Big Data Storage

Big Data Storage

Big Data Storage

Big Data Storage

This section reviews the following issues-

- Storage System for Massive Data
- Distributed storage systems
- Storage mechanism for Big data
  - Database Technology: Key-Value Databases, Column-Oriented Databases, Document Databases, Platform for Nimble Universal Table Storage
  - Design Factors
  - Database Programming Model: MapReduce, Dryad, All-Pairs, Pregel

The storage infrastructure needs to provide information storage service with reliable storage space, and it must provide a powerful access interface for query and analysis.

Storage infrastructure generally consists of hardware infrastructure and storage mechanisms

Big Data Storage-Storage System for Massive Data

4.1 Storage System for Massive Data
Data storage refers to storage and management of large-scale datasets, while achieving reliability and availability.

A data storage system consists of two parts: infrastructure and data storage methods or mechanisms.

A large number of storage systems emerged to meet the demands of big data. Existing storage technologies can be classified as -

- Direct Attached Storage (DAS)
- Network Storage
  - Network Attached Storage (NAS)
  - Storage Area Network (SAN)

Big Data Storage-Storage System for Massive Data

4.1.1 DAS
In DAS, disc drives are directly connected with servers.

DAS applies to a few server environments but, when the storage capacity is increased, the efficiency of the storage supply will be quite low, and the upgradeability and expandability will be greatly limited.

DAS is mainly used in personal computers and small-sized servers, which only support such applications requiring low storage capacities and does not directly support multicomputer shared storage.

Tap drivers and RAID (redundant array of independent disks) are classic DAS equipment.

Big Data Storage

4.1.2 Network storage

Network storage is to utilize the original network or a specially-designed storage network to provide users with uniform information access and sharing services of information systems.

Network storage equipment includes special data exchange equipment, disk array, tap library, and other storage media, as well as special storage software.

It is characterized with mass data storage, limited data sharing, full utilization of data mining and information, data reliability, data backup and safety, as well as simplified and unified data management.

Big Data Storage

NAS is actually an auxiliary storage equipment of a network. It is directly connected to a network through a hub or switch, communicating with the TCP/IP protocol. (TCP-Transmission Control Protocol)

SAN focuses on data storage with a flexible network topology and high-speed optical fiber connections. It allows multipath data switching among any internal nodes.

DAS, NAS, and SAN can all be divided into three parts:

- disk array: it is the foundation of a storage system and the fundamental guarantee for data storage;
- connection and network sub-systems, which provide connection among one or more disc arrays and servers;
- storage management software, which handles data sharing, disaster recovery, and other storage management tasks of multiple servers.

Big Data Storage-Distributed Storage System

4.2 Distributed Storage System

To use a distributed system to store massive data, the following factors should be taken into consideration: Consistency, Availability, Partition Tolerance (CAP).

Consistency: A distributed storage system requires multiple servers to cooperatively store data. As there are more servers, the probability of server failures will be larger. Usually, data is divided into multiple pieces to be stored at different servers to ensure availability in case of server failure. However, server failures and parallel storage may cause inconsistency among different copies of the same data. Consistency refers to assuring that multiple copies of the same data are identical.

Big Data Storage-Distributed Storage System

Availability: A distributed storage system operates in multiple sets of servers. As more servers are used, server failures are inevitable/unavoidable. It would be desirable if the entire system is not seriously affected with respect to serving the reading and writing requests from customer terminals. This property is called availability.

Partition Tolerance: Multiple servers in a distributed storage system are connected by a network. The network could have link/node failures or temporary congestion. The distributed system should have a certain level of tolerance to problems caused by network failures. It would be desirable that the distributed storage still works well when the network is partitioned.

Big Data Storage

Eric Brewer proposed a CAP theory in 2000, which indicated that a distributed system could not simultaneously meet the requirements on consistency, availability, and partition tolerance; at most two of the three requirements can be satisfied simultaneously.

Seth Gilbert and Nancy Lynch from MIT proved the correctness of CAP theory in 2002. Since consistency, availability, and partition tolerance could not be achieved simultaneously, we can have a CA system by ignoring partition tolerance, a CP system by ignoring availability, and an AP system that ignores consistency, according to different design goals.

Big Data Storage

CA systems do not have partition tolerance, i.e, they could not handle network failures. Therefore, CA systems are generally deemed as storage systems with a single server, such as traditional small-scale relational databases.

CP systems ensure partition tolerance. CP systems generally maintain several copies of the same data in order to ensure a level of fault tolerance. CP systems also ensure data consistency, i.e., multiple copies of the same data are guaranteed to be completely identical. BigTable and Hbase are two popular CP systems. BigTable is designed in the way similar to GFS, a distributed file system of Google, where one Master and several Tablet Servers constitute a star structure in a system.

Data in BigTable is sequenced in the lexicographic order of rows. During data modification, we shall insert a record in a sequential table, find a position to be inserted, and then move the original data to make room for the newly inserted data.

Big Data Storage

AP systems, also ensure partition tolerance. However, AP systems are different from CP systems in that AP systems also ensure availability. However, AP systems only ensure eventual consistency rather than strong consistency in the previous two systems. Dynamo and Cassandra are two popular AP systems.

Cassandra, with sound expandability, is used for storing massive textual data by mainstream commercial online SNS (Social Networking Services) companies, such as Facebook and Twitter.

In order to support the storage of textual data of users, Cassandra inherits the column family model of BigTable to aggregate data with similar features into a column family.

Big Data Storage: Storage Mechanism for Big Data

4.3 Storage Mechanism for Big Data

4.3 Storage Mechanism for Big Data

Database Technology
Design Factors
Database Programming Model

Database Technology

□ Key-Value Databases
□ Column Oriented Databases-BigTable, Cassandra, Derivative Tools of BigTable
□ Document Databases-MongoDB, SimpleDB, CouchDB
□ Platform for Nimble Universal Table Storage

Big Data Storage: Storage Mechanism for Big Data

Design Factors

Data Model, Data Storage, Concurrency Control, Consistency, CAP Option.

Database Programming Model

□ MapReduce
□ Dryad
□ All-Pairs
□ Pregel

Big Data Storage: Storage Mechanism for Big Data

Storage Mechanism for Big Data

Existing storage mechanisms of big data may be classified into three bottom-up levels: (a) file systems, (b) databases, and (c) programming models.

File systems are the foundation of the applications at upper levels. Google's GFS is an expandable distributed file system to support large-scale, distributed, data-intensive applications. GFS uses cheap commodity servers to achieve fault tolerance and provides customers with high-performance services. GFS supports large-scale file applications with more frequent reading than writing.

GFS also has some limitations, such as a single point of failure and poor performances for small files. Such limitations have been overcome by Colossus, the successor of GFS.

Big Data Storage: Storage Mechanism for Big Data

In addition, other companies and researchers also have their solutions to meet the different demands for the storage of big data. For example, HDFS and Kosmosfs are derivatives of open-source codes of GFS.

Microsoft developed Cosmos to support its search and advertisement business.

Facebook utilizes Haystack to store the large amount of small-sized photos. Taobao also developed TFS and FastDFS.

Big Data Storage: Storage Mechanism for Big Data

4.3.1 Database Technology

The database technology has been evolving for more than 30 years. Various database systems are developed to handle datasets at different scales and support various applications.

NoSQL databases (i.e., non-traditional relational databases) are becoming more popular for big data storage. NoSQL databases feature flexible modes, support for simple and easy copy, simple API, eventual consistency, and support of large volume data. (API: Application Programming Interface)

NoSQL databases are becoming the core technology for big data.

The following three main NoSQL databases are: Key-value databases, column-oriented databases, and document-oriented databases, each based on certain data models.

4.3.1.1 Key-Value Databases

Key-value Databases are constituted by a simple data model and data is stored corresponding to key-values. Every key is unique and customers may input queried values according to the keys.

Over the past few years, many key-value databases have appeared as motivated by Amazon's Dynamo system.

Several other representative key-value databases: Dynamo, Voldemort.

Dynamo is a highly available and expandable distributed key-value data storage system. It is used to manage the store status of some core services in the Amazon e-Commerce Platform. Amazon e-Commerce Platform provides multiple services and data storage that can be realized with key access.

Key-Value Databases.....

Voldemort is also a key-value storage system, which was initially developed for and is still used by LinkedIn. Key words and values in Voldemort are composite objects constituted by tables and images.

The Voldemort interface includes three simple operations: reading, writing, and deletion, all of which are confirmed by key words.

Other key-value storage systems include Redis, Tokyo Cabinet and Tokyo Tyrant, Memcached and MemcacheDB, Riak and Scalaris.

Big Data Storage: Storage Mechanism for Big Data

4.3.1.2 Column-Oriented Databases

The column-oriented databases store and process data according to columns other than rows. Columns and rows are segmented in multiple nodes to realize expandability. The column-oriented databases are mainly inspired by Google's BigTable.

Big Data Storage: Storage Mechanism for Big Data

BigTable

BigTable is a **distributed, structured data storage system**, which is designed **to process the large-scale (PB class) data** among thousands of commercial servers. The basic data structure of BigTable is a multi-dimension sequenced mapping with sparse, distributed, and persistent storage.

Indexes of mapping are key words of rows, key words of columns, and timestamps, and every value in mapping is an unanalyzed byte array. The keywords of rows in BigTable are **64KB character strings**, in which the **rows are stored according to the lexicographical order** and are continually segmented into Table.

BigTable is based on many fundamental components of Google, including GFS, cluster management system, SSTable file format, and Chubby. GFS is used to store data and log files.

Big Data Storage: Storage Mechanism for Big Data

Cassandra
Cassandra is a distributed storage system to manage the huge amount of structured data distributed among multiple commercial servers. The system was developed by Facebook and became an open-source tool in 2008.

It adopts the ideas and concepts of both Amazon Dynamo and Google BigTable, especially integrating the distributed system technology of Dynamo with the BigTable data model.

Tables in Cassandra are in the form of distributed four-dimensional structured mapping, where the four dimensions include row, column family, column, and super column.

Big Data Storage: Storage Mechanism for Big Data

Derivative Tools of BigTable
Since the BigTable code cannot be obtained through the open source license, some open source projects compete to implement the BigTable concept to develop similar systems, such as HBase and Hypertable.

HBase is a BigTable clone programmed with Java and is a part of Hadoop of Apache's MapReduce framework.

HBase replaces GFS with HDFS. It writes updated contents into the RAM and regularly writes them into files in discs.

4.3.1.3 Document Databases

Compared with key-value storage, document storage can support more complex data forms. Since documents do not follow strict modes, there is no need to conduct mode migration.

The three important representatives of document storage systems, MongoDB, SimpleDB, and CouchDB.

MongoDB is an open-source document-oriented database. MongoDB stores documents as Binary JSON (BSON) objects, which is similar to objects.

Every document has an ID field as the main key word.

(JSON stands for JavaScript Object Notation)

SimpleDB is a distributed database and a web service of Amazon. Data in SimpleDB is organized into various domains in which data may be stored, acquired, and queried. SimpleDB allows users to use SQL to run queries.

Apache CouchDB is a document-oriented database written in Erlang. Data in CouchDB is organized into documents that consist of fields named by key words/names and values, and are stored and accessed as JSON objects. Every document is provided with a unique identifier. CouchDB allows access to database documents through the RESTful HTTP API.

Big Data Storage: Storage Mechanism for Big Data

4.3.1.4 Platform for Nimble Universal Table Storage

Platform for Nimble Universal Table Storage (PNUTS) is a large-scale parallel geographically-distributed system for Yahoo!'s web applications.

It relies on a simple relational data model in which data is organized into a property record table.

Big Data Storage: Design Factors

4.3.2 Design Factors

Of the various database systems, there is not a single system that can achieve the optimal performance under all workload circumstances.

In each database system, some performance goals have to be compromised to achieve optimized operation for specific applications.

Cooper et al. discussed the trade-offs confronted by data management systems based on cloud computing, including reading performance and writing performance, delay and durability, synchronous and asynchronous copies, and data segmentation, among others.

In the following, the several prominent features of the existing database systems (rather than analyzing the design goals of a specific system): Data Model, Data Storage, Concurrency Control, Consistency, CAP Option.

Big Data Storage: Storage Mechanism for Big Data

4.3.3 Database Programming Model

The massive datasets of big data are generally stored in hundreds and even thousands of commercial servers.

Apparently, the traditional parallel models (e.g., Message Passing Interface (MPI) and Open Multi-Processing (OpenMP)) may not be adequate to support such large-scale parallel programs.

Some parallel programming models have been proposed for specific fields. These models effectively improve the performance of NoSQL and reduce the performance gap between relational databases. Therefore, these models have become the corner-stone for the analysis of massive data.

Big Data Storage

MapReduce

MapReduce is a simple but powerful programming model for large-scale computing using a large number of clusters of commercial PCs to achieve automatic parallel processing and distribution.

In MapReduce, the computational workloads are caused by inputting key-value pair sets and generating key-value pair sets. The computing model only has two functions, i.e., Map and Reduce, both of which are programmed by users.

The Map function processes input and generates intermediate key-value pairs.

The Reduce function receives the intermediate key and its value set, merges them, and generates a smaller value set.

Big Data Storage

Dryad

Dryad is a general-purpose distributed execution engine for processing parallel applications of coarse-grained data.

The operational structure of Dryad is a directed acyclic graph, in which vertexes represent programs and edges represent data channels.

Dryad executes operations on the vertexes in computer clusters and transmits data via data channels, including documents, TCP connections, and shared-memory FIFO.

During operation, resources in a logic operation graph are automatically mapped to physical resources. The operation structure of Dryad is coordinated by a central program called job manager, which can be executed in clusters or workstations of users.

Big Data Storage

All-Pairs

All-Pairs is a system specially designed for biometrics, bio-informatics, and data mining applications. It focuses on comparing element pairs in two datasets by a given function.

The All-Pairs problem may be expressed as a three-tuples (Set A, Set B, and Function F), in which Function F is utilized to compare all elements in Set A and Set B.

The comparison result is an output matrix M. It is also called the Cartesian product or cross join of Set A and Set B.

All-Pairs is implemented in four phases: system modeling, input data distribution, batch job management, and result collection.

Big Data Storage

Pregel
The 'Pregel system of Google' facilitates the processing of large-sized graphs, e.g., analysis of network graphs and social networking services.

A computational task is expressed by a directed graph constituted by vertexes and directed edges, in which every vertex is related to a modifiable and user-defined value.

Directed edges are related to their source vertexes and every edge is constituted by a modifiable and user-defined value and an identifier of a target vertex.

After the graph is built, the program conducts iterative calculations, which are called super steps among which global synchronization points are set until algorithm completion and output completion.