

GlusterFS: One Storage Server to Rule Them All

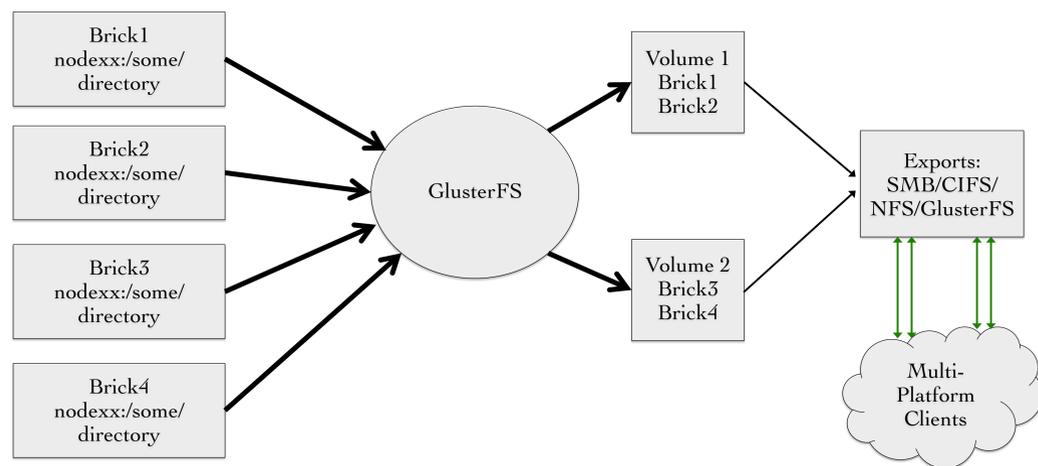
Eric Boyer (Michigan Tech), Matthew Broomfield (New Mexico Tech),
Terrell Perrotti (South Carolina State University)

Mentors: David Kennel (DCS-1), Greg Lee (DCS-1) Instructor: Dane Gardner (NMC)



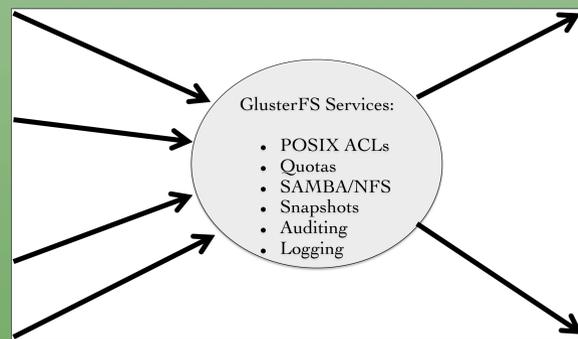
Objective

GlusterFS is an open source distributed filesystem that is designed to be capable of scaling to several petabytes of storage and serving thousands of clients. Clusters with GlusterFS can be comprised of commodity hardware combined with a TCP/IP or Infiniband interconnect to form one large parallel network file system. In this project we evaluate the usability of the GlusterFS storage system as both a high performance storage system and a general purpose storage system.



Procedure

Our team built an 8 node cluster running CentOS 6.2 to test the latest version of GlusterFS, which is 3.3. The nodes were interconnected with 10 Gbps Ethernet and a 1Gbps administrative network. We began by researching and implementing common enterprise storage services and features, as described below. We then tested the read and write performance of 6 different GlusterFS setups for both a 4 node and 8 node GlusterFS volume and averaged the respective data together. We ran many iterations of each test with 1, 4 and 8 nodes mounting the volume. The purpose of these different tests was to observe how performance of the GlusterFS changes as more nodes are added to the GlusterFS and as more clients simultaneously access the volume.



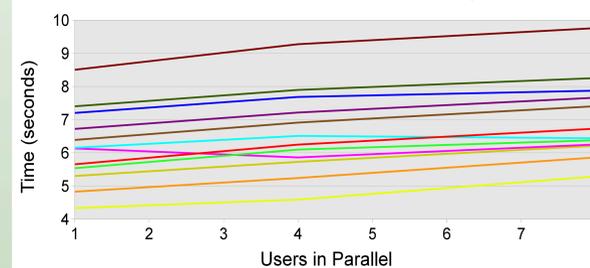
Services

There are several desirable services for enterprise storage that we implemented on GlusterFS. These services were implemented once a volume was mounted on a head node. One concern was controlling who can access certain data, and this was accomplished with POSIX ACLs. These access control lists allow permissions to be set on directories and files. Limiting how much storage users can be given was accomplished through GlusterFS's built-in quotas, which are given on directories. Backups of GlusterFS were achieved through the rsnapshot utility, which can backup entire volumes or specific directories at any given interval. An administrator's ability to see what is happening to the GlusterFS itself and which users are accessing the GlusterFS is very important. We used GlusterFS's built-in logging and the auditd utility to track these things.

Performance

Legend		
	4 Nodes	8 Nodes
Base	■	■
Striped-2	■	■
Striped-4	■	■
Replicate-2	■	■
Replicate-4	■	■
Hybrid-2-2	■	■

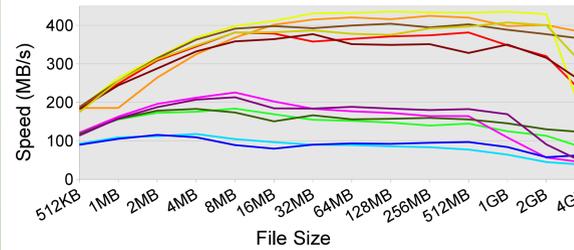
Time to ls a Recursive Directory



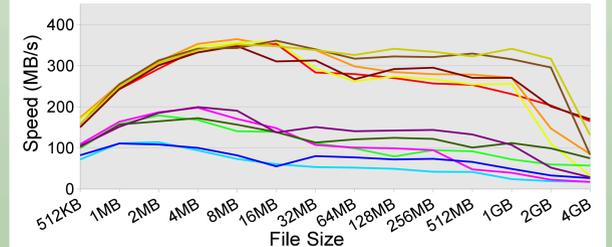
Test Considerations

GlusterFS's performance is greatly affected by the underlying hardware. Our setup used Jumbo Frames, which could produce overhead for smaller file sizes. 6 of our nodes used software RAID 0 partitions for GlusterFS bricks, but two of our nodes did not. This dampens the scalability of any tests that used all 8 nodes. It should also be noted that the client nodes were, in most cases, also held the bricks for our test volumes.

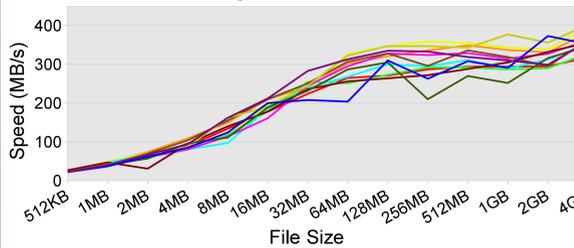
GlusterFS Write Speeds with 4 Users in Parallel Using The dd Command



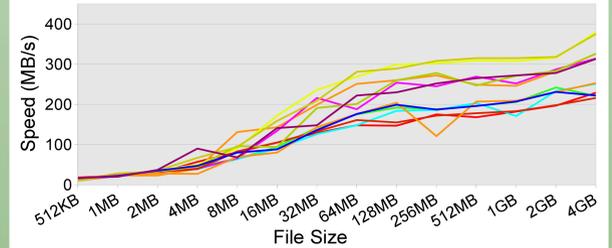
GlusterFS Write Speeds With 8 Users in Parallel Using The dd Command



GlusterFS Read Speeds with 4 Users in Parallel Using the dd Command



GlusterFS Read Speeds with 8 Users in Parallel Using the dd Command



Results

Writing to our GlusterFS test volumes started out slow for small file sizes and peaked around 8MB. From this point, the write speeds for each setup level off and eventually drop off due to each system running out of cache. All striped volumes performed better than all replicated volumes. Read speeds also started out slow, but continued to increase for larger file sizes. Read speeds also showed very little drop off for large files. When testing the time to ls a recursive directory, each test had a unique initial time to complete. All volumes increased at the same linear rate as more users simultaneously executed the command.

Conclusion

GlusterFS, with its built in abilities and easy connection to other services, proved to have widespread capabilities as a virtual file system. It's performance was, however, not quite what we expected. The scalability of GlusterFS is very dependent upon the underlying hardware. Another downfall to GlusterFS is it's current lack of built-in encryption and IP based security paradigm. Although GlusterFS still has future potential for high performance computing, it would currently be best suited in a general purpose computing environment.

Future Research

It is important to research how GlusterFS scales when used across a greater number of nodes. Furthermore, Geo-replication is a lucrative feature of GlusterFS that allows offsite replication of data over the internet. There is also support for Unified File and Object Storage with OpenStack's Swift and Apache Hadoop using GlusterFS as a storage backend. Our limited hardware and time prevented us from testing how the performance of GlusterFS is impacted with changes in RAID types, over Infiniband, and with using other filesystems on top of GlusterFS.