2011 10th IEEE/ACIS International Conference on Computer and Information Science

A Novel Solution of Distributed Memory NoSQL Database for Cloud Computing

Jing Han, Meina Song, Junde Song
School of Computer, Beijing University of Posts and Telecommunication
Beijing 100876, China
babyblue110128@hotmail.com; mnsong@bupt.edu.cn; jdsong@bupt.edu.cn

Abstract

The traditional relational database to online storage are becoming increasingly problematic: the low performance do not gracefully meet the needs of mass data, the storage approaches of massive data are still not perfect presents. NoSQL and distributed memory database technologies have the potential to simplify or eliminate many of these challenges. NoSQL database technologies can provide Key-value style of data storage and largely ensure high performance. Distributed memory database technologies provide a means for easily store mass data in cloud in a dynamic and scalable manner.

This paper argues for a new architecture called CDSA, which is a distributed memory NoSQL database architecture for Cloud computing to improve the performance of querying data and ensure mass data storage in cloud by using rational strategy. Furthermore, add or deleted any node from the distributed database cluster, the others node can work without stop service. We believe that CDSA can provide durable storage with high throughput and lower access latency

Keywords: cloud computing; NoSQL; cache; memory database; distributed

I. INTRODUCTION

In order to store and manage data in an efficient way, database management system (DBMS) was conceptualized by industry in the 1960s and developed the early DBMS from file system. With the expansion of applications for handling more complex data model, more complex query operation and more extensive database, gradually, distributed database was developed, parallel database, database cluster, data warehouse and so on. All of them were met the application requirements in some ways, but the support of massive data are still not perfect. At the same time, Internet becomes a huge repository of information and vast amounts of data generated every day. If want to retrieve the data, you need massive storage space and large scale computing, which existing database system cannot afford, and large server has issues such as limited performance and high price. Therefore, Google established a large scale cluster using a lot of cheap PC, designed and implemented a distributed file system named GFS[1], a storage system named BigTable[2] and a parallel programming environment (MapReduce[3][4]), Which constitutes Google's "cloud computing" environment, other companies have also proposed and implemented similar cloud system, including Amazon's EC2[5], S3[6], and IBM's "Blue Cloud" and so on.

If faced mass data, the existing database system would be lacked performance and cannot afford the high prices of database software, in future mass data must be migration to the cloud, but the ACID properties which database system required will lead to serious performance reducing of part of operation when stored the data distributed.

To ensure high availability, high reliability and economy, the cloud data storage must be stored redundant to ensure reliability. In addition, the cloud computing system needs to meet the needs of large number of users simultaneously, provide service to users paralleled. Therefore, data storage technology of cloud computing requires high throughput and high transfer rate.

This paper presents high performance data storage architecture for Cloud Computing to improve the performance of storing distributed data by using more rational strategy.

This paper is structured in the following way: Section 2 describes related work. Section 3 introduces the internal architecture of Cloud Data Storage Architecture (CDSA). Section 4 concludes the paper.

II. RELATED WORK

A. Cloud Computing

Cloud Computing[7] was mainly proposed by Google, Amazon and IBM and so on. There are several definitions currently, and each described cloud computing from a different point of view.

In which a comprehensive definition[8] is: “cloud computing is a platform (system) or a type of application. A cloud computing platform provision, configuration, reconfigure and provision on-demand and so on. In a cloud computing platform, the server can be physical server or virtual server. Advanced computing clouds usually contain a number of other computing resources, such as Storage Area Networks, network equipment, firewall and other safety equipment. Cloud computing describes a scalable applications which can access through the Internet. ‘Cloud applications’ use large-scale data centers and a strong server, web application and network services can run in it. Any user can access a cloud computing applications through appropriate equipment and a standard Internet browser. ”

This work is supported by the National Key project of Scientific and Technical Supporting Programs of China (Grant Nos.2008BAH24B04, 2008BAH21B03); the Program of the Co-Construction with Beijing Municipal Commission of Education of China.

978-0-7695-4401-4/11 $26.00 © 2011 IEEE
DOI 10.1109/ICIS.2011.61

IEEE computer society

Despite there exist a variety of understanding of cloud computing, it's mainly the following three basic characteristics: infrastructure on a large scale low-cost server clusters; applications developed in collaboration with the underlying services to maximize the use of resources; to ensure high availability by the redundancy among multiple low-cost servers.

The feature of data storage in cloud computing is that, data is automatically distributed in different storage nodes in the cluster, each data storage node only preserved a fragment of whole data, at the same time, user can set to store data in different storage nodes to ensure that a single point failure will not cause data loss.

B. Cloud Database

As the development of large-scale distributed service and distributed storage which clouds computing needs, traditional relational database are facing many new challenges, especially in scenarios of the large scale and high concurrent SNS applications, by the use of a relational database to store and inquires users' dynamic data has been ragged and exposes many problems to overcome, such as there need high real-time insert performance; Need mass data storage capacity, still need high retrieval speed; Need to store data seamless spreading throughout the distributed cluster environment, and have ability of online expansion, etc. According this, NoSQL[9] is made.

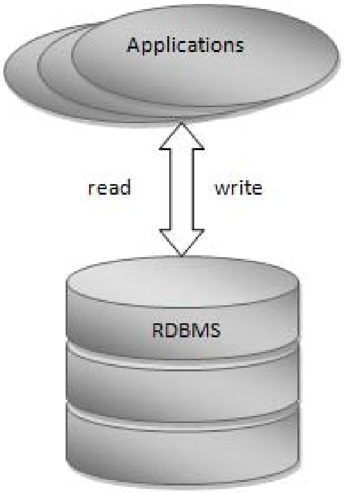

Figure 1. Simple Data R&W Process

NoSQL database broke the paradigm constraint of traditional relational database. From a storage perspective, NoSQL is not a relational database, but a hash database with format of Key-value[10]. Due to the abandoned powerful SQL query language, transaction-consistent and constraints in paradigm of relational database, NoSQL database largely solved challenges which the traditional relational database is facing.

Challenges traditional relational database facing in cloud computing[11]:

1) High Performance - demand on the high concurrent literacy of database

In cloud computing, users' complicated needs require real-time formation of dynamic pages and provide dynamic information, it cannot use technology to staticized dynamic pages, so database concurrent is too high and tend to thousands of times reading request per second. A relational database is hard to cope with thousands of times SQL data written request, also the hard disk cannot bear.

2) Huge Storage – demand on the efficient storage and visit of mass data

The mass data generated dynamically in cloud computing, for relational database, recorded hundreds of millions of records in the storage in a table and the efficient to run SQL query is extremely low.

3) High Scalability & High Availability

In the web-based framework, database is too hard to extend horizontally. When the volume and traffic of an application system is growing, for Web Server and App Server, adding more hardware and service node to expand the performance and load capacity, but for database there have no simple way like that. For a lot to application system deployed in cloud should be provided 24-hour uninterrupted service, upgrading and extension of database system are very difficult, often require downtime and data migration, but cannot through adding database services node continuously to implement expansion.

Meanwhile, many of the main characters of relational database in the cloud often have no use.

1) Database transaction consistency

Many system deployed in the clouds does not require strict database transaction, and have low requirements of read consistency; some occasions write consistency requirement is not high also. Therefore, database transaction management became high load of an under-heavy-burden database.

2) Complex SQL query, especially demand on multi-table association query

Any system with mass data has difficulty on multiple associated queries among big tables, and complex data analysis and complex data report. In cloud computing, simple conditions paging query of single table with primary key is more often, association query of relational database query have been are greatly weakened.

As cloud computing is an Internet-based application (environment), the database used in cloud computing are key / value model which is for Internet.

III. CDS ARCHITECTRUE

This paper presents a solution which oriented to high performance cloud computing for distributed-memory database NoSQL. The solution was used to solve the problems like data storage and access issues in the massive highly concurrent systems of cloud computing environment。

According to the functions of various components, the system logic can be abstracted as three layers: Data Cache layer, Memory Database layer and Disk Database Layer.

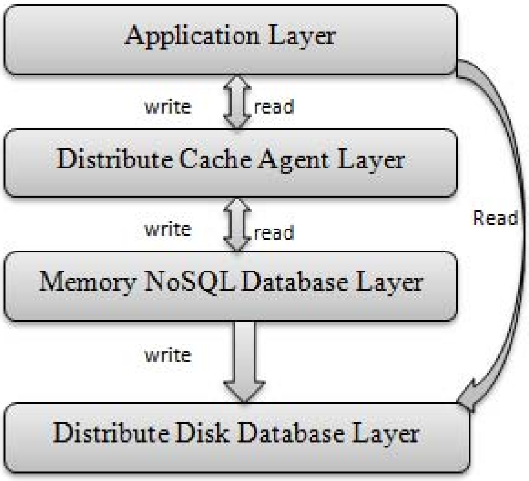
Application Layer
write
read
Distribute Cache Agent Layer
write
read
Read
Memory NoSQL Database Layer
write
Distribute Disk Database Layer

Figure 2. CDS Archetectrue.

A. Data Cache Layer（DCL）

Currently, the way of centralized deployment was used to handle the problem of cache. The distributed deployment of database will lead to the change of cache when the data in any node was updated.

Particularly the larger of distributed database will takes the problems like the bigger load of current cache in the middle tier, too expensive to maintain the system consistently, and weak practically of system. In addition, the frequent consistency Maintenance will reduce the utilization of cache. Therefore, we propose a new cache system which was applied in distributed environment. And the primary feature of this cache system is maximizes the role of the cache. The definition of distributed cache is all the service nodes just store and maintain the cache data in current database and its corresponding query information. During the data updating of all databases will not affect the cache information of other databases. From the task of Agent in the middle layer, the cache data of each service node can be used to summary the query results.

The structure of cache stored should same as table structure in disk, and the data cannot have redundancy. Although the rational use of this method can improve the efficiency of database queries, but some common requests are also not timely respond. For example, everyday users statistic the sales the day before, at the first query in cache, they must not hit in the cache, so the first response time of query is often very long, which is a shortcoming which cache technology cannot overcome.

We use "Optimal Prefetch(OP)" method to resolve this problem. Work presented in [12] discusses the Prefetch method, but the data pattern should be input in advance. To meet the need of unknown data pattern of cloud database, the core ideas of OP method is to update distributed cache before the first query of user, the specific method is: record all the queries of users, predict queries may be ran, queries at the appropriate time to active cache to work, stored the relevant information in each distributed cache. This method will solve the problem of the first hit to cache and reduce the user delay when the first query, so that cache can play a better role.

OP cache is a method to capture change of data; it is based on a prediction algorithm, in cache layer OP is used to improve query response speeds. Record the user's query history, extract the appropriate query script, abstracted the query with different parameter values as a template; statistic from the historical probably query parameters of each query and sort, OP objects is the high probability template and its parameters. However, typical applications of this method are: the use of access the database structure to create dynamic Web page, because the mode of database is predefined, it is appropriate for simple statistical analysis and access prediction.

B. Memory Database Layer（MDL）

Before the persistence of data, the data in the cloud were stored in memory firstly, for the efficiency of memory storage is much higher than the performance of disk storage.

Combined with the requirements of data storage which previous analyzed for high performance data and mass storage in the cloud computing environment, NoSQL came to the only solution.

Therefore, these two considerations can be deduced from an idea: memory NoSQL database, use "memory" as a "disk", read and write operations no longer interact directly with the disk database but with memory databases, which avoid the time delay and the same issues in simple NoSQL read-write- separation structure.

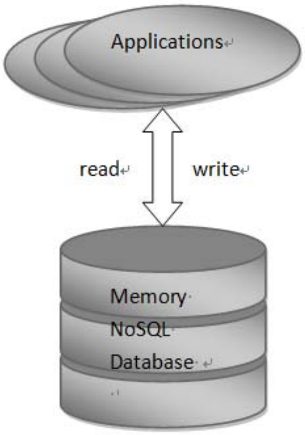

Figure 3. Read and write operations in Memory Database

In the transactional database system, the memory database required for a consistency with disk database in the startup. Therefore, the memory database need to have the same table definition with library database; and in the first time startup, it required to load all data in table of disk database into memory database.

C. Disk Database Layer（DDL）

Key-Value database is mainly characteristic of high concurrently read and write performance. Key-Value memory database load the entire database in memory, which flush data from memory database to disk through the asynchronous operation regularly. In addition, by setting expire time of Key-Value can also capture the changed data in memory database. Because operations is ran purely in memory, Key-Value memory database's performance is very good, more than million times read and write operations can be handled per second.

Changes of data in memory database need to be copied to the disk database. Then copy the data from memory to disk can be seen as the process of an original asynchronous writing operation, obviously, the asynchronous write operation make faster.

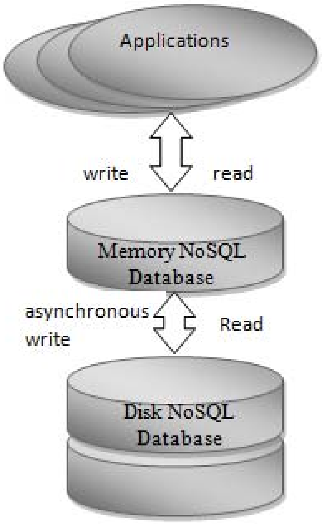

Figure 4. Data synchronization from Memory Database to Disk Database

D. Distributed storage in CDSA

Distributed database in cloud computing is a database service system which constitute from database distributed in different nodes, and according this distributed architecture to provide online flexible scalability. For example, more data nodes can be added or deleted without stop service and so on. Cloud database is not just a database, but a distributed database network service which was constituted with a large number of database nodes. A write operation of cloud database will be copied to other nodes, read request will be routed to a node in cloud. For a cloud database cluster, the scalability is become easy; just add a node in the cluster inside cloud. Cloud distributed database is necessarily a multiple-node system; its performance of concurrent depends on the number of nodes in the system and routing efficiency.

By separated the operation of reading and writing, vertical and horizontally segmentation, mass data can be stored Cloud memory NoSQL database cluster. As shown below, the database cluster distributed proxy is in charge of controls such as data routing.

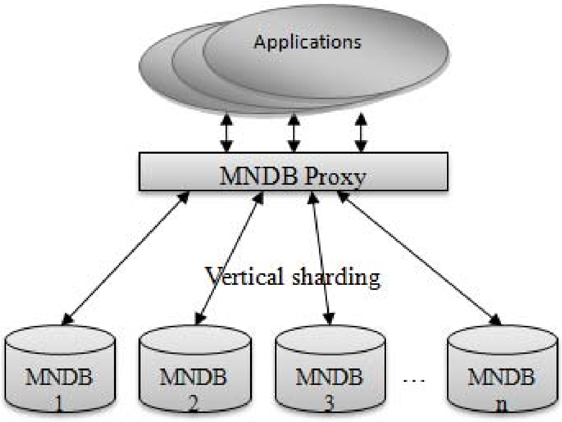

Figure 5. Memory NoSQL Database cluster

When copy data from memory database to disk database, it will use hybrid partition method. The partition which was committed by a NoSQL node in Memory Database Layer will be committed by a memory node and a common NoSQL node. The database schema will be formed by the two databases.

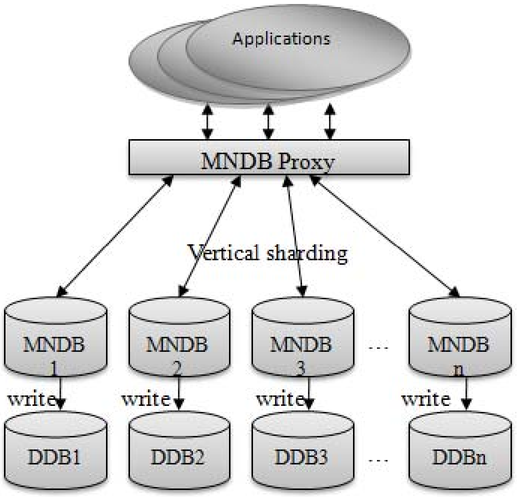

Figure 6. H-Shard cluster

IV. PERFORMANCE OF CDSA

MDL of CDSA is created by aggregating the main memories of servers. The CDSA can provide durable and available storage with 100x the throughput of only use disk database and lower access latency. The combination of low latency and large scale will enable a new breed of data-intensive applications.

This section focused on the test of CDSA access patterns. Through the basic operations of insert, read, update, and delete, to test both the throughput and average response time, at the same time to verify the correctness and effectiveness of CDSA. Using the test environment specifically as follows:

To test throughput and response time of CDSA, firstly nmoll is used as tools to capture the performance of the test program and write to a file, and then use the nmoll as an analyser to analyze the data file, finally we select the test results of CPU and I/O of disk to create a graphical report.

Test program insert, update, read or delete 500 million records respectively, the time consuming was recorded. First the test program opened the database, the records were read from a disk database to memory corresponds, and then run the Insert. Update, Read or Delete operations, finally synchronized data from memory to disk file.

TABLE I. TEST RESULT-DATA OPERATION

Operations	open database (s)	run (s)	close database (s)
insert	2	17	4
update	0.7	8	2
read	1	5	3
delete	1	5	2

TABLE II. TEST RESULT-AVG RESPONSETIME&THROUGHPUT

Operations	average response time (μ s)	throughput
insert	3	316000
update	1.5	562000
read	1.2	753000
delete	1	671000

Table I and table Ⅱ show the test result: the average response time is in the level of microseconds. Because of logic is relatively simple, the test results may be relative better than the practical, but it is certain that the CDSA is a good choice for cloud.

V. CONCLUSION

According to the characteristics of distributed storage in cloud, by introducing memory database and NoSQL, we propose an architecture of distributed memory NO-SQL database for cloud, to address massive data storage and high concurrent access issues cloud in computing environments. The core idea is:

1) *High performance:*

By using a cache layer and a memory NoSQL database, to provide high-performance database access service, which is the main goal of this architecture;

2) *Persistence*

By using of memory database and disk database to implement asynchronously copy from memory database to disk database;

3) *Mass data storage*

By created a distributed cloud database cluster in the cloud to achieve mass data storage;

4) *High Availability*

Based on the distributed database proxy, by using MDB master node and slave node to implement high availability further; the secondary disk database can achieve fast data recovery。

The statistical data shows that this architecture performs well and brings a lot of benefits. However, there is limit in the management of user-defined data resources, the flexibility of cloud service controls, and search in different data resources. We plan to address these problems in the future releases.

REFERENCES

[1] S. Ghemawat, H. Gobioff, and S. Leung The Google File System In: Proc. of the Nineteenth ACM Sym-posium on Operating Systems Principles, October, 2003

[2] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E. Gruber Bigtable: A Distributed Storage System for Structured Data. In Proceedings of OSDI'06: Seventh Symposium on Operating System Design and Implementation, November, 2006

[3] Amazon. Amazon elastic compute cloud (Amazon EC2). 2009. http://aws.amazon.com/ec2/

[4] Amazon Simple Storage Service (S3). 2009. http://www.amazon.com/s3/.

[5] Dean J, Ghemawat S. MapReduce: Simplified Data Processing on Large Clusters. In OSDI '04: Sixth Symposium on Operating System Design and Implementation, 2004

[6] Yang HC, Dasdan A, Hsiao RL, Parker DS. Map-Reduce-Merge: Simplified relational data processing on large clusters. In: Proc. of the 2007 ACM SIGMOD Int' Conf. on Management ofData. New York: ACM Press,2007. 1029-1040.

[7] P. Mell and T. Grance, "Draft nist working definition of cloud computing - v15," 21. Aug 2009, 2009

[8] Chen Kang, Zheng Weimin Cloud Computing: System Instances and Current Research[J] Journal of Software, Vol.20, No.5, May 2009, pp.1337−1348.

[9] "NoSQL Relational Database Management System: Home Page". Strozzi.it. 2007-10-02. Retrieved 2010-03-29

[10] DeCandia G, Hastorun D, Jampani M, Kakulapati G, Lakshman A, Pilchin A, Sivasubramanian S, Vosshnll P, Vogels W. Dynamo: Amazon' s highly available key−value store. In: Proc. of the 21st ACM Symp. on Operating Systems Principles. New York: ACM Press, 2007. 205-220.

[11] Eelco Plugge, Tim Hawkins, Peter Membrey The Definitive Guide to MongoDB: The NoSQL Database for Cloud and Desktop Computing Publisher Apress Berkely, CA, USA, 2010

[12] Wang D, Xie J. An approach toward web caching and pre-fetching for database management system. Paris, France: SIGMOD, 2001: 204-216.