Testing of several distributed file-system (HadoopFS, CEPH and GlusterFS) for supporting the HEP experiments analisys.

Giacinto DONVITO
INFN-Bari

Agenda

- Introduction on the objective of the test activities
- HadoopFS
- GlusterFS
- CEPH
- Tests and results
- Conclusion and future works

Introduction on the objective of the test activities

- The aim of the activity is to verify:
  - Performance
  - Reliability
  - Features
- Considering solutions that provides software redundancy
  - A site could use commodity hardware to achieving high level of data availability
- The scalability should be guaranteed at the order of few PetaByte

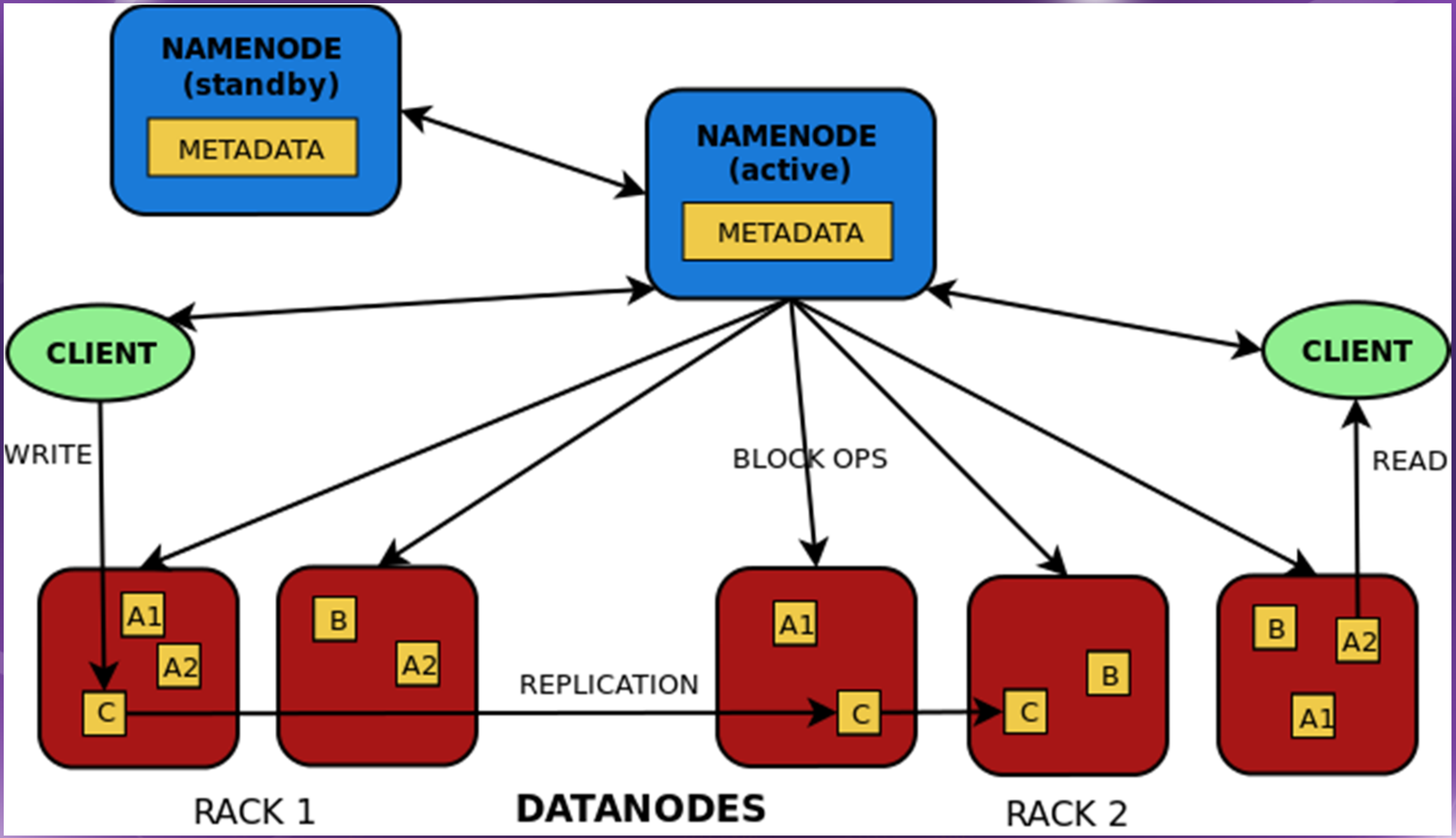
Introduction on the objective of the test activities

- The focus is to serve typical Tier2/Tier3 sites for LHC experiments
  - Supporting interactive usage
  - Running data analysis
  - Supporting SRM, gridftp, Xrootd
  - Being prepared for the new cloud storage techniques
- Open Source solutions

HadoopFS

- Apache Hadoop Distributed File System:
  - Open-source
  - Developed in Java
  - Large dataset
  - Fault tolerant
  - Commodity hardware
  - High throughput
  - Scalable
  - Rack awareness

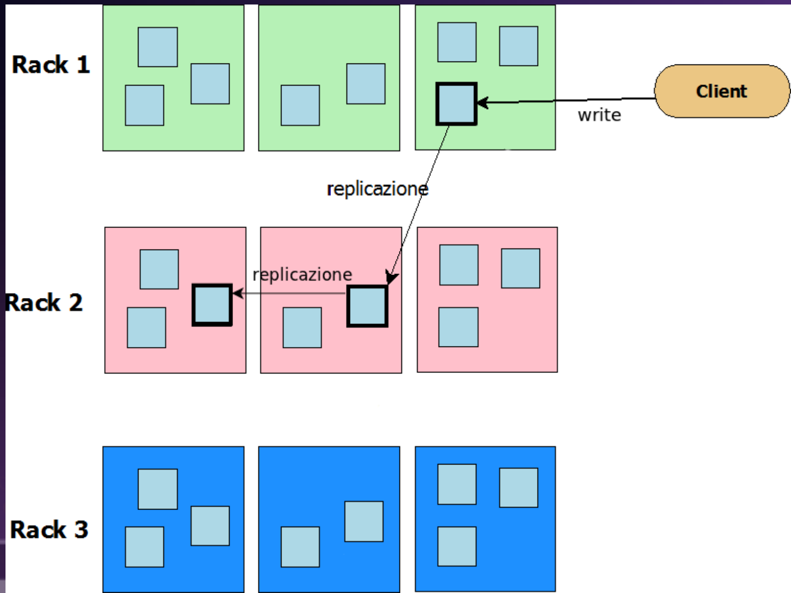
HadoopFS



NAMENODE (standby)
METADATA
NAMENODE (active)
METADATA
CLIENT
CLIENT
WRITE
BLOCK OPS
READ
A1
A2
C
B
A2
REPLICATION
A1
C
C
B
B
A2
A1
RACK 1
DATANODES
RACK 2

HadoopFS

- "The primary objective of HDFS is to store data reliably even in the presence of failures" (Hadoop documentation)
  - File are split in chunk (default 64MB)
    - dfs.blocksize
  - Placement policy (default):
    - 3 replicas
      - 1 replica in the local rack
      - 2 replicas in the remote rack
    - dfs.replication

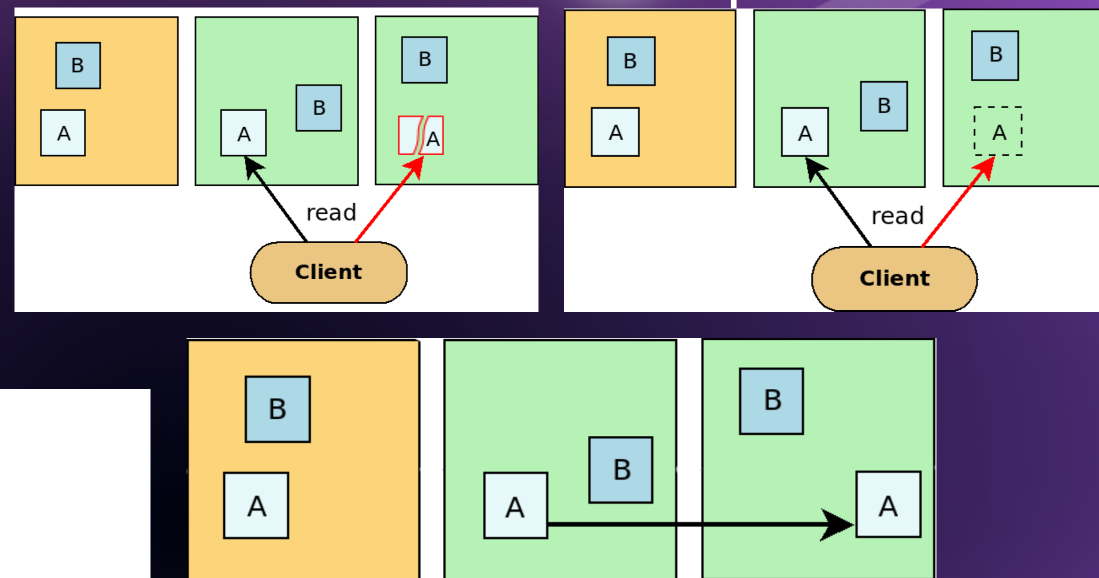


HadoopFS functionality test

- We have executed several test to check the behavior at several types of failures:
  - Metadata failures
    - Client retry, active-standby namenode
  - Datanode failures:
    - During write operation, during read operation, in case of data corruption, mis-replicated blocks, under and over replicated blocks
- We always succeeded to fulfill the expected behavior and no (un-expected) data-loss were registered

HadoopFS

Data Corruption



NameNode 'pccms61.ba.infn.it:9000' (active)

Started:	Thu Aug 22 11:45:05 CEST 2013
Version:	2.0.0-cdh4.1.1, 581959ba23e4af85afd8db98b7687662fe9c5f20
Compiled:	Tue Oct 16 10:39:59 PDT 2012 by jenkins from Unknown
Upgrades:	There are no upgrades in progress.
Cluster ID:	CID-9b734b6d-3611-4eb7-ab5e-49419f75dc3a
Block Pool ID:	BP-1130807058-212.189.205.51-1340275038748

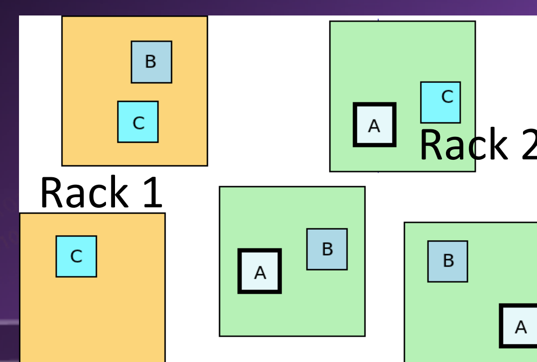
Browse the filesystem
NameNode Logs

Cluster Summary

Security is *OFF*
719577 files and directories, 798359 blocks = 1517936 total.
Heap Memory used 3.29 GB is 69% of Commited Heap Memory 4.71 GB. Max Heap Memory is 8.89 GB.
Non Heap Memory used 42.98 MB is 71% of Commited Non Heap Memory 60.38 MB. Max Non Heap Memory is 130 MB.
WARNING : There are 1642 missing blocks. Please check the logs or run fsck in order to identify the missing blocks.
See the Hadoop FAQ for common causes and potential solutions.

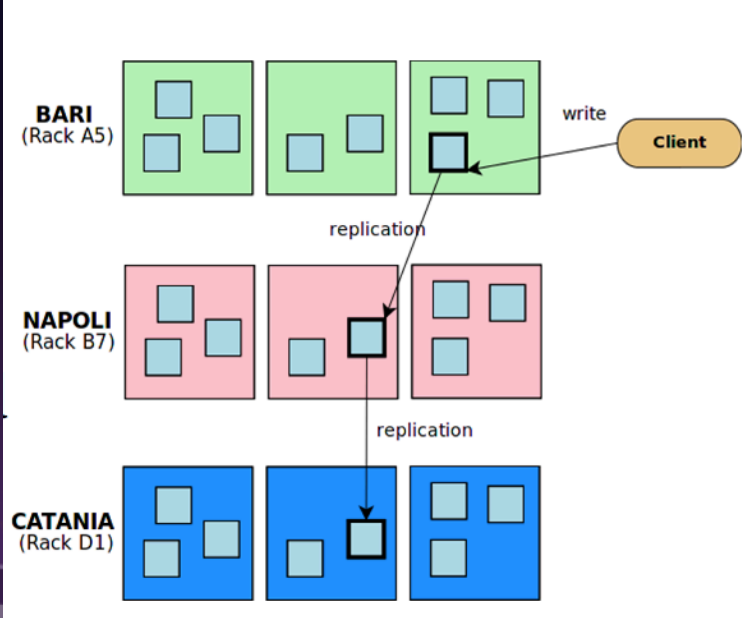
Configured Capacity	:	135.31 TB			
DFS Used	:	23.48 TB			
Non DFS Used	:	6.66 TB			
DFS Remaining	:	105.16 TB			
DFS Used%	:	17.36 %			
DFS Remaining%	:	77.72 %			
Block Pool Used	:	23.48 TB			
Block Pool Used%	:	17.36 %			
DataNodes usages	:	Min %	Median %	Max %	stdev %
		0 %	25.59 %	100 %	32.31 %
Live Nodes	:	155 (Decommissioned: 9)			
Dead Nodes	:	97 (Decommissioned: 36)			
Decommissioning Nodes	:	0			
Number of Under-Replicated Blocks	:	77			

Mis-replicated blocks



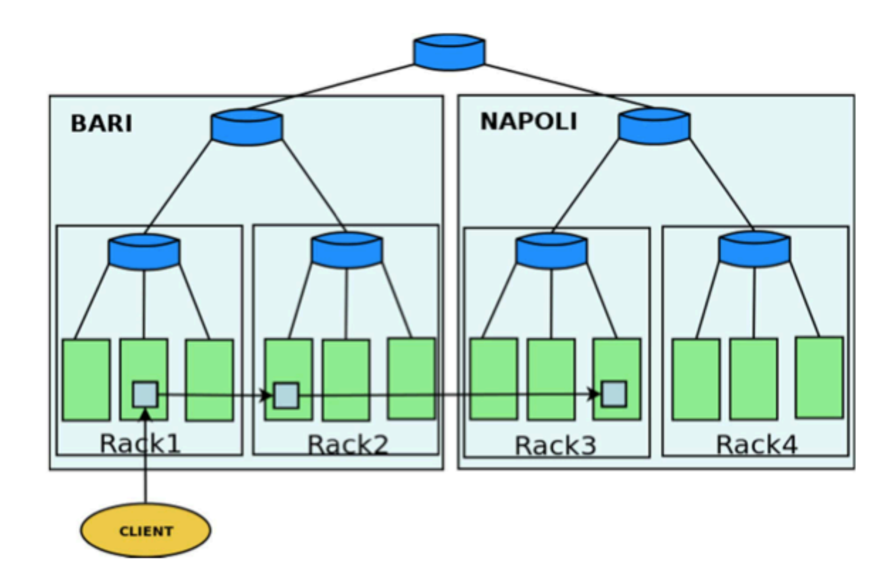
HadoopFS: our development

- One Replica Policy
  - 1 replica per rack
    - Increasing the reliability
    - Increasing the available bandwidth for read operation



HadoopFS: our development

- Hierarchical Policy
  - It is able to exploit a geographically distributed infrastructure
  - 2 replicas in the source site in 2 different racks
- The data will survive also to the loss of a complete site



HadoopFS pros&cons

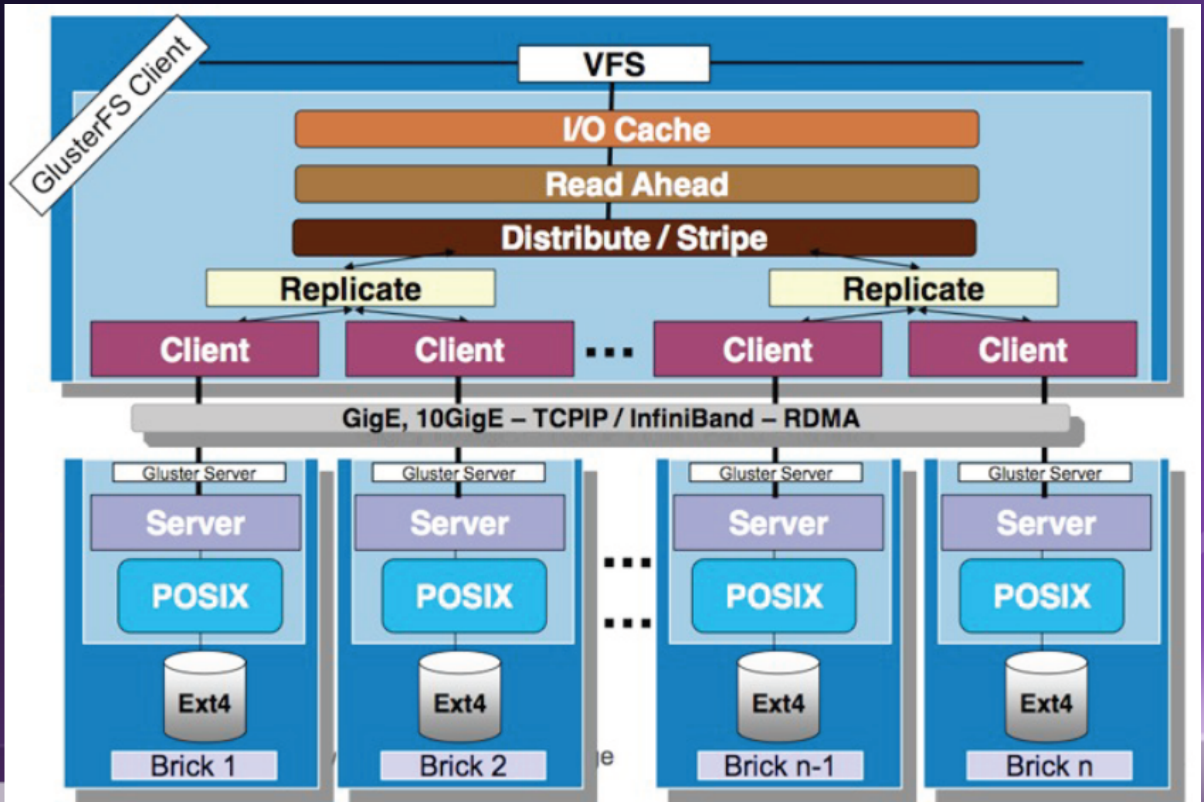
- MapReduce
- Dynamic self-healing
- Great Scalability
  - Already tested in few big Tier2 in LHC and many companies
- Web monitoring interface
- Support for SRM (Bestman) and gridftp/xrootd (Nebraska)
- Non strictly-posix compliance
  - Fuse based
- No support for new cloud storage technologies

GlusterFS

- OpenSource solution acquired by RedHat
- Could be used both with disk in the WN and with standard infrastructures based on disk servers (SAN/DAS)
- Written in C under GPLv3
- Posix compliance
- Exploit NFS protocol
- Available on many platforms (RedHat, Debian, MacOS, NetBSD, OpenSolaris)
- Support also new storage cloud technologies (Block Storage, Object Storage, etc)
  - It is based on Swift (OpenSource Object Storage developed within OpenStack framework)

GlusterFS

- Working behavior:
  – The client exploit a FUSE module to access file and implement advanced policy (Distribute/Stripe/Replica, etc)
- The client and server could exploit both TCP/IP and infiniband connections
- The server hosts data on standard file-systems (ext4, xfs, etc)



GlusterFS

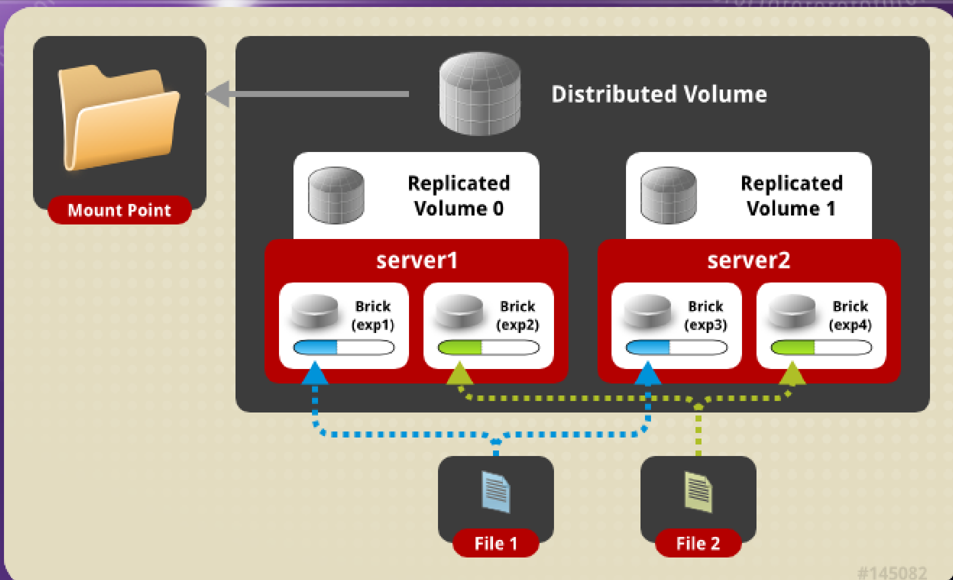
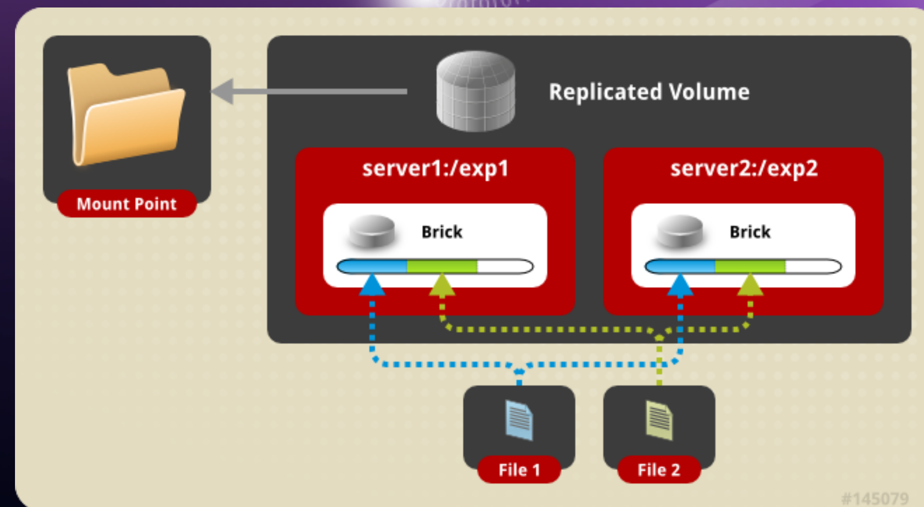


Figure 5.5. Illustration of a Distributed Replicated Volume



- Working behavior:
  - Striped Volume
  - Replicated Volume
  - Distributed Volume
  - Striped Replicated Volume
  - Distributed Replicated Volume

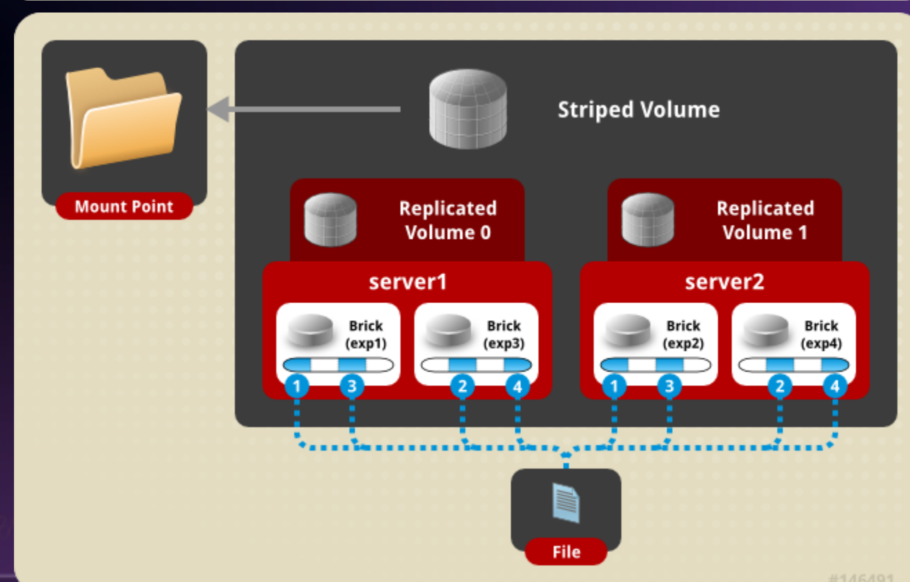
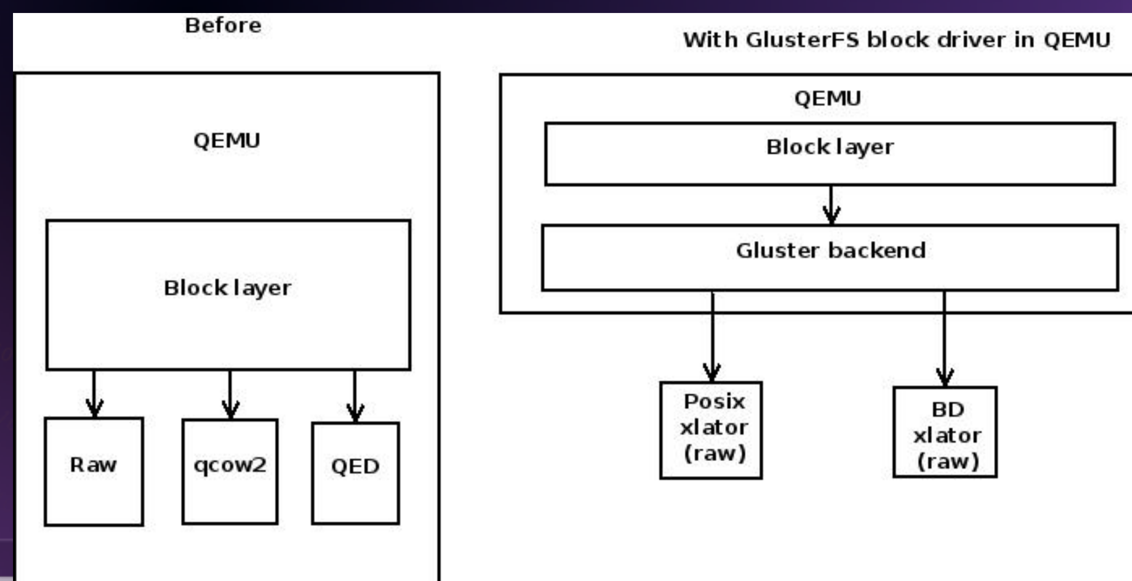


Figure 5.6. Illustration of a Striped Replicated Volume

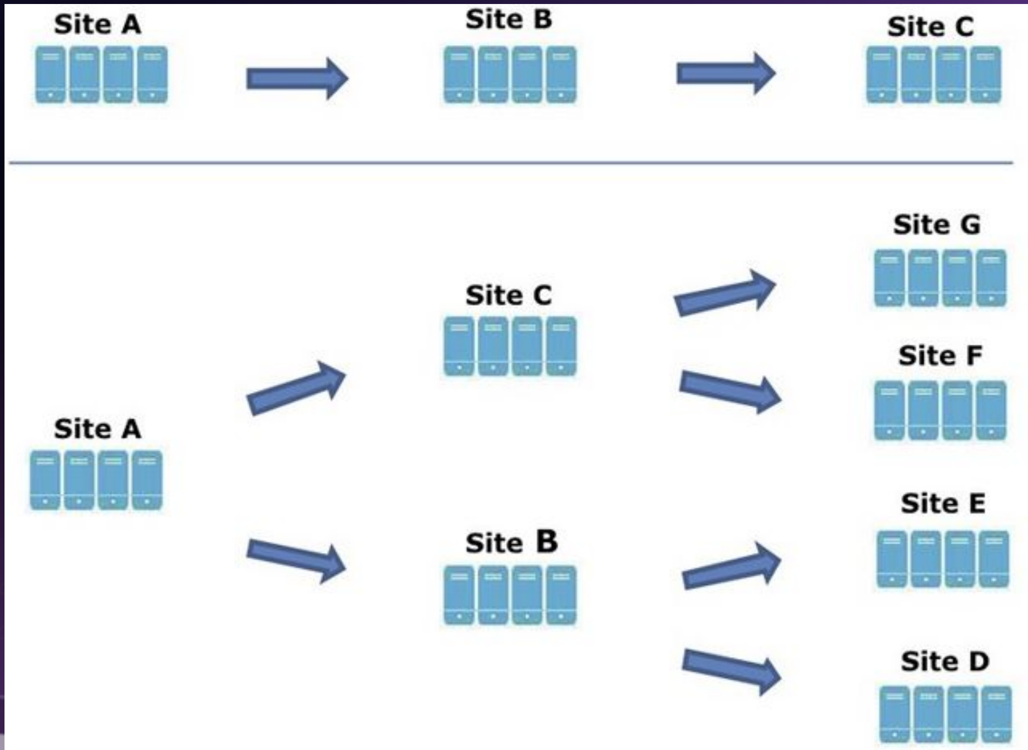
GlusterFS

- POSIX ACL support over NFSv3
- Virtual Machine Image Storage
  - qemu – libgfapi integration
    - improvements in performance for VM image hosting
- Synchronous replication improvements
- Distributed-replicated and Striped-replicated are very important in the contest where performance and data availability is important



GlusterFS

- GlusterFS provides a geographical replication solution
  - Could be useful as disaster recovery solution
  - It is based on the paradigm of active-backup
  - It is based on rsync
  - It is possible to replicate the whole file-system or a part of it
  - It could be used also from one site to multiple instances of GlusterFS on different sites



Glusterfs pros&cons

- Easy to install and configure
- Fully posix compliance
- Many available configuration
- Great performance
- Provides interesting cloud storage solutions
- Some instabilities and data loss in some specific situations
- There are no many scalability reports beyond petabyte

CEPH file-system

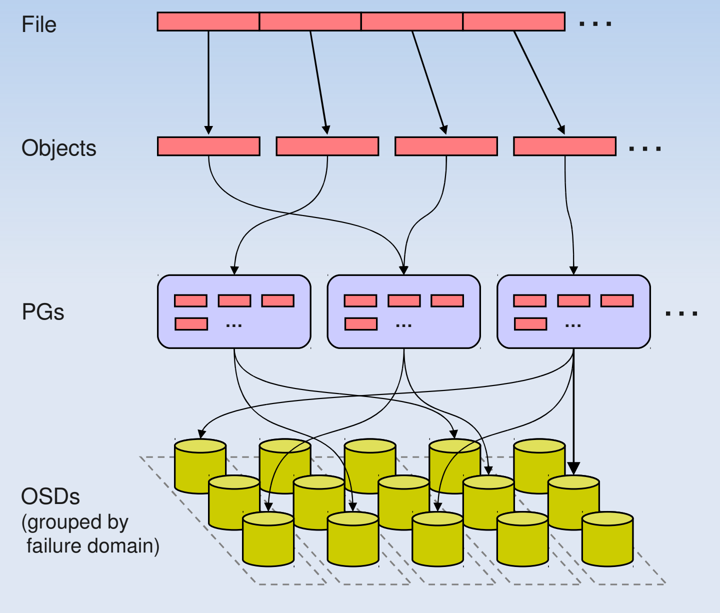
- Development started in 2009
- Now it is acquired by a company (Inktank) also if it is still an completely OpenSource projects
- It is integrated by default in the Linux Kernel since 2.6.34 release (may 2010)
- It could use, although not already at “production level”, BTRFS (B-tree fle system) as backend
  - Several interesting features (Raid0/1, and soon Raid5/6, data deduplication, etc) implemented at software level

CEPH file-system

- Designed to be scalable and fault-tolerant
  - In order to support >10’000 disk server
  - Up to 128 metadata server (could serve up to 250kops/s aggregate)
- CEPH can provide three different storage interfaces: Posix (both at kernel level and using fuse), Block and Object storage
- Several IaaS cloud platforms (i.e.: OpenStack, CloudStack) officially supports CEPH to provides Block Storage solution
- The suggested configuration do not require/suggest the use of any hardware raid: the data availability is implemented at software level

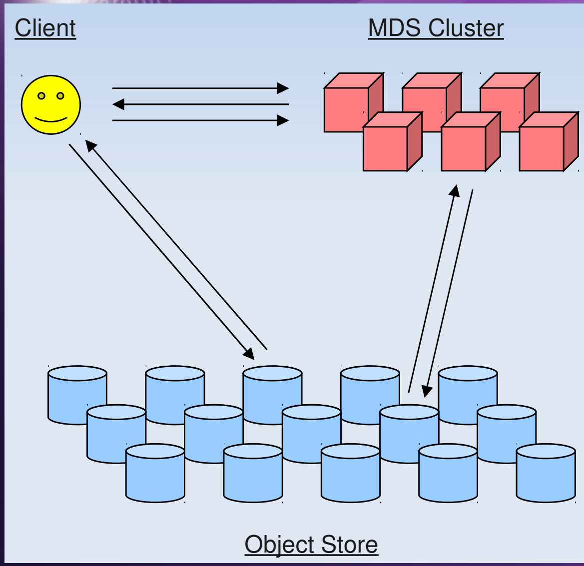
CEPH file-system

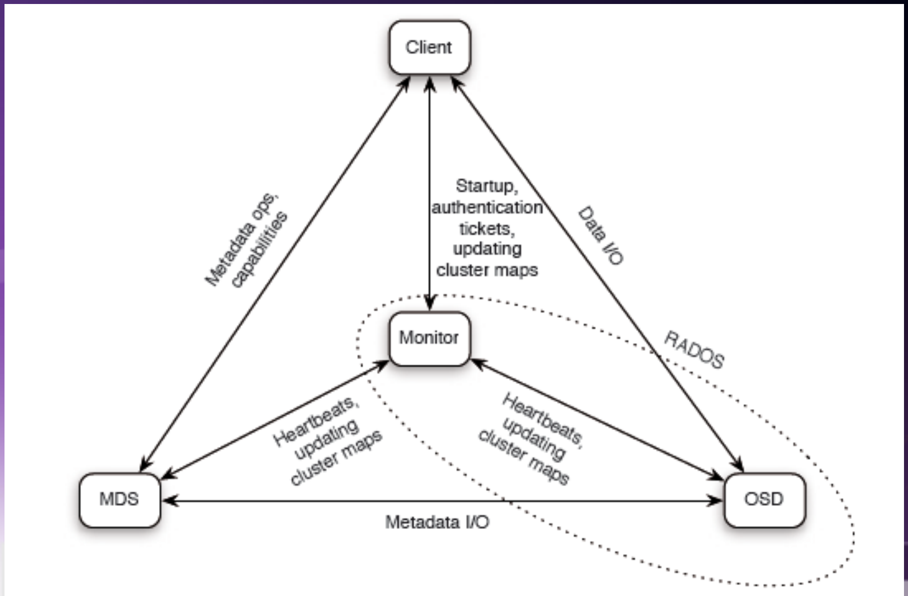
- The data distribution is based on an hash function
  - No query needed to know the location of a given file
- This means that the mapping is “unstable”:
  - Adding a disk server, mean that the whole cluster need to reshuffling the location of the data
- It is possible to define “failure domain” at the level of: disk, server, rack
- Data placement rules could be customized:
  - “tre different copies of the same file in three different racks”
- All the datanodes knows the exact location of all the files in the cluster

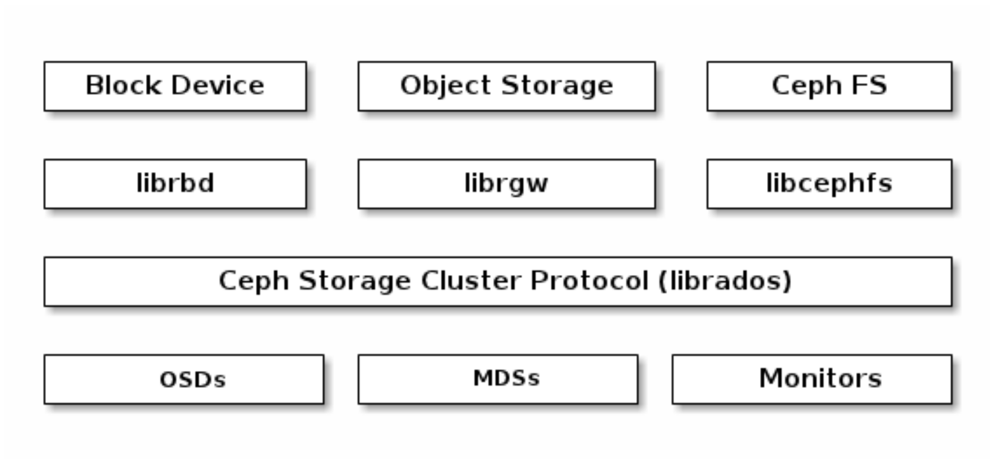


CEPH file-system

- Monitor: manages the heartbeats among nodes
- MDS: manages l'I/O on metadata
- OSD: contains the objects
- The client will interact with all the three services
- A 10 node cluster will be composed by:
  - 3 monitor node
  - 3 MDS node
  - 7 OSD node

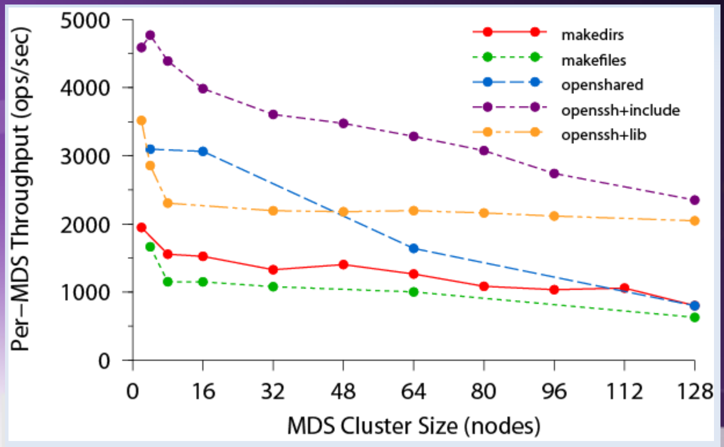


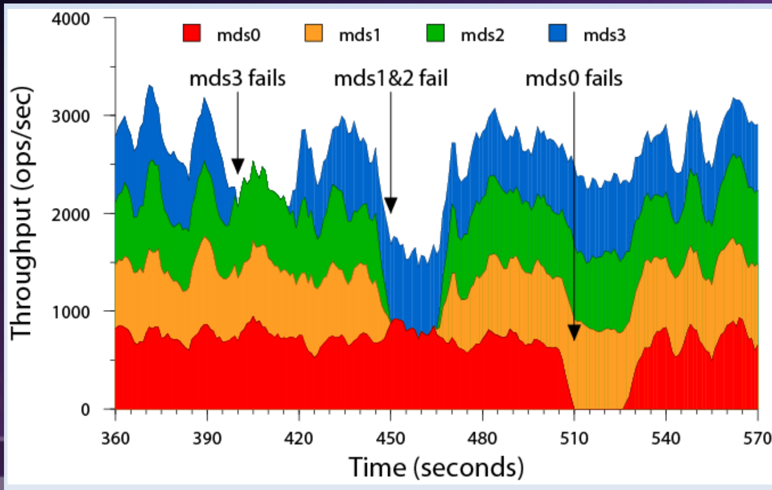




CEPH file-system

- The three storage interfaces (posix, block and object) are different gateways on the same objects APIs
- The object could be stored also “striped” in order to increase the performances
  - Object Size, Stripe Width, Stripe Count
- Data Scrubbing: it is possible to periodically check the data consistency (to avoid inconsistencies between data and metadata, and or data corruptions)





CEPH functionalities test

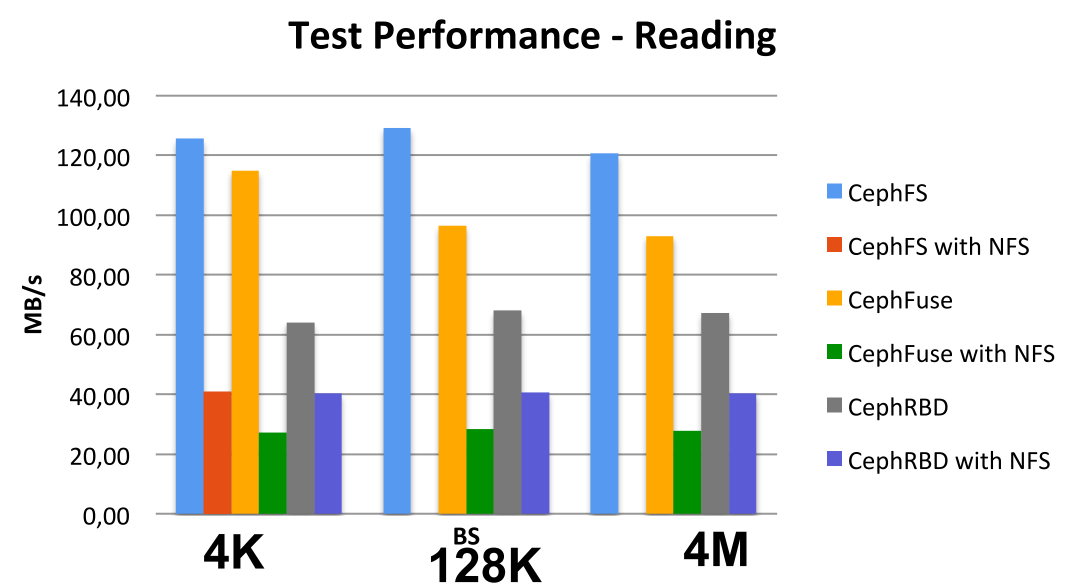
- The “quorum” concept is used for each critical service (there should be odds numbers of instances):
  - If 2 over 3 services are active the client could read and write. Is only one is active the client could only read
- We verified the behaviour in case of failure of each service:
  - The High Availability worked always as expected
  - We tested both failure in data and metadata services
  - Both using posix and RBD interfaces
- We tested also the possibility to export the storage using standard NFS protocols
  - It works quite well both using RBD and POSIX interface
    - Was very unstable using kernel interface

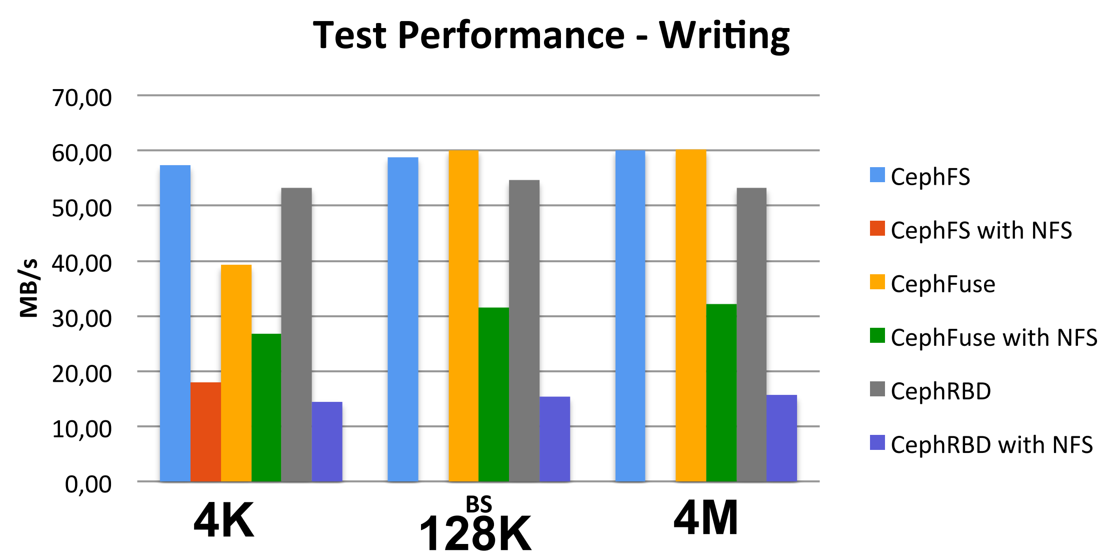
CEPH RBD

- CEPH RBD features:
  - Thinly provisioned
  - Resizable images
  - Image import/export
  - Image copy or rename
  - Read-only snapshots
  - Revert to snapshots
  - Ability to mount with Linux or QEMU KVM clients
- In OpenStack it is possible to use CEPH both as device in Cinder (Block storage server) and for hosting virtual images in Glance (Image Service)
- CEPH provides an Object Storage solution that has interfaces compatible with both S3 (Amazon) and Swift (OpenStack)

CEPH Performance test

BS
4K
128K
4M





Virtual Machine

CEPH pros&cons

- Complete storage solution (supports all the storage interfaces: posix, object, block)
- Great scalability
- Fault-tolerant solution
- Difficult to install and configure
- Performance issues
- Some instabilities while under heavy load

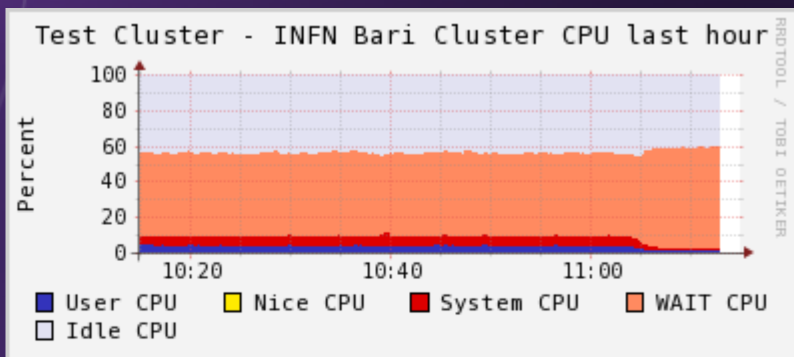
HDFS v2 CDH 4.1.1 (by USCMS Nebraska)

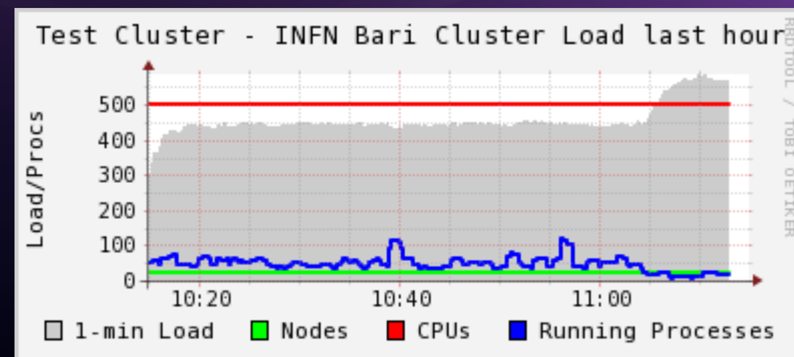
- 20 datanodes, 1 namenode
- Chunk size: 128MB, Rdbuffer: 128MB, Big_writes active
    - # iozone -r 128k -i 0 -i 1 -i 2 -t 24/36 -s 10G

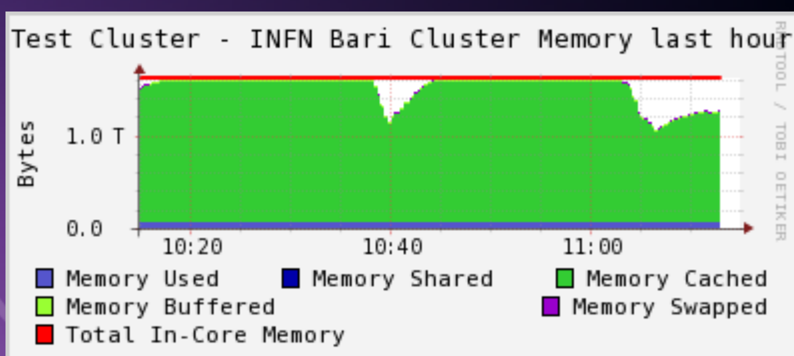
MB/s

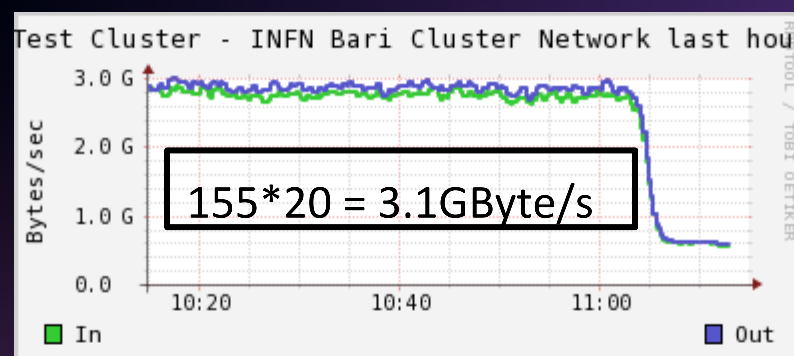
	24 Threads	36 Threads
Initial Write	239.72	
Re-write		
Random Write		
Initial Read	155.18	193.65
Re-read	151.33	207.43
Random Read	29.06	39.98

HDFS – 24 threads

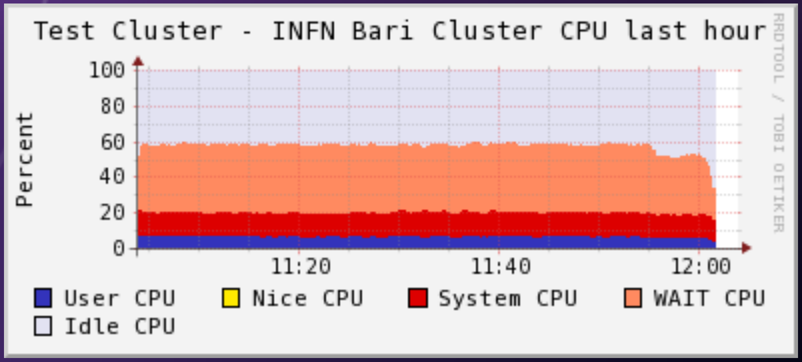


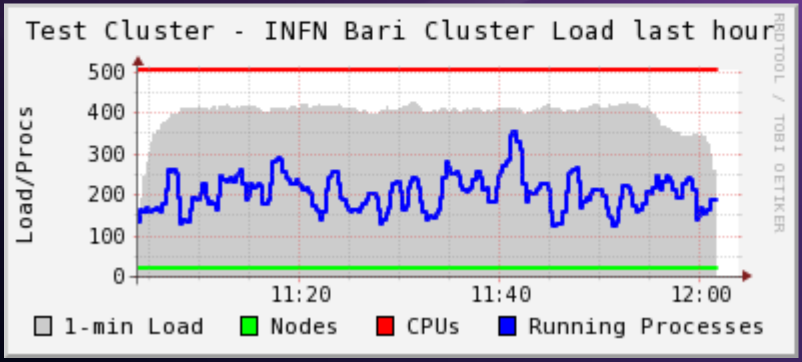


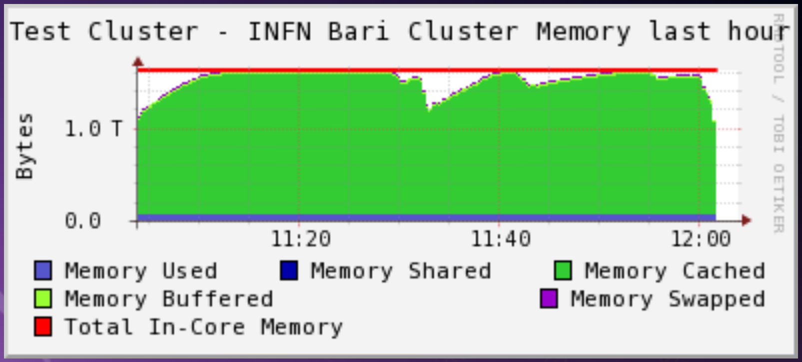


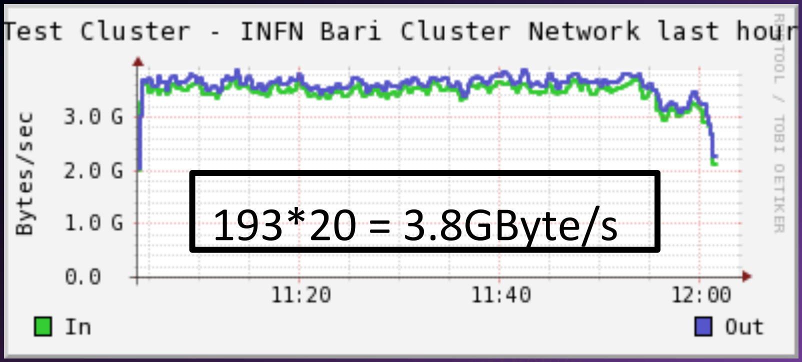


HDFS – 36 threads









Ceph Cuttlefish (0.61)

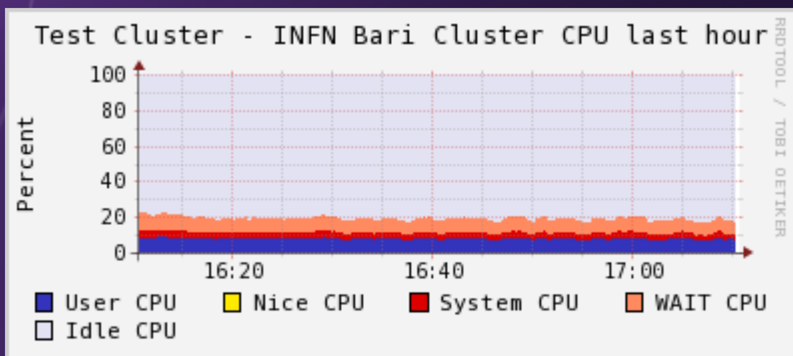
- 3 Mon, 1 Mds, 120 osd (6osd * 20nodi)
- On all the nodes (SLC6)
  - `# iozone -r 128k -i 0 -i 1 -i 2 -t 24 -s 10G`

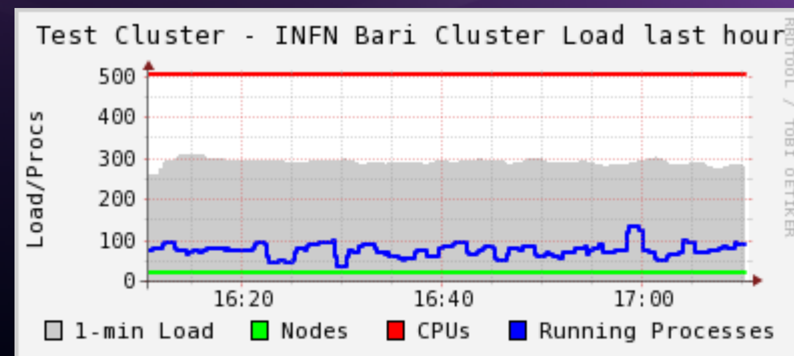
MB/s

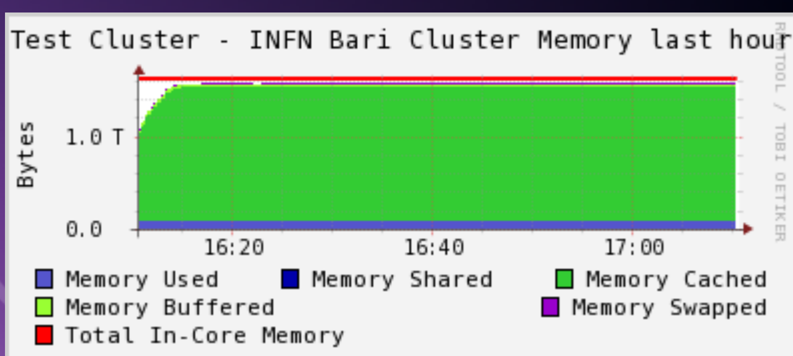
	24 Threads
Initial Write	52.49
Re-write	54.05
Random Write	ERROR
Read	95.38
Re-read	102.04
Random Read	ERROR

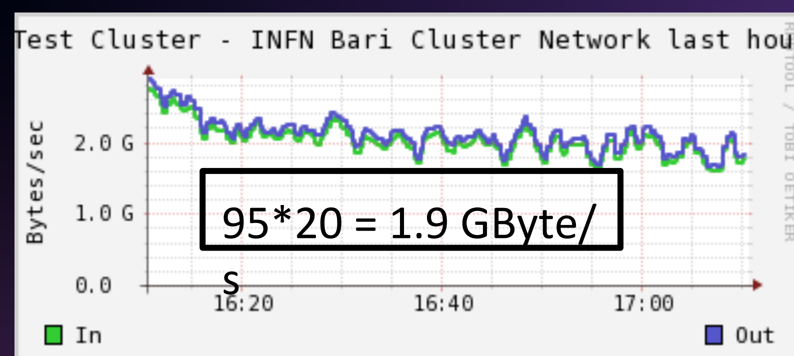
95*20 = 1.9 GByte/s

Ceph Cuttlefish (0.61)









Ceph Dumpling (0.67.3)

- 3 Mon, 1 Mds, 95 osd (5osd * 19nodi)
- On all the nodes (SLC6)
  - `# iozone -r 128k -i 0 -i 1 -i 2 -t 24 -s 10G`

MB/s

	24 Threads
Initial Write	18.93
Re-write	19.31
Random Write	13.96
Read	53.40
Re-read	57.29
Random Read	5.13

53*19 = 1.0 GByte/s

Ceph-Dev (0.70)

- 3 Mon, 1 Mds, 15 osd (5osd * 3nodi)
- On all the nodes (Ubuntu 12.04)
  - `# iozone -r 128k -i 0 -i 1 -i 2 -t 24 -s 10G`

MB/s

	24 Threads
Initial Write	51,06
Re-write	60,05
Random Write	7,00
Read	101,58
Re-read	133,61
Random Read	12,05

Gluster v3.3

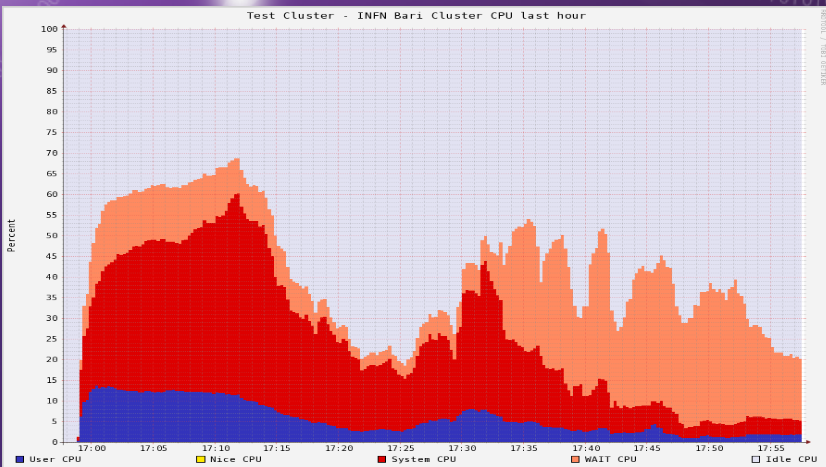
- 21 nodes, 6 brick per node
- On all the nodes (SLC6)
  - ```
    # iozone -r 128k -i 0 -i 1 -i 2 -t 24 -s 10G
    ```

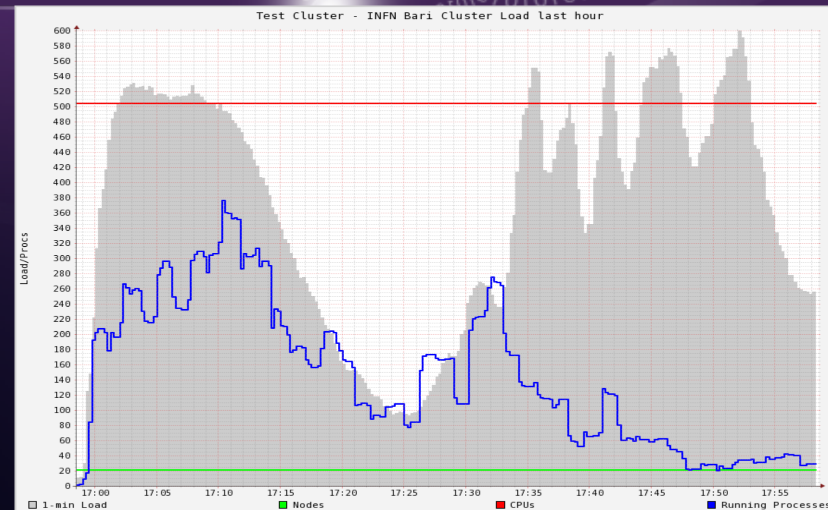
MB/s

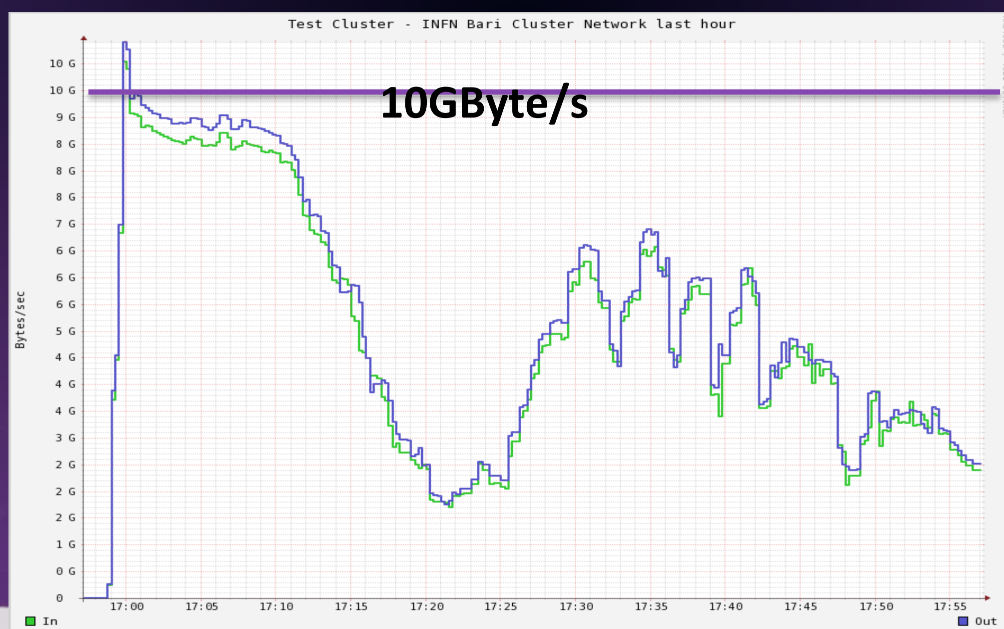
	24 Threads
Initial Write	234.06
Re-write	311.75
Random Write	326.89
Initial Read	621.08
Re-read	662.92
Random Read	242.75

621*21 = 13 GByte/s
    ```

Gluster v3.3


Test Cluster - INFN Bari Cluster CPU last hour
Percent
17:00 17:05 17:10 17:15 17:20 17:25 17:30 17:35 17:40 17:45 17:50 17:55
User CPU
Nice CPU
System CPU
WAIT CPU
Idle CPU


Test Cluster - INFN Bari Cluster Load last hour
Load/Procs
17:00 17:05 17:10 17:15 17:20 17:25 17:30 17:35 17:40 17:45 17:50 17:55
1-min Load
Nodes
CPUs
Running Processes


Test Cluster - INFN Bari Cluster Network last hour
10GByte/s
Bytes/sec
17:00 17:05 17:10 17:15 17:20 17:25 17:30 17:35 17:40 17:45 17:50 17:55
In
Out

Gluster v3.4

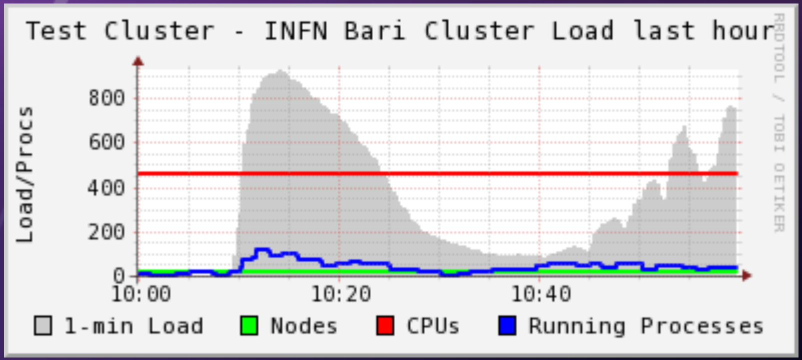
- 20 nodes, 6 brick per node
- On all the nodes (SLC6)
  - # iozone -r 128k -i 0 -i 1 -i 2 -t 24 -s 10G

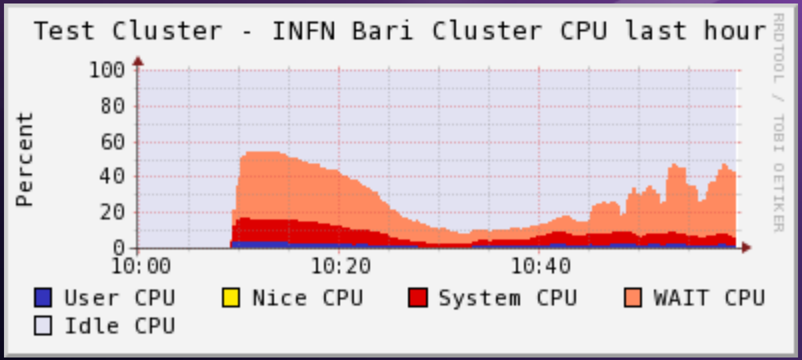
MB/s

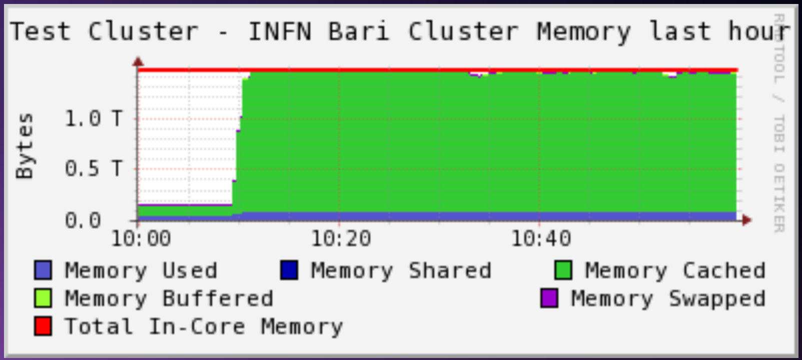
	24 Threads
Initial Write	306.34
Re-write	406.90
Random Write	406.33
Read	688.06
Re-read	711.46
Random Read	284.00

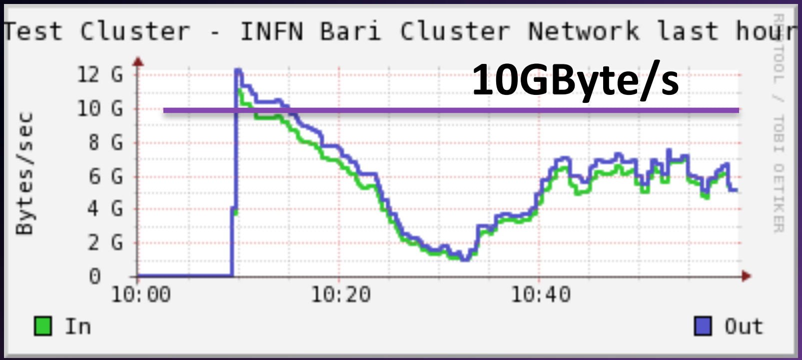
688*20 = 13 GByte/s

Gluster v3.4


Test Cluster - INFN Bari Cluster Load last hour
Load/Procs
800
600
400
200
0
10:00
10:20
10:40
1-min Load
Nodes
CPUs
Running Processes


Test Cluster - INFN Bari Cluster CPU last hour
Percent
100
80
60
40
20
0
10:00
10:20
10:40
User CPU
Nice CPU
System CPU
WAIT CPU
Idle CPU


Test Cluster - INFN Bari Cluster Memory last hour
Bytes
1.0 T
0.5 T
0.0
10:00
10:20
10:40
Memory Used
Memory Shared
Memory Cached
Memory Buffered
Memory Swapped
Total In-Core Memory


Test Cluster - INFN Bari Cluster Network last hour
10GByte/s
Bytes/sec
12 G
10 G
8 G
6 G
4 G
2 G
0
10:00
10:20
10:40
In
Out

Using dd for comparing them all

24 dd in parallel - 10GB file - bs 4M

MB/s	HDFS	CEPH CF	GLUSTER
read	220.05	126.91	427.3
write	275.27	64.71	268.57

Average per single host (the cluster is made by 20 hosts)

Conclusions …

- We have tested, from a point of view of the performance and functionalities, three of the main known and diffused storage solution …
- … trying to focus on the possibility not to use an hardware raid solution
- taking into account the new cloud storage solution that are becoming more and more interesting

Conclusions …

- Hadoop
  - looks very stable, mature and scalable solution
  - Not fully posix compliance and not the fastest
- GlusterFS:
  - Very fast, posix compliant, and easy to manage
  - Maybe not as scalable as the others, still have few reliability problems
- CEPH:
  - Looks very scalable, complete and technological advanced
  - Still not very mature and stable, performance issues

... and future works

- We will continue this activity of testing storage solution in order to follow the quite fast evolution in this field
- In particular CEPH looks quite promising if/when stability and performance issues will be solved.
- The increasing interest in cloud storage solution are forcing the developers to put effort in providing both block and object storage solutions together with the standard posix

People Involved

- Domenico DIACONO (INFN-Bari)
- Giacinto DONVITO (INFN-Bari)
- Giovanni MARZULLI (GARR/INFN)