dCache Ceph Integration
Paul Millar for dCache Team

ADC TIM at CERN
2016-06-16
https://indico.cern.ch/event/438205/

Many slides stolen from donated by Tigran Mkrtchyan
dCache as Storage System

- Provides a single-rooted namespace.
- Metadata (namespace) and data locations are independent.
- Aggregates multiple storage nodes into a single storage system.
- Manages data movement, replication, integrity.
- Provides data migration between multiple tiers of storage (DISK, SSD, TAPE).
- Uniquely handles different Authentication mechanisms: X.509, Kerberos, username+password, OpenID-Connect.
- Provides access to the data via variety of access and management protocols (WebDAV, NFSv4.1/pNFS, xxxFTP, xrootd, DCAP, SRM).
dCache as Storage System

- Provides a single-rooted namespace.
- Metadata (namespace) and data locations are independent.
- Aggregates multiple storage nodes into a single storage system.
- Manages data movement, replication, integrity.
- Provides data migration between multiple tiers of storage (DISK, SSD, TAPE).
- Uniquely handles different Authentication mechanisms: X.509, Kerberos, username+password, OpenID-Connect.
- Provides access to the data via variety of access and management protocols (WebDAV, NFSv4.1/pNFS, xxxFTP, Xrootd, DCAP, SRM).
Road Map

a story in three phases
Storage in dCache (what we have now)

- dCache services
  - Namespace, PoolSelection, protocol support, Authn/Authz
- Replication/Migration
- Pool service
  - Block device
- Pool service
  - Block device
- Pool service
  - Block device
Phase 1: abstracting from “block device”

- Each pool has its own independent 'partition'
- Each 'partition' attached to its own block device
Phase 2: quorum storage

- Any pool can deliver data
- Object store takes care of replication
Phase 3: mixed deployment

- Support multiple islands: a CEPH cluster or regular block device,
- dCache can move data between islands.
Not only CEPH!

• Other object store can be adopted
  DDN WOS, Swift, S3, CDMI, …

• Work will also add support for cluster file systems:
  Luster, GPFS, GlusterFS, …
Current status: phase 1

- **Functional prototype** only
  - no performance testing, not for production
- Focus on **stability and functionality** first
  - all dCache features must be available
- Based on **RADOS Block Device** (RDB):
  - supports striping, alterable content, resilience, placement.
  - Each dCache pool is a CEPH pool.
  - Each dCache file is a RBD image.
- **Object interface** will be evaluated as well
Next steps...

• Phase 1
  • functional prototype is **being tested by sites**
  • **HSM integration** changes being evaluated,
  • Preparing for **regular code-review**,
  • Will be part of **dCache release**.

• Phase 2 & 3
  • Our priority depends on **user demand**,
  • Depends on **operational overhead**, if any
  • Also watching **support overhead**, if any
Backup slides
Pool internals

- cell communication
- mover queue
- flush queue

Data Mover

metadata

virtual repository

data repository
Pool internals

- cell communication
- mover queue
- flush queue

Data Mover

virtual repository

metadata

../meta

XFS/ext4

data repository

../data

POSIX IO

..../data

..../meta
Pool internals

- cell communication
- mover queue
- flush queue

Data Mover

virtual repository

metadata

POSIX IO

XFS/Ext4
data repository

../data

../meta
Pool internals

- cell communication
- mover queue
- flush queue

Data Mover

virtual repository

metadata

../meta

data repository

librados RDB
Why CEPH

• You ask for it!
• No specific hardware support
• Runs on commodity hardware
• Scalable to exabytes of data
• Deployed at sites as storage system for OpenStack
• Provides Object, Block and File interfaces
How the prototype works

- Pool still has `meta` directory.
- IO handled with CEPH RBD interface
- Each dCache pool has a corresponding CEPH pool
  - resilience
  - placement group
- Each dCache file is a `rbd image` in CEPH
CEPH (extremely simplified)

- OSD ~ a physical disk
- CRUSH - determines how to store and retrieve data by computing data storage locations.
- RADOS - distributes objects across the storage cluster and replicates objects
- librados - provides low-level access to the RADOS service.

Diagram:
- APP
- RDB
- CEPH FS
- LIBRADOS
- RADOS
- CRUSH
- OSD
- OSD
- OSD
HSM script

- file:/path/to/pnfsid
  - hsm.sh put <pnfsid> rbd://ceph/...
  - hsm.sh get <pnfsid> rbd://ceph/...

- checksum command
  - hsm.sh checksum <pnfsid> rbd://ceph/...
dCache's data management

- Automatic migration
  - Tape/disk/disk
  - HotSpot detection
  - Permanent migration jobs
  - Checksumming on transfer
- Manual migration
- Data replication
  - multiple copies
  - same host/rack/site policy
Software-defined storage

• Abstraction of logical storage services and capabilities from the underlying physical storage systems

• Automation with policy-driven storage provisioning with service-level agreements replacing technology details.

• Commodity hardware with storage logic abstracted into a software layer.
Links

- https://www.dcache.org/
- https://en.wikipedia.org/wiki/Software-defined_storage
- http://ceph.com/
Storage in dCache (outsourcing, phase 1)

- dCache provides high level service
- Data replication and management core dCache service
- Each pool has its own 'partition' on shared storage
- Each 'partition' attached to its own block device

![Diagram of dCache services and block devices]
Phase 1 (changing IO layer)

- Single data server owns the data
- Single data server manages data
  - flush to tape
  - restore from tape
  - removal
  - garbage collection
Storage in dCache (outsourcing, phase 2)

- dCache provides high level service
- All pool see all 'partition' on shared storage
- Any pool can deliver data from any partition
- Object store takes care about replication
Phase 2 (Changing core philosophy)

- All data managed by 'quorum'
  - group decision who interact with tape
  - group decision who/when file is removed
  - File location is always 'known'
Storage in dCache (outsourcing, phase 3)

- dCache provides high level service
- dCache can move data between regular and OS pools
Phase 3 (mixed environment)

- Mixed setup
- Islands of storage servers
- Replication and data movement between islands