Design of Data Storage and Management System Framework for Big Data for Internet of Things

Luo Jiawei

Dalian University of Science and Technology ,LiaoNing DaLian,116052

Keywords: Internet of things; big data; data storage

Abstract. The Internet of Things is a new network technology that has developed rapidly in recent years. Persistent storage of IoT big data and statistical analysis of the stored data can better manage the IoT system and save IoT applications. cost. However, the big data of the Internet of Things has massive characteristics, and traditional data storage technologies and management systems are difficult to meet actual needs. In addition, the need to quickly find data requires a more efficient new storage architecture. Based on this, this paper designs the DMFS system architecture and file writing process based on the Hadoop file system, and builds a distributed file system for massive small files. An aggregate query data system is designed on the basis of probability-oriented data OLAP, and the type and implementation of the query are designed. Based on the above research, an HDFS dynamic copy management strategy based on fragile storage is proposed.

Introduction

In recent years, the number of wireless sensor components of the Internet of Things system has been increasing, and the data tasks performed on these sensor components have become more diverse. With the development of modern Internet systems, the data space of Internet big data will reach a larger scale. Extracting valuable data from the Internet of Things system to improve management efficiency, office efficiency and improve daily life are important directions for the development of modern science and technology. For statistical analysis of the data stored in the Internet of Things system, a new system architecture is needed to manage the entire information space, and new storage and computing technologies are required to complete data statistics and analysis. It is also necessary to ensure real-time response of service information. Data storage technology and management systems put forward more stringent requirements. This paper uses distributed thinking to build a data storage and management system framework for IoT big data, which aims to improve system operation efficiency.

Design of Distributed File System for Massive Small Files

Considering the common problems of HDFS, in order to change the current situation of writing large numbers of small files, this study uses two designs: "write cache" and "cluster write". The former is to write files in memory and use cluster writing to improve files. Write throughput rate; the latter is a process of clustering small files from different sensors into large files, and writing these synthesized files into system memory[1]. The system architecture design of Sensor FS is shown in Figure 1. DMFS uses the "top" method, and it can be installed on the server of the system's main node storage node.

DOI: 10.25236/ISMHI.2019.133

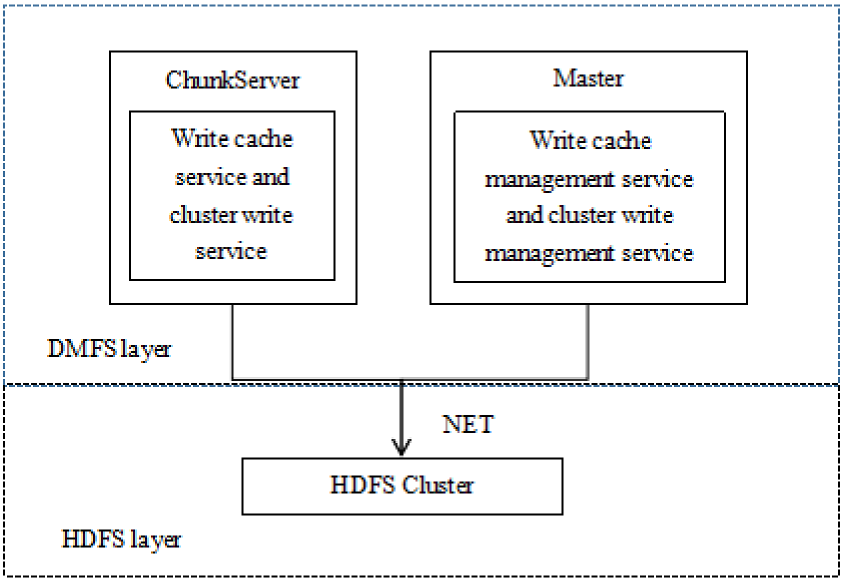
ChunkServer
Write cache service and cluster write service
Master
Write cache management service and cluster write management service
DMFS layer
NET
HDFS Cluster
HDFS layer

Figure 1 System architecture design of Sensor FS

The DMFS system architecture is shown in Figure 2. It consists of a master node and different storage nodes. The master node includes a write and transfer magic chopsticks module and a sensor clustering module. The former mainly receives the sensor's write request and monitors the client's request command; the latter mainly sends the write request data and receives the returned information[2]. Each storage node includes a write buffer and a write merge module, which is mainly responsible for cooperating with the master node to fully cluster the sensors.

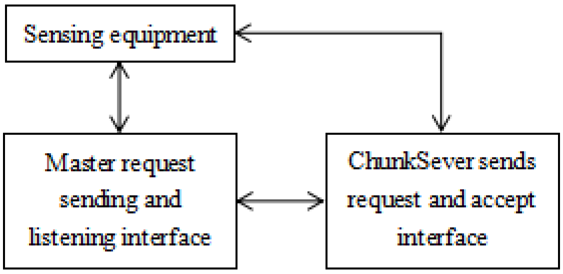
Sensing equipment
Master request sending and listening interface
ChunkSever sends request and accept interface

Figure 2 DMFS system architecture diagram

The workflow of DMFS is as follows: First, write scheduling. When the sensor in the system establishes a connection with the DMFS for the first time, the sensor connects to the write scheduling module of the manager, and the write scheduling module returns information such as the main storage address to the sensor according to the memory capacity. Second, file writing. The sensor establishes a connection with the main storage. The main storage receives a large number of small files in the form of TCP. After completing the write operation, it returns a command to the sensor to write successfully, and the corresponding data is also transmitted to each storage node. Third, cluster writing. When the total amount of files in the system reaches the threshold, the DMFS system initializes the files that need to be stored and calculates the sensor level accumulation. The corresponding management module summarizes the calculation results and completes the initial dependency graph initialization. The result determines the combined write position of the cluster.

Aggregated query data system architecture and replica dynamic management

The traditional method of expressing the fact table of data is to use the PWS model, but the probability data scale of this model may be smaller than the actual number, which affects the aggregated value. Based on this, this research redesigned the query framework for aggregated data, and built a data cube oriented to probability data (see Table 1).

Table 1 Probability data cube

C	D1	D2	Measure
(D1,D2)	D1	D2	{(I, 0.1), (2, 0.26), (3, 0.32), (4, 0.3)}
	D1	D2	{(I, 0.3)}
(D1)	D1	*	{(I, 0.I), (2, 0.26), (3, 0.32), (4, 0.3)}

After establishing a data cube oriented to probabilistic data, we need to study the materialization operations of the data cube. In order to balance the relationship between query efficiency and storage cost, this article adopts a partial materialization method, which uses a linear cost model ($Cost\left(c, c^{A}\right) = Cost_{10}\left(C, C^{A}\right) + Cost_{comp}\left(C, C^{A}\right)$) to evaluate the benefits of CUboid materialization operations, and selects some Cuboid for materialization operations based on the evaluation results[3-4]. In the query process, if the Cuboid has not been materialized into the storage device, the aggregation operation is performed on the materialized data first, and then the query is performed.

With the passage of time, the popularity of Internet big data will change to a certain extent, and the static copy management strategy has been difficult to meet the actual data management needs. Based on the need to improve data access performance, this study proposes a replica dynamic management strategy, that is, by dynamically adjusting the storage location of the replica to improve data access performance, it can also make data storage more secure. The adjustment of the copy dynamic management strategy is shown in Figure 3. For datanode internal data copy adjustment, the access frequency of data copies within a certain period is used as an index to evaluate the popularity of the data, and this index guides the data adjustment operation within the same Datanode.

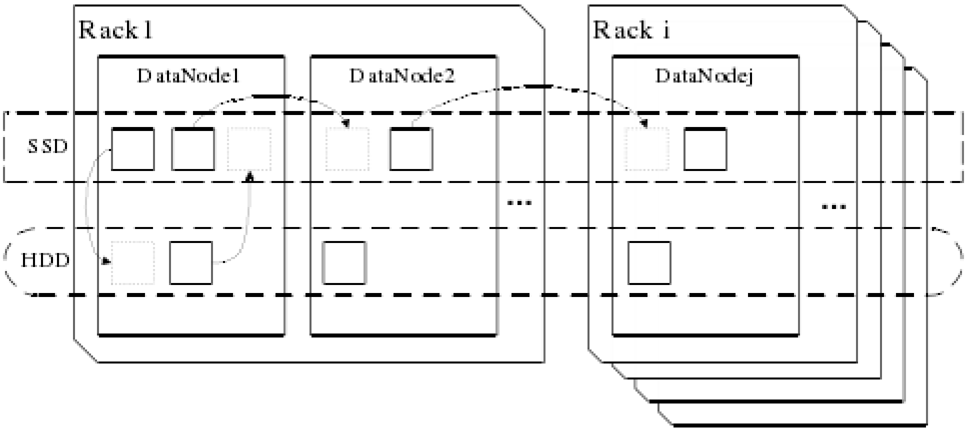

Figure 3 Tuning operations in a copy dynamic management strategy

Conclusion

In the era of big data, the pressure on the Internet to store massive amounts of data is constantly increasing. In the past, traditional persistent storage systems such as distributed and HDFS have been unable to meet the storage requirements of massive files, nor can they complete real-time, high-performance data storage. At the same time, statistical analysis of data stored in the Internet of Things system requires a new system architecture to manage the entire information space. Therefore, based on the existing research, this paper designs a data storage and management system framework for the Internet of Things big data, and designs and researches a distributed file system for massive small files, an aggregate query data system, and dynamic management of replicas. This system framework can better make up for the basic problems of traditional persistent storage systems, improve data access performance, and to a certain extent can reduce the storage and management pressure of the Internet of Things big data.

References

[1]Hongming Cai, Boyi Xu, Lihong Jiange, et al. IoT-Based Big Data Storage Systems in Cloud Computing: Perspectives and Challenges[J]. IEEE Internet of Things Journal, 2017, 4(1):75-87.

[2]Yang Yang, Xianghan Zheng, Wenzhong Guo, et al. (Revised Version) Privacy-preserving Smart IoT-based Healthcare Big Data Storage and Self-adaptive Access Control System[J]. Information Sciences, 2018, 479.

[3]Yuqing Mo. A Data Security Storage Method for IoT Under Hadoop Cloud Computing Platform[J]. International Journal of Wireless Information Networks, 2019, 26(2):51-52.

[4]Wei Wang, Peng Xu, Laurence T. Yang. Secure Data Collection, Storage and Access in Cloud-Assisted IoT[J]. IEEE Cloud Computing, 2018, PP(99):1-1.