A v4
- 256 GB of RAM
- 5x HGST Ultrastar SS200 1.9T
- HPE SmartHBA H240ar

HPE ProLiant DL380 Gen9 SFF
- 2x Intel E5-2697 v4
- 256 GB of RAM
- 12x HGST Ultrastar He 10TB
- HPE SmartArray P440ar

OpenShift Network (10 GbE)
Storage Network (10GbE)

VMware vSphere 6.0
OpenShift Setup

OpenShift Workers

OpenShift Infrastructure + Workers

HDD Storage Nodes

OpenShift Network (10 GbE)

Storage Network (10GbE)

VMware vSphere 6.0
OpenShift Setup (HDD Test)

OpenShift Workers

- Worker VM Specs:
  - 12 vCPUs
  - 96 GB of RAM
  - 2x 10GbE NICs

Support VM Specs:

- 2 vCPUs
- 8 GB of RAM

OpenShift Infrastructure + Workers

- Master VM Specs:
  - 8 vCPUs
  - 32 GB of RAM
  - 2x 10GbE NICs

- Infra VM Specs:
  - 2 vCPUs
  - 8 GB of RAM

HDD Storage Nodes

- CNS VM Specs v1:
  - 4 vCPUs
  - 8 GB of RAM

- CNS VM Specs v2:
  - 16 vCPUs
  - 32 GB of RAM

Worker VM Specs:

- 12 vCPUs
- 96 GB of RAM
- 2x 10GbE NICs

Support VM Specs:

- 2 vCPUs
- 8 GB of RAM

Master VM Specs:

- 8 vCPUs
- 32 GB of RAM
- 2x 10GbE NICs

Infra VM Specs:

- 2 vCPUs
- 8 GB of RAM

Utilities

- Grafana
- HAProxy

VMware vSphere 6.0

#redhat #rhsummit
OpenShift Setup (SSD Test)

OpenShift Infrastructure + Workers

Master VM Specs:
- 8 vCPUs
- 32 GB of RAM
- 2x 10GbE NICs

Infra VM Specs:
- 2 vCPUs
- 8 GB of RAM

SSD Storage Nodes

CNS VM Specs v1:
- 4 vCPUs
- 8 GB of RAM

CNS VM Specs v2:
- 16 vCPUs
- 32 GB of RAM

OpenShift Worker Nodes

Worker VM Specs:
- 12 vCPUs
- 96 GB of RAM
- 2x 10GbE NICs

Support VM Specs:
- 2 vCPUs
- 8 GB of RAM

VMware vSphere 6.0

OpenShift Network (10 GbE)

Storage Network (100GbE)
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

<table>
<thead>
<tr>
<th>DS2</th>
<th>DS3</th>
</tr>
</thead>
<tbody>
<tr>
<td>• stresses the database with lots of independent queries</td>
<td>• Optimized database access with newer client libraries, stored procedures</td>
</tr>
<tr>
<td>• known to be IO centric</td>
<td>• Known to be less demanding on the database</td>
</tr>
<tr>
<td>• “headless” test mode: load driver queries database directly</td>
<td>• Load driver accesses application via API (simulating end users)</td>
</tr>
<tr>
<td>• No product reviews</td>
<td>• 5 GB database</td>
</tr>
<tr>
<td>• 20 GB database</td>
<td>• Think time: 300 ms</td>
</tr>
<tr>
<td>• Think time: 10ms</td>
<td>• 6 concurrent users / application stack</td>
</tr>
<tr>
<td>• 8 concurrent users / database</td>
<td>• OCP stock mysql:5.7/php:5.6 images</td>
</tr>
<tr>
<td>• OCP stock mysql:5.7 images</td>
<td>OPM numbers are not comparable!</td>
</tr>
</tbody>
</table>
BENCHMARK RESULTS
DVDStore2 - Driving the RDBMS
DVDSStore2 - Database only - CNS Spec #1

Average Transaction Response Time

![Graph showing Average Transaction Response Time over Number of DS2 database instances. The graph compares HDD and SDD performance. The UX Tolerance Limit is 700ms.](image-url)
DVDStore2 - Database only - CNS Spec #1

Aggregate Order Throughput results

Number of DS2 database instances vs Orders per Minute

- HDD
- SDD

#redhat #rhsummit
DVDStore2 - Database only - CNS Spec #1

Investigating performance bottleneck - CPU utilization CNS nodes

![Graph showing CPU utilization over time with statistics for different CNS nodes.](image)
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!
Removing the bottleneck on CNS Nodes on SSD after going to 16 vCPUs
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
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
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

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

10GbE starts to become a bottleneck
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
Average Transaction Response Time

- HDD
- SSD

UX Tolerance Limit 700ms

Number of DS3 application stacks
DVDStore3 - LAMP Stack - CNS Spec #2

Aggregate Order Throughput results

- HDD
- SSD
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
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