Distributed Metadata Management of Mass Storage System in High Energy Physics

Distributed Metadata Management of Mass Storage System in High Energy Physics

Qiulan Huang[1], Ran Du[1], YaoDong Cheng[1], Jingyan Shi[1], Gang Chen[1], Wenxiao Kan[1,2]
[1]IHEP computing center, P.O.Box 918-7, 19B Yuquan Road, Beijing 100049, China
[2]University of Chinese Academy of Sciences, Beijing, China

E-mail: huangql@ihep.ac.cn

Abstract. In this contribution, we design and implement a dynamic and scalable distributed metadata management system(StarFS) to High Energy Physics (HEP) mass storage system. Particularly, we discuss the key technologies of the distributed metadata management system. We propose a new algorithm named Adaptive Directory Sub-tree Partition(ADSP) for metadata distribution. ADSP divides the filesystem namespace into sub-trees with directory granularity. Sub-trees will be stored on storage devices in flat structure, whose location and file attributes are recorded as extended attributes. The placement of sub-tree is adjusted adaptively according to the workload of metadata cluster so that the load balance could be improved and metadata cluster could be extended dynamically. Experiments show that ADSP achieves higher metadata performance and scalability compared to Gluster and Lustre. We also propose a new algorithm called Distributed Unified Layout(DULA) to improve dynamic scalability and efficiency of data positioning. A system with DULA could provide uniform data distribution and efficient data positioning. DULA is an improved consistent hashing algorithm which is able to locate data in O(1) without the help of routing information. Experiments prove that the better uniform data distribution and efficient data access can be achieved by DULA. This work is validated in YangBaJing Cosmic Ray(YBJ) experiment.

1. Introduction

Metadata management is quite important to overall system performance in large-scale distributed storage systems, especially in the big data era[1]. Metadata performance would produce a big effect on the scalability, availability and high performance of the massive storage system. In order to manage metadata effectively, so that data can be allocated and accessed efficiently, we design and implement a dynamic and scalable distributed metadata management system(StarFS) to HEP mass storage system based on Gluster[2]. In this contribution, the architecture of StarFS are introduced. Then, we discuss the key technologies of the distributed metadata management system. We propose a new algorithm named Adaptive Directory Sub-tree Partition(ADSP) for metadata distribution. ADSP is an improved sub-tree partition algorithm with low computational complexity, also easy to be implemented. We also propose a new algorithm called Distributed Unified Layout(DULA) to improve dynamic scalability and efficiency of data positioning. A system with DULA could provide uniform data distribution and efficient data positioning. DULA is an improved consistent hashing algorithm which is able to locate data in O(1) without the help of routing information.

The rest of the paper is organized as follows. Section 2 introduces the architecture of StarFS. Section 3 presents ADSP algorithm. Section 4 discusses the DULA method. Section 5 illustrates the system evaluation. And we conclude the paper in Section 6.

2. System Architecture

One of the key design features of StarFS is the separation of metadata from data management. By decoupling metadata operations, the system is able to provide the metadata service with high performance, high scalability and high availability, which can easily solve the issues of mass storage system in IHEP[3] like single point failure, load balance dynamically and so on. Figure 1 shows the architecture of StarFS. StarFS implements distributed metadata management. Its main components are listed as followings.

(1) User Interface.

It offers POSIX semantics and API to users.

(2) Metadata Management

It organizes and distributes metadata in clusters and grants access control to make sure the data security.

(3) Data Service

Data Service is responsible for the data location and placement, data replication and data migration.

(4) Resource

Resource is the device layer, which divides logical and physical layer. The physical one is the disk media like SSD, SATA and SAS. The logical one is the logical volume organizing the disk media by RAID.

(5) Monitoring

Monitoring checks the status of StarFS and reports the warning/critical information to administrator.

(6) Configuration

Configuration module provides two ways to configure the system including command line and web portal.

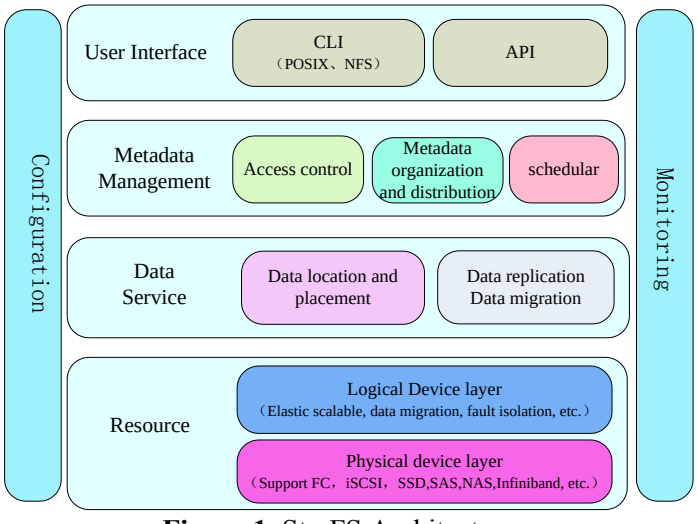

Figure 1. StarFS Architecture

3. Adaptive Directory Sub-tree Partition algorithm(ADSP)

ADSP divides the filesystem namespace into sub-trees with directory granularity. Sub-trees will be stored on storage devices in flat structure, whose locality information and file attributes are recorded as extended attributes. The local storage structure of metadata is consistent with the directory structure of LINUX operating system. The root directory is named with the unique identifier GFID (Global File Identification) of the root directory of the sub-tree. The distribution information of the migrated sub-tree is also recorded in the extended attribution. We can collect by looking up all the distribution

information of subtrees into a complete directory hierarchy. The placement of sub-tree is adjusted adaptively according to the load of metadata cluster so that the load balance could be improved and metadata cluster could be extended dynamically. ADSP is an improved sub-tree partition algorithm with low computational complexity, also easy to be implemented.

3.1. Directory Tree Partition

The principle of ADSP is the directory tree partition, which divides the namespace of file system into subtrees with directory granularity. Each directory is in charge of the metadata management under the directory. And the metadata across the metadata cluster. Each metadata server only care about their own part, which greatly improves the metadata performance.
For example, the namespace is showed in Figure 2.

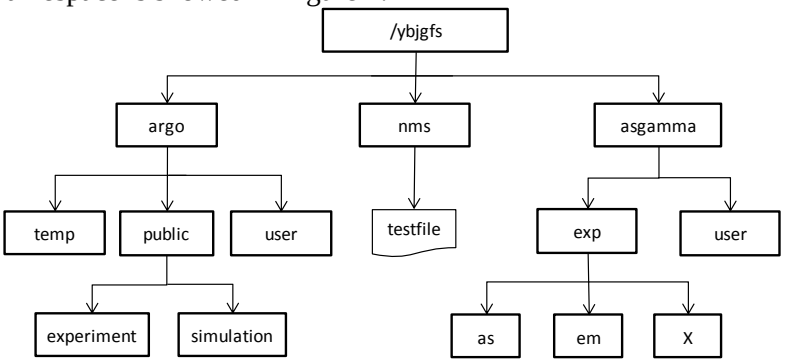

Figure 2. The directory hirerarchy

In this example, we know the mount point is /ybjgfs, which is not included in the directory structure in local storage. argo/public locates in one metadata server and argo/user locates in another one. The sub directories of them inherit the distribution of their parents. On metadata server side, each directory is an independent tree with the unique identification ID. The ID is the GFID of the root directory. Figure 3 shows the internal structure of argo/public on server side (/data1/gdata2/data05/295574a6-45f7-4c8e-84b4-cdfd7659c8dd). The prefix /data1/gdata2/data05 is the local storage partition on metadata server. The string 295574a6-45f7-4c8e-84b4-cdfd7659c8dd is the GFID of argo/public in the /ybjgfs file system.

```
[root@prfserver01 ~]# ll /data1/gdata2/data05/
total 88
drwxr-xr-x 2 root    root 4096 Dec 18 17:58 0d93a2dd-e031-4a22-b0e8-fe335b44f516
drwxr-xr-x 9 huangql root 4096 Dec 29 23:15 295574a6-45f7-4c8e-84b4-cdfd7659c8dd
drwxr-xr-x 2 root    root 4096 Dec 18 17:22 6cd1dffb-4661-47e4-9bd9-77ec78676be7
drwxr-xr-x 3 root    root 4096 Dec 20 11:18 851de119-faa4-4471-9429-f58a9077c966
drwxr-xr-x 2 huangql u07  4096 Dec 29 23:05 9d31f59f-1eaa-4095-bbae-6c58f3e79138
drwxr-xr-x 2 root    root 4096 Dec 18 16:28 a08ef6fa-5d12-4373-9c7f-d428812262f3
drwxr-xr-x 2 root    root 4096 Dec 18 18:13 ab9a7768-bfa2-49b7-b593-31bc375b7c9a
drwxr-xr-x 3 huangql u07  4096 Dec 29 23:14 e537c53e-67a4-4c0b-8d07-ebc411554381
```

Figure 3. The internal structure on metadata server

Figure 3 illustrates the flat structure of the metadata on server side with EXT3/EXT4[4][5] file system. If it is a file, StarFS will create an empty file, otherwise create a directory. Take /ybjgfs/argo/public for example, the storage on metadata server is showed in Figure 4. Files under this directory exist with size 0 and directories exist as usual.

```
[root@prfserver01 ~]# ll /data1/gdata2/data05/295574a6-45f7-4c8e-84b4-cdfd7659c8dd/
total 84
drwxr-xr-x 2 huangql u07  4096 Dec 29 22:48 bei
drwxr-xr-x 2 huangql u07  4096 Dec 29 19:58 chenchen
drwxr-xr-x 2 huangql u07  4096 Dec 29 20:19 dl
-rw-r--r-- 1 huangql root    0 Dec 14 21:28 ff_0
-rw-r--r-- 1 huangql root    0 Dec 14 21:28 ff_1
-rw-r--r-- 1 huangql root    0 Dec 14 21:28 ff_2
-rw-r--r-- 1 huangql root    0 Dec 14 21:28 ff_3
drwxr-xr-x 3 huangql root 4096 Dec 18 23:37 hql
-rw-r--r-- 1 huangql u07     0 Dec 26 17:31 ii
-rw-r--r-- 1 huangql u07     0 Dec 26 17:33 jj
drwxr-xr-x 2 huangql root 4096 Dec 22 15:49 lan
drwxr-xr-x 2 huangql u07  4096 Dec 29 19:57 mydir
drwxr-xr-x 2 huangql u07  4096 Dec 29 20:24 orange
-rw-r--r-- 1 huangql u07     0 Dec 29 19:58 zz
```

Figure 4. The storage of sub directory on metadata server

To satisfy POSIX directory access semantics, the metadata server cluster must traverse all the ancestor directories containing a requested part of metadata to provide the directory hierarchy as a logical view to clients.

3.2. Metadata Definition

The metadata describes the directory hierarchy of file system. In StarFS, metadata definition includes file/directory create time, modify time, access time, size, GFID and so on.

GFID	UID	GID	SIZE	PATH	NAME	ATIME	MTIME	CTIME	OTHERS

Figure 5. Metadata Definition

GFID is the unique identification number of file/directory, which plays an important role in distinguishing different files/directories. UID (User ID) and GID(Group ID) illustrate the owner of files/directories. SIZE records file size. PATH is the logical path of files/directories. ATIME is the access time. MTIME is the modified time. CTIME is the create time. There are some other items like INO(Inode number) and nlink(number of links).

3.3. Dynamic Meatadata management

In order to adapt the load of metadata cluster, the placement of sub-tree must adjust adaptively so that the load balance could be improved and metadata cluster could be extended dynamically. A dynamic metadata distribution is necessary because both file/directory numbers and popularity of pieces of directory tree change over time in an unpredictable fashion.

In ADSP, we design a monitoring module to collect the status of metadata cluster every 2 seconds to know the real-time CPU, Wait CPU, network IO, memory and disk state. We define the weights of those values to evaluate the sequence of metadata servers sorted by the performance. Thus, it is possible to migrate the metadata between servers according to the sequence. When some pieces of metadata are migrated to other server, the distribution information has to be changed correspondingly, that is to set the extended attributes (trusted.zefs.md.layout) as the new server ID. Experiments show that it takes about 1.387s to migrate the metadata of a directory (depth=5, 30000 files/directories) from one server to another, which is optimistic.

4. Distributed Unified Layout(DULA)

The data distribution across data servers is important for system's dynamic scalability and efficiency. We propose a new algorithm called DULA. DULA can provide uniform data distribution and efficient data positioning. DULA is an improved consistent hashing algorithm[6] which is able to locate data in O(1) without the help of routing information. The main idea of DULA is to abstract the storage device as a interval management node. The size of the management node is decided by the performance of the

device. In general, the better performance, the bigger size of the management ranges. This can smoothly shield the storage device heterogeneity, and eliminate the bucket effect to maximize the cluster performance.

The management range calculated as following. As mentioned above, the range size is related to the storage device performance. So we define the performance evaluation field *P*(same as section 3.3). In our system, we assume that the parameters of data server like CPU, memory, network IO and disk IO are same, and *P* is related to the storage disk capacity *V*, then the weight $W_i = V_i \Big/ \sum_{i=1}^{i=N} V_i, (1 \le i \le N)$.

In DULA, all the storage device are mapped into a ring 2^m (m=32), the range R_i of each of them is $R_i = 2^m \times W_i \ (1 \leqslant i \leqslant N)$. For instance, we have 8 storage nodes with the same configuration, then the range is $R_i = 2^m / 8$.

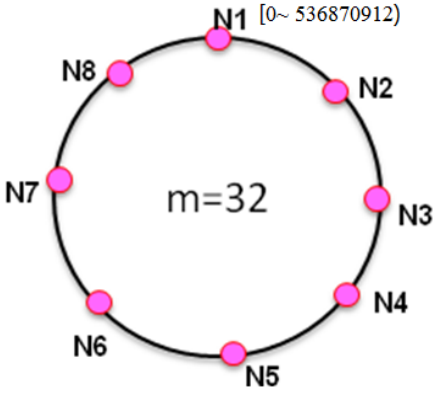

Figure 6. The ring of DULA

To provide good load balancing across data servers, DULA distribute files across servers based on Davies-Meyer[7] hash of the file GFID to determine the location of data. This strategy can bring a number of advantages. Clients can locate and contact the responsible data server directly in O(1). And the workloads are evenly distributed across the cluster.

This work is validated in YBJ experiment[8]. We deployed the production system StarFS in computer center, IHEP. The system has capacity 347TB with 44 storage devices. There are 49,113,284 files in total. Figure 7 there are about one million inodes in each storage devices.

Filesystem	Inodes	IUsed	IFree	IUse%	Mounted on
/dev/sdb1	549322752	1151920	548170832	1%	/data1
/dev/sdb2	549322752	1133465	548189287	1%	/data2
/dev/sdb3	549322752	1129776	548192976	1%	/data3
/dev/sdb4	549322752	1118090	548204662	1%	/data4
/dev/sdb5	549322752	1129550	548193202	1%	/data5
/dev/sdb6	549322752	1143333	548179419	1%	/data6
/dev/sdb7	550650096	1144696	549505400	1%	/data7
/dev/sdc1	549322752	1112975	548209777	1%	/data8
/dev/sdc2	549322752	1108084	548214668	1%	/data9
/dev/sdc3	549322752	1149500	548173252	1%	/data10
/dev/sdc4	549060608	1159473	547901135	1%	/data11

Figure 7. The distribution of data across storage cluster

5. Evaluation

We evaluate our prototype with the common benchmark tool Mdtest to demonstrate its metadata operation performance and scalability. In all tests, metadata/data servers are running on HP Gen 5 with Intel(R) Xeon(R) CPU E5430, 8GB RAM and 12TB disk array attached.

In order to evaluate the advantages of ADSP, we have implemented a simulation environment to compare the metadata operation performance among Lustre[9], Gluster, CEPH[10] and ADSP. We simulated a directory (depth=5, 20000 files/dirs) to test the metadata operation (mkdir, rmdir, statdir, createfile, delfile, statfile, readfile).

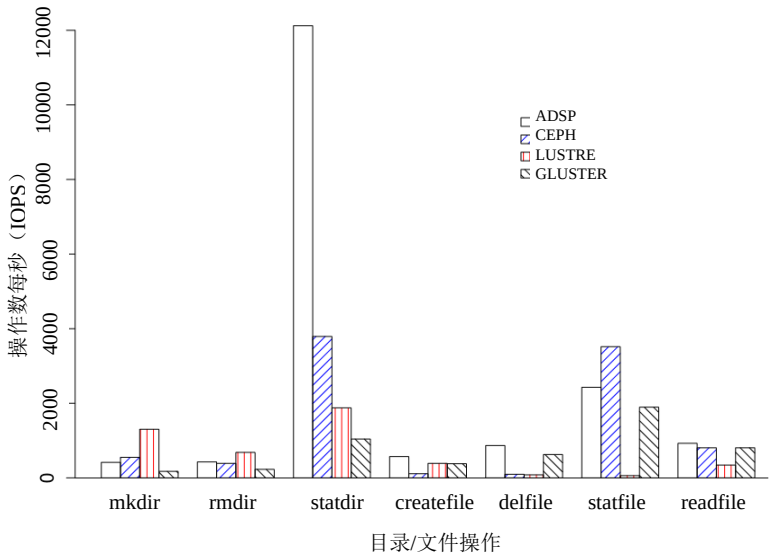

Figure 8. Metadata performance comparison among Lustre, Gluster, CEPH and ADSP

Figure 8 shows the Metadata performance is greatly improved by ADSP. From the Figure 8, we know the performance in directory operation of ADSP is about 2~3 times than Gluster and the performance in file operation of ADSP is about 2 times than Gluster. Compared ADSP and CEPH, the operations in mkdir and readfile are quite the same, however, the performance gap in statdir and createfile is obvious. ADSP behaves better than CEPH in those operations. The results illustrate CEPH with complex metadata partition and lock mechanism worse the performance in writefile. And ADSP is better than Lustre in file operation, which is because the file attributes are recorded in data servers like size, create time, access time, modified time. Even though access the file metadata, the requests must communicate the responsible data server.

Furthermore, we do the system scalability evaluation. We test the metadata operation under the storage system have been expanding from 18, 60, 120, 240, 480 to 600. In Figure 8, chart A is the the statdir performance, chart B is the mkdir/rmdir performance, chart C is the readfile/statfile performance and chart D is the createfile/delfile performance. Figure 9 illustrates the Gluster metadata performance reduces obviously when the system scale expands, otherwise, StarFS with ADSP algorithm behaves more stably. This validates that StarFS with ADSP algorithm has the better scalability than Gluster.

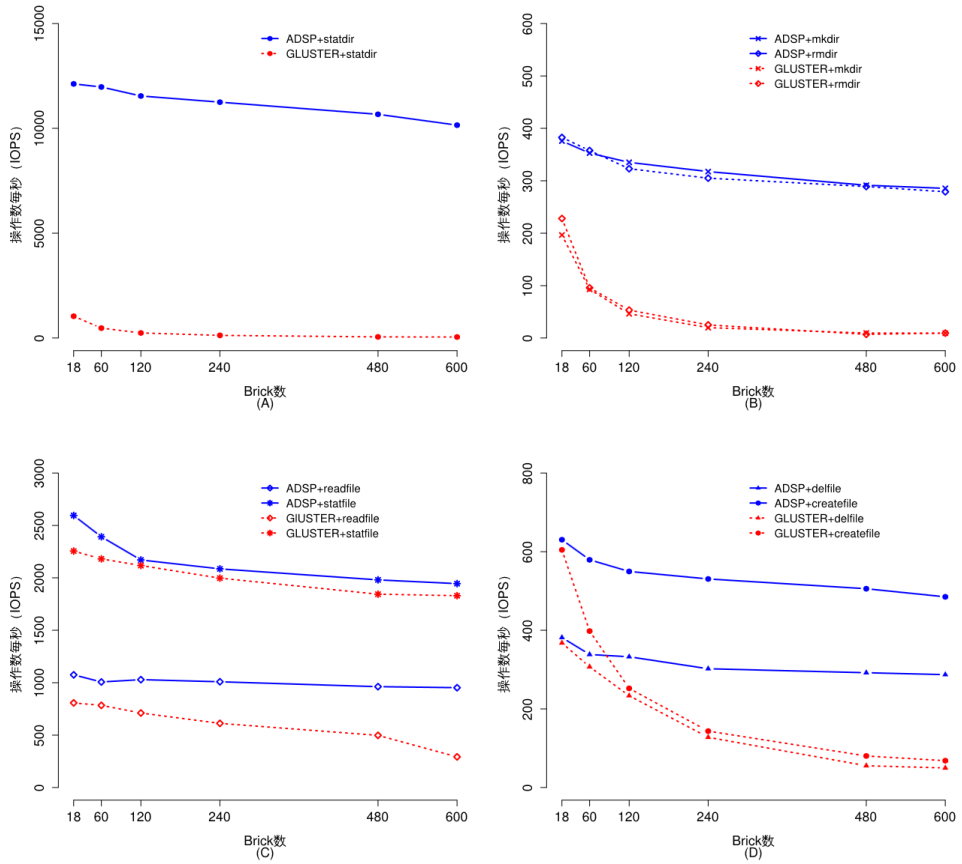

Figure 9. Scalability Evaluation between StarFS and Gluster

6. Conclusion

We have presented in this paper our works of designing and implementing a distributed metadata management to service HEP mass storage system. Our system proposes a new metadata management algorithm ADSP and a data storage method DULA to distribute metadata/data across the metadata/data cluster to achieve the load balance and high scalability. ADSP divides the filesystem namespace into sub-trees with directory granularity. Metadata will be stored on storage devices in flat structure, whose location information and file attributes are recorded as extended attributes. ADSP can adjust adaptively according to the workloads of metadata cluster so that the load balance could be improved. Experiments show that ADSP achieves higher metadata performance and scalability compared to Gluster and Lustre. DULA is an improved consistent hashing algorithm which is able to locate and contact the responsible data server directly in O(1). And the workloads are evenly distributed across the cluster.

Acknowledgments

This work was supported by the National Natural Science Foundation of China(NSFC) under Contracts No. 11305192.

References

[1] Quanqing Xu, Rajesh Vellore Arumugam, Khai Leong Young, et al. Efficient and Scalable Metadata Management in EB-scale File Systems. JOURNAL OF LATEX CLASS FILES, VOL. 6, NO. 1, Jul 2013.
[2] Gluster file system, http://www.gluster.org/
[3] IHEP (Institute of High Energy Physics, Chinese Academy of Sciences), http://english.ihep.cas.cn/
[4] Tweedie S.,EXT3, Journaling File system, Ottawa Linux Symposium, Ottawa Congress Centre, Ottawa, Ontario, Canada, 20 July, 2000.
[5] M. Avantika et al., The new ext4 filesystem: current status and future plans, Proceedings of the Linux Symposium, Ottawa, Ontario Canada, June 2007.
[6] STOICA I, MORRIS R, KARGER D, et al. Chord: A scalable peer-to-peer lookup service for internet applications; proceedings of the ACM SIGCOMM Computer Communication Review, F, 2001 [C]. ACM.
[7] WINTERNITZ R S. A Secure One-Way Hash Function Built from DES; proceedings of the IEEE Symposium on Security and Privacy, F, 1984 [C].
[8] P Bernardini. ARGO-YBJ experiment in Tibet. 2008 J. Phys.: Conf.Ser. Vol.120, 06
[9] Lustre file system, http://www.lustre.org
[10] WEIL S A, BRANDT S A, MILLER E L, et al. Ceph: A scalable, high-performance distributed file system; proceedings of the Proceedings of the 7th symposium on Operating systems design and implementation, F, 2006 [C]. USENIX Association.