PERFORMANCE ANALYSIS OF CLUSTER FILE SYSTEM ON LINUX

Cheng yaodong[#], Xu dong, Chen gang, Wang lu, Wu wenjing
Institute of High Energy Physics, Beijing 100039, China

Abstract

With the development of Linux and improvement of PC's performance, PC cluster used as high performance computing system is becoming much popular. The performance of I/O subsystem and cluster file system is critical to a high performance computing system. In this work the basic characteristics of cluster file systems and their performance are reviewed. The performance of four distributed cluster file systems, AFS, NFS, PVFS and CASTOR, were measured. The measurements were carried out on CERN Linux 7.3.3 using standard I/O performance benchmarks. Measurements show that for single-server single-client configuration, NFS, CASTOR and PVFS have better performance and write rate slightly increases while the record size becomes larger. CASTOR has the best throughput when the number of write processes increases. PVFS and CASTOR are tested on multi-server and multi-client system. The two file systems nicely distribute data I/O to all servers. CASTOR RFIO protocol shows the best utilization of network bandwidth and optimised to large data size files. CASTOR also has better scalability as a cluster file system. Based on the test some methods are proposed to improve the performance of cluster file system.

INTRODUCTION

Cluster systems are popular architectures in the field of high performance computing. In cluster systems, each compute node has its I/O subsystem. With the continual improvement of hard drive technology, network technology and Linux software performance, we start to investigate how to use our existing hardware and software more efficiently, how to combine I/O subsystems of all the compute nodes into a global system image to make the most of I/O subsystems in each node. Figure 1 illustrates the architecture of a cluster system we are studying.

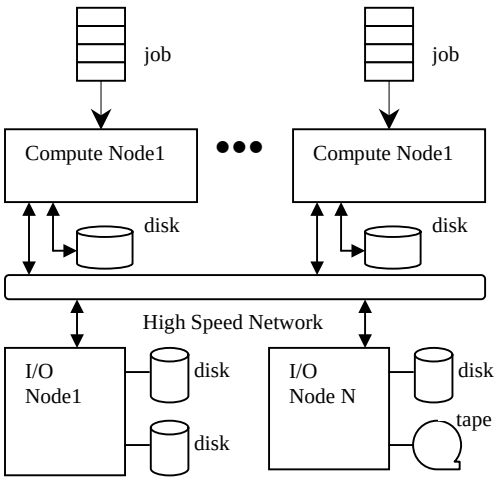

Figure 1: Architecture of a cluster system

The cluster system shown in figure 1 is made up of compute nodes and I/O nodes. Jobs are running on compute nodes, which have their own disks. I/O nodes provide storage capacity with disks and tapes to compute nodes. In such a cluster system, cluster file system is critical to making full use of all the I/O subsystems in each node. There exist many different cluster file systems. Their performance, suitable environment, scalability are different and the performance is the research emphasis of this work. First, this paper reviewed the basic characteristics of cluster file system. To analyse the performance of cluster file system, we built a data access model. Then the paper proposed performance analysis formulae and discussed different cases. The performance of four distributed cluster file system, NFS [1], AFS [2], PVFS [3], CASTOR [4], were measured. The result will be given in this paper. At the end of this paper, based on the test some methods are proposed to improve the performance of cluster file system.

CLUSTER FILE SYSTEM

Cluster file system is one of the most important methods to share information of cluster system, which must meet the features of cluster computing environment. In general, a cluster file system should have the following characteristics:

- Single-system image: The file system integrates all the storage devices in cluster system, such as local and global disks, tape etc, and presents users with a uniform, enormous file system image.
- Transparency: Local and remote files can be accessed by same system calls; The host name or IP address of file server isn't one part of the full path of a file; If the location in which a file is stored changes, the full path of the file doesn't change.
- Good scalability: The cluster file system can adapt to the increase of numbers of nodes and users in cluster system.
- High performance.

Distributed/cluster file systems are usually classified into three types by structure: traditional client/server distributed/cluster file system (e.g. NFS), share-disk distributed/cluster file system (e.g. GFS [5]), and virtual share-disk distributed/cluster file system (e.g. Frangipani [6]).

[#]chyd@mail.ihep.ac.cn

PERFORMANCE ANALYSIS

Data access model

We set up a data access model presented in figure 2. The storage servers that offer storage services to clients through network are divided into I/O nodes and management nodes. When clients access data, clients first send requests to management such as open file and management nodes allocate appropriate I/O nodes. Then clients exchange data with I/O nodes such as read/write data.

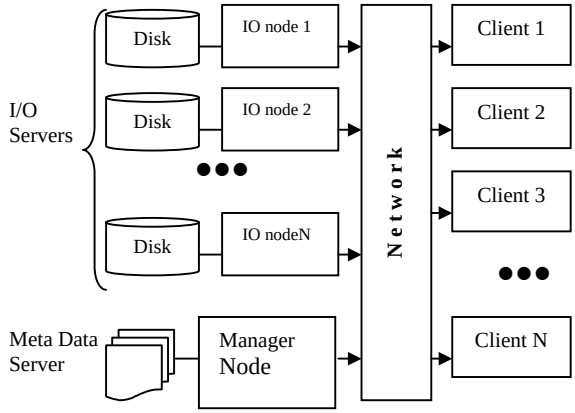

Figure 2: Data access model of cluster file

In the figure 2, the management nodes store the metadata of file system, e.g. file sizes、access permissions、time stamp and so on. File entities are stored in I/O nodes, to which one or more disks are attached. A client reads or writes files in cluster file system.

Performance analysis formula

In the data access model above mentioned, we assume that data is processed in each client. We also assume that management nodes only deal with data access requests of clients and I/O nodes only perform operations of reading/writing files and don't provide computing service. The traffic between clients and management nodes is very small in this way. We assume that the time that I/O nodes spend dealing with requests of clients is much smaller than the time consumed by transferring a lot of data. Thus the access time to data is:

$$T = \max (D*c/N, D/(N*I), D/(M*I), D/(P*R))$$

$$S = D/T = \min (N/c, N*I, M*I, P*R)$$

In these equations, D presents the total quantity of data; c presents the CPU time spent on disposal of each byte; T presents the minimum access time to total data; S is the maximum aggregation bandwidth; R is access speed of each disk; I is the network interface speed; M is the number of I/O nodes; N is the number of clients; P is the number of disks which are read/written in parallel. The limitation is $P/M \geqq 1$, which means that an I/O server has at least one disk.

If the complexity of data process is simple (that is: c is very small) the time that clients spend processing data is very short. So the time cluster file system spends in accessing to all the data is:

$$T = \max (D/(N*I), D/(M*I), D/(P*R))$$

$$S = D/T = \min (N*I, M*I, P*R)$$

Now we discuss the different cases that maybe happen:

1) N (the number of clients) is 1,and the M (the number of I/O nodes) is equal to 1 or larger than 1, the speed of each disk is greater than the speed of network transmission. So the entire bandwidth of the system depends on the network transmission speed I.

2) N (the number of clients) is 1,and the M (the number of I/O nodes) is equal to or larger than 1, the speed of each disk is slower than the speed of network transmission speed. So the entire bandwidth of the system S is equal to the minimum of I and P*R.

3) N (the number of clients) is equal to or larger than 1,and M (the number of I/O nodes) is 1; the speed of each disk is greater than the speed of network transmission. So the entire bandwidth of the system depends on the network transmission speed I.

4) N (the number of clients) is larger than 1, and the M (the number of I/O nodes) is larger than 1, the speed of each disk is greater than the speed of network transmission. So the aggregate bandwidth S is equal the minimum of N*I and M*I.

PERFORMANCE TEST

Test Environments

Twelve computers constitute our test environment. These computers are classified into three types: management nodes、I/O nodes and client nodes, just like in the above data access model. The configurations are as follow:

CPU：Intel(R) Pentium(R) 4 CPU 2.80GHz，cache size：512 KB

Memory：512MB

Disk：WD Caviar® SE，capacity: 80G，cache 8M，rotate speed: 7200RPM

Operation system：CERN Linux7.3.3

Kernel：2.4.20-18.7.cernsmp

Local file system：ext3

All of the servers and the clients are connected through 100M Ethernet. To avoid the influences that CPU loads impact on I/Os, no other job is running on the servers and clients.

The following cluster file systems were measured: OpenAFS 1.2.9，NFS v3，PVFS，CASTOR 1.6.1.2.

Results

We use Netperf [7] version 2.2pl3 to measure network performances and use Iozone [8] version 3.217 to test performances of the local file system before we measure the performance of cluster file systems. We find that the transmission speed of 100M Ethernet is about

94.11Mbits/sec. Figure 3 illustrates the write performance of local file system.

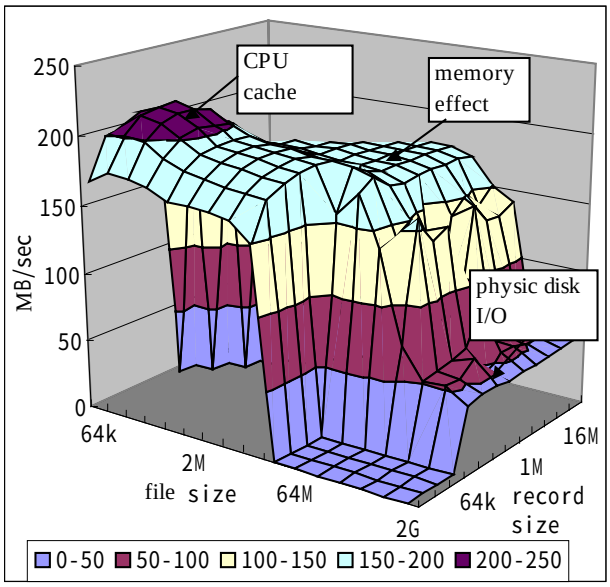

Figure 3 Write performance of local file system

Figure 3 shows the write performance result of local file system ext3 where file size and request size are changed. The place on the lower right that touches the floor of the graph is not actual data. Excel graphs empty cells as containing a zero. We can see explicitly that CPU cache and system cache affect the performances. When the file size exceeds the cache size, the real speed of physical disk I/O is a bit below 50MB/s.

IOzone cannot test CASTOR directly because CASTOR doesn't support traditional UNIX file system interface. We include CASTOR's header file <shift.h> into the IOzone source codes, link it together with CASTOR RFIO libraries and produce a new executable IOzone codes. Our test results are as follows.

Test 1: one-client-to-one-server write performance measurement, file size: 512MB, record size: 64KB-16MB, output unit: KB/sec. This test results are presented in Table 1.

Table 1：Write performance of one client one server

File system		NFS	AFS	PVFS	CASTOR
Record size (KB)	64	11101	5173	9953	10209
	128	10803	5342	10158	10335
	256	11054	5239	10103	10530
	512	11125	5137	10239	10622
	1024	11083	5148	10759	10697
	2048	11042	5335	10603	10722
	4096	11045	5212	10662	10723
	8192	11109	5175	10948	10705
	16384	11047	5353	10976	10678

From Table 1, we can conclude that the write performance of NFS、PVFS and CASTOR is a bit bigger than 10MB/s, smaller than the network bandwidth (about 94.11Mbits/s) and far smaller than the local I/O rate (a bit below 50MB/s). This complies with the first case in the above performance analysis. The performance of cluster file system is limited by the network speed in this case. The write bandwidth of AFS is about a half of other file system and is perhaps affected by other factors.

Test 2: Multi-process measurement, file size：100MB, record size: 512KB, process numbers: 1 to 10, output unit: KB/sec. The results are presented in Table 2.

Table 2: multi-process measurement

File System		NFS	AFS	PVFS	CASTOR
Number of processes	1	10372	7878	10806	10680
	2	10362	7889	10752	11255
	3	10323	10841	10751	11221
	4	10311	1020	10686	11450
	5	10257	9358	10707	11430
	6	10258	9142	10690	11441
	7	10255	8120	10696	11390
	8	10173	8545	10697	11440
	9	10240	8652	10696	11442
	10	10250	7305	10698	11430

From Table 2, it can be concluded that as the number of processes (the maximal number here is 10) increases, I/O bandwidths of these file systems don't obviously decrease. Especially the bandwidth of CASTOR is on the increase, up to 11.4MB/s. The bandwidths of these file systems don't exceed the network bandwidth (about 94.11Mbits/s) and don't yet exceed the local read/write speed. These are accord with the first case in the above performance analysis.

Test 3: multi-client-to-multi-server measurement.

Here we only test PVFS and CASTOR. We use the IOzone's cluster measurement function. Each file size is 200MB, record size is 2MB and output unit is MB/sec. The results are presented in figure 4.

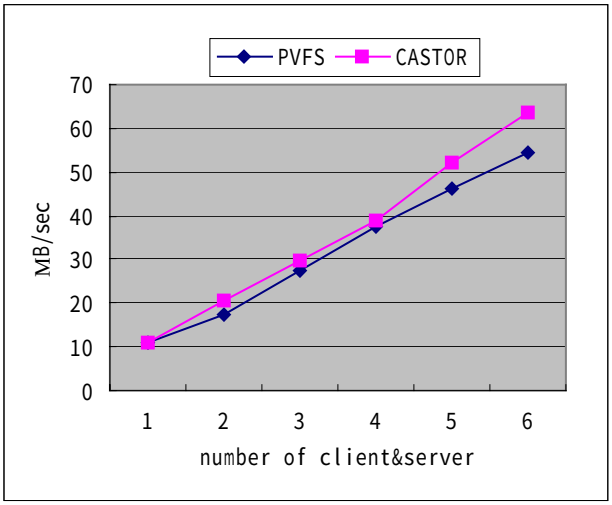

Figure 4 cluster testing (multi-client & multi-server)

CASTOR and PVFS can keep good balance in distributing the data into all I/O servers. PVFS stripes file

data across multiple disks of I/O nodes in a cluster. CASTOR doesn't stripe files, however, it evenly distributes file requests to each disk server. In this way all disk servers are macroscopically utilized efficiently. So good aggregation bandwidths can be obtained. We can come to a conclusion from the results that PVFS and CASTOR have good scalability. The aggregation bandwidth almost linearly increases as the number of clients and server increases and CASTOR has higher performance. The measurement is in accord with the fourth case in the above performance analysis.

CONCLUSION

From the analysis and measurement results, it can be concluded the performance of NFS for single client access is good. However, NFS performance is limited by contention at the single server. AFS, which affords single, shared name space and uses local cache mechanism, enhances the efficiency of utilization of cache files. It is suitable for sharing files across local network or wide network. But I/O bandwidth is low when files are transferred through network. PVFS stripes file data across multiple disks in I/O nodes. Its performance and scalability are good. But its fault-tolerant system need be improved. CASTOR combines disks of cluster into disk pools and forms a uniform file system. It distributes automatically file operation requests into different disk servers and makes full use of every disk. It implements automatic load balances of I/O requests and has the good scalability. CASTOR RFIO protocol has good utilization of network bandwidth and optimises access to large size files. So performance of CASTOR is higher than other three file systems while transferring large size files. It also affords HSM functions. So it is very fit for applications on high-energy physics. However, so far it has not the interface to local file system and can't be operated by original programs and standard shell commands. In addition, if network speed is fast enough, the performance of access single file is limited by the I/O speed of disk because CASTOR doesn't support file striping.

In theory, if the data is physically balanced among the I/O devices, the data requirements are balanced among the application's tasks, and the network has enough aggregate bandwidth to pass the data between the two without saturating, such a cluster file system should provide good I/O performance. Based on the above analysis and measurement results, the following methods can be used to optimise the performance of cluster file system:

1) Designing and using high performance network file transfer protocol, making full use of network bandwidth.

2) Connecting the whole cluster system by high-speed network, for example Gigabit Ethernet or Myrinet.

3) Distributing file requests into different machines, different storage devices in order to achieve a good aggregation bandwidth because network bandwidth is the main factor that limits the performance of cluster file system so far (Connecting all storage devices to a single server is not good way).

4) Finding some way to increase the read/write speed of disk, such as using the technology of RAID. Disk is one of main factor that influence the performance of cluster system because disk is mechanic equipment and increase is 20% per year in its I/O rates, which is much slower than roughly 60% per year of processor performance.

5) Using the technology of file striping and parallel I/O operations when I/O speed of single disk is lower than the network bandwidth.

6) Making good use of some traditional file system design technology such as caching、pre-fetching.

7) Applying the technology of multithreads and asynchronous I/O to enhance the ability of parallel processing in management nodes because management nodes in the cluster file system always become the performance bottleneck in the large cluster system.

REFERENCES

[1] Sun Microsystems, Inc. NFS: Network File System Protocol Specification. RFC1094

[2] OpenAFS Home Page: http://www.openafs.org/

[3] P. H. Carns, W. B. Ligon III, R. B. Ross, and R. Thakur, "PVFS: A Parallel File System for Linux Clusters," In Proceedings of the 4th Annual Linux Showcase and Conference, pages 317–327, Atlanta, GA, Oct. 2000, USENIX Association

[4] CERN CASTOR web: http://castor.web.cern.ch/castor/

[5] Kenneth W Preslan, Andrew P Barry, Jonathan E Brassow *et al*, A 64-bit, Shared Disk File System for Linux. The Sixteenth IEEE Mass Storage Systems Symposium.

[6] Chandramohan A. Thekkath, Timothy Mann, and Edward K. Lee. Frangipani: A Scalable Distributed File System. In Proceedings of the Symposium on Operating Systems Principles, 1997, pages 224-237

[7] Public Netperf Homepage: http://www.netperf.org

[8] IOzone Filesystem Benchmark Web: http://www.iozone.org