Building a cluster filesystem

Using distributed, directly-attached storage

Peter van Heusden
pvh@sanbi.ac.za

Information and Communication Services
University of the Western Cape
Bellville, South Africa

November 2014

UNIVERSITY *of the* WESTERN CAPE

Executive Summary

- UWC Astrophysics cluster stores working data on distributed, directly attached storage, not SAN
- Two systems (both open source) in use: Ceph and GlusterFS
  - Ceph allows for great flexibility in storage configuration but CephFS lacks stability at present
  - GlusterFS is easy to deploy and stable but has slow metadata performance and lacks flexibility in storage configuration
- Both options install with ease on CentOS filesystem and allow for *scale out* construction of storage for cluster
- 170TB+ of storage in active use, with GlusterFS (133 TB) rock solid since deployment (mid 2014) and resilient to transient node failure

UNIVERSITY *of the* WESTERN CAPE

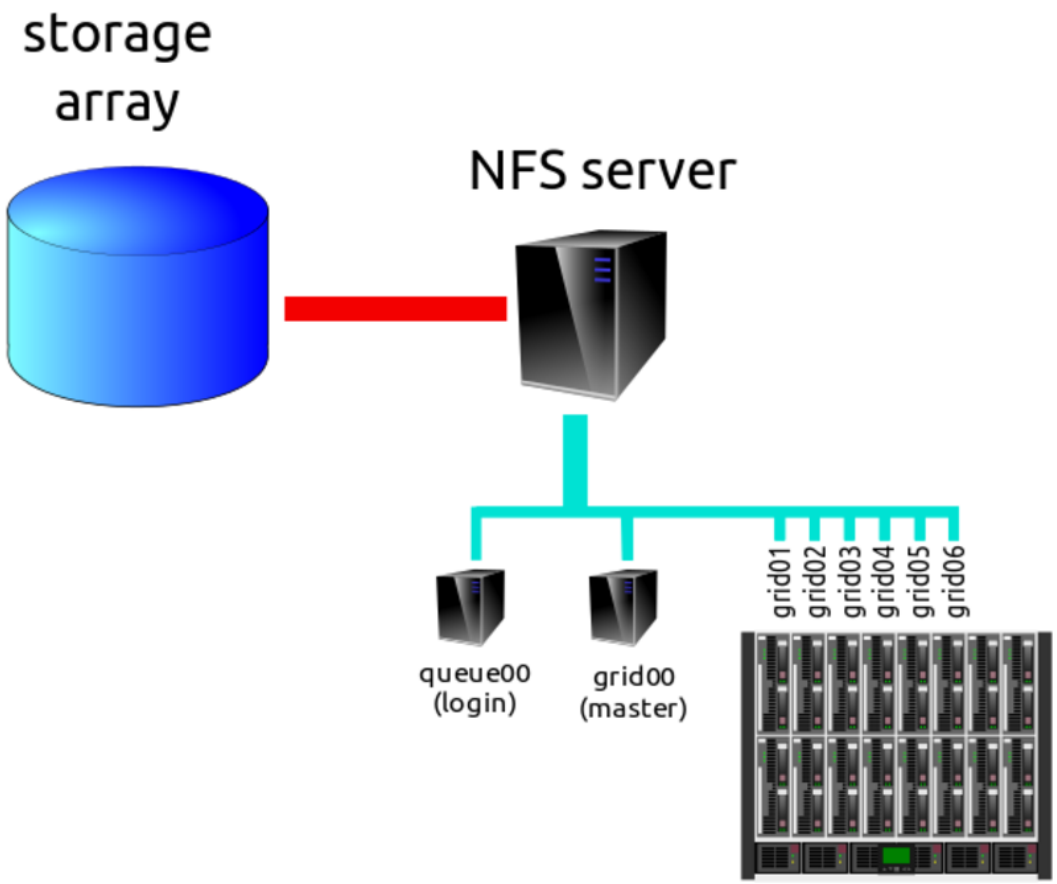
The UWC Astrophysics cluster

- Two clusters, built from SuperMicro servers
  - 4 AMD Opteron CPUs per node, total of 48 cores
  - 256 GB RAM per node
  - 15 TB available disk per node
  - 10 GbE networking
  - Management node with 24 GB RAM, dual Opteron, 6 TB disk for each cluster

1. *Timon:* 4 nodes
2. *Pumbaa:* 19 nodes

UNIVERSITY *of the* WESTERN CAPE

Traditional cluster archicture: cluster + SAN

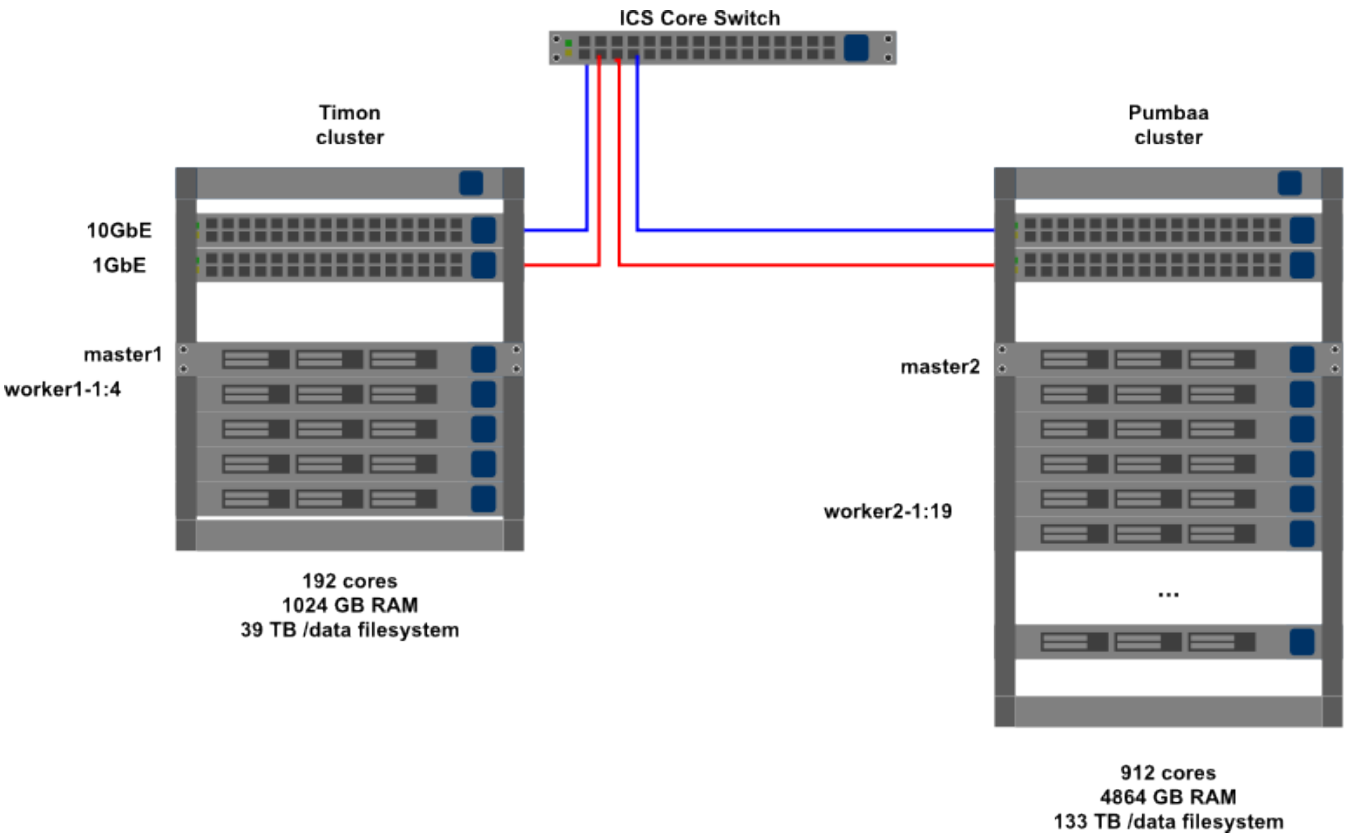


UNIVERSITY of the WESTERN CAPE

Traditional cluster architecture: cluster + SAN (2)

- SAN device shared by nodes on cluster
- Proprietary hardware providing resilience using RAID + redundant hardware on SAN
- Upgrade path: buy a bigger SAN

UWC Astrophysics architecture



UNIVERSITY of the WESTERN CAPE

What, no SAN?

- Budget issues meant that a SAN wasn't available for the Astrophysics cluster
- Forced to utilise direct-attached storage but...
- want to access storage from any node in the cluster thus:
- *Distributed, direct-attached storage*
- Core requirement: POSIX semantics filesystem accessible anywhere, resilient to failure of single cluster node
- Investigated three open source distributed filesystems: *MooseFS*, *CephFS* and *GlusterFS*

UNIVERSITY *of the* WESTERN CAPE

MooseFS

- “A fault tolerant, distributed file system”
- Single metadata (master) server, multiple metadata backup servers, multiple storage (chunk) servers
- Filesystem mount is FUSE (filesystem in userspace) based
- Resilience is provided by per-directory or per-goal “replica goal”
- Large files can be striped across multiple chunk servers and retrieved in parallel
- Compression can be configured on a per-directory basis
- Available as source, compiles and installs easily on CentOS 6.5

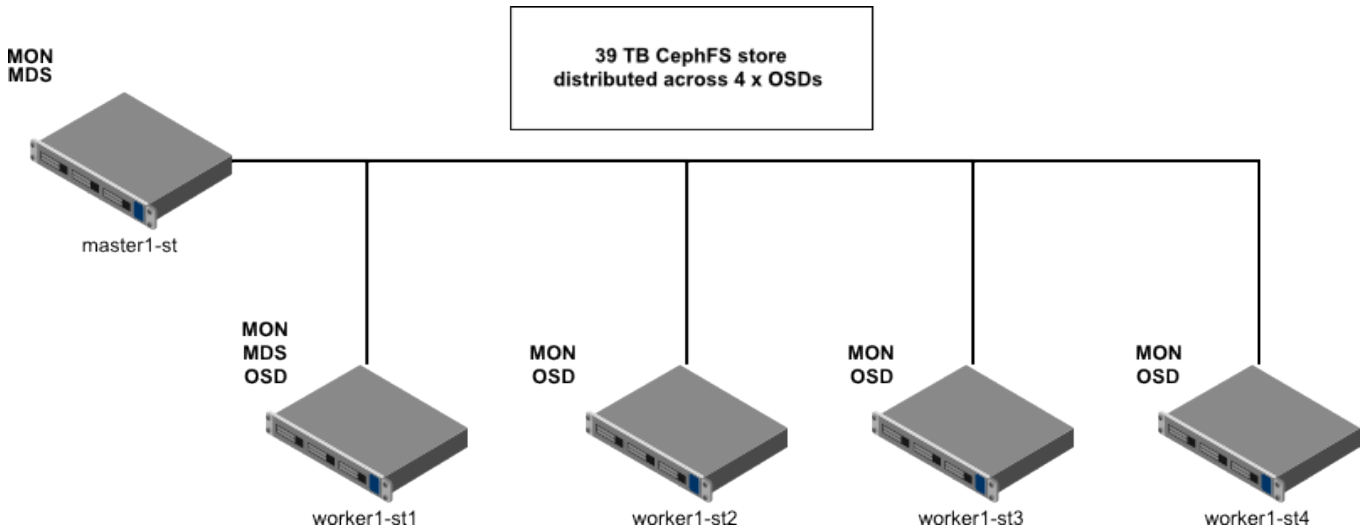
UNIVERSITY of the WESTERN CAPE

CephFS

- A “distributed object store and filesystem”
- Object store based on two daemon types:
  1. monitor daemons to maintain state of object pools
  2. object storage daemons to store and retrieve objects
- Filesystem is built on top of object storage and filesystem semantics are provided by metadata server (can be configured in master/slave group with failover support)
- Historical focus has been on providing block storage interface (RBD) and filesystem labelled “not production ready”
- During deployment on Timon cluster uncovered kernel bug in 3.10 kernel that resulted in inconsistent view of filesystem
  - Bug fixed by patch in 3.12 Linux kernel, but Timon is running on kernel-lt release (v. 3.10)
  - Ceph/CephFS is *new* and it can hurt. Expect rapid evolution of CephFS over next 12 months (due to Redhat purchase).

UNIVERSITY *of the* WESTERN CAPE

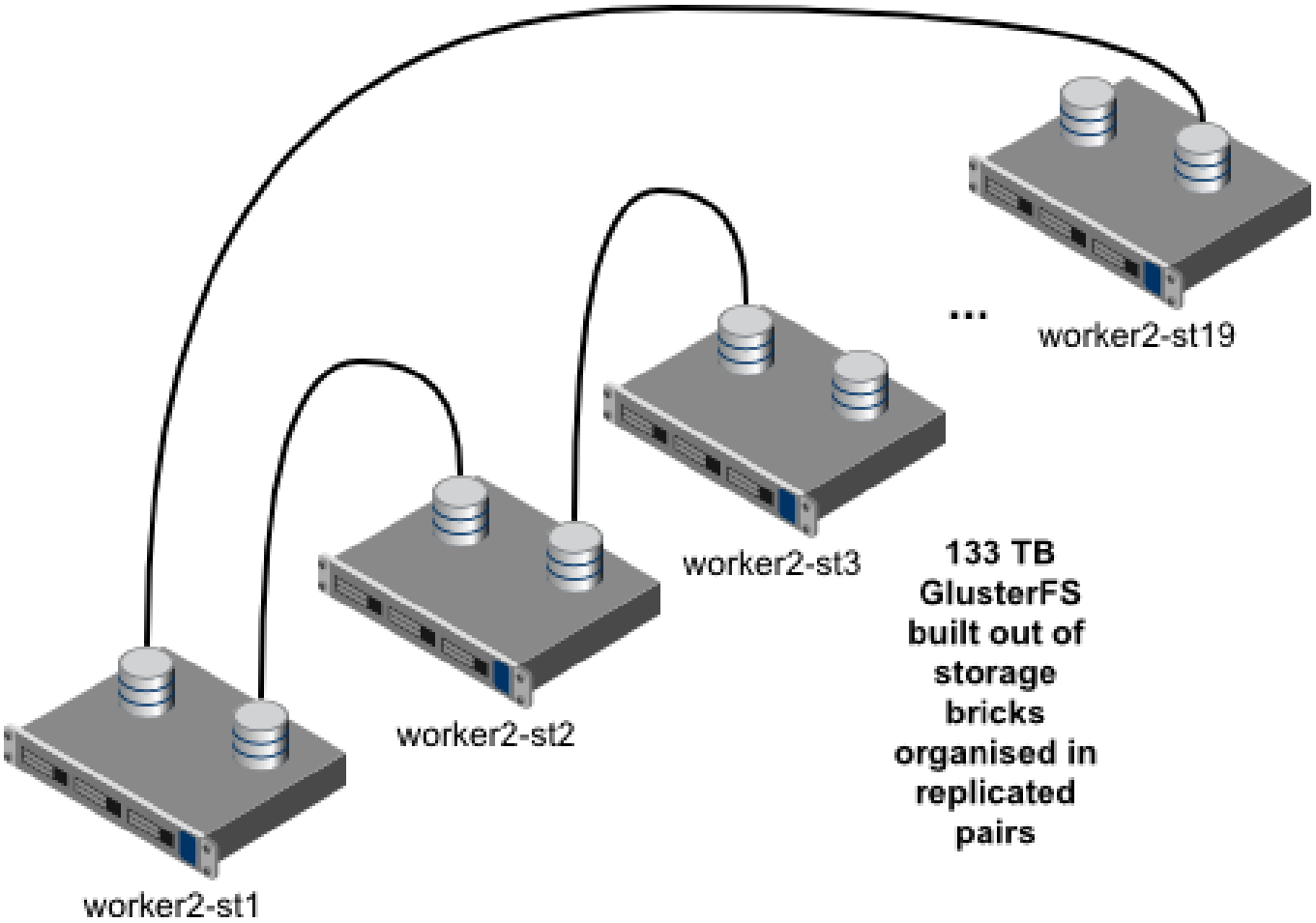
Ceph deployment



GlusterFS

- A “unified, poly-protocol, scale-out filesystem”, very easy to deploy, just two daemons:
  1. The *glusterd* storage daemon: provides interface to on disk storage and stores metadata in filesystem (typically XFS) “extended attributes”
  2. The *glusterfsd* filesystem daemon that coordinates a stack of “translators” and ultimately client access
- Filesystem mount is via NFS, CIFS or as a FUSE mount
- Built around a distributed hash table (DHT), metadata stored along with data
- Replication policy is set per-volume at volume create time and replication largely happens at write time
  - If new servers are added, manual intervention is needed to rebalance replicas across storage
- Metadata lookup potentially queries the entire cluster (see Jeff Darcy’s blog on distribution)

GlusterFS deployment

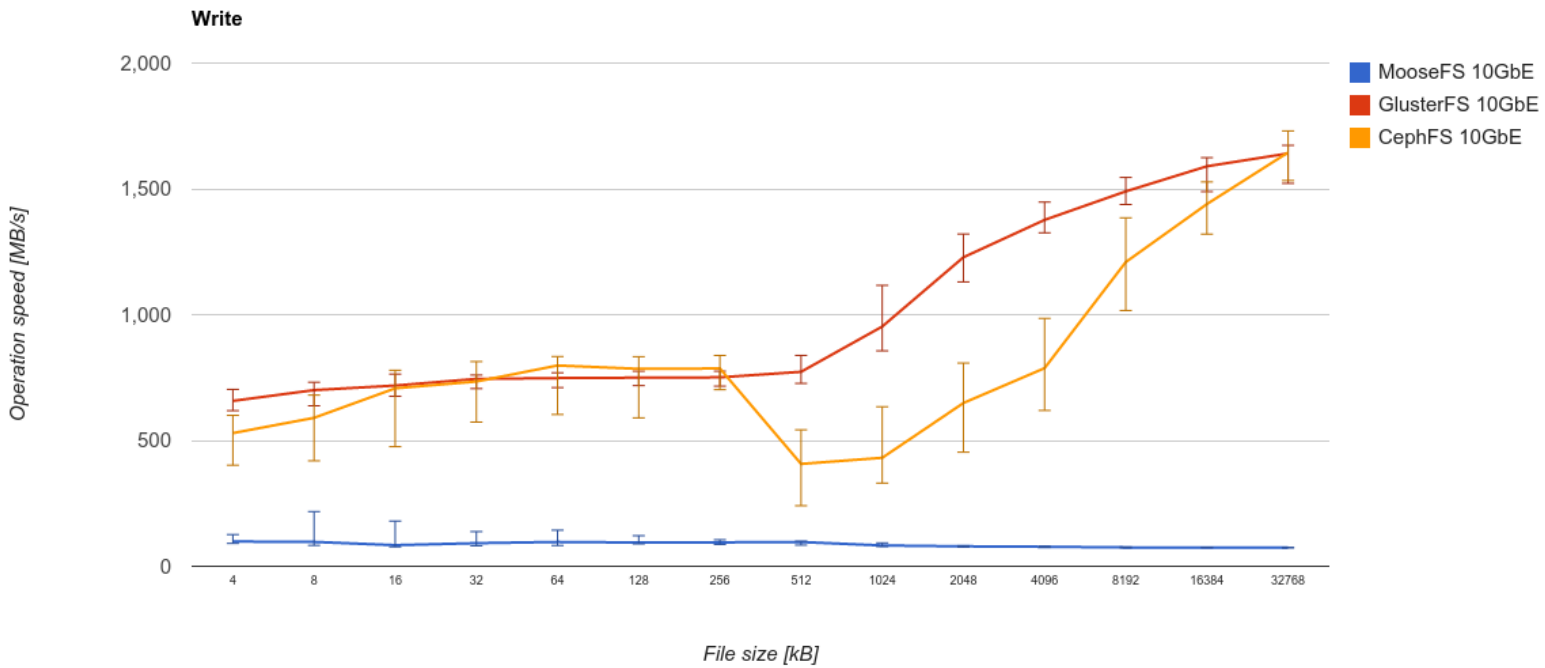


Write and read throughput comparisons

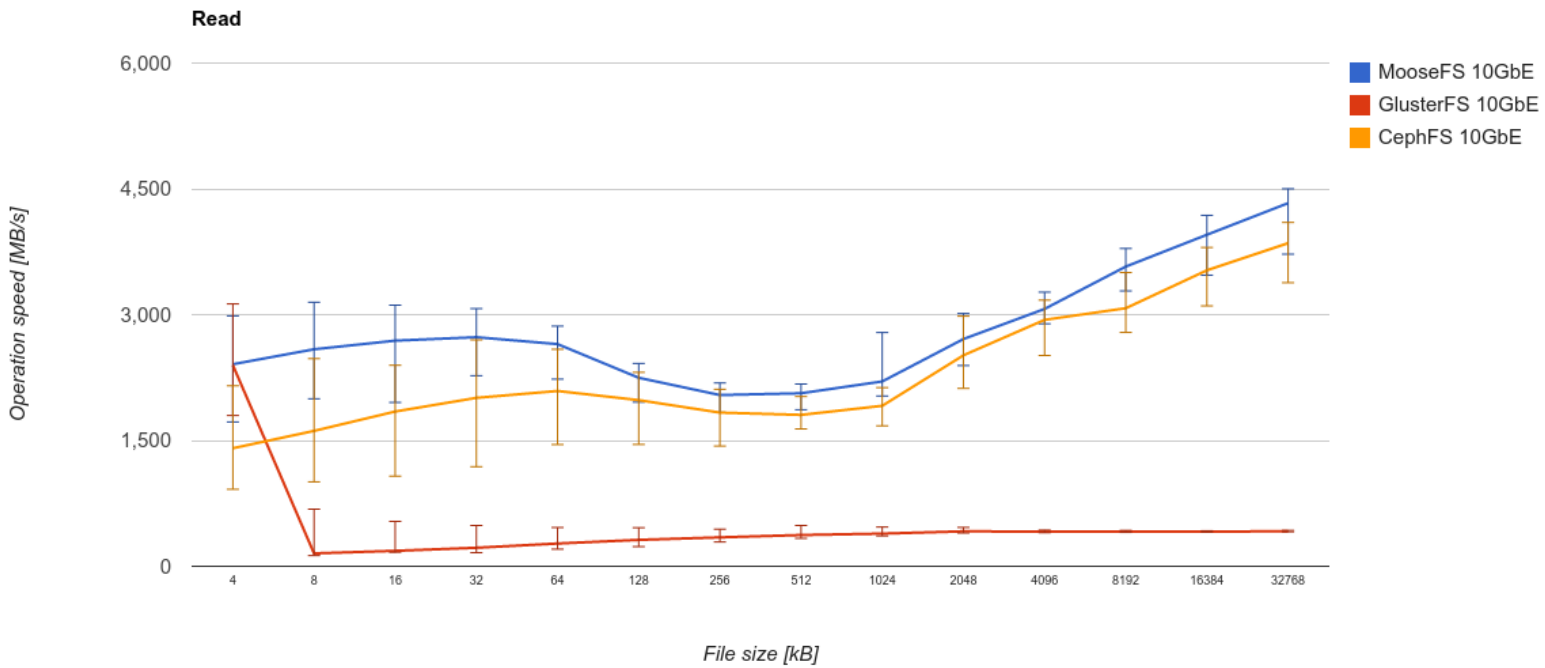
- Tests were done with *iozone*
- Each test was repeated 10 times, and results were compared with modified version of iozone-results-comparator
- Results are still preliminary and require validation and investigation

UNIVERSITY *of the* WESTERN CAPE

Write comparison



Read comparison



Conclusion

- Distributed, direct attached storage offers a viable alternative to a dedicated SAN for a compute cluster
  - Impact of storage CPU usage on HPC / scientific output hasn't been quantified yet
- *CephFS* shows promising performance but currently lacks stability and requires recent Linux kernel
- More research is needed on tuning to maximise performance under different workloads
  - Examining implementation and tradeoffs in DFS would make a good computer science research project

UNIVERSITY *of the* WESTERN CAPE