CDFS: A high-efficiency Data Access System for Storage Federations

Shiyuan Fu[1,2,*] , *Qi* Xu[1,2] , *Yaodong* Cheng[1,2,3], *Gang* Chen[1,2]

[1]Institute of High Energy Physics, CAS, 100049 Beijing, China

[2]University of Chinese Academy of Sciences, 100049 Beijing, China

[3]Tianfu Cosmic Ray Research Center, Institute of High Energy Physics, Chinese Academy of Sciences, 610041 Chengdu, China

Abstract. High energy physics (HEP) experiments, such as LHAASO, produce a large amount of data, which is usually stored and processed on distributed sites. Nowadays, the distributed data management system faces some challenges such as global file namespace, efficient data access and storage. Focusing on those problems, this paper proposed a cross-domain data access file system (CDFS), applying data deduplication and compression as the storage-optimized engine, aiming at dynamically building an aggregate view of multiple distributed storages and accessing data in a fast and efficient way. The test based on the raw data of LHAASO experiment showed that the CDFS could present a unique repository based on distributed sites in LHAASO. And the storage-optimized engine reduces the storage consumption of the raw data by more than 50%.

1 Introduction

The Large High Altitude Air Shower Observatory (LHAASO)[1], aiming at exploring the origin of high-energy cosmic rays.

The amount of data collected from detectors is huge. The largest amount of data among them is generated by WCDA, 12Gb data per second, 50PB in a year. The DAQ system gets

[*] Corresponding author: fusy@ihep.ac.cn

raw data, and stores them directly without compression. The raw data is subsequently compressed by another program in off-line mode, which needs a lot of storage and time. The measures taken so far is removing old raw data weekly (monthly) to ensure the preservation of new raw data. A more efficient data storage strategy would allow more data to be stored.

In addition, nowadays LHAASO has three sites: Beijing, Chengdu and Daocheng, connected by private network, shown in fig 1. So the data is scattered among different sites. The data should be accessed conveniently without knowing which site it is located in and transferred between sites in a fast and efficient way.

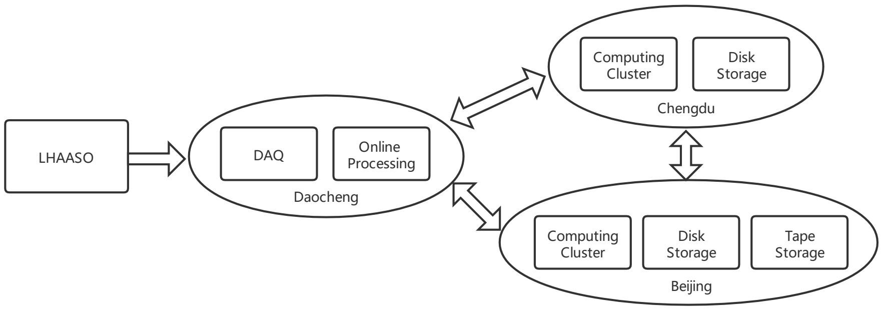

Fig. 1. Distributed sites in LHAASO

2 Related work

2.1 Global file namespace

Now widely used ways to build a global file namespace are centralized registration and dynamic federation. At the first mode, the system has its name services providing global file namespace, such as Rucio[2]. The second mode can automatically aggregate the global files view based on broadcast, such as Dynafed[3]. Although its speed of file-searching is slower than the first mode, it is more simple and dynamic.

2.2 Data access tools

In the process of cross-site data access, data access protocol and data transfer protocol often exist together. Application layer data transfer protocols include HTTP, FTP, GridFTP, etc.. Data access protocols, such as file system interface provided by AFS, NFS, Lustre, FUSE, provide direct and valid data access. At present, the XrootD framework, a sophisticated and scalable data access architecture[4] based on XRootD, has been widely used in high energy physics, including ATLAS, CMS.

2.3 Deduplication and compression

Data deduplication has been widely used to save storage space and network bandwidth, while lossless compression algorithms allows accurate reconstruction of the original content with less data. For lossless compression, expect widely used tools such as gzip, [5] uses the recurrent neural network to predict the probability distribution of the next character with knowing the previous characters to decrease the compressed data size.

3 The design and implementation

3.1 Design conception

CDFS is a cross-site data access system, which can be deployed on multiple sites, aiming at accessing data in an efficient and fast way, building global directory among different sites and storing more data in the limited storage space. According to the requirements of the LHAASO experiment, the design concept is listed as below.

1) Unified file directory among sites. Based on Dynafed, it is achieved by real-time update during file access with a database saving the accessed directory information.

2) Cross-site data transmission. Based on XrootD, the data is divided into small blocks and transmitted in multi-stream to realize the data access on demand.

3) Efficient storage of data. Reducing the space required for data storage is implemented by deduplication and compression.

3.2 Architecture

The architecture of CDFS is depicted in Figure 2. There are four components including TransferD, CacheD, DCFile and file plugin API. The DCFile is deployed in remote site and local site. Others are deployed in local site. Some details in [6].

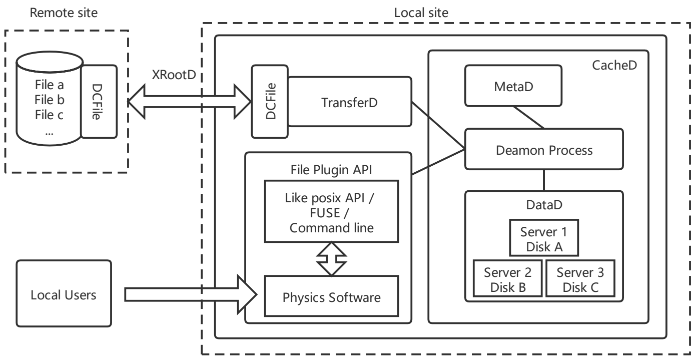

Fig. 2. The architecture of CDFS

3.3 Implementation

3.3.1 Global namespace in CDFS

Global namespace is built in Metadata. We use the dynamic federation mode with adding a metadata server to build a data federation among multiple sites.

As shown in figure 3, in this federation, each site has its own file namespace. Using Dynafed, all of the file namespace among different sites will be real-time aggregated into a global file namespace while accessing data. Users can easily view directories and find files without knowing their actual location. The metaD, which is implemented based on RocksDB, caches local metadata information, and further speeds up file searching locally.

So, when users access data at remote site, metaD will firstly check whether the location of the data is in local. If not, then the Dynafed will find it on broadcast and the location will be cached in metaD. In addition, bitmap in metaD can use less space to store the blocks information in the file has already been cached locally, for accessing remote files on demand.

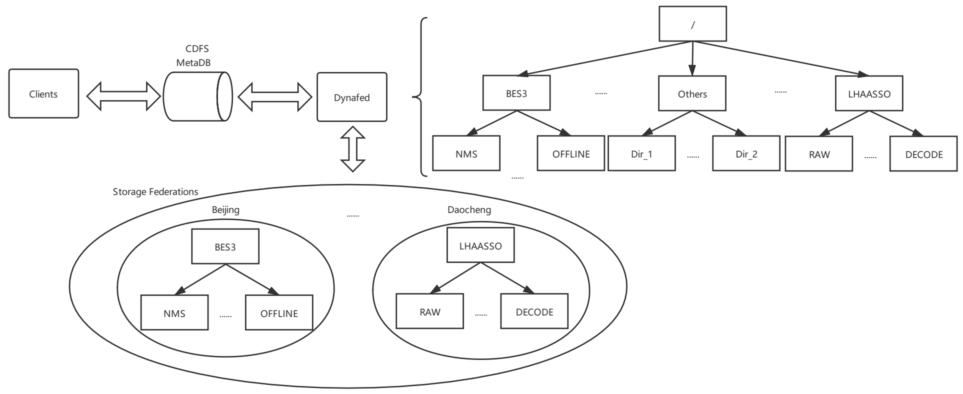

Fig. 3. Global namespace building

3.3.2 Data transfer in CDFS

Data transfer in independent blocks among sites in CDFS is based on Xrootd protocol. CDFS can handle multiple client requests at the same time, merge the same requests, work with CacheD to ensure on-demand access and transfer of data, and with DCFile to ensure that only blocks that do not exist at the target site are transferred.

3.3.3 DCFile in CDFS

DCFile includes deduplication and lossless compression. Although deduplication and lossless compression are both intended to reduce the amount of space for data storage with no loss of information, the granularity of their processing objects can be different. Deduplication will find out the same blocks. Compression is used to find out the redundancy in one block, which is on a smaller scale than duplication. So, data deduplication can be viewed as a pre-processing of compression.

3.3.4 Data deduplication

By computing sha256 based on data content, CDFS will assign a unique data ID to each data block. Different data block content will produce different data ID.

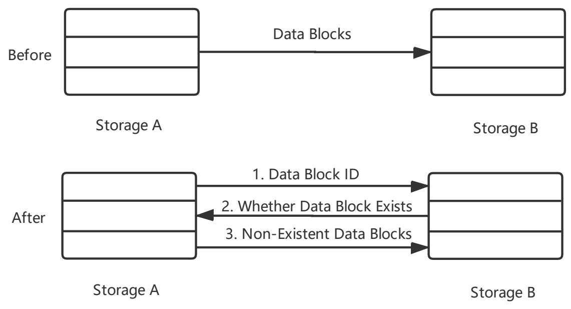

Fig. 4. The data flow before and after using deduplication

As shown in Figure 4, when storage side A receives the remote file transfer request from site B, the data ID is passed firstly. CDFS will check whether the data ID exists by searching in MetaD. If the data block ID has been in existence, we think the storage site B already has the same data block. So there is no actual transmission to the current data block. Otherwise, the compressed data will be transferred to the site B and stored. And the data ID will be stored in MetaD. So, data deduplication can eliminate the storage redundancy of the same data block in the same site. The data transfer tool is based on Xrootd.

3.3.5 Data compression

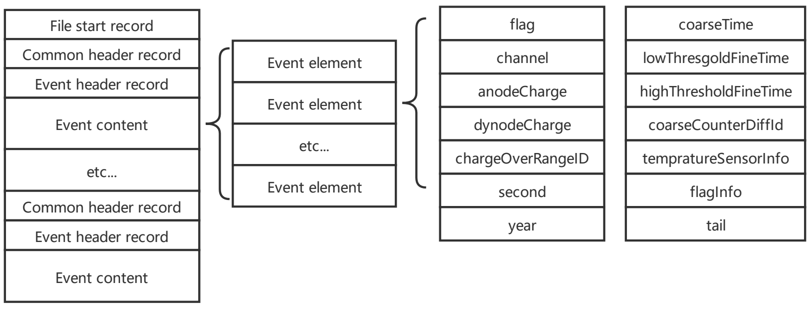

Fig. 5. The structure of WCDA data

Raw data are collected periodically. For example, WCDA collect multiple attributes at a time every 200ns. The attributes are stored as a file in sparse structure, as shown in figure 5, the content of WCDA file loops in the same format. Every event content has many events sequentially stored and each event has 14 attributes.

Using gzip, the file is compressed in whole and in 14 different attributes by column. Figure 6 is the comparison between the two methods. The compression ratio (CR) is the ratio of compressed file size to the original file size. The CR of compressed in columns is lower than in whole by about 20%.

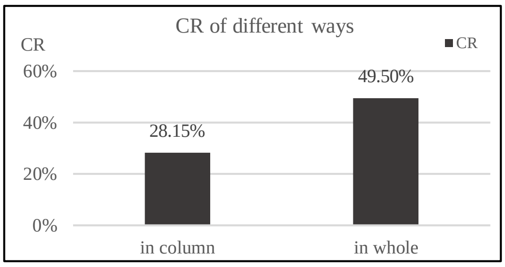

Fig. 6. Compression ratio of different ways

We believe that the reduction in compression is due to the fact that the change of the structure improves the correlation between the data before and after, which means the temporal relationship of the data is enhanced. So, DeepZip[6] is used to lower the compression ratio, using neural network to find the relationship in data. Three models are used as the predictor, LSTM, GRU and FC.

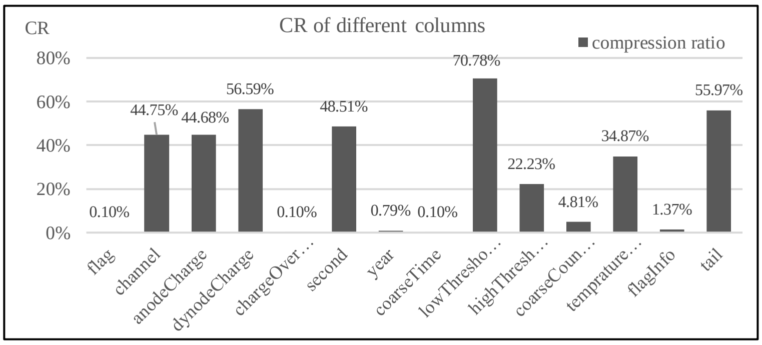

Fig. 7. Compression ratio of different columns

Here we set a simple standard that only taking into account columns with a compression ratio of more than 20%. Figure7 is the compression ratio among different columns. Column “lowThresholdFineTime” has the highest compression ratio, while column “lowThresholdFineTime” has the best compression ratio.

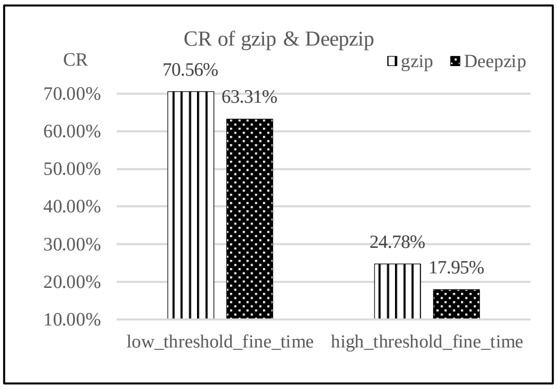

Fig. 8. Compression ratio of gzip and Deepzip

For the sake of simplicity, we use DeepZip to compress these two columns of WCDA data and find that GRU works best among the three model. Result is in Figure 11. The compression ratio of column "lowThresholdFineTime" has been improved more than seven percent, while that of the column "highThresholdFineTime" also has been improved six point eight percent.

4 Conclusion

In this paper, we introduce a cross-domain data access system CDFS. The CDFS dynamically builds an aggregate view of multiple distributed storage, and uses DCFile to eliminate redundant storage of the same data blocks at one site and minimize the space that one data block required. Finally, we try to compress data using neural network method, which is better than gzip. In the future, we will do more research on the relationship among different attributes by using neural network to compress the data better.

This work was supported by the National Key Research and Development Program of China (No. 2017YFB0203200) and National Natural Science Foundation of China (No. 11575223) .

Reference

1. Z. Cao, Chin. Phys. C **34**, 249 (2010)
2. C. Serfon, M. Barisits, T. Beermann, V. Garonne, L. Goossens, M. Lassnig, A. Nairz, R. Vigne, J. Phys. G. Nucl. Partic **273–275**, 969 (2016)
3. F. Furano, R. Rocha, A. Devresse, O. Keeble, A. Alvarez Ayllon, P. Fuhrmann, ISGC 2013 **179**, 15(2013)
4. A. Dorigo, P. Elmer , F. Furano , A. Hanushevsky, WSEAS Transactions on Computers **4(4)**, 348 (2005)

5. M. Goyal, K. Tatwawadi, S. Chandak, I. Ochoa, DCC 2019, 575 (2019)
6. Q. Xu, Z.J. Cheng, Y.D. Cheng, G. Chen, Lecture Notes in Computer Science **11473**,154 (2018)