Gluster V/s Ceph

-Presented By : Rajeshwari Chatterjee

Professor-Andrey Shevel

Course: Computing Clusters Grid and Clouds

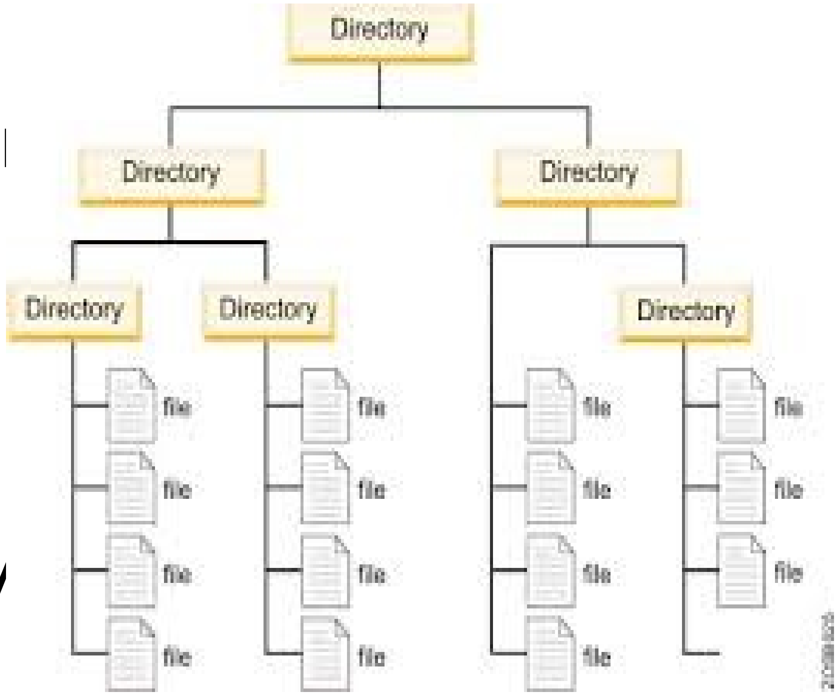
ITMO University, St. Petersburg

Contents

- Introduction File System
- Enterprise Needs
- Gluster –Revisited
- Ceph –Revisited
- Gluster and Ceph
- Results

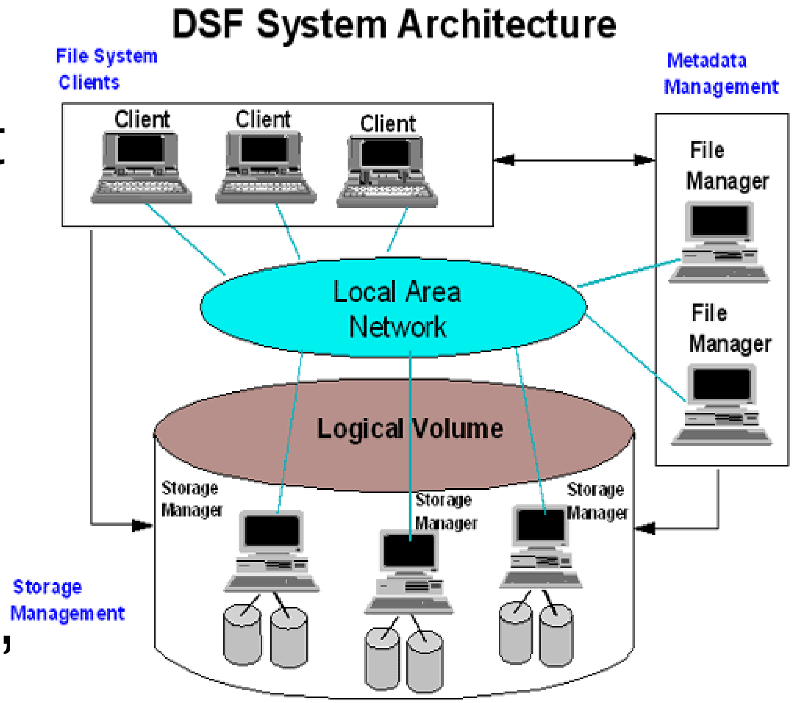
What is a Filesystem?

- It is organizing and storing files on a hard drive, flash drive or other storage devices
- Separates Data ,Provides Meta data information
- OS has own File system
- Differ in storage capacity speed, security,
- Ex:NTFS,FAT32,HFS,ext2, ext3,ext4,Btrf



What is a Distributed File System

- Files stored amongst one or more servers which can be accessed by remote clients wit proper authorization rights.
- Namespace, mapping Scheme to emulate a virtual view of local file system
- Pay as you Use Based Infrastructure
- Differ in read write operations, performances, permanent or temporary loss of storage resource
- Example –Gluster FS, Ceph, Nutanix



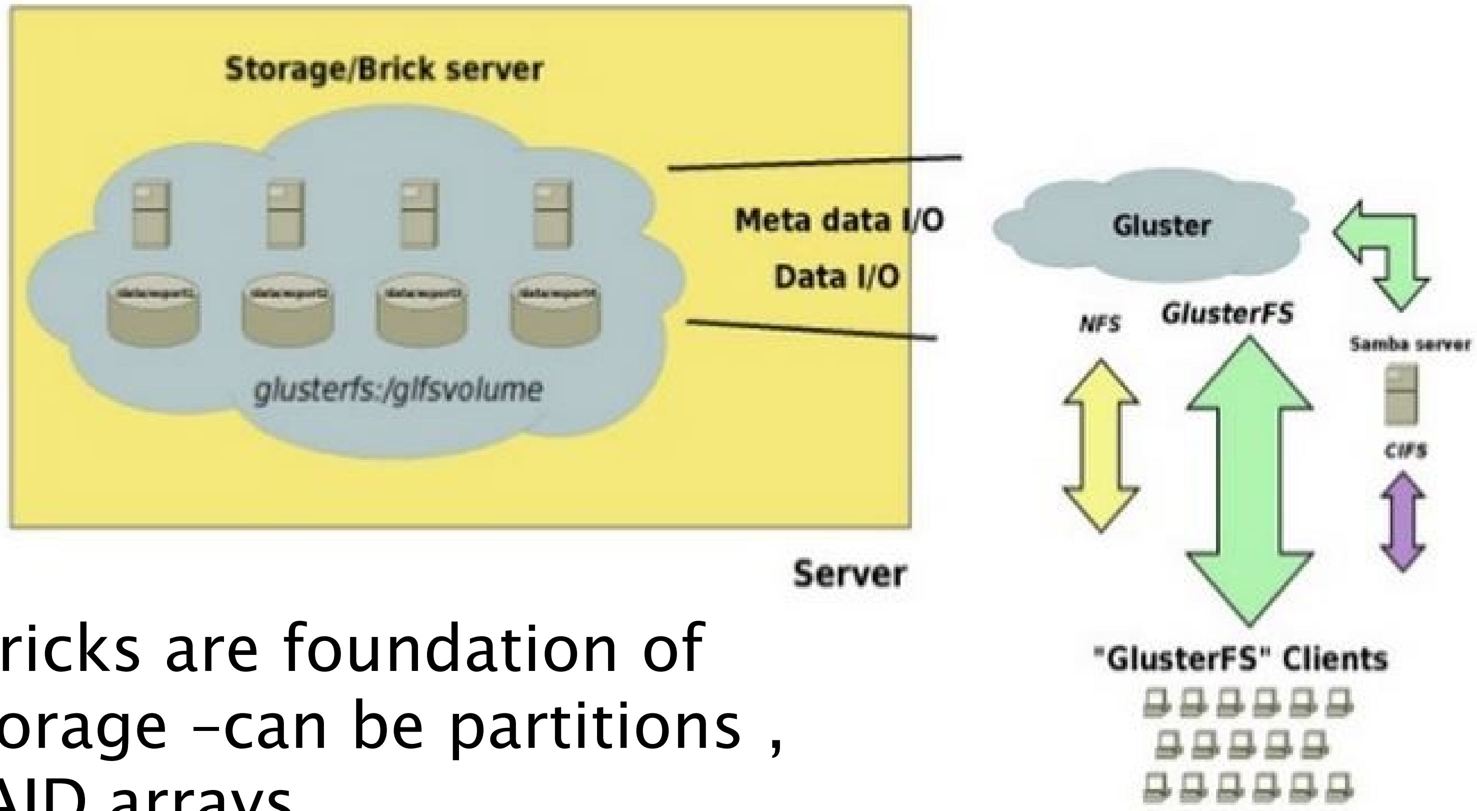
Enterprise Goals for Data Storage

- Information is the Key to Success
- Massive amount of data generated
- Data Intensive Applications
- Limited Storage Capacity
- Fault Tolerance and Disaster Recovery
- Reliability of Communication System
- Cost effective Solution– Minimum Proprietary reliance
- Scalability and Elasticity
- Data Security
- Data Sharing
- Decentralize and limit Failure points

GLUSTER

- Described as open source Scale-out Storage File System
- File System written in User Space which uses FUSE to connect to VFS layer
- File Systems- ext4, btrfs,xfs.
- Based on Commodity Hardware
- Scale storage size to peta- byte of data.
- Client Access Mechanisms based on using NFS, SAMBA, HTTP, REST, FUSE, libgfapi
- Hashing Algorithm–Davis Mayer's Algorithm used for file placement

Gluster- Contd.



•Bricks are foundation of storage –can be partitions , RAID arrays

•Translators

•No Separate Metadata Server

CEPH

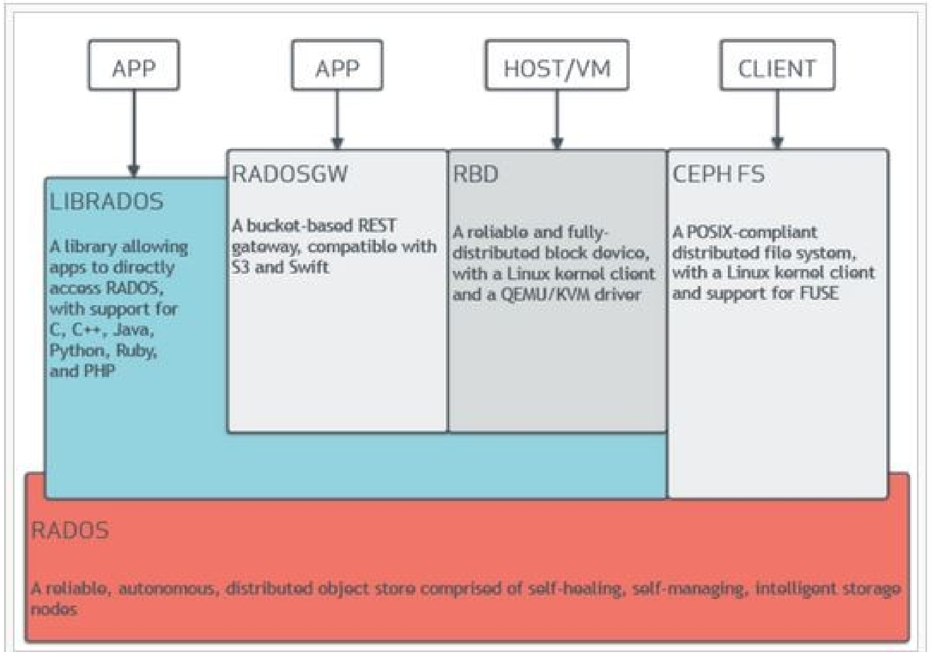
- Distributed Scale out system with POSIX semantics
- Supports Block Storage, Object Storage, File System
- Storage cluster based on RADOS
- No Single Point of Failure,
- Scalable to exa-byte level
- Physical Storage of Data handled using CRUSH maps-Storing and Retrieving of Data
- OSD-Data Silos
- Enables Hyperscaling Feature-Ceph OSD daemons are cluster aware and interact with other OSDs

CEPH Contd.

- RBD –access to RADOS interface

- CEPH-FS Linux File system which allows access to Ceph Storage

- RADOS Gateway –offers RESTFUL storage in CEPH accessible via Amazon S3 or Swift


APP
APP
HOST/VM
CLIENT
LIBRADOS
A library allowing apps to directly access RADOS, with support for C, C++, Java, Python, Ruby, and PHP
RADOSGW
A bucket-based REST gateway, compatible with S3 and Swift
RBD
A reliable and fully-distributed block device, with a Linux kernel client and a QEMU/KVM driver
CEPH FS
A POSIX-compliant distributed file system, with a Linux kernel client and support for FUSE
RADOS
A reliable, autonomous, distributed object store comprised of self-healing, self-managing, intelligent storage nodes

Extensions and Interfaces

- Since version 3.4, it's finally possible to access the data on GlusterFS directly via a libgfapi
- Gluster is Modular and Extensible through use of Translators
- Extensions Via Plug-ins :Ceph does not provide any run-time extensibility currently
- Ceph supports RESTFUL Api -has been storing binary object store
- RADOS gateway offers openstack swift object store Supports Amazons S3 storage
- Gluster is the new kid in the block in terms of object store

Meta Data Server

- Performance of Meta Data Server often is bottleneck

- Ceph uses load balancing across multiple servers

- Gluster FS does not use any meta data server but uses distributed hashing algorithm

- File name changes then need to redistribute file again, which could be a performance degradation

Replication

- Gluster FS-Translators used
- Bricks used multiple of replication factor
- Replication transparent to user.

- Gluster FS server has trust relationship
- Qurum mechanism
- Ceph- Once user uploads binary object OSD replicates
- Uses CRUSH algorithm , OSD and MON server which to which OSD it needs to replicate
- Self healing process in case of failure

Comparison

Test Goals: Scalability, Performance

Environment :

- Different Compute Nodes
- Different VM counts

Four Compute Nodes Running RDO Open Stack
Four Storage Nodes either running Ceph or Gluster
Red Hat Storage in Replication Level 2
Ceph used with Replication level 2

Read /Write Operations

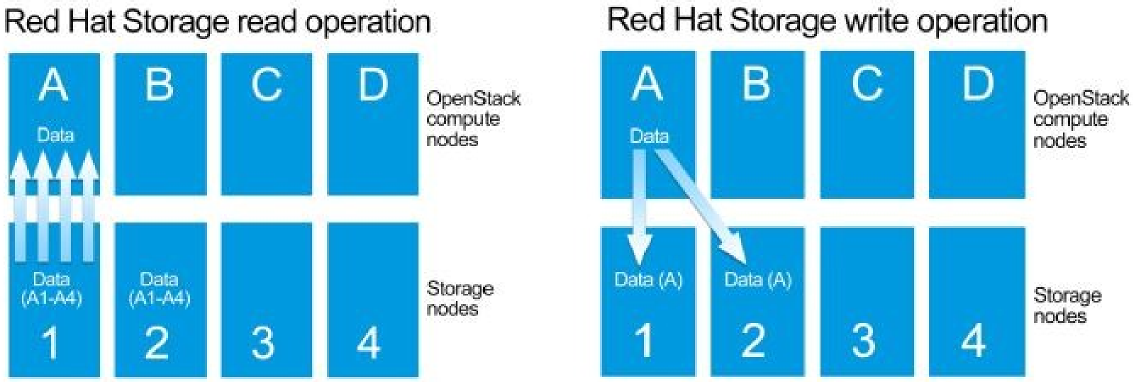


Figure 1: Read and write IO of Red Hat Storage.

Red Hat Storage or Gluster Read/Write Operations

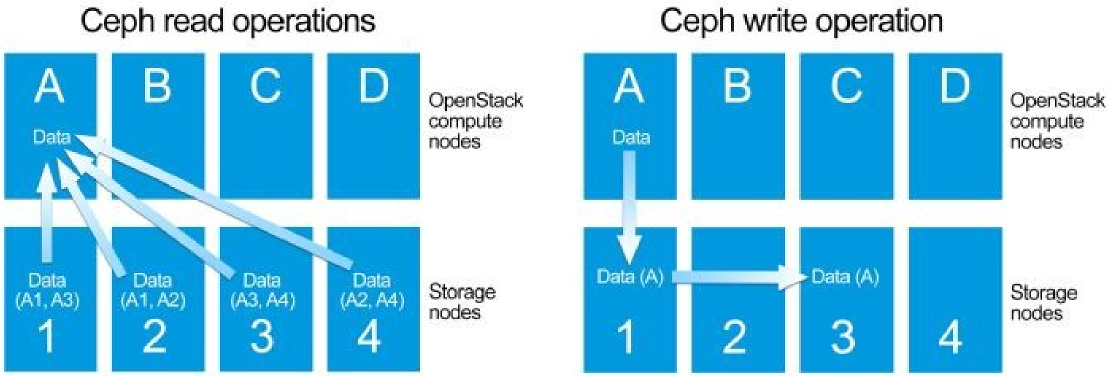
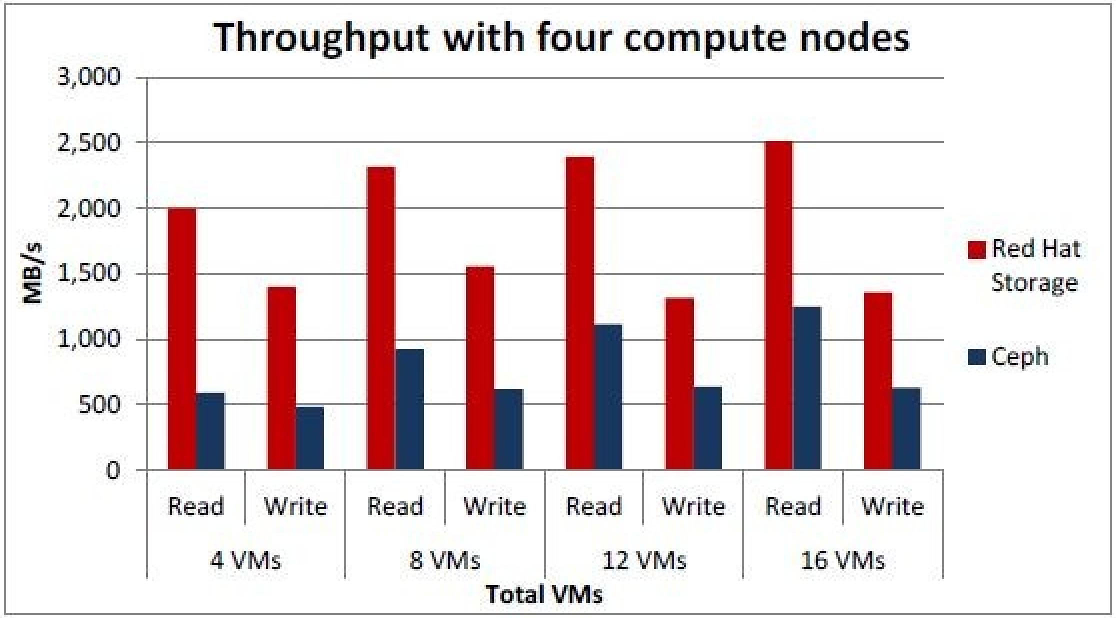


Figure 2: Read and Write IO of Ceph Storage.

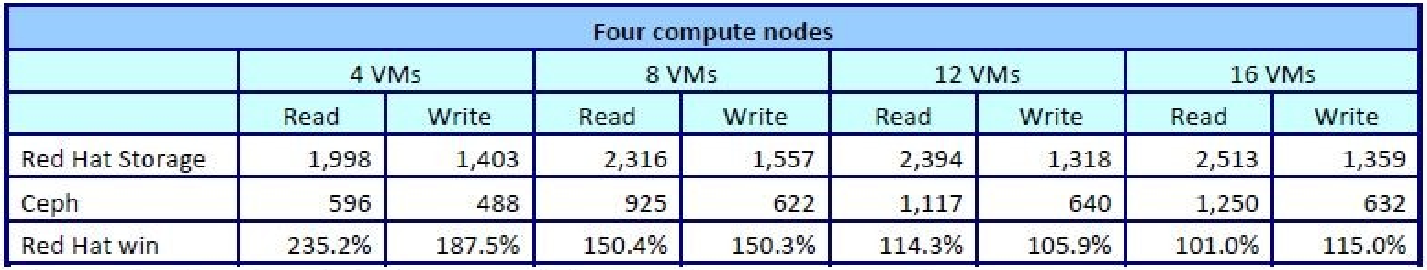
Ceph I/O
Operation Diagram

Which is better



At four node throughput 101.0 % and 235.2 % read write solution

Higher performance and scalability results



Four compute nodes								
	4 VMs		8 VMs		12 VMs		16 VMs	
	Read	Write	Read	Write	Read	Write	Read	Write
Red Hat Storage	1,998	1,403	2,316	1,557	2,394	1,318	2,513	1,359
Ceph	596	488	925	622	1,117	640	1,250	632
Red Hat win	235.2%	187.5%	150.4%	150.3%	114.3%	105.9%	101.0%	115.0%

Figure 8: Throughput, in MB/s, for the storage solutions at varying VM counts across four compute nodes.

Conclusion

- Ceph is rooted in object store
- Has its strength in the RADOS layer
- Gluster based on NAS system has strength in file system domain,
- Gluster FS leaner filesystem helps enables debugging and recovery

THANK YOU !

Contact Email Id
Rajeshwari.Chatterjee@student.lut.fi