Running Docker on Lustre

An architectural overview

Blake Caldwell
OLCF/ORNL

LUG 2016
Portland, Oregon
April 6, 2016

ORNL is managed by UT-Battelle
for the US Department of Energy

OAK RIDGE National Laboratory | OAK RIDGE LEADERSHIP COMPUTING FACILITY

About Docker

- What is it?
  - A toolset for packaging, shipping, and running containers (user environment)
- What is it good for?
  - Consistent user environments
    - Rapid prototyping, proof of concepts (development)
    - Reproducible research
  - Application isolation
  - Server consolidation

A Conversation on Image Distribution

HPC User:
`furious_mccarthy`

Docker Oracle:
`goofy_blackwell`

A Conversation on Image Distribution

HPC User:
`furious_mccarthy`

Docker Oracle:
`goofy_blackwell`

1. How do I run the same image on 50 different nodes?

A Conversation on Image Distribution

HPC User:
`furious_mccarthy`

1. How do I run the same image on 50 different nodes?

Docker Oracle:
`goofy_blackwell`

Push it to Docker Hub

A Conversation on Image Distribution

HPC User:
`furious_mccarthy`

1. How do I run the same image on 50 different nodes?
2. My images can't leave the local network

Docker Oracle:
`goofy_blackwell`

Push it to Docker Hub

A Conversation on Image Distribution

HPC User: `furious_mccarthy`	**Docker Oracle:** `goofy_blackwell`
1. How do I run the same image on 50 different nodes?	Push it to Docker Hub
2. My images can't leave the local network	Create a private repository

A Conversation on Image Distribution

HPC User:
`furious_mccarthy`

Docker Oracle:
`goofy_blackwell`

1. How do I run the same image on 50 different nodes?

   Push it to Docker Hub

2. My images can't leave the local network

   Create a private repository

3. I have a lot of compute nodes and 1 registry (bottleneck)

A Conversation on Image Distribution

HPC User: `furious_mccarthy`	**Docker Oracle:** `goofy_blackwell`
1. How do I run the same image on 50 different nodes?	Push it to Docker Hub
2. My images can't leave the local network	Create a private repository
3. I have a lot of compute nodes and 1 registry (bottleneck)	Load balance the registries

A Conversation on Image Distribution

HPC User:
`furious_mccarthy`

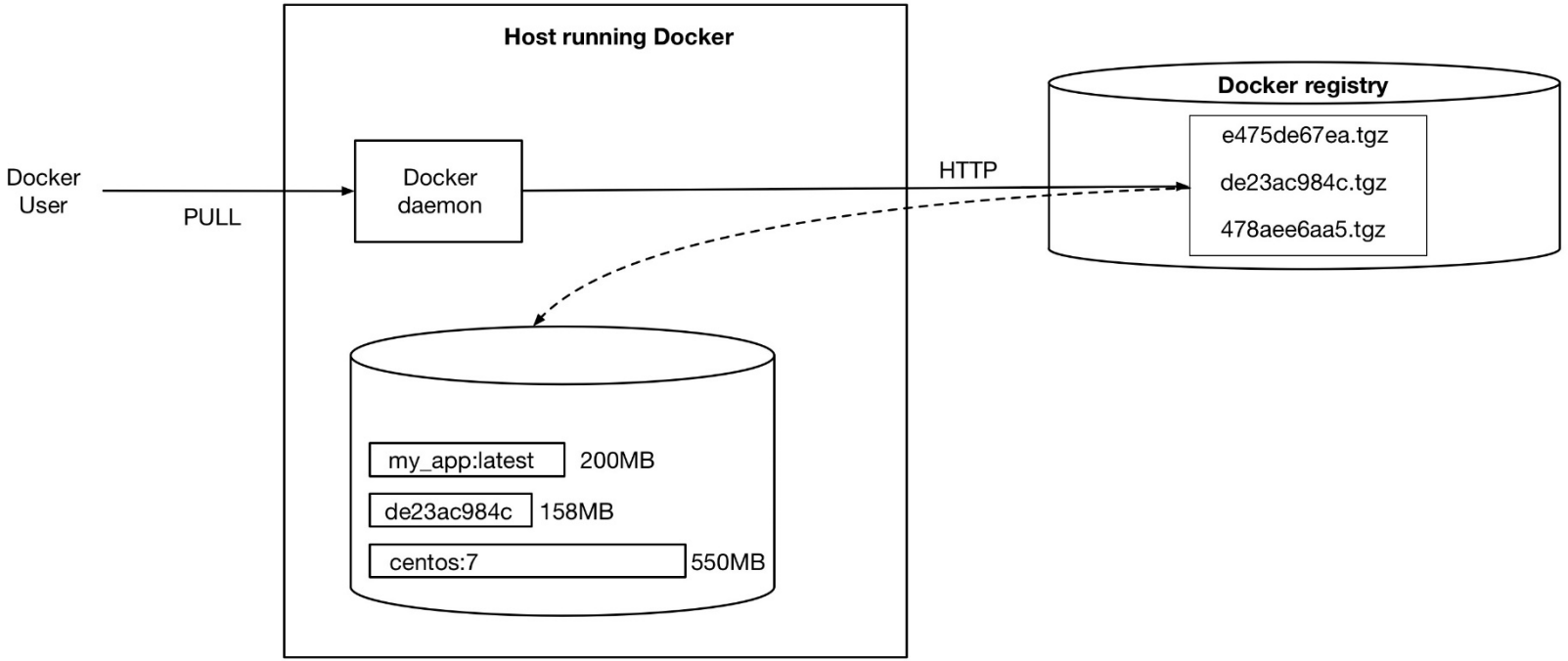
Docker Oracle:
`goofy_blackwell`

1. How do I run the same image on 50 different nodes?
   Push it to Docker Hub
2. My images can't leave the local network
   Create a private repository
3. I have a lot of compute nodes and 1 registry (bottleneck)
   Load balance the registries
4. The images are inconsistent!

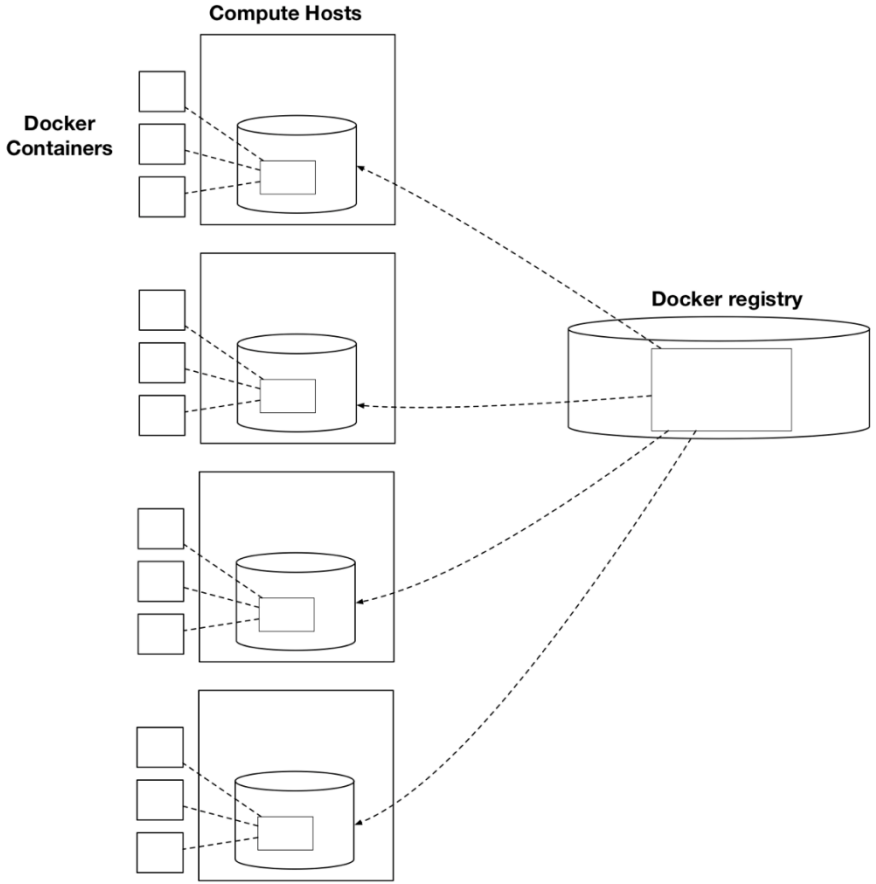
A Conversation on Image Distribution

HPC User: `furious_mccarthy`	**Docker Oracle:** `goofy_blackwell`
1. How do I run the same image on 50 different nodes?	Push it to Docker Hub
2. My images can't leave the local network	Create a private repository
3. I have a lot of compute nodes and 1 registry (bottleneck)	Load balance the registries
4. The images are inconsistent!	Redeploy! Cattle vs. pets…

Normal Docker Pull



Parallel Docker Pull



Why Does Docker Need a Distributed Image Store?

- Deployment means waiting on disk I/O
- Copies are everywhere!
- Consistency
- Security

Why Lustre?

- A shared, persistent, filesystem already present in many cluster computing environments
- We're addressing the speed of Docker when using the same image across many nodes in parallel

Docker Images vs. Volumes

- Images: the base filesystem image of the container (*chroot*)
  - Stored in Docker registries (push/pull)
  - Made up of layers (copy-on-write)

Docker Images vs. Volumes

- Images: the base filesystem image of the container (*chroot*)
  - Stored in Docker registries (push/pull)
  - Made up of layers (copy-on-write)
- Volumes: filesystem mounts added at container creation time
  - Bind-mounts from host
  - Plugins exist for volumes on distributed storage (Ceph, Gluster, S3)
    - No Lustre volume driver

Docker Images vs. Volumes

- Images: the base filesystem image of the container (*chroot*)
  - Stored in Docker registries (push/pull)
  - Made up of layers (copy-on-write)
- Volumes: filesystem mounts added at container creation time
  - Bind-mounts from host
  - Plugins exist for volumes on distributed storage (Ceph, Gluster, S3)
    - No Lustre volume driver
- What options exist for storing images on Lustre…

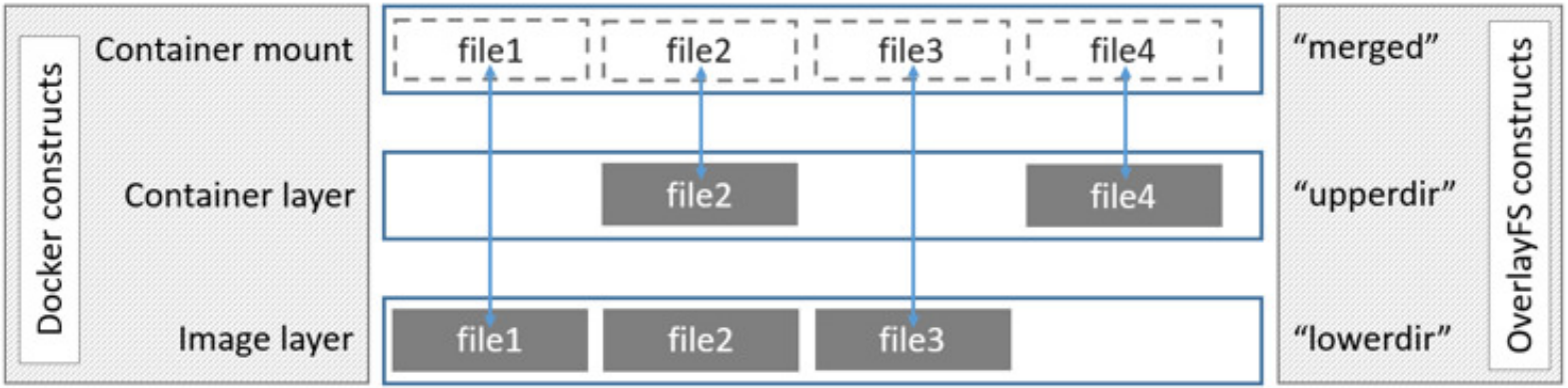
Devicemapper + Loopback

- The dm-loopback implementation is Docker’s fallback storage driver
  - Devicemapper in RHEL, Ubuntu, SLES
  - No block device configuration required
  - Thinp snapshots are copy-on-write
- Metadata operations are handled on VFS locally
- But it’s quite slow
  - Jason Brooks – Friends Don't Let Friends Run Docker on Loopback in Production
    http://www.projectatomic.io/blog/2015/06/notes-on-fedora-centos-and-docker-storage-drivers/

OverlayFS

- Upstream since Linux 3.18
  - Hasn’t always supported distributed file systems
- Presents a union mount of one or more r/o layers and one r/w layer
  - Layers are directories
  - Modified files are copied up

OverlayFS Union Mounts



https://docs.docker.com/engine/userguide/storagedriver/overlayfs-driver/

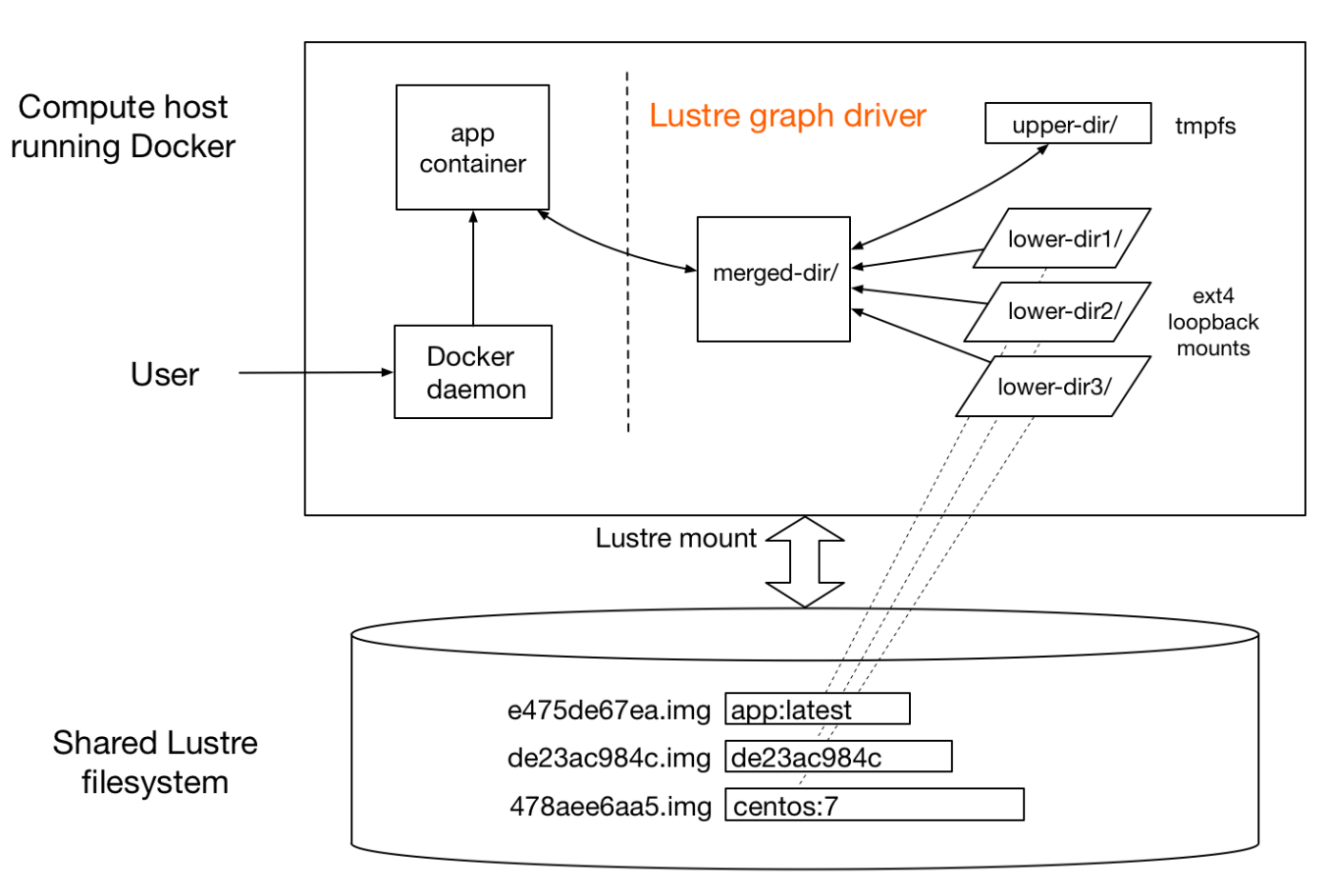
OverlayFS

Pros:

+ Page cache entries shared for all containers
+ Natively supports copy-on-write

- Cons:
  - Copy-up penalty on write
  - Docker’s implementation uses hard links for chaining image layers
  - FS-only so lots of files, and metadata operations

OverlayFS + Loopback



OverlayFS + Loopback: Implementation

- https://github.com/bacaldwell/lustre-graph-driver

Conclusions

- Loopback devices on Lustre could support cluster computing workloads
  - No image pulls, just run
  - Read-only layers on filesystem
  - Ephemeral layers node-local
- Work remains
  - Upstream Docker overlayfs driver with multiple lower layers
  - Loopback device performance (LU-6585 lloop driver)

Resources

- Jeremy Eder – Comprehensive Overview of Storage Scalability in Docker
  http://developers.redhat.com/blog/2014/09/30/overview-storage-scalability-docker/
- Jérôme Petazzoni – Docker storage drivers
  http://www.slideshare.net/Docker/docker-storage-drivers
- Reproducible environments
  http://nkhare.github.io/data_and_network_containers/storage_backends/
  https://github.com/marcindulak/vagrant-lustre-tutorial

Questions...

Pull to Lustre