Ceph: An Open Source Object Store

Evan Harvey
Gustavo Rayos
Nick Schuchhardt

Mentors: David Bonnie, Chris Hoffman, Dominic Manno

LA-UR-15-25907
What is an Object Store?

- Manages data as objects
- Offers capabilities that are not supported by other storage systems
- Object Storage vs. Traditional Storage
What is Ceph?

- An object store and filesystem
- Open source and freely available
- Scalable to the Exabyte level
Basic Ceph Cluster

- Monitor Node
  - Monitors the health of the Ceph cluster
- OSD Node
  - Runs multiple Object Storage Daemons (One daemon per hard drive)
- Proxy Node
  - Provides an object storage interface
  - Can interact with cluster using PUT/GET operations
  - Provides applications with a RESTful gateway to the Ceph storage cluster
Basic Ceph Cluster

Data → Object

Client

PUT/GET

Proxy

Monitor

HDD / JBOD

OSD Node
But Why?

- Campaign Storage
- More reliable than other file systems
- POSIX compliant
- Scales better than RAID
- Cost efficient
Project Goals

• Build a Ceph storage cluster
  – 1 Monitor node
  – 6 OSD nodes (Around 20 OSD daemons each)
  – 3 proxy nodes

• Erasure coding profiles

• Single vs. Multiple proxies
Test Environment

- CentOS 6.6
- Ten HP ProLiant D380P Gen8 Servers
- Three Supermicro 847jbod-14 (45 disks each)
- Mellanox Infiniband 56 Gb/s
- Two SAS cards 6 Gb/s
  - 8 ports at 600 MB/s
- Four Raid cards 6 Gb/s
  - 8 PCI Express 3.0 lanes
Our Set Up

Ceph Object Store

PUT/GET

Client A  Client B  Client C

Monitor Node

Proxy Node

Proxy Node

Proxy Node

OSD  OSD  OSD  OSD  OSD  OSD

JBOD  JBOD  JBOD
Pools and PGs

OSD.1

OSD.2

OSD.3

OSD.4

OSD.5

OSD.6

Pool A
Rep Size = 3

Pool B
Rep Size = 3

Pool C
Rep Size = 2

Client

Objects

PGs
Pools and Placement Groups

- An object belongs to a single placement group
- Pools group placement groups
- Placement groups belong to multiple OSDs
CRUSH!

- Controlled Replication Under Scalable Hashing (CRUSH)
- Algorithm finds optimal location to store objects
- Stripes objects across storage devices
- On the OSDs
CRUSH (pgid) --> (osd1, osd2, ... osdn)
Erasure Coding

- High resiliency to data loss
- Smaller storage footprint than RAID
- Data is broken up into object chunks
- Striped across many hard drives
- K + M values used to stripe
- Various erasure profiles
Erasure Coding

Object name: \( \text{rbd} \)

Object content: ABCDEFGHIJKL

\[ K + M = 4 + 2 \]

Encoded(4,2)

Object name:
- \( \text{rbd} \)
- \( \text{rbd} \)
- \( \text{rbd} \)
- \( \text{rbd} \)
- \( \text{rbd} \)
- \( \text{rbd} \)

Chunk #:
- 1
- 2
- 3
- 4
- 5
- 6

Contents:
- ABC
- DEF
- GHI
- JKL
- UVW
- XYZ

OSD

OSD

OSD

OSD

OSD

OSD
Results

- Difficult to install and configure Ceph on CentOS 6.6
- Multiple proxies write faster than a single proxy
- Replicated profile was faster than the erasure coded profiles
- $K + M$ values did not significantly affect read and write speeds
Three Proxy Write Performance of Ceph

- **replicated (x2)**
- **jerasure (4+2)**
- **isa (4+2)**

**Pool Configuration**

**Object Size**
- 4 MB
- 8 MB
- 16 MB
- 32 MB
Single Proxy Sequential Read Performance of Ceph

MB/Sec

Object Size
4 MB
8 MB
16 MB
32 MB

replicated (x2)
jerasure (4+2)
isa (4+2)

Pool Configuration
Ceph Headaches

- Documentation is inaccurate
- Nodes **must** be configured in specific order
  - Monitor → OSDs → Proxies
- Ceph was unable to recover after hardware failure
- Could only use one out of the four Infiniband lanes
- Unable to read in parallel
Conclusion

- Ceph is difficult to install and configure
- Stability of Ceph needs to be improved
- Unable to recover from hardware failures during benchmarking
- Performance was promising
Future Work

• Investigate bottleneck of tests
• Further explore pool configurations and PG numbers
• Look into Ceph monitoring solutions
• Test differences between ZFS/BTRFS vs XFS/EXT4
Acknowledgements

- Mentors: David Bonnie, Chris Hoffman, Dominic Manno
- Instructors: Matthew Broomfield, assisted by Jarrett Crews
- Administrative Staff: Carolyn Connor, Gary Grider, Josephine Olivas, Andree Jacobson
Questions?

- Objects stores?
- Ceph and our object store?
- Installation and configuration?
- Pools and Placement groups?
- CRUSH?
- Erasure coding?
- $K + M$?