Performance in a Gluster System

Versions 3.1.x

TABLE OF CONTENTS

LIST OF FIGURES

1.0 INTRODUCTION TO GLUSTER

Gluster is a software-only platform that provides scale out NAS for cloud and virtual environments. With Gluster, enterprises can turn commodity compute and storage resources (either on-premise or in the public cloud) into a scale-on-demand, virtualized, commoditized, and centrally managed storage pool. The global namespace capability aggregates disk, CPU, I/O and memory into a single pool of resources with flexible back-end disk options, supporting direct attached, JBOD, or SAN storage. Storage server nodes can be added or removed without disruption to service-- enabling storage to grow or shrink on-the-fly in the most dynamic environments. Gluster is modular and can be configured and optimized for a wide range of workloads. While manufacturers of proprietary storage hardware often trumpet excellent storage performance numbers, the tests that those numbers are based upon are often geared towards a highly idealized environment. The tests will often reflect a single, highly specific workload. Furthermore, the tests are often run on configurations which have been created without regard to cost, using (for example) large amounts of solid state storage, large amounts of CPU and RAM, and multiple, redundant, high speed network cards.

As a scale-out, software only system, Gluster was designed to provide good performance across a wide variety of workloads, and was designed to enable customers to economically achieve very good performance levels under less than ideal conditions. Gluster provides enterprises the ability to easily adjust configurations to achieve the optimal balance between performance, cost, manageability, and availability for their particular needs.

This paper discusses Gluster's approach to performance, the factors that impact performance, and the results that customers should expect when adjusting those factors. The paper includes detailed results from both on-premise and Amazon Web Services configurations. The paper assumes that the reader has already read "An Introduction to Gluster Architecture" available at www.gluster.com.

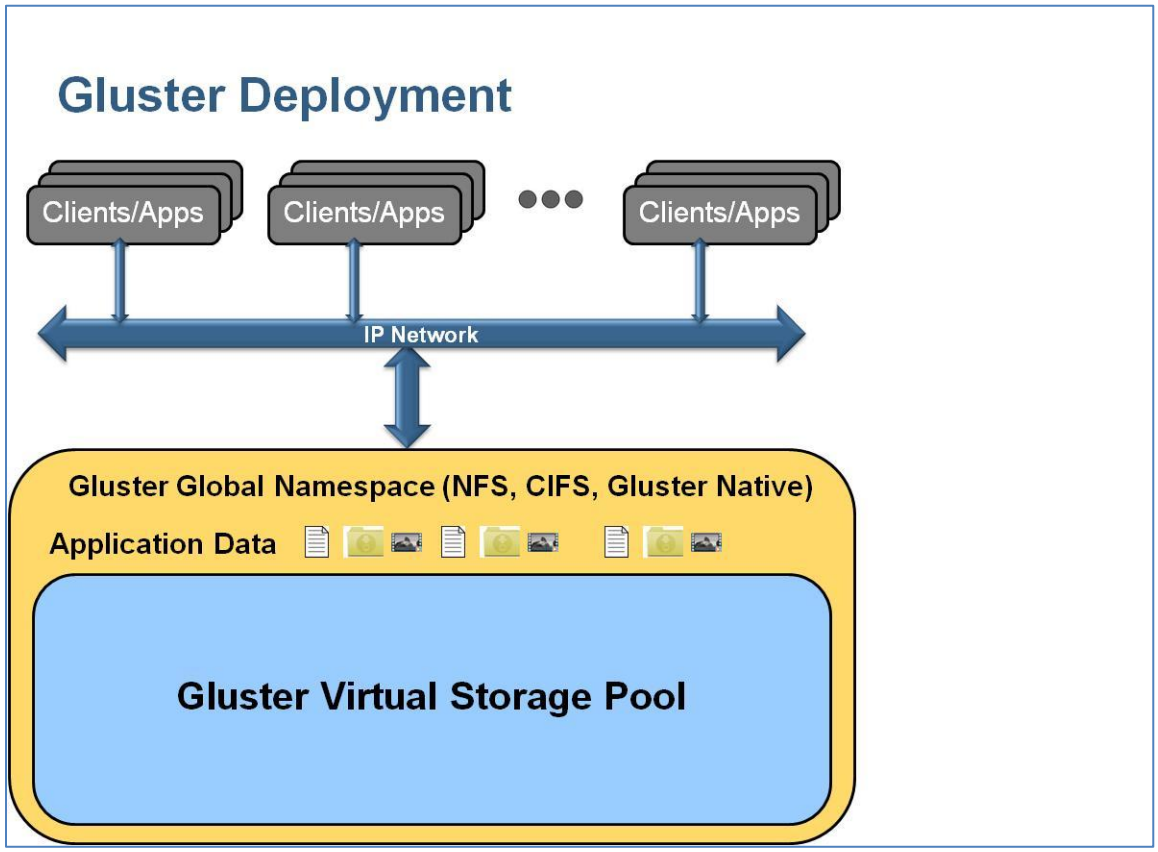


Figure 1: Gluster Deployed – Leverages Global Namespace and Gluster Virtual Storage Pool

2.0 GLUSTER VIEW OF PERFORMANCE

Gluster was designed to achieve multiple performance goals, as outlined below.

2.1 GOOD PERFORMANCE ACROSS A WIDE VARIETY OF WORKLOADS

Gluster is designed to provide a virtualized, commoditized, and centrally managed pool of storage that can be used for a wide variety of storage needs. Since such a pool is generally used for a wide variety of applications, we designed Gluster to perform well across a variety of workloads, including:

- Both large numbers of large files and huge numbers of small files
- Both read intensive and write intensive operations
- Both sequential and random access patterns
- Large numbers of clients simultaneously accessing files

While Gluster's default configuration can handle most workloads, Gluster's modular design allows it to be customized for particular and specialized workloads.

2.2 SCALE OUT PERFORMANCE

As a scale out system, Gluster is designed to distribute the workload across a large number of inexpensive servers and disks. This reduces the impact of poor performance of any single component, and dramatically reduces the impact of factors that have traditionally limited disk performance, such as spin time. In a typical Gluster deployment, relatively inexpensive disks can be combined to deliver performance that is equivalent to far more expensive, proprietary and monolithic systems at a fraction of the total cost. To scale capacity, organizations need simply add additional, inexpensive drives and will see linear gains in capacity without sacrificing performance. To scale out performance, enterprises need simply add additional storage server nodes, and will generally see linear performance improvements. To increase availability, files can be replicated n-way across multiple storage nodes.

A simplified view of Gluster's architecture deployed on-premise, and within a virtualized environment, can be seen in Figure 2, below. For a more detailed explanation please see "An Introduction to Gluster Architecture" available at www.gluster.com

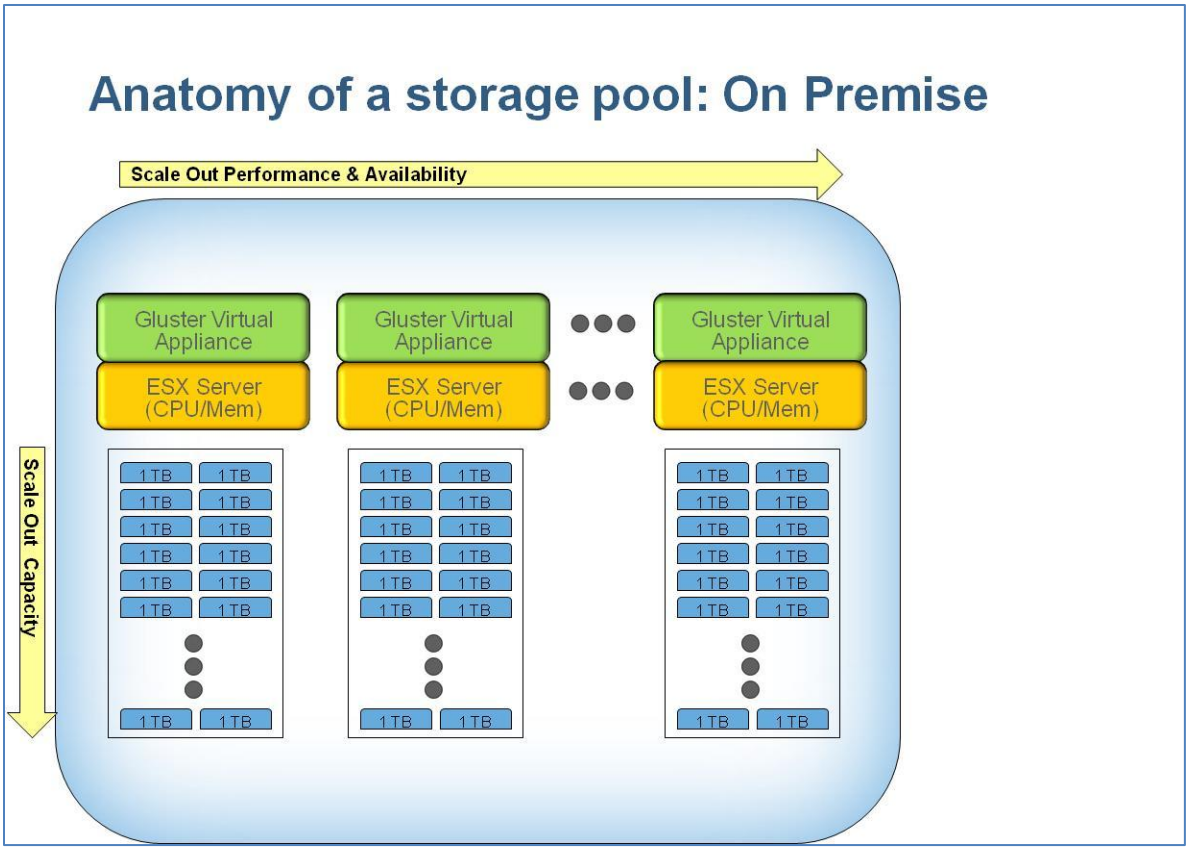


Figure 2: Gluster Storage Pool: On Premise

As Gluster does not rely upon hardware to optimize performance, we have implemented a number of techniques within our software architecture to improve performance. The most important of these is our unique, no metadata model, described more fully in "An Introduction to Gluster Architecture." Most other scale-out systems separate metadata from data, using either a centralized or distributed configuration. This not only creates a point of failure, it also creates a performance bottleneck that grows worse as the number of disks, storage nodes, or files increase. It also creates severe performance issues for workloads with large numbers of small files, as smaller the average file size results in a higher ratio of metadata to data. Indeed, such systems often simply cannot work with files below 1 MB in size.

In addition to the no metadata model, other performance enhancing techniques employed by Gluster include:

- read ahead
- write behind
- asynchronous I/O
- intelligent I/O scheduling
- aggregated operations

For implementations using Gluster's Native Filesystem in User Space (FUSE) client, performance is further enhanced for most applications via the use of parallelism (as opposed to NFS or CIFS). Each client accesses each storage server node, with no need for inter-node communication. In this sense, Gluster's native FUSE protocol is similar to the proposed—but not yet released—parallel NFS (pNFS) standard.

2.3 PRICE PERFORMANCE

By aggregating the disk, CPU, memory, and I/O resources of large numbers of inexpensive components, Gluster aims to make it possible to deliver a wide variety of performance levels at a fraction of the cost of proprietary, scale up systems. Further, Gluster's unique no-metadata model eliminates the need for us to use expensive solid state or multiple storage interconnects to maintain performance and coherency. While Gluster can be configured to offer speeds of up-to 2 GB/s per storage node, we think it is far more important to be able to deliver great price performance across a wide range of performance levels.

2.4 RELIABILITY TRUMPS PERFORMANCE

Gluster has chosen to avoid certain techniques that enhance performance at the risk of introducing issues with data integrity or availability. For example, Gluster Native client avoids the use of write caching, as it introduces cache coherency problems. Instead, we leverage safer techniques such as those discussed in 2.2.We also have implemented a number of techniques to improve reliability, such as

- **No separation of meta-data and data:** As a filesystem scales, the meta-data database is the part, most prone to corruption. Gluster eliminates meta-data servers all together.
- **User Space:** Gluster is implemented as a user space storage operating system. Its advantages are similar to virtual machine implementations.
- **Stackable design:** Most of the functionalities (from volume manager to networking stack) are implemented as self-contained stackable modules. Deploying as a modular architecture enforces a very tight discipline while adding new features. Parts of the file system can be selectively turned on and off at run time,
- **Proven disk filesystem backend:** Gluster uses proven disk file systems such as Ext3/4 to store data persistently.

3.0 FACTORS IMPACTING PERFORMANCE

Several factors impact performance in a Gluster deployment.

3.1DISKS, STORAGE NODES, NETWORK SPEEDS

3.1.1 DISK SPEED

In most scale up systems, disk speed and seek time are the biggest determinant of performance. While the use of solid state drives obviously eliminates the impact of spin time, such drives are generally 3-10 times as expensive per GB as low end SATA drives.

In a typical Gluster configuration, workload is spread across a large number of 1 or 2 TB drives. Thus, the spin time of any individual drive becomes largely irrelevant. For example, Gluster has been able to achieve upwards of 16 GB/s read throughput and 12 GB/s write throughput in an 8 node storage cluster created entirely using 7.2 K RPM SATA drives. For most customers, therefore, the scale out approach eliminates the need for expensive drives or complicated tiering systems.

In certain circumstances characterized by very high performance requirements and very low capacity requirements (e.g. risk modeling), Gluster can be used in combination with SSDs for exceptionally high performance. In a recent configuration with Gluster in conjunction with Fusion I/O, the customer was able to achieve 5.6 GB/s write throughput per node (2.3 GB/s replicated).

3.1.2 # OF DISKS

In most scale-out systems with a centralized or distributed metadata server, adding disks often leads to performance degradation or to non-linear gains in capacity. With Gluster, adding additional disks to a storage node will result in linear gains in effective capacity, and will generally result in either neutral to moderately positive impacts on performance.

3.1.3 # OF STORAGE NODES

Gluster performance is most directly impacted by the number of storage nodes. Generally speaking, distributing the same number of disks among twice as many storage nodes will double performance. Performance in a Gluster cluster increases near-linearly with the number of storage nodes; an 8 storage node cluster will deliver approximately 4 times the throughput of a 2 storage node cluster.

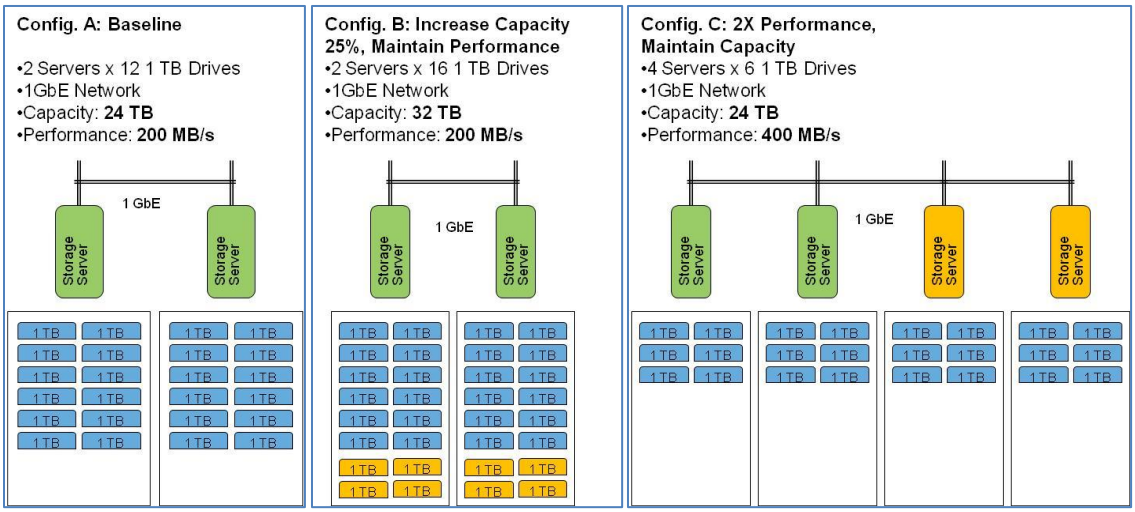
3.1.4 NETWORK

Generally speaking, by the time a system has 8 storage nodes, the network becomes the bottleneck, and a 1 GbE system will be saturated. By adding a 10GbE or faster network, you will achieve faster per node performance. As noted above, we have been able to achieve 16 GB/s read throughput and 12 GB/s write throughput in an 8 storage node cluster using low end SATA drives when configured with a 10GbE network. The same cluster would achieve approximately 800 MB/s of throughput with a 1 GbE network.

3.1.5 SUMMARY IMPACT OF ADJUSTING DISKS, STORAGE NODES, AND NETWORK

To illustrate how Gluster scales, the figures below show how a baseline system can be scaled to increase both performance and capacity. The discussion below uses some illustrative performance and capacity numbers; detailed results can be found in section 4.

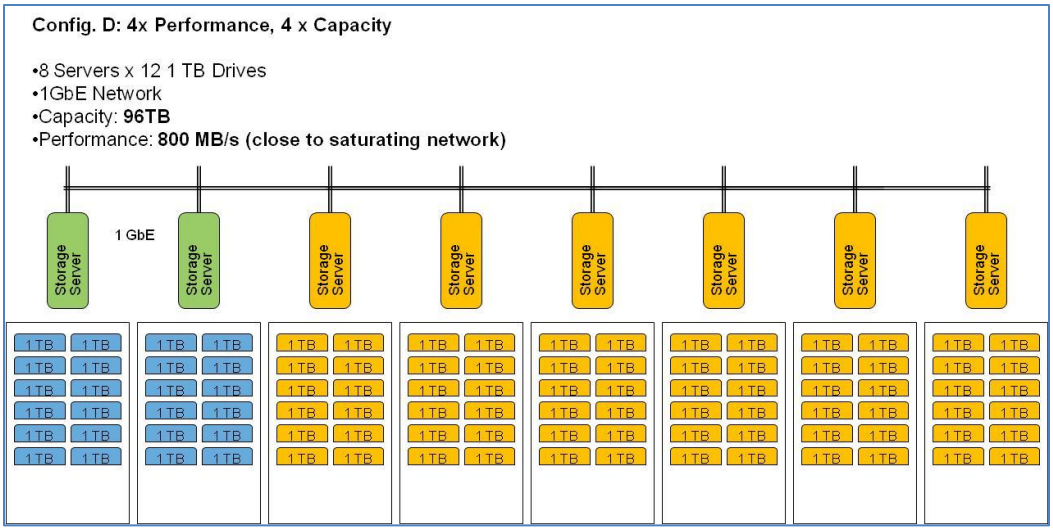
A typical direct attached Gluster configuration will have a moderate number of disks attached to 2 or more server/storage nodes which act as NAS heads. For example, to support a requirement for 24 TB of capacity, a deployment might have 2 servers, each of which contains a quantity of 12 X 1 TB SATA drives. (See Config A, below).



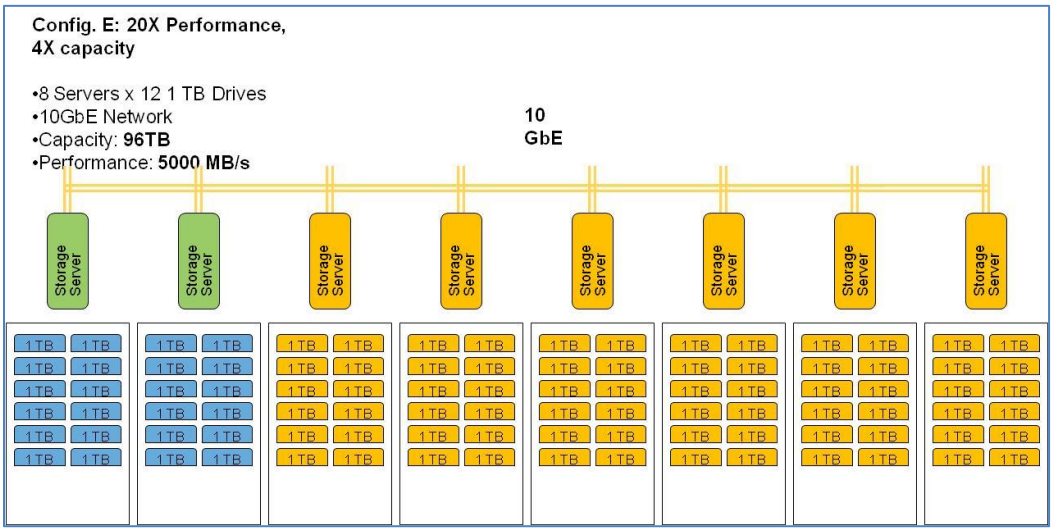
If a customer has found that the performance levels are acceptable, but wants to increase capacity by 25%, they could add another 4 X 1 TB drives to each server, and will not generally experience performance degradation. (i.e., each server would have 16 X 1 TB drives). (See Config. B, above). Note that an enterprise does not need to upgrade to larger, or more powerful hardware to increase capacity; they simply add 8 more inexpensive SATA drives.

On the other hand, if the customer is happy with 24 TB of capacity, but wants to double performance, they could distribute the drives among 4 servers, rather than 2 servers (i.e. each server would have 6 X 1 TB drives, rather than 12 X 1 TB). Note that in this case, they are adding 2 more low-priced servers, and can simply redeploy existing drives. (See Config. C, above)

If the enterprise wants to both quadruple performance and quadruple capacity, they could distribute among 8 servers (i.e. each server would have 12 X 1 TB drives). (See Config. D, below)



Note that by the time a solution has approximately 10 drives, the performance bottleneck has generally already moved to the network. (See Config. D, above). Therefore, in order to maximize performance, we can upgrade from a 1 Gigabit Ethernet network to a 10 Gigabit Ethernet network. Note that performance in this example is more than 20x the performance of the baseline. (See Config. E, below)



As you will note, the power of the scale-out model is that both capacity and performance can scale linearly to meet requirements. It is not necessary to know what performance levels will needed 2 or 3 years out. Instead, configurations can be easily adjusted as the need demands.

3.2 OTHER FACTORS IMPACTING PERFORMANCE

3.2.1 RAM AND CPU ON THE NODES

Generally speaking, Gluster does not consume significant compute resources from the storage nodes themselves. (Our reference configuration uses single quad core 2.6 GHz Xeon E5640 processor with 8GB of RAM). However, if you have a cache-intensive application (e.g. multiple reads of the same of the same file) adding additional memory can help.

Similarly, since Gluster runs in user space, if you want to run other applications concurrently on the same physical storage node, it may make sense to provide more CPU and memory.

In a cloud environment, Gluster recommends running on an “m-1 large” equivalent or more powerful configuration.

3.2.2 # OF CLIENTS

Gluster is designed to support environments with large numbers of clients. Since the I/O of individual clients is often limited, system throughput is generally greater if there are 4 times as many clients as servers.

3.2.3 USE OF GLUSTER VIRTUAL APPLIANCES VS. BARE METAL

For most commercial, on-premise customers, Gluster recommends deploying in a virtual appliance rather than as a file system on bare metal. The advantages of a virtual appliance are generally greater manageability and much lower risk of issues due to mis-configuration. We have found that the per-storage node impact of using Gluster in a virtual appliance is generally less than 5% per node for the most common operations. Performance in a virtual appliance generally scales linearly across nodes, which reduces any performance impact from virtualization to negligible levels.

3.2.4 USE OF CLOUD VS. ON-PREMISE

Gluster is designed to work both on-premise and in public cloud environments, such as Amazon Web Services. While the use of public cloud services brings many advantages, special care must be taken to ensure that acceptable performance levels can be achieved. In many cloud environments, the performance of both server instances and storage instances is often highly variable, especially if those instances are shared with other users of the cloud.

Gluster’s commercial cloud offerings (e.g. the Gluster Amazon Machine Image) reduce the impact of storage block variability (e.g. EBS) by distributing the storage for any given instance across 8 EBS blocks. In a properly functioning environment, Gluster performance in the cloud is approximately 40- 50% per server instance (e.g. EC2) of what one can expect per physical node for a comparably configured on-premise solution.

Gluster performance in the cloud can also be impacted by the performance of compute instances. As noted above, node RAM and CPU is generally not a performance-determining factor. However, if a compute instance is shared with others users and applications in the cloud, there is a possibility that the shared instance performance will be a factor. In such cases, Gluster provides tools for diagnosing and migrating to new instances (which does not require data migration.)

Since there is virtually no limit to the number of instances or blocks that can be provisioned inside a Gluster Cloud Cluster, the scale out performance combined with the on-demand nature of cloud services makes Gluster an attractive choice even for high performance applications, especially if those applications are elastic in nature (e.g. running simulations).

Of course, if clients or applications are not also located in the cloud, Internet latency itself is likely to be a significant factor.

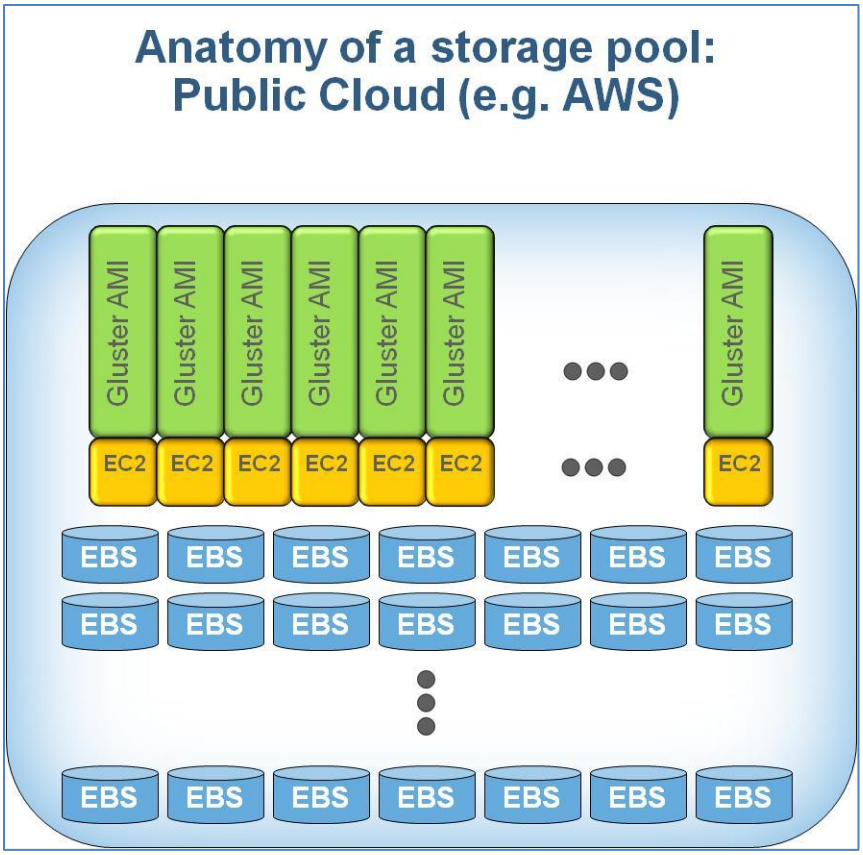


Figure 3: Representative Cloud Deployment

3.2.5 READ VS. WRITE

As is to be expected, write operations are generally slightly slower than read operations for small number of storage nodes. At large numbers of storage nodes, however, writes can be faster than reads, especially with non-sequential reads.

Gluster Native protocol does *not* implement write caching, as we believe that the modest performance improvements from rite caching do not justify the risk of cache coherency issues.

3.2.6 REPLICATION

Gluster is designed to support n-way synchronous replication. If a system is configured for two-way, active/active replication, write throughput will generally be half of what it would be in a non-replicated configured. However, read throughput is generally improved by replication, as reads can be delivered from either storage node.

3.2.7 FILE SIZE

Most storage systems rely on a centralized or distributed metadata store for location metadata. In such an environment, as file and block size is reduced, the ratio of metadata to data increases, and performance degrades significantly. Many such systems require file size to be 1MB or larger. In fact, many systems simply cannot work with files below a certain size, as the metadata would exceed available limits.

Gluster's unique no-metadata model eliminates this bottleneck. Gluster's single storage node performance with Gluster Native Protocol is generally consistent down to file sizes of about 64K. For size smaller than this, Gluster provides good IOPS results. Changing file and block impact Gluster NFS and Gluster Native FUSE differently as discussed in the next section.

With very small file sizes, network speed becomes a factor much sooner than at larger file sizes. Therefore, for high performance applications involving small files, especially those that use NFS or CIFS, we recommend the use of 10GbE rather than 1GbE.

3.2.8 BLOCK SIZE

Gluster generally delivers consistent throughput performance down to block sizes of about 64K.

3.2.9 NFS VS. GLUSTER NATIVE

Generally speaking, Gluster's Native FUSE client will deliver better read and write performance than Gluster NFS across a wide variety of block sizes. However, Gluster NFS will deliver better write performance at small (<32 K) block sizes because of its kernel based write caching.

Native client write performance is sensitive to changes in block size, and is not sensitive to changes in file size. For NFS, both read and write performance are not sensitive to changes in block size, but are sensitive to changes in file size. The reason that this happens, is that NFS clients perform write caching in addition to read caching, while the Gluster native client only caches reads (i.e. we do this to guarantee data integrity).

3.2.10 OTHER OPERATIONS

While read and write operations are generally the most important operations for day-to-day performance, you may want to look at the results in 4.6 to understand the relative performance of other operations. Although it is common to copy a large number of files in order to assess performance, it is important to note that a typical copy operation has many more I/O operations than a typical write, and results should be understood in this context. Also, the copy operation is serial and involves 4k block I/O. Most file systems cache these operations and write them in background. Gluster avoids write caching for coherency reasons and often performs additional data-integrity checks. This performance penalty outweighs its benefits. Gluster 3.2 will introduce new features to instantly query the file system to list changes to files and folders and disk usage metrics without crawling Petabytes of information. Cloud developers can utilize this new API instead of slow and serial Unix commands such as "rsync", "du" and "find".

3.2.11 MISCONFIGURATION

As should be clear from the discussion above, there are many factors that impact performance. In addition to the factors discussed above, operating system choice, network firmware, bad NIC cards, etc. can all play a factor in performance. Gluster comes with tools to help diagnose such issues. In addition, we offer performance tuning as a paid professional service.

4.0 RESULTS OF TESTS

4.1 CONFIGURATION DETAILS

The test results shown below were obtained for the 3.1.2 version of Gluster, installed variously on bare metal, in the Gluster Virtual Storage Appliance for VMWare, and in Amazon Web Services as the Gluster Amazon Machine Image. The test runs were generally either IOZONE or the Lawrence Livermore National Lab IOR test. Details for most of the individual test can be found below.

Unless otherwise indicated, the interconnect was 1GbE. There were generally 8 clients in the test, except for the high speed tests which had 32 clients.

Configuration for Gluster in a non-virtualized environment (bare metal).

- Dell PowerEdge 1950
- 2 x Quad-core Intel Xeon E5430 @ 2.66GHz, 8GB memory.
- Storage controller: Dell PERC 6/E
- Datastores: 6 x 1.8TB, each datastore is 2 disks, RAID-0 Broadcom Corporation NetXtreme II BCM5708 Gigabit Ethernet
- CentOS 5.5

Configuration for Gluster Virtual Storage Appliance for VMWare:

- One Gluster Virtual Appliance was installed per physical ESX host.
- ESX Host configuration:
    - Dell PowerEdge 1950
    - 2 x Quad-core Intel Xeon E5430 @ 2.66GHz, 8GB memory.
    - Storage controller: Dell PERC 6/E
    - Datastores: 6 x 1.8TB, each datastore is 2 disks, RAID-0 Broadcom Corporation NetXtreme II BCM5708 Gigabit Ethernet
    - ESX version: 4.1.0
- Client configuration:
- Dell PowerEdge 1950
- Quad-core Intel Xeon E5405 @ 2.00GHz, 2GB memory.
- Broadcom Corporation NetXtreme II BCM5708 Gigabit Ethernet

Configuration for Amazon Web Services:

- Each server is an m1.large instance running the Gluster AMI.
- Linux software RAID-0 (md) 8 x 10GB EBS volumes.

- Each client is an m1.large instance running the Gluster AMI.

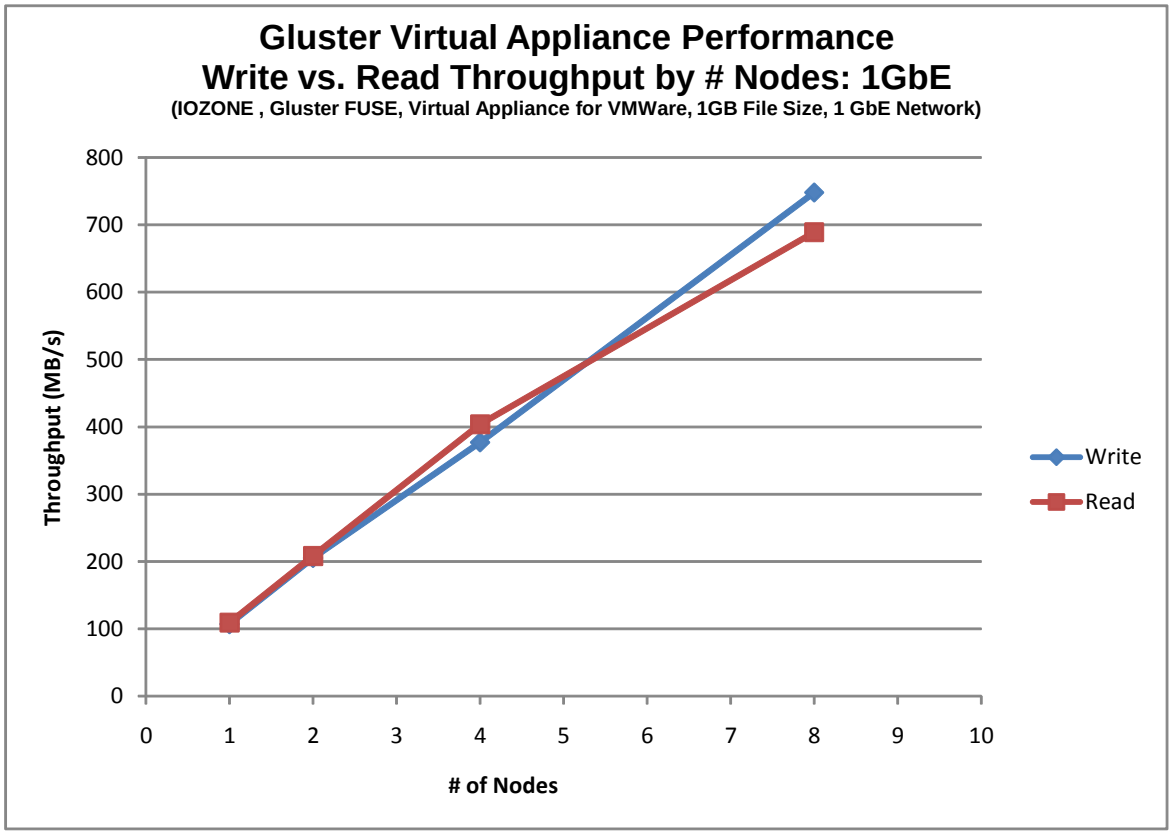
Gluster Virtual Storage Appliance for VMWare was installed in an environment with 1 virtual appliance per physical nodes, 6 X 1TB 7.2K RPM SATA drives per node, and 8 clients. 1GbE interconnect was used for the trials, which used the IOR tests described in section 5.0.

High Speed Network Test

For the highspeed network test, the above configuration (bare metal) was conducted using IOZONE, as described in section 5.0. The 10GbE interconnect was a Dell PowerConnect 8024, 24 10 GbE Ports, Four Combo Ports

4.2 SCALE OUT READ & WRITE (VIRTUAL APPLIANCE)

As can be seen in the test results in Figure 4 below, write throughput scales linearly to 8 storage nodes and saturates the network at 8 nodes. Read scales near linearly, and has almost saturated a 1 GbE network by 8 nodes.

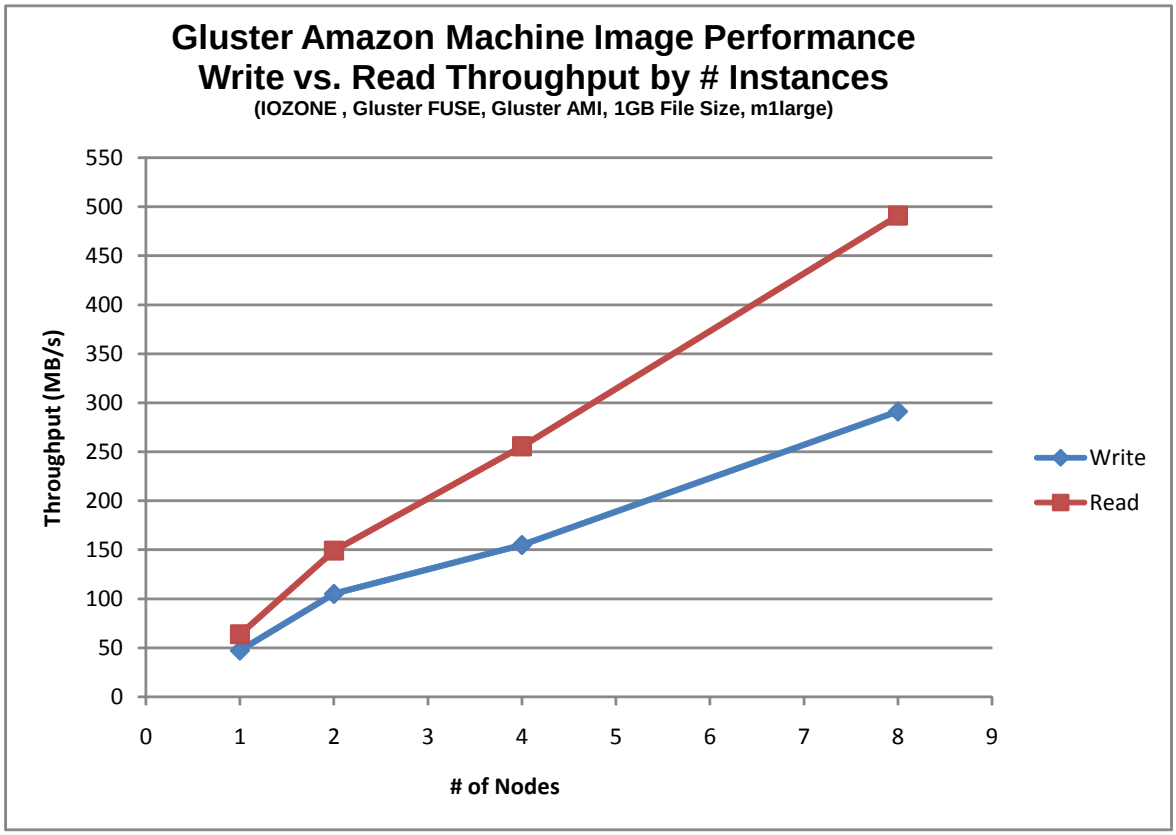


# Nodes	Write	Read
1	107.08	109.5
2	205.31	208.23
4	377.05	404.08
8	748.23	689.21

Figure 4: Scale Out Read & Write: Virtual Appliance

4.3 SCALE OUT READ & WRITE (AMAZON MACHINE IMAGE)

As can be seen in the test results in Figure 5 below, for Gluster in an Amazon Web Services environment, read throughput scales linearly to 8 instances, at a slope greater than 1.0. Write throughput scales near linearly to 8 instances, albeit at a slope of approximately 0.8..

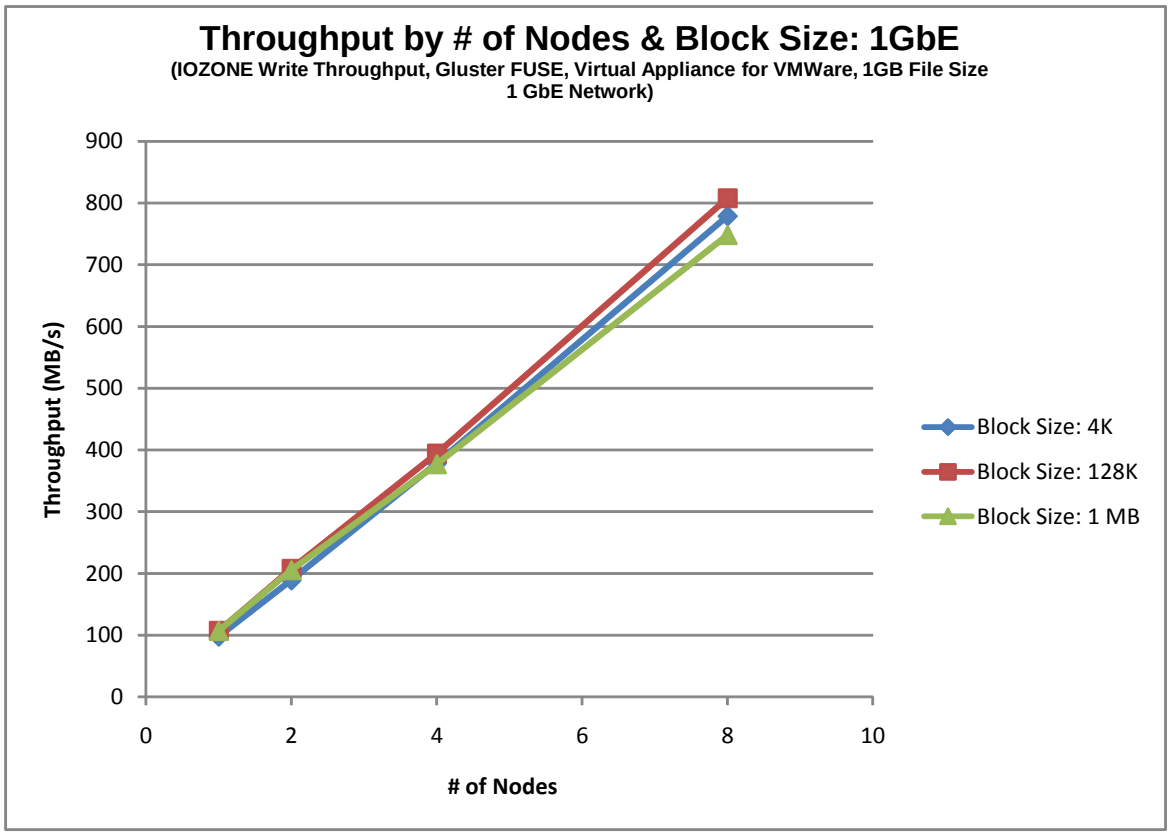


# Instances	Write	Read
1	47.5	64.05
2	105.25	149.35
4	155.1	255.74
8	291.3	490.95
slope	0.8	1.0

Figure 5: Scale Out Read & Write: Amazon Machine Image

4.4 IMPACT OF BLOCK SIZE

As can be seen by the results in Figure 6 below, changing block size generally has a negligible impact on scale out performance, assuming that the block size is smaller than the file size.

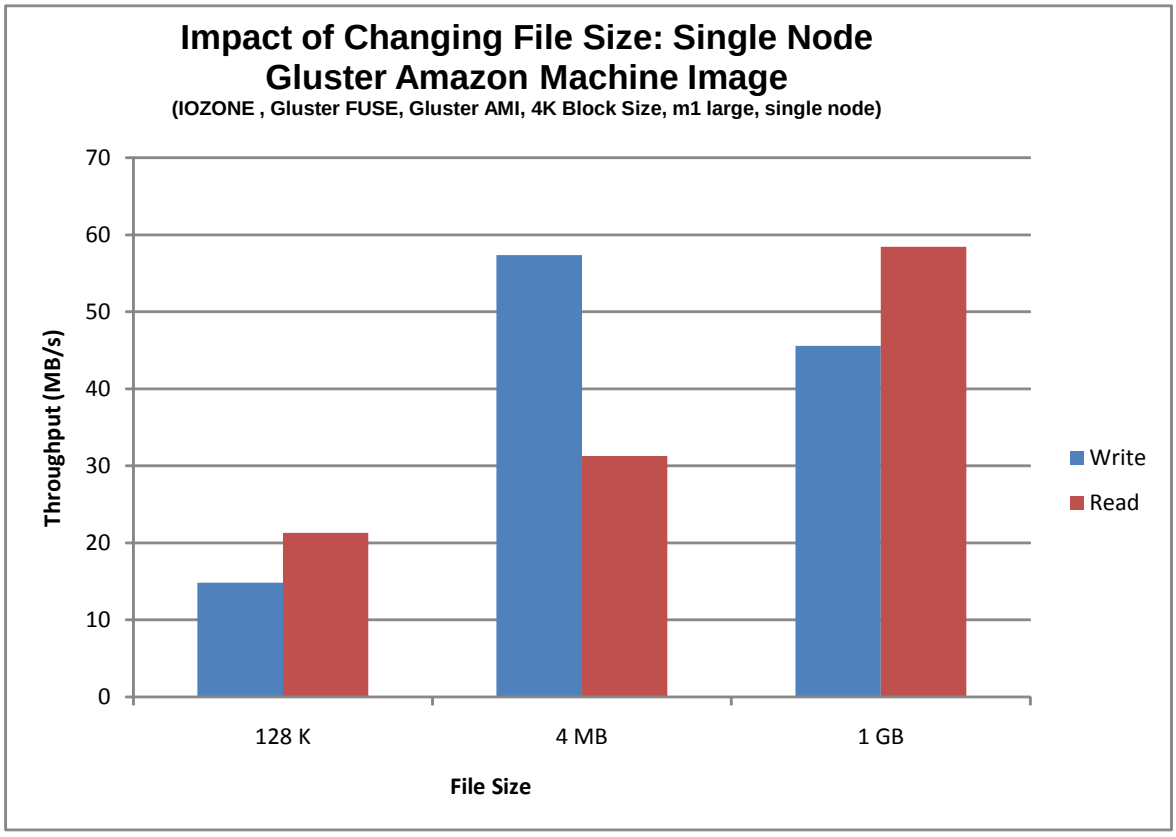


# Nodes	Block Size: 4K	Block Size: 128K	Block Size: 1 MB
1	97.96	107.19	107.08
2	189.08	207.41	205.31
4	378.56	394.4	377.05
8	779.07	807.8	748.23

Figure 6: Impact of Changing Block Size

4.5 IMPACT OF FILE SIZE

As can be seen in Figure 7 , below, changing file size has a material impact on Gluster Native FUSE performance per storage node. However, single node performance is still very good for small file sizes, even in the Amazon Environment, and increases with the number of storage nodes.

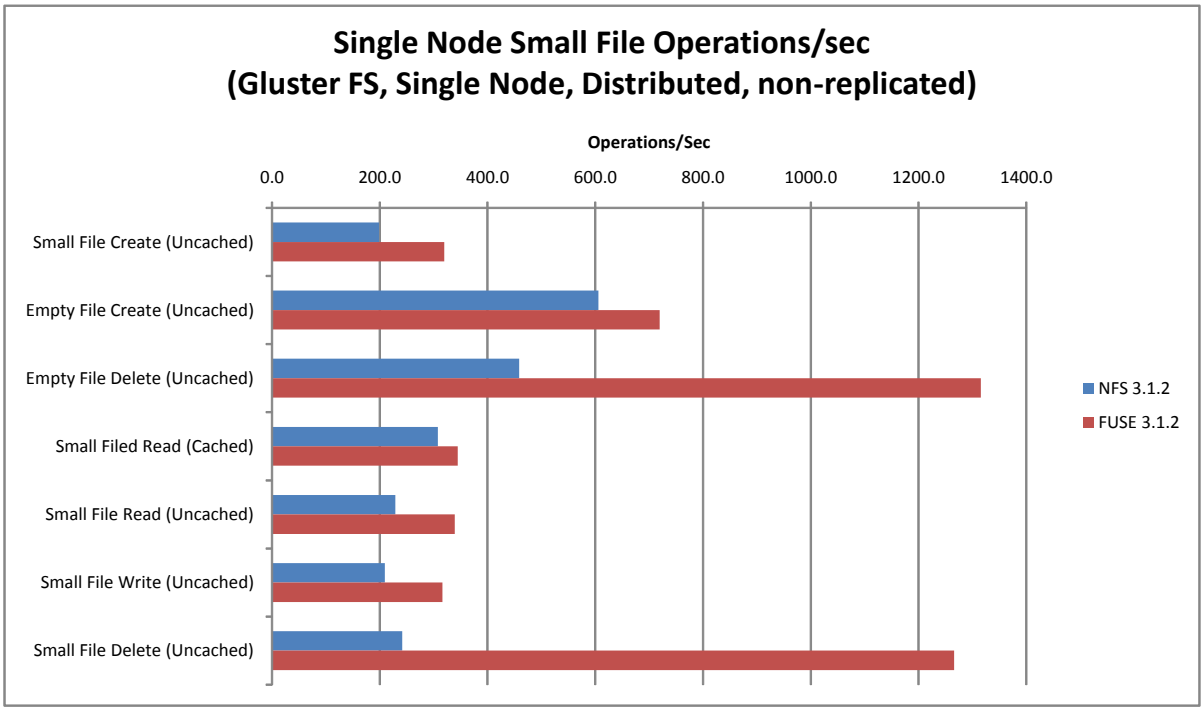


File Size	Write	Read
128 K	14.85	21.31
4 MB	57.37	31.29
1 GB	45.59	58.46

Figure 7: Impact of Changing File Size (AWS)

4.6 SMALL FILE OPERATIONS

As can be seen in Figure 8 below, Gluster delivers good single storage node performance for a variety of small file operations. Generally speaking, Gluster Native FUSE will deliver better small file performance than Gluster NFS, although Gluster NFS is often better for very small block sizes. Perhaps most important, IOPS performance in Gluster scales out just as throughput performance scales out.

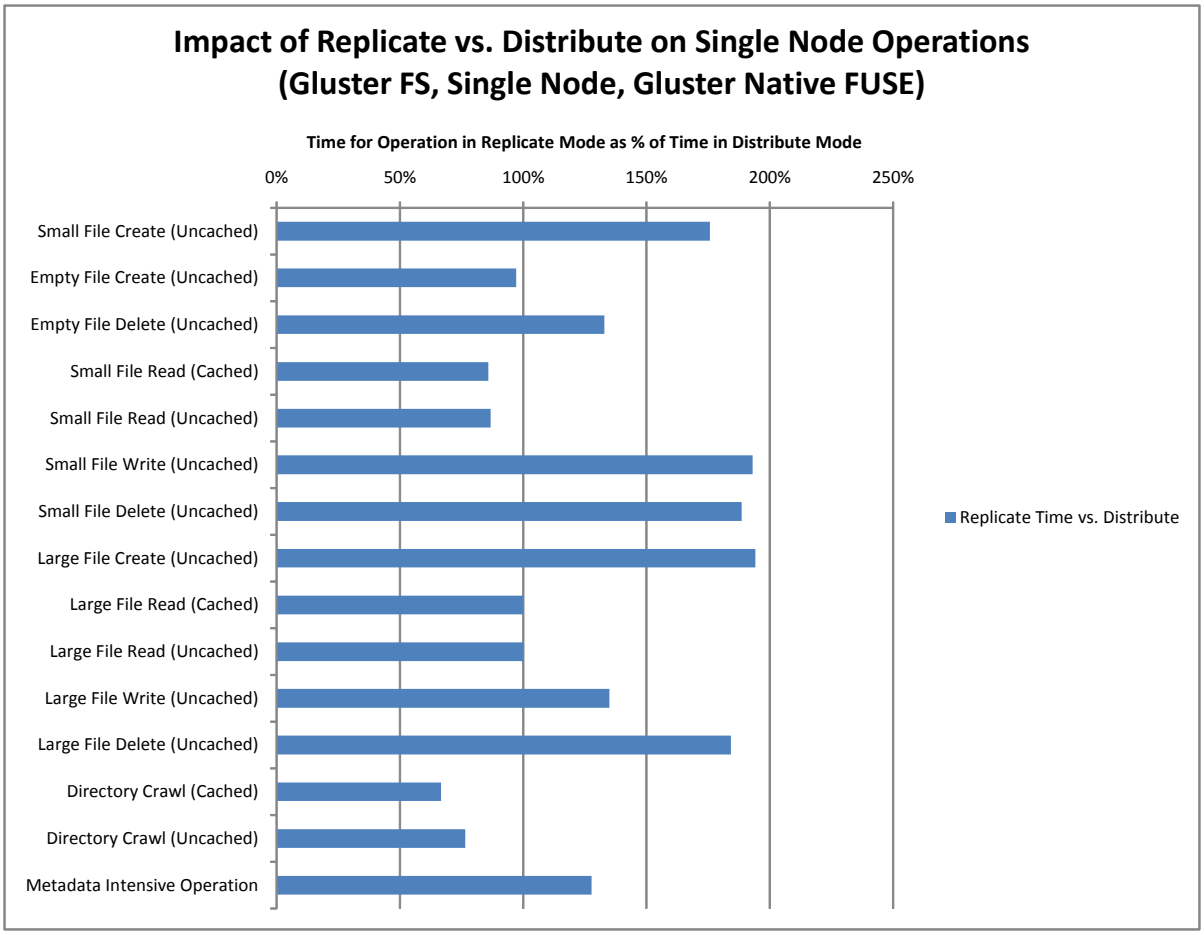


Small File (4k) performance workload	Operations/Sec	
	NFS 3.1.2	FUSE 3.1.2
Small File Create (Uncached)	198.8	319.5
Empty File Create (Uncached)	606.1	719.4
Empty File Delete (Uncached)	458.7	1315.8
Small Filed Read (Cached)	307.7	344.8
Small File Read (Uncached)	228.8	339.0
Small File Write (Uncached)	209.6	316.5
Small File Delete (Uncached)	241.5	1265.8

Figure 8: Small File Operations, FUSE vs. NFS

4.7 IMPACT OF REPLICATION

As can be seen in Figure 9 below, write and create operations are generally half the speed in a replicated environment than in a purely distributed environment. Read and delete operations are generally unaffected to slightly faster in a replicated environment.



Workload	Replicate Time vs. Distribute
Small File Create (Uncached)	176%
Empty File Create (Uncached)	97%
Empty File Delete (Uncached)	133%
Small File Read (Cached)	86%
Small File Read (Uncached)	87%
Small File Write (Uncached)	193%
Small File Delete (Uncached)	189%
Large File Create (Uncached)	194%
Large File Read (Cached)	100%
Large File Read (Uncached)	100%
Large File Write (Uncached)	135%
Large File Delete (Uncached)	184%
Directory Crawl (Cached)	67%
Directory Crawl (Uncached)	76%
Metadata Intensive Operation	128%

Figure 9: Impact of Replication

4.6 IMPACT OF FAST NETWORK

As can be seen in 10, below, at 8 storage nodes, the network is usually more than saturated. Use of higher speed networking gear (10 GbE or Infiniband) can significantly improve performance.

HIGH SPEED NETWORK PERFORMANCE: 8 NODE CLUSTER, GLUSTER FS on BARE METAL

CONFIGURATION		Gluster NFS	Gluster FUSE 10 GbE	Gluster FUSE Infiniband
Network		10 GbE	10GbE	20Gb Infiniband
Nodes		8	8	8
Server		Dell R510, single Quad Core, 4GB RAM	Dell R510, single Quad Core, 4GB RAM	Dell R510, single Quad Core, 4GB RAM
Storage per Node		12 X 1 TB 7.2K RPM SATA DAS	12 X 1 TB 7.2K RPM SATA DAS	12 X 1 TB 7.2K RPM SATA DAS
Total Storage		96 TB	96 TB	96 TB
IOZONE BENCHMARK				
Concurrent Stream Write (MB/s)	Coalesced	2718	4226	3773
Concurrent Stream Read (MB/s)	Uncached	2938	4758	8788
	Cached	3138	4822	8788

Figure 10: High Speed Network Configurations

5.0 CONDUCTING YOUR OWN TESTS

No set of standardized tests can serve as a perfect replacement for testing Gluster in your own environment with your own workloads. As an open source product, GlusterFS can—of course—be tried for free. Gluster's Virtual Storage Appliance for VMWare is also available for download as a free 30 day trial. Gluster's Amazon Machine Image is also available for a free 30 day trial, although you may have to pay Amazon for the use of AWS services. The use of the Gluster AMI is often a convenient way to test Gluster at scale, as you can deploy a system of several storage nodes and hundreds of terabytes in a few minutes, and need only pay for the hours of Amazon Web Services time that you use. However, as noted above, you should not expect to draw direct comparisons between per storage node performance in a physical environment and per instance performance in the cloud. It is also important to take into consideration any internet latency if your applications are also not running in the cloud.

For on-premise testing, the notes below can provide some guides to testing.

Single stream benchmark:

It is important to find out real disk and network speeds before measuring Gluster's performance.

"dd" disk dump utility comes handy to quickly measure your uncached disk performance..

- 1GB uncached write test, 1MB block size

  $ dd if=/dev/zero of=/path/to/file bs=1M count=1024 oflag=direct

- 1GB uncached read test, 1MB block size

  $ dd if=/path/to/file of=/dev/null bs=1M iflag=direct

Now measure your network performance between client and server using "iperf" TCP/IP bandwidth benchmark tool.

On the server node, start iperf server.

$ iperf -s

On the client node, start iperf client,

$ iperf -c <Server IP>

Notice the bandwidth. On a physical 1GigE network you should expect 90-110MB/s.

Now perform the same "dd" tests on the GlusterFS mount point. You should see performance close to network or disk which ever is lower. Gluster 2 way replicated volumes will show half of the network performance for writes, because of synchronous write operations. To measure cached performance, simply ignore "iflag=direct" or "oflag=direct" arguments.

Multi Stream Benchmark with IOZone:

Iozone is a powerful file system wide benchmarking utility, capable measuring aggregated performance across your entire storage cluster. You need to setup passwordless ssh connections between the client machines before you launch iozone. To measure the full potential of your storage cluster, you need to produce maximum load. A

decent server to client ratio is 1:4. For example, if you have 8 servers, start iozone from 32 clients. If you have fewer clients, try to increase number of streams per client until you saturate client network ports.

This command launches parallel iozone tests across the clients, each client will create 1G files with 128k block size. Clients IP addresses are listed in clients.txt file.

```
$ iozone -o -c -t $(wc -l clients.txt) -r 128k -s 1g -+m clients.txt
```

Multi Stream Benchmark with LLNL's IOR:

IOR is more scalable and accurate than iozone. IOR requires an MPI environment, which is fairly easy to set up.

The command below invokes IOR tests over 32 clients, each creating 1GB file with 128k block size. "-F" is 1 file per process. "-m" is "multiple file mode" (needed when specifying -F). "-t" is block size. "-o" is output test directory. Hosts file lists one IP or hostname per line.

```
$ mpirun --np 32 --hostfile hosts --mca btl tcp,self IOR -C -N 32 -b 1g -F -m -t 128k -o /path/to/gluster/ior
```

6.0 CONCLUSION

Gluster provides scale out NAS for cloud and virtual environments, and delivers excellent performance across a wide variety of workloads. Since performance, capacity, and availability all scale out using commodity components, users can economically configure their storage solutions to achieve the optimal balance between performance, cost, capacity, and availability.

A wide variety of factors impact performance, including the number of storage nodes, the speed of the network, file size, and the nature of the workload. This paper provides some guidelines to understanding the impact of these factors, so that performance levels can be anticipated under ideal and less-than-ideal circumstances. Guidance is also provided for conducting your own tests of Gluster. For further information about Gluster please visit us at www.gluster.com, or at our community site www.gluster.org.