SSD and Container Native Storage for High-Performance Databases

Earle F. Philhower, III
Sr. Technical Marketing Manager, Western Digital

August 2018
Databases ∩ Containers = Null Set?

\[ \text{Integral(VMs } dt) = \text{DevOps(Containers)} \]

Software Defined Storage + Containers > SAN

\[ \text{Lim Performance(CNS + SSD)} | DBs->128 \]

\[ \text{Databases / (SSD + Containers)} = \infty \text{ Possibilities} \]

* But there *is* a study guide available at:
Databases ∩ Containers = Null Set?

*Challenge: Databases in containers*

- Software ate the world. Containers ate software...

- But the databases that containers rely on are still on dedicated hardware.
  - Performance
  - Persistence

*Challenge: Integrate Containers and Database*

- Enable DevOps advances for databases
- Provide persistent, high performance storage
- Remove need for database system silos
Integral(VMs dt) = DevOps(Container)

From VMs to Containers and DevOps

• VMs (Virtual Machines)
  – VMware®, Hyper-V, KVM, Xen
  – Emulate entire hardware and run full software stack
    • Full Operating System and support software!
    • One VM host could have 100 copies of the Linux® kernel active
  – Per-VM limits on CPU, memory, network, I/O
  – Maximizes isolation of apps

• Containers
  – Split a single OS image into multiple domains (containers)
    • Only one kernel (Windows, Linux) active
  – Each application or executable can have CPU, mem, etc. limits
  – Maximizes density of apps / hardware
  – “DevOps” focused, very fast to deploy and manage
**SDS + Containers > SAN**

*Architecture for persistent storage in a unified cluster*

- Software defined storage system (SDS) for persistence
  - Keep storage control and management in-cluster
  - Remove need for external persistent storage
  - Keep control in DevOps’ hands

- Run everything under Virtualization
  - KVM, VMware (this example), public cloud, etc.
  - One software-defined storage VM/node with SSD connection
    - Ensure appropriate CPU, memory, networking resources
  - One container node with ephemeral and SDS connections
SDS + Containers > SAN²

*Red Hat® OpenShift and Ultrastar® SS200 Under VMware*

- **Software**
  - VMware vSphere® virtualization management
  - Red Hat OpenShift Container Storage (OCS)
    - SDS = GlusterFS, optimized for containers
  - Red Hat OpenShift Container Platform (OCP)
    - Container orchestration
  - Oracle MySQL™
    - Running DVDStore2 Test (part of VMmark™ testbed)
    - I/O intensive, transactional operations
  - 3 SDS VMs

- **Hardware**
  - 9 HPE ProLiant Servers
  - 15 Western Digital Ultrastar SS200 SAS SSD (5 per SDS VM)
  - (also tested with 36 Ultrastar He10 HDDs)
Container Native Storage + SSD = f(ast)

System Diagram

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

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

VMware vSphere
Red Hat OpenShift
Lim Performance (CNS + SSD) | DBs -> 128

Test Results

Transaction Time = 91ms
Databases / (SSD + Containers) = ∞ Possibilities

• Challenge: Integrate Containers and Database

• Solution: SDS with SSD in an integrated environment
  – Red Hat OpenShift Container Storage (OCS)
  – Red Hat OpenShift Container Platform (OCP)
  – Western Digital Ultrastar SSDs

• Benefits:
  – Enable DevOps for databases, with performance
  – Provide persistent, high performance storage
  – Remove need for database hardware silos
