The Feasibility Research of Cloud Storage based on Global File System

Kefei Cheng
College of Computer Science, Chongqing University of Posts and Telecommunications
Chongqing China

Nan Wang
College of Computer Science, Chongqing University of Posts and Telecommunications
Chongqing China

***Abstract*—Cloud storage is the extension and development based on the cloud computing, which should not only provide the conventional file access just like POSIX, but also must be able to support the massive data management and the public service support functions. GFS (Global File System) is a solution for massively parallel computing and cloud storage, and its high availability and good scalability can assure the data that it can be more secure and reliable stored and more efficiently scheduled. In this paper we test the GFS environment, and study the file system recovery when node damage happening, and compare the GFS with ext3, NFS system in read and write performance. The experiment result and the practical system performance shows, GFS can provide effective support for the cloud storage, and provide good storage and calculation for cloud computing.**

***Keywords* - Cloud Storage; GFS; File system recovery; Read and write performance.**

I. INTRODUCTION

The concept of cloud storage is the same as cloud computing, it refers to the system connect a large variety of different types of storage devices together, using cluster , distributed file system and grid technologies, to provide data outside storage and access functions.

Distributed file system is the core part of cloud storage, and it is also the most difficult part to achieve. With the cluster, distributed file system and grid computing technologies, cloud storage can work among multiple storage devices, and provide a better data access performance.

GFS (Global File System), developed by the University of Minnesota, is a SAN (Storage Area Network)-based shared storage cluster file system. GFS was an open source software in a very long time, and later as Sistina company`s plan to provide support and service was not succeed. In 2011 Sistina GFS became a proprietary software, in order to promote their own financial income.

After the acquisition of Sistina, Red Hat company open the source code of GFS following the GPL (General Public License) agreement. Now, the full name of GFS is called "red hat global file system", each server charge an annual fee of $ 2200.

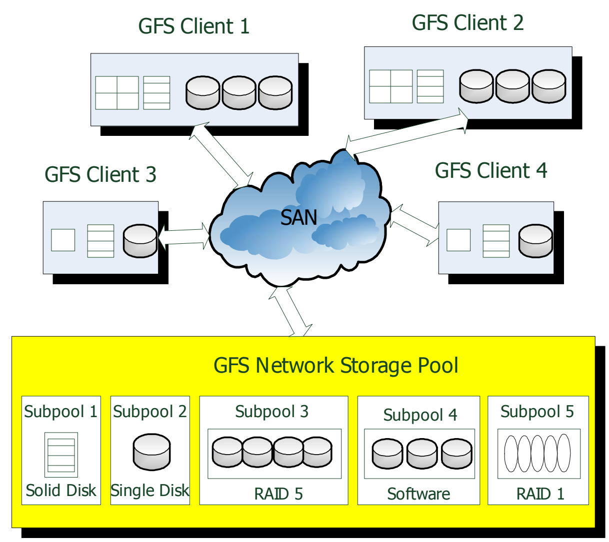


Figure 1. GFS architecture

With a wide range of IT applications and increasing data of the demand for shared accessing, Linux cluster technologies become more and more important, in many cases there will be multiple servers to simultaneously access the shared data needs, in general, through the traditional solution NFS/CIFS, etc. to achieve, or the method of data replication to ensure data consistency. But usually with the data and the massive increase in the number of clients increases, NFS/CIFS performance will be poor, and through data replication cannot guarantee real-time data consistency. To overcome this weakness, Red Hat offers its own solution through the GFS to the same partition on shared storage, so that each node as the service to access your local file system, the use of the direct I/O connectivity and access method, greatly improved access to the same partition share the performance data.

II. RELATED TECHNOLOGY ANALYSIS

GFS is a POSIX-compliant cluster file system and volume management system, and it runs on Red Hat Enterprise Linux system, supporting for SAN storage system. GFS can supports all major servers and storage devices on Red Hat Enterprise Linux system, and it is a technology-leading cluster file system on Linux.

With the use of the GFS, multiple servers can share a file system to store data. The information can be stored on the server or in a storage area network. Compared to other cluster file systems, GFS is a more powerful, better broad-based supported and better value for money. GFS system has three kinds of architectures:

A. GFS with SAN

This architecture has high performance and high scalability, see Figure 2. Multiple GFS nodes connected directly to the SAN storage unit via switches, and applications run directly on GFS nodes, to avoid the traditional NAS (Network Attached Storage) schema file server nodes of various bottlenecks and network delay. This architecture can support up to 300 GFS nodes.

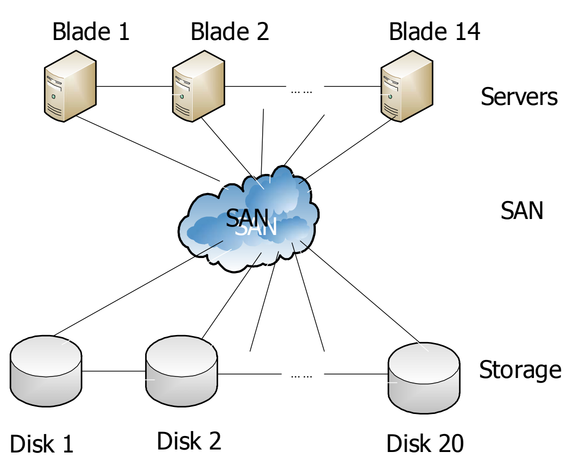


Figure 2. GFS with SAN

B. GFS and GNBD with SAN

This architecture blend of performance, scalability, price, see figure 3. The SAN storage is shared to the traditional network through GNBD (Global Network Block Device), each GNBD has a pool of threads in memory, thread for processing to the SAN store one block of data path mapping, therefore, every SAN memory can be accessed by any GNBD node, achieving a balanced load and high availability.

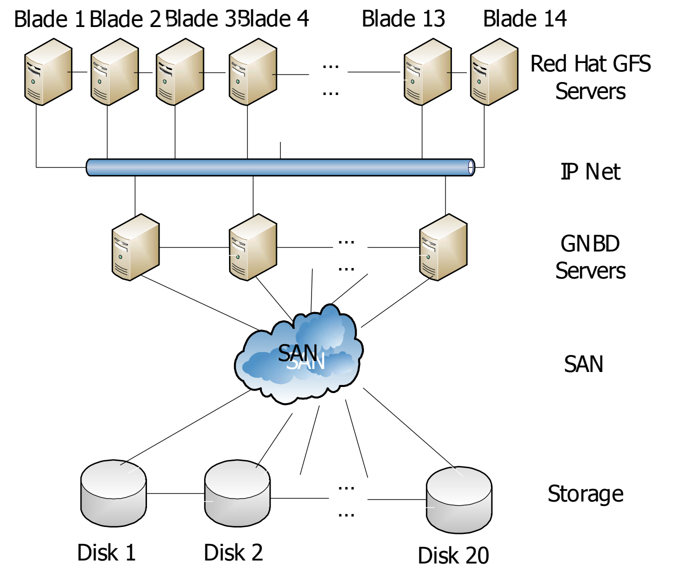


Figure 3. GFS and GNBD with SAN

C. GFS and GNBD with Directly Connected Storage

This architecture is more economic and giving more attention to performance, see figure 4. The biggest difference between this architecture and architecture B is without the SAN storage, storage is DAS (Direct-Attached Storage). This approach can be easily achieved in the traditional environment, but need to consider the redundant of GNBD and DAS.

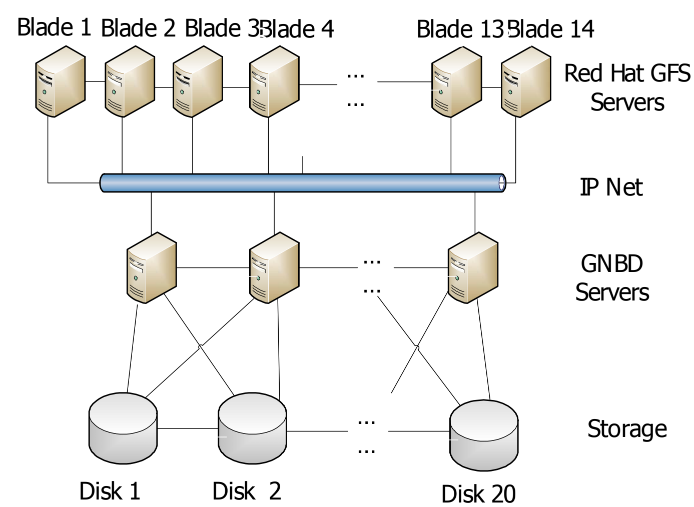


Figure 4. GFS and GNBD with Directly Connected Storage

III. GFS SYSTEM ENVIRONMENT

A. System Environment

Hardware environment: IBM BladeCenter JS22, IBM System x3650 M3, IBM System Storage DS5020;

Software environment: CentOS Linux 5.5, Kernel version Linux 2.6.18-194.el5.

B. *Install Cluster Suite/GFS Packages*

The Linux kernel 2.6 has been integrated GFS kernel module, so we only need to install several rpm packages. The whole GFS system is comprised of cman, system-config-cluster, gnbd, etc. So, on the CentOS 5.5 system we need to install these packages:

1. rgmanager -- Management of cluster system services and resources;

2. system-config-cluster -- The graphical cluster configuration tool, and you can configure the host on the cluster name, resources, fence and services;

3. ccsd -- Cluster configuration service daemon and the related documents;

4. magma -- Include the cluster lock management, and the database interface;

5. cman -- Cluster manager including the control of cluster members, the information, and the notification;

6. dlm -- Include the distributed lock management;

7. fence -- Management of cluster nodes between network switching, optical network switch and integrated power management interface of I / O protection system;

8. iddev -- Used to identify the device includes a formatted file system database;

9. GFS -- Red Hat Global File System module;

10. gnbd -- Red Hat Global Network Block Device module;

11. lvm2-cluster -- Logical volume management cluster expansion module;

12. pexpect -- A pure Python module used to start the subroutine and the automatic control.

After install these packages by rpm command, reboot the system.

C. *System Config Process*

1. Config /etc/hosts setting at each node to ensure that all nodes of GFS are in a LAN segment;

2. Config the cluster.conf file by system-config-cluster, loading nodes, and config the fence service. Then config the failover, domains and resources, and send the cluster.conf files to the other nodes through the scp command;

3. Stop the iptables service at each node, and then start DLM, ccsd, cman, fence services;

4. Export the shared storage device of the gnbd-server, and import the device in each node;

5. Load the GFS service, and use gfs_mkfs commands to create the GFS file system on the gnbd-server.

6. Establish a folder with name gfs on each node, and use the mount command on each node to load this file folder, and then nodes can share the storage of gnbd-sever.

```
[root@node6 gnbd]# gfs_mkfs -p lock_dlm -t GFS:gfs -j 2 /dev/gnbd/gfs
This will destroy any data on /dev/gnbd/gfs.
  It appears to contain a gfs filesystem.

Are you sure you want to proceed? [y/n] y

Device:                    /dev/gnbd/gfs
Blocksize:                 4096
Filesystem Size:           1259688
Journals:                  2
Resource Groups:           20
Locking Protocol:          lock_dlm
Lock Table:                GFS:gfs

Syncing...
All Done
```

Figure 5. Create GFS completed

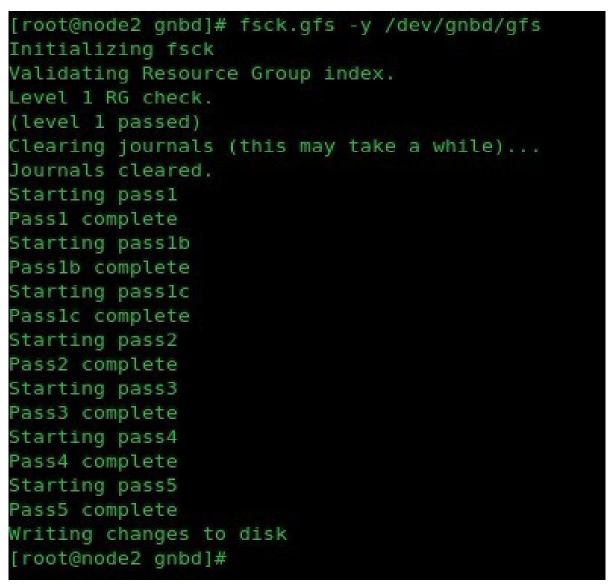
IV. FEASIBILITY RESEARCH

A. *File System Recovery when Partial Node Fail*

When the file system fails to mount a node, the file log system allows rapid recovery, and we use the fsck.gfs command to repair GFS file system. The principle of fsck.gfs in cluster is when a node or a few nodes fail to be mounted, GFS can repair the file system and data by using redundancy data. But the memory occupancy of fsck.gfs process is very high.

At the present era of increasing popularity of cloud computing, is block-based cloud storage model able to do in terms of security as safe as GFS? Whether the file system can always assure data integrity and security rely on the redundant data, this is the most worrying issues that all storage companies and users mostly concerning with.

Simulate a node was damaged, and then use fsck.gfs to recover the file system. It only takes about 10 seconds to finish the recovery in the 10M LAN environment.



```
[root@node2 gnbd]# fsck.gfs -y /dev/gnbd/gfs
Initializing fsck
Validating Resource Group index.
Level 1 RG check.
(level 1 passed)
Clearing journals (this may take a while)...
Journals cleared.
Starting pass1
Pass1 complete
Starting pass1b
Pass1b complete
Starting pass1c
Pass1c complete
Starting pass2
Pass2 complete
Starting pass3
Pass3 complete
Starting pass4
Pass4 complete
Starting pass5
Pass5 complete
Writing changes to disk
[root@node2 gnbd]#
```

Figure 6. Fsck.GFS recovery

GFS allows multiple Linux machines shared storage devices through the network. Each machine can use the share disks on line as a local disk, and the GFS also appears as a local file system. Once a server problem occurs, the user can still access to the data by other computers in the network.

B. File System Read and Write Performance Test

When using GFS to do experiment, there are some differences between the GFS I/O read and write performance and the ext3 system of CentOS. So I use iozone to test ext3, GFS, and NFS system, test file size 300M, with 5 concurrent threads.

Table 1. File System Read and Write Test Results

	Large file read	Large file write	Random read
ext3	135MB/s	117MB/s	101M/s
GFS	43.4MB/s	41.7MB/s	36M/s
NFS	9.3MB/s	9.2MB/s	6.3M/s

Overall, GFS performance is better than NFS, especially it can support more computers to share storage and cluster replication function. In a small concurrent read and write case, NFS can be infinitely close to the limit of local read and write, but in a high concurrent read and write case, NFS performance decreases obviously, while GFS performance can be more linear. The only drawback is that if there is a GFS node executes a large number of write operations, it is easily to pause when you access to the file system

V. Conclusions

The outstanding performance of GFS in the availability and scalability determines that it can be used as a cloud storage file system. After the verification of data security and file system read and write ability, GFS can take a space in the cloud storage tide, and cloud storage based on the GFS is feasible.

GFS file system has been verified in college teaching experimental platform for cloud computing currently. The actual operation shows that GFS can provide reasonable support for cloud computing environments in storage, and can provide good storage guarantee for cloud computing.

Acknowledgment

This paper is sponsored by the major science and technology programs of the information industry of Chongqing key technologies R&D programs (CSTC2010AB2003).

References

[1] Red Hat GFS 6.0(2011), http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/3/html/GFS_6.0_Administration_Guide/.

[2] Install and Configure GFS with Vmvare, http://www.5dlinux.com/article/7/2009/linux_35793.html.

[3] Red Hat Enterprise Linux 5 - Global File System , http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5/html/Global_File_System/.

[4] Tang Jian, “Research and Application of the Cloud Computation Storage System”, Computer Knowledge and Technology, May, 2009.

[5] Jiabin Deng, JuanLi Hu, Anthony Chak Ming Liu, Juebo Wu, “Research and Application of Cloud Storage”, Intelligent Systems and Applications (ISA), May 2010.

[6] Install and configure GFS on CentOS, http://bbs.linuxtone.org/thread-1286-1-1.html.

[7] Shuang Liang Weikuan Yu Dhabaleswar K. Panda, “High Performance Block I/O for Global File System (GFS) with InfiniBand RDMA”, 2006 International Conference on Parallel Processing (ICPP'06).