CEPH FILE SYSTEM

BY: MARIE LESLIE MELANIE PITTUMBUR

EMAIL: MARIE.PITTUMBUR@STUDENT.LUT.FI

03 JUNE 2015

COURSE: COMPUTING CLUSTERS, GRIDS & CLOUDS

COURSE AUTHOR: PROFESSOR ANDREY Y. SHEVEL

ITMO UNIVERSITY, RUSSIA

OUTLINE

- Introduction
- Basic Terminologies & concepts
- Features of Ceph File System
- Architecture of Ceph File System
- Ceph FS Fundamental Design Principles
  - Decoupled MetaData & Data Management
  - Dynamic Distributed MetaData Management
  - Reliable Autonomic Distributed Object Storage
- Client Operation
- Conclusion

INTRODUCTION

- Ceph created by Sage Weil as a PhD project in 2007.
- Ceph is a distributed file system that features: data replication and fault tolerance while maintaining POSIX compatibility.
- Foremost advantages: Excellent performance, Reliability, and Scalability for Petabytes scale, dynamic and distributed systems.
- It employs object-based storage & conventional hard disks are replaced with intelligent object storage devices (OSDs).
- Ceph has excellent I/O performance and scalable metadata management, supporting more than 250,000 metadata operations per second.

BASIC CONCEPTS & TERMINOLOGIES (1)

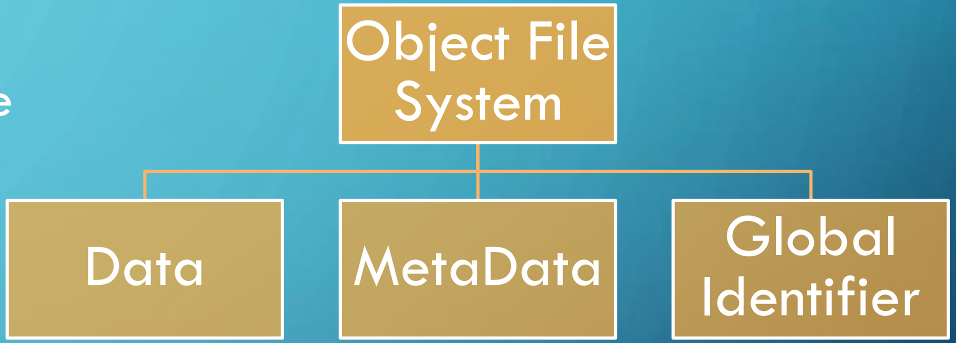
- Components of a file: MetaData, Mechanism to access & store the file & Data
- Filesystem finds out which blocks of disk space belongs to which file to append data and create new files.

User
---File System = Abstraction---
Data Blocks

- MS-DOS FAT FS: Allocation tables to store the location of the next block storing the data cluster of the file.
- Unix Fast FS: Uses Inode blocks to store all file metadata & references to data blocks
- Block-based file systems: Files are segmented into evenly sized blocks of data.
- Apart from block addresses, no context information about the file is provided

BASIC CONCEPTS & TERMINOLOGIES (2)

- Object-based file systems:
  - ❑Data for each file is stored in a single object
  - ❑MetaData is expandable and provides contextual about file
  - ❑Global identifier: To locate object over a distributed system



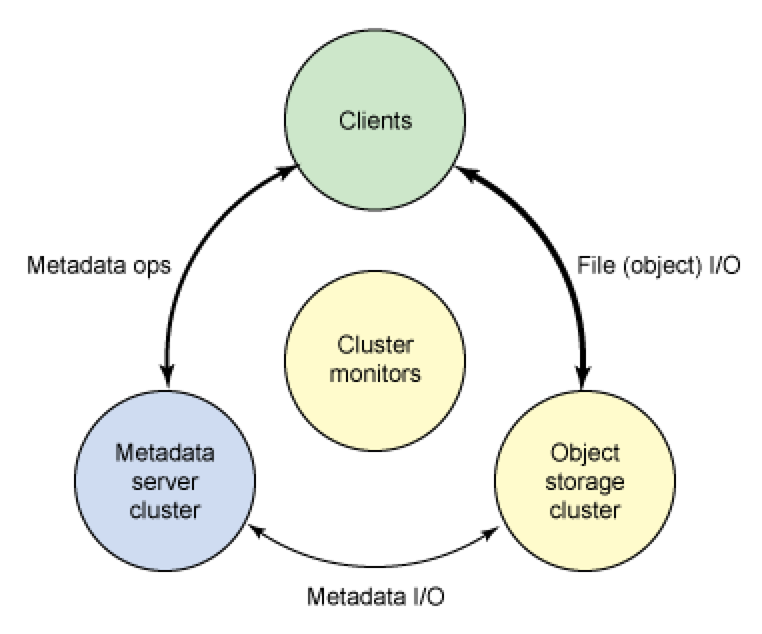
- MetaData servers perform metadata operations such as file open, file rename
- Low-level file I/O operations such as block allocation decisions for read & write operations are delegated to intelligent OSDs.
- Object based file systems are adapted to deal with data growth

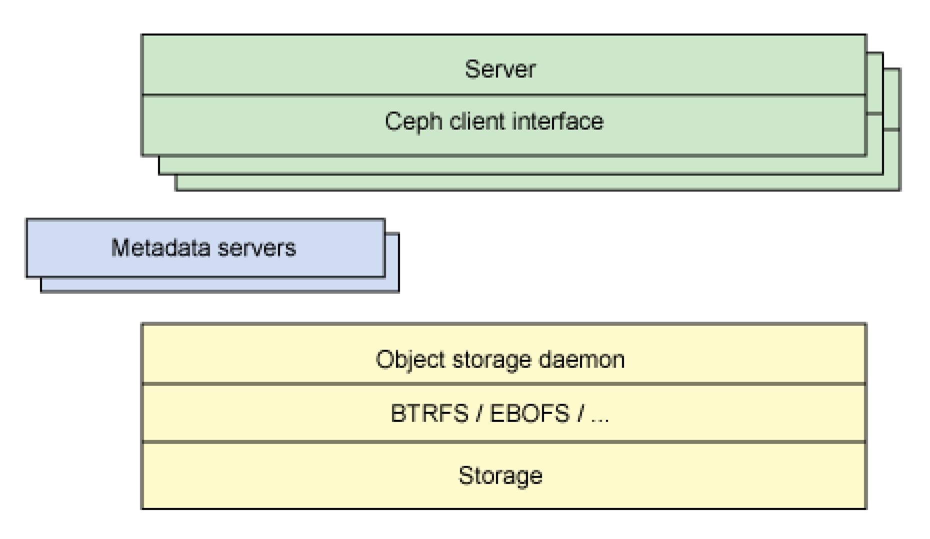
FEATURES OF CEPH FILE SYSTEM

- Primary goals driving design of Ceph File system:
  - ❑Scalability: Includes the overall storage capacity and throughput of the system
  - ❑Performance: Access to files or directories by clients
  - ❑Reliability: Self-healing and dynamic file system for no single point of failure
- Ceph maximizes decoupling of metadata & data management by eliminating allocation or inode lists. Data distribution algorithms used.
- Ceph provides extremely efficient metadata management and seamlessly adapts to various workloads for different computing requirements.
- By leveraging OSDs intelligence: Semi-autonomous, fault tolerant and recovering file systems

ARCHITECTURE OF CEPH FILE SYSTEM

- Components of Ceph File System:
  - ❑A client instance that exposes a POSIX file system interface to a host
  - ❑A cluster of OSDs storing both data and metadata
  - ❑A metadata cluster managing the namespace (file names & directories), security, consistency & coherence
  - ❑Cluster monitors: Manage the cluster map of the OSDs in case devices are added or removed.





CEPH FS FUNDAMENTAL DESIGN PRINCIPLES (1)

- **Decoupled MetaData & Data Management**
  - ❑Management of the metadata & storage of the actual file data is separated
  - ❑Long block lists (each of 512 bytes) are replaced with shorted object lists
- Unlike other object-based file system, Ceph eliminates any allocation or inode lists.
- File data is striped onto predictably named objects -> Boosting performance
- Uses random data distribution function, CRUSH to assign objects to storage devices.
- Through calculation any party can access the object’s name and location -> file contents

CEPH FS FUNDAMENTAL DESIGN PRINCIPLES (2)

- Dynamic Distributed Metadata Management
  - ❑Metadata operations take up about half the workload of filesystems
  - ❑Efficient management is critical to system performance
- Ceph metadata cluster architecture: Dynamic sub-tree partitioning -> Single authoritative MDS + Adaptive distribution of cached metadata across nodes
- Current Access patterns to objects are used to distribute workload among MDSs accordingly.
- Effective use of OSDs resources.
- Predict Scalability requirements in the future number of OSDs

CEPH FS FUNDAMENTAL DESIGN PRINCIPLES (3)

- **Reliable Autonomic Distributed Object Storage**
  - ❑Petabyte scale systems are highly dynamic and nodes fail regularly.
- Filesystem is implemented incrementally: new devices are added with time while old devices are removed.
- Data distribution has to be dynamic to adapt to availability of resources and to maintain appropriate level of data replication.
- Large volume of data constantly created, deleted or moved.
- Ceph FS benefits from increase in reliability and availability of storage: OSDs manage data migration, replication or recovery on their own.

CEPH CLIENT

- Client interface for Ceph file system incorporated into the Linux kernel (since 2.6.34)
- Abstraction of the underlying metadata servers, monitors, and individual object storage devices
- Client's point of view: Only a mount point to the user's filesystem which can be accessed for normal I/O operations.
- To run a ceph file system:

❑A running Ceph Storage cluster

❑A running Ceph metadata server

❑Mount the Ceph filesystem: Either as mounted device in /mnt/cephfs or using FUSE or directory in user's space using FUSE: /home/user/cephfs.

CONCLUSION

- A Comparison with other Large Scale Distributed Systems:

Large Scale systems	Parallel file & data systems	Metadata & Data decoupling systems
OceanStore & Farsite FS	Vesta, Galley & Swift	StorageTanks, GPFS
Offer Petabytes of reliable storage space	High transfer rates by data striping	Scalability limited by the use of block-based disks &
Poor file access performance due to use of allocation and inode lists for file name lookup	Reliability & Scalability issues due to lack of scalable metadata access & robust data distribution algorithms	Metadata & data distribution functions not sophisticated enough.

REFERENCES

- Sage A. Weil, Scott A. Brandt, Ethan L. Miller, Darrell D. E. Long, and Carlos Maltzahn. 2006. Ceph: a scalable, high-performance distributed file system. In Proceedings of the 7th symposium on Operating systems design and implementation (OSDI '06). USENIX Association, Berkeley, CA, USA, 307-320.
- http://www.ibm.com/developerworks/library/l-ceph/
- http://www.snia.org/sites/default/education/tutorials/2009/fall/file/CraigHarmer_Object-based_File_Systems_An_Overview.pdf
- http://ceph.com/docs/master/cephfs/