Gluster file system optimization and deployment at IHEP

GlusterFS Features

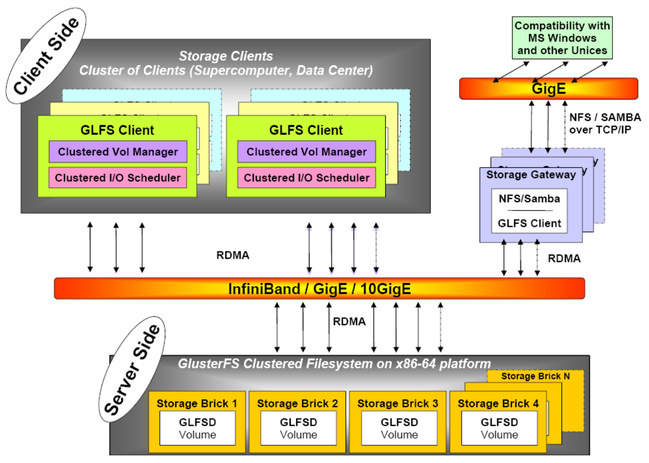
No central metadata
- No Performance Bottleneck
- Eliminates risk scenarios

Linear Scaling

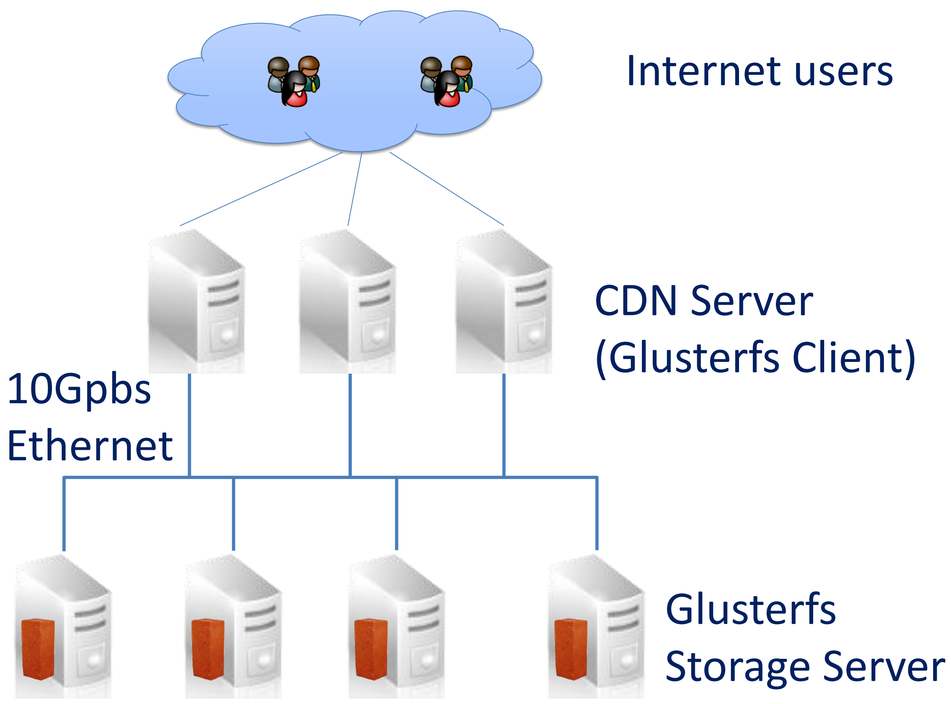
High Redundancy

Good performance
- Striping
- I/O accelerators: I/O threads, I/O cache, read ahead and write behind

Simple and Inexpensive Deployment



Architecture of Glusterfs



Glusterfs Deployment of CDN solution

CASE 1: Internet Content Distribution

Challenges
- billions of small files (average 30kB per file)
- thousands of concurrent access on each client

Technical problems
- limited number of files (~1700/sec) to access concurrently on one client
- IOPS bottleneck of SATA disk

Solutions
- change single thread to multi-thread mechanism in client fuse module to read/write on /dev/fuse
- cache inodes in server memory as many as possible to improve lookup efficiency and IOPS

Results
- stores 50 billion files in 50 servers
- ~2400 RPS each client, ~40 thousand RPS totally

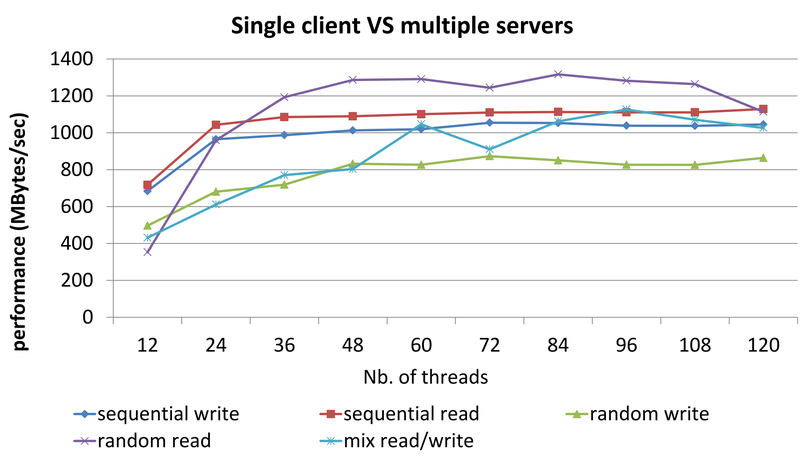
CASE 2: Home directories and HPC

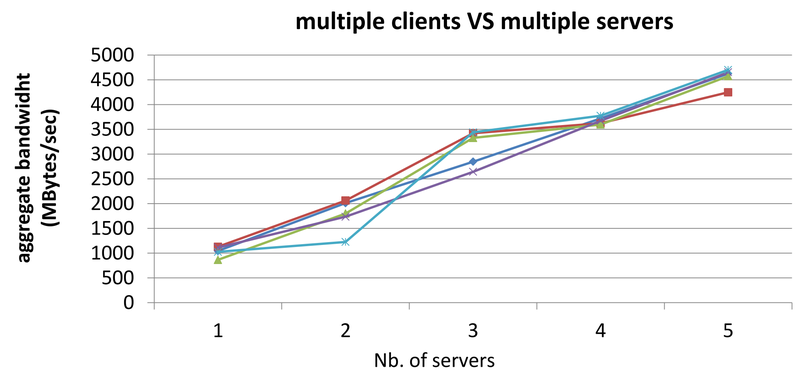
Challenges
- list directories or files quickly
- mkdir or rmdir the same directory frequently
- new space should be used immediately after adding new storage device without performing rebalance operation

Technical problems
- send too many lookup requests (N*M) to locate a file if there are N levels of directory tree and M bricks. DHT efficiency is wasted!!
- No global lock between different clients. Split-brain is troublesome!!
- DHT layout will not change before rebalance unless new directory

Solutions
- introduce unify layout to reduce the number of requests N*M to N
- synchronize the directory attribute among all bricks periodically (split-brain problem is not yet solved completely!)
- create a new version of layout after adding or deleting brick(s)

Results
- deployed in YBJ cluster farm, 4 servers , ~200TB totally
- work well for computing job, not so good for HOME directory





2 Xeon CPU, 16GB Mem, 12*2TB SATA disks per server;
10Gbps Ethernet interconnected; Iozone benchmark; file size : 400MB

Glusterfs performance in YBJ cluster farm

Conclusion

1) more safe than other metadata-server based file systems because it adopts DHT algorithm to locate file
2) more suitable in the case of stable directory system, for example, media or CDN
3) slow to list directory even if one brick server is busy because readdir operation will travel all bricks
4) difficult to keep consistency of the same directory in all bricks, which will cause different gfid or split-brain
5) In general, it is well designed, but should be optimized according to different application requirements.

the Institute of High Energy physics, Chinese Academy of Sciences

Yaodong CHENG (Email: chyd@ihep.ac.cn)