Integrating Kubernetes Persistent Volumes into a Composable Infrastructure Platform

Brian Pawlowski
DriveScale Inc.
Agenda

- Composable Infrastructure
  - The problem with Scale-out Apps
  - DriveScale Solution
- Kubernetes and Scale-out Apps
- Container Storage Interface (CSI)
  - Persistent vs. non-persistent volumes
- DriveScale plug-in for Kubernetes
  - Feature summary
  - In operation
  - Failover/migration (wording…)
- Future
The Problem with Scale-out Applications

- Widely deployed scale-out apps (Cassandra, Hadoop, Spark, etc) are defined primarily as local DAS reference architectures
  - Scale compute/storage together
  - Use commodity platforms
- Issues arise in practice
  - Getting the compute/storage ratio right is hard – leads to overprovisioning
  - Coupled lifetimes lead to unwanted tech refresh
- What about legacy enterprise storage?
  - Aggregate/virtualize media behind small number of network interfaces, introducing performance issues
Scale-Out Applications
Scale-out Applications Recovery

- Applications like Cassandra can withstand the loss of a node with local storage
  - Triggers a rebuild of the data from the replicas
- Composable Infrastructure (DriveScale) lets you separate the storage from the compute and reattach the data to a new node
  - Avoiding rebuild time and overhead
- Disaggregated storage for persistent volumes brings this capability to Kubernetes
Terminology
## Terminology

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSI</td>
<td>The Container Storage Interface provides a standard interface for any Container Orchestration systems (like Kubernetes) to expose arbitrary storage systems to their container workloads.</td>
</tr>
<tr>
<td>Volume</td>
<td>A unit of storage made available inside of a CO-managed container, via the CSI. Ephemeral or Persistent.</td>
</tr>
<tr>
<td>Persistent Volume</td>
<td>A volume that persists beyond the lifetime of a pod.</td>
</tr>
<tr>
<td>Block Volume</td>
<td>A volume that will appear as a block device inside the container. (not supported currently by DriveScale)</td>
</tr>
<tr>
<td>Mounted Volume</td>
<td>A volume that will be mounted using the specified file system and appears as a directory inside the container.</td>
</tr>
<tr>
<td>FlexVolume</td>
<td>Obsolete Kubernetes-specific volume API preceding CSI.</td>
</tr>
<tr>
<td>CO</td>
<td>Container Orchestration system, communicates with Plugins using CSI service RPCs.</td>
</tr>
<tr>
<td>SP</td>
<td>Storage Provider, the vendor of a CSI plugin implementation.</td>
</tr>
<tr>
<td>Drive</td>
<td>Any of HDD, SSD, or SSD slice</td>
</tr>
<tr>
<td>HDD</td>
<td>Hard Disk Drive (spinning disk)</td>
</tr>
<tr>
<td>SSD</td>
<td>Solid State Drive or Device</td>
</tr>
<tr>
<td>SSD slice</td>
<td>A right-sized virtual slice of an SSD</td>
</tr>
<tr>
<td>RAID</td>
<td>Redundant Array of Independent Drives (RAID 0 – striped, RAID 1 – mirror, RAID 5 and 6 – parity)</td>
</tr>
<tr>
<td>Pod</td>
<td>Group containers, share an IP address, have a Kubelet agent</td>
</tr>
<tr>
<td>Kubelet</td>
<td>An agent that runs on each node in the cluster. It makes sure that containers are running in a pod</td>
</tr>
<tr>
<td>gRPC</td>
<td><a href="https:">Google Remote Procedure Call</a>]</td>
</tr>
<tr>
<td>Node</td>
<td>A host where the user workload will be running, uniquely identifiable from the perspective of a Plugin by a node ID.</td>
</tr>
<tr>
<td>Plugin</td>
<td>Aka “plugin implementation”, a gRPC endpoint that implements the CSI Services.</td>
</tr>
<tr>
<td>Plugin Supervisor</td>
<td>Process that governs the lifecycle of a Plugin, MAY be the CO.</td>
</tr>
<tr>
<td>Workload</td>
<td>The atomic unit of “work” scheduled by a CO. A container or a collection of containers.</td>
</tr>
</tbody>
</table>
Enter Kubernetes
Kubernetes

- <fill in intro>
## History of Kubernetes

<table>
<thead>
<tr>
<th>Year</th>
<th>Events</th>
</tr>
</thead>
<tbody>
<tr>
<td>Old Days</td>
<td>Google Borg, etc.</td>
</tr>
<tr>
<td>2008</td>
<td>• cgroups introduced to mainstream Linux</td>
</tr>
<tr>
<td>2014</td>
<td>• Extreme-early adopters of Google Kubernetes</td>
</tr>
</tbody>
</table>
| 2015 | • CNCF launches, Google’s managed Kubernetes GKE opn K8s to a wider community  
       • Red Hat’s OpenShift launches |
| 2016 | • Focus moves to other clouds or on prem to adopt Kubernetes |
| 2017 | • Managed Kubernetes begins to appear on AWS and Azure  
       • Docker and Mesosphere announce support for Kubernetes  
       • Kubernetes becomes the de facto standard  
       • Production deployments at scale begin |
| 2018 | • Additional managed services (DigitalOcean, Oracle)  
       • Investment shifts to the lifecycle management in Kubernetes |
Kubernetes Architecture

Kubernetes Architecture (DELETE)
Container Storage Interface (CSI)

- Consistent, orchestration independent volume mgmt API
- Dynamic provisioning and deprovisioning of a volume.
- Attaching or detaching a volume from a node.
- Mounting/unmounting a volume from a node.
DriveScale in Three Slides
Evolution of Workloads

Traditional Workloads
Many apps managed on each server with shared enterprise storage

- vSphere
- Shared NAS or SAN Storage

Data Intensive Workloads
Apps running on clusters of servers with local storage

- VMs
- NAS
- SAN

- Hadoop
- Cassandra
- Spark
- Kubernetes
Rethink data center architecture

Data Intensive Workloads

Apps running on clusters of servers with local storage

- Tightly coupled, fixed resources
- Commonly over provisioned
- Stranded compute and storage resources
- Server SKU sprawl
- Lifecycles tied together
The DriveScale Composable Platform

DriveScale Composer

- Policy
- Orchestration
- Monitoring

Automated creation of logical servers
Automated end-to-end set up for NVMe/TCP, NVMe/RDMA or iSCSI
Patented load balancing

ToR Ethernet switches
10G, 25G, 100G, 400G

Diskless servers (boot drive)
DriveScale Server Agent

eBODs, JBODs (Flash, HDD)
DriveScale Adapter Software

Each resource treated as independent from its enclosure
DriveScale CSI Plug-in
Overview of DriveScale CSI

- Scalable Persistent Storage for Containers
  - Location Aware (performance and failure domain optimization)
  - Shared Nothing (maintains local drive benefits)
  - Enterprise scale (10,000’s compute, 100,000s drives)
- Direct Attached performance
  - Deliver native performance of the devices to containers
  - Equivalent to applications running in Bare Metal servers
- Data locality for Containers
  - Delivers Data Locality/IO scalability/performance for containerized applications
- Logical connection between drives and containers that can move with containers
  - Storage and Data Persistence
  - Transparent to applications running in containers
  - Container mobility critical to deliver efficiency gains (avoid copies and rebuilds)
Kubernetes on DriveScale

DriveScale Domain

Kubernetes

K1

K2

K3

Inventory
Kubernetes on Drivescale

- A Kubernetes cluster maps to a single DriveScale cluster
  - Kubernetes is the software/container orchestrator
  - DriveScale is the hardware/physical orchestrator
- Multiple Kubernetes clusters can run in a DriveScale domain (data center)
  - DriveScale clusters are created dynamically when Kubernetes requests PVs
Dynamic PV Lifecycle with DriveScale

CreateVolume → Controller → Node → Node
  | PublishVolume  | Stage Volume  | Publish Volume
  |               |              |
DeleteVolume → Controller → Node
  | UnpublishVolume |
  |               | Unstage Volume
  |               | Unpublish Volume
Creating a DriveScale Volume

Kubernetes CSI

CreateVolume

Controller
PublishVolume

Node
Stage Volume

Node
Publish Volume

DriveScale API

• Add Logical Node
• Add Drives
• Map Drives to Logical Node

Result

• 1 Logical Node ↔ 1 PV

• Add Server
• Map Server to Logical Node
Creating a DriveScale Volume

- **create volume** (only needed when the PV does not exist) - create a logical node and assign drives to it based on request (can include topology info). (First PV also creates a DriveScale cluster)
- **controller publish volume** (runs from outside the destination node - so that it’s possible to actually remove a drive from a node if the node is down) - will attach the target node to the volume logical node. The drives are visible as block devices, they are formatted, encrypted block device is created if using encryption but not mounted.
- **node stage volume** (runs on the node - called once per node using the volume) - will mount the volume in a common location (does nothing if using a block volume rather than mount volume). The mount operation is done by the CSI driver, not through the API (because the mount is supposed to happen inside the container the CSI driver runs in which the agent has no visibility on)
- **node publish volume** (runs on the node - called once per pod using the volume - note there can be multiple containers in a pod) - will bind mount from the common location to the pod specific mount point. For block volumes, it will create a block device file corresponding to the drive in the right location. Here as well the mount operation / mknod is done by the CSI driver for the same reason as in the previous step
Types of DriveScale PVs

- HDD
- SSD
- RAID 0 HDD or SSD
- SSD slices
Scale-out Arguments for DriveScale PVs

- <Discuss target applications and their storage>
- Subhead
- Example 1
- Example 2
- Subhead
- Subhead
  - Example 1
  - Example 2
- Subhead
<add picture extending Kubernetes architecture to show PVs and DriveScale composer>
Creating a DriveScale Volume (APIs)

- <show the APIs invoked in creation process>
Avoiding copies or rebuilds

- <Motivation for DS PV support>
Migrating a DriveScale Volume

- <Show the steps taken to move a DS volume>
Failover a DriveScale Volume

- <Show the steps taken to failover a DS volume>
Failover a DriveScale Volume

- <graphic comparison of performance hit>
Future Work

- <What will we be doing in future for plug-in>
Backup
References

- https://github.com/container-storage-interface/spec/blob/master/spec.md
- https://kubernetes.io/docs/concepts/storage/volumes/#csi
- https://kubernetes-csi.github.io/docs/
- https://github.com/DriveScale/k8s-plugins
- Volumes: https://kubernetes.io/docs/concepts/storage/volumes/
- https://kubernetes.io/docs/concepts/storage/persistent-volumes/
- https://kubernetes.io/docs/concepts/workloads/pods/pod/
- https://grpc.io/blog/principles/
- Children’s Guide to Kubernetes: https://www.youtube.com/watch?v=4ht22ReBjno
- Kubernetes: https://en.wikipedia.org/wiki/Kubernetes
- Persistent volumes by example: http://kubernetesbyexample.com/pv/
DriveScale makes it easy to adopt new technologies

- NVMe over fabrics (TCP, RDMA, iSCSI)
- Accelerators: GPUs and FPGAs
- Scale-out micro services, cloud-native applications
- Faster processors
- Latest drive technology
Gain the transformative economics of composable

- 50% lower cost than legacy and cloud
- 69% less data center footprint
- 44% savings for upgrades
- 2 minutes to deploy infrastructure
- No more overprovisioning
• Bullet one
  • Bullet two
    • Bullet 3
      • Bullet 4