ISSN : 0974 - 7435 Volume 10 Issue 22

2014

BioTechnology

An Indian Journal

FULL PAPER

BTAIJ, 10(22), 2014 [13798-13803]

Research on multidimensional indexing based on cloud computing system

Xinquan Yang[1,2]

[1]**School of Computer and Information Engineering, Heze University, Heze, 274015, (CHINA)**

[2]**Institute of Embedded System and Internet of Things, Heze University. Heze, 274015, (CHINA)**

ABSTRACT

Rapid development of cloud computing technology realizes the storage and management of mass data, but changes of data model make the indexing technology of traditional database management system cannot process data directly, which means research of indexing technology in cloud environment is needed to provide efficient support for retrieval tasks of mass data in cloud environment. With the poor research of multidimensional indexing based on cloud computing system, this paper studies the multidimensional indexing in cloud computing system, introduces it, describes the concept of multidimensional indexing and introduces the generally used multidimensional data processing method. Then it researches multidimensional indexing based on cloud computing system. This paper includes brief introduction of distributed storage system and something related to consistency Hash and the process of multidimensional data indexing drops to one-dimensional indexing, proposes multidimensional system of M-Index for the problem of current distributed storage system data indexing doesn't support more than one program, phases in pyramid technique, proceeds dimensionality reduction on two dimensional space, so as to offer reference for mass data storage indexing in cloud environment.

KEYWORDS

Mass data; Multidimensional indexing; Distributed storage; Pyramid technique.

INTRODUCTION

With the development of network and internet, traditional data management technology cannot satisfies the users' needs of mass data management any more, while cloud computing can solve this problem well which provides a big impetus to the development of mass data management[1]. As a new network technology, cloud computing attracts many users because of its huge storage and low cost[2]. Users can share and store resources anytime and anywhere through cloud service on internet. As the important representation of cloud computing, servitization of computing and resources can provide mass data management, computing and application deployment to users[3]. There are two indexing techniques in cloud environment: unidimensional data indexing and multidimensional data indexing. This paper focuses on multidimensional indexing.

MULTIDIMENSIONAL INDEXING IN CLOUD ENVIRONMENT

In cloud environment, mass data processing program will be divided into n small subprograms automatically through net, mass data of users will be stored in server by net distribution and be processed by systematic search and computing analysis consisting of many servers, and the results will be returned to users. In cloud computing environment, subset of distributed data is stored in network nodes, its search strategy is deployed in need-satisfying characteristic for users to rapidly search data by corresponding column. Many data queries can be realized in distributed network, including setting up relationship, one-dimensional query, neighbor query, range query and point location query.

Multidimensional indexing means regarding all the data attributes information of products in cloud computing as multidimensional data, such as the size of e-book, publishing time and price, transforming these multidimensional attributes into multidimensional data, building a multidimensional indexing based on these multidimensional data in order to proceed multidimensional indexing and range queries at the same time rather than unidimensional query in different dimensions. Figure 1 shows the most used multidimensional data processing method.

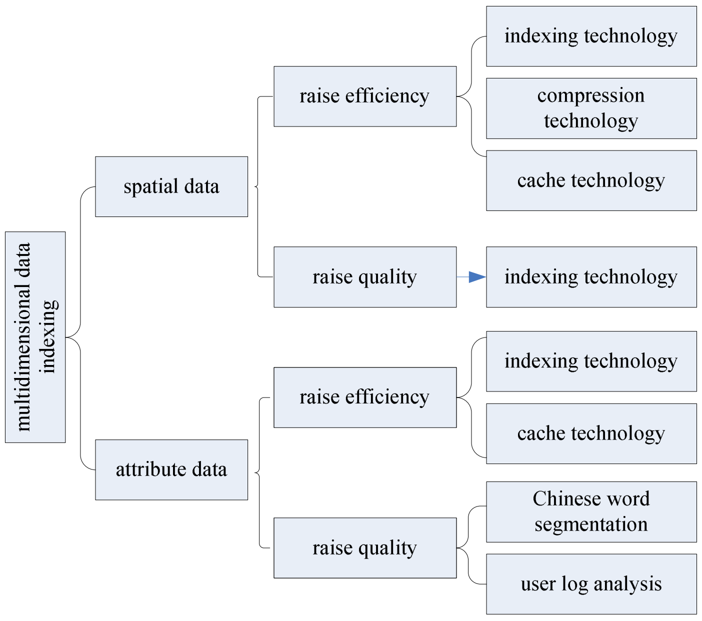

Figure 1 : Multidimensional data processing method

The major issue of the establishment of multidimensional data indexing structure in cloud environment is to ensure load balance between space and host process. When proceeding centralized routing inquiry, carrying out inserts, updates and deletes, it can ensure low consumption of mutual information. Usually, cloud computing system expects data keeps smooth distribution in network so as to ensure mostly the same spatial load of each host. The main form of multidimensional data indexing structure is to map multidimensional space to one-dimensional space, sort the multidimensional spatial objects by one-dimensional value, divide indexing space into lots of little pieces and code every model by z-transform.

MULTIDIMENSIONAL INDEXING IN CLOUD COMPUTING ENVIRONMENT

Relevant work introduction

Distributed storage system

GFS is a distributed file system developed by Google, its operation mode is to divide data files into pieces and store them distributively. It supports parallel files reading[4]. On the basis of GFS, Google developed Bigtable, a distributed storage system with similar functions of distributed database. Bigtable is mainly used to store structural data or semi-structural data and provide data storage and query service for Google search engine and Google map[5].

Dynamo developed by Amazon is also a representative distributed storage system with the feature that users can adjust the scope of data center dynamically according to work loan[6] and it uses P2P structure in the system to reduce query time.

Cassandra is a structural data storage system integrating Dynamo and Bigtable, it is used broadly. Data storage platforms of some applications like Facebook and Digg are all Cassandra[7].

PLUTS, a distributed database system developed by YaHoo, can process massive distributed data.

Consistency hash

MIT proposes consistency Hash algorithm for distributed hash table, realizing the real application of DHT in P2P environment[8]. Many algorithms can realize Hash, including Chord, CAN and Pastry, this paper focuses on Hash algorithm with data center storage node built with Chord. Identifiers of each node and key word in Chord system are M bit binary string acquired through hash operation of node IP and keyword which sorts all nodes clockwise by the size of identifiers to form an annular topological structure. Figure 2 shows the Chord network when m=4.

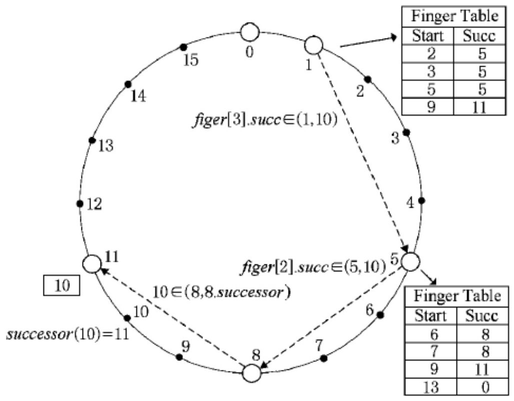

Figure 2 : Chord network when m=4

Multidimensional data indexing

Brief introduction of multidimensional data indexing mechanism which can execute complex query, including introduction of topological and indexing mechanism of overlay network in data design center and definition of multidimensional data indexing built with pyramid technology.

Overlay network topology

In data center based on shared nothing structure, storage device of every node is independent. Figure 3 shows overlay network topology, Figure 4 divides $N_i (0 \le i < n)$ into two kinds of nodes: storage nodes SN_i and indexing nodes IN_i. SN_i is used to store data allocated to N_i, and IN_i is used to manage metadata of data in SN_i. Data center will build an annular overlay network of IN_i according to Chord protocol. At the same time, there are two data indexes for maintenance on each IN_i, namely multidimensional index M-Index and local data index L-Index.

In overlay network topology shown in Figure 3, main task of M-Index is to maintain route information transmitted in overlay network, while L-Index is in charge of managing metadata information so as to avoid failure of single point and save storage space.

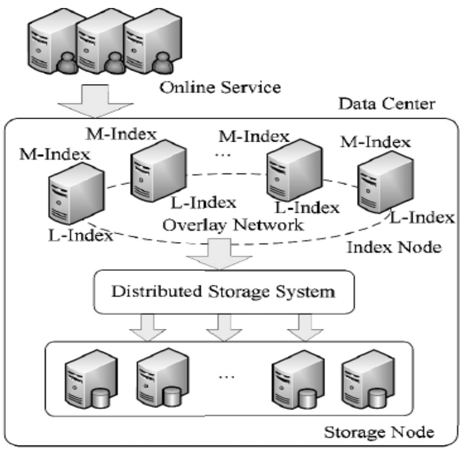

Figure 3 : Overlay network topology

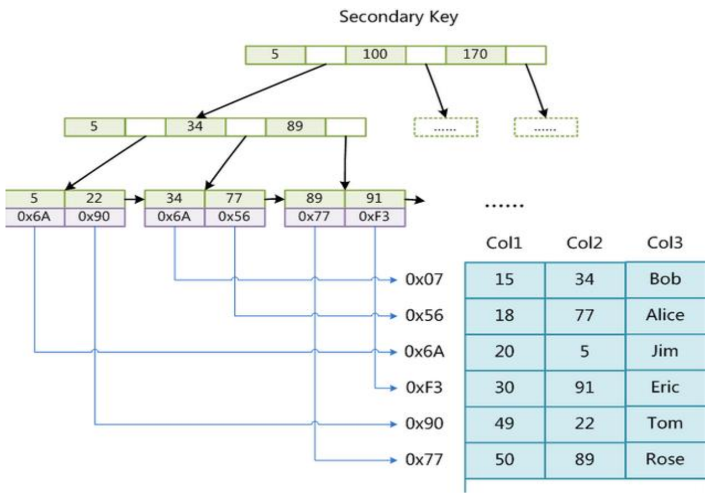

Figure 4 : Application of B-tree

Indexing mechanism

Data stored in cloud computing environment are mostly multidimensional data, for example, the metadata of an audio file has such information as memory, duration and compression ratio. This paper plans data in a unified way by abstracting it to a binary key-value pair, namely metadata attribute and attribute value represented as:

$$D =< a_j, v_j > (0 \le \mathrm{j} < \mathrm{d}) \qquad \textbf{(1)}$$

Primary key of data in consistency Hash only contains a certain metadata, making distributed storage system cannot carry out multidimensional query. Establishment of multidimensional data indexing by pyramid technology in this research is targeted to M-Index by reducing dimensionality of d-dimensional vector $M\left(v_0,\ldots,v_{d-1}\right)$ into one-dimensional index id with pyramid technology, while M-Index ensures no loss of data to keep data integration and avoid omission in data query.

M-Index sets point $P\left(0.5,\ \ldots,\ 0.5\right)$ as center point, then allocates d-dimensional space $\left[0,1\right]^d$ into 2d subspaces named pyramid with P on the top and d-1-dimensional space at the bottom. Coding rule of pyramid is: any one-dimensional vector $D_m\left(0 \le m < d\right)$ in d-dimension is perpendicular to the bottom of two pyramids, when m-dimensional coordinate $vm < 0.5$ in one pyramid, this pyramid is coded as m pyramid, when $vm \ge 0.5$, this pyramid is coded as $m+d$ pyramid.

M-Index defines d-dimensional vector as point V in d-dimension, and one-dimensional index after dimension reduction as the distance between point and point P. Dimension reduction may produce the same distance between point and point P, generating redundance in query. To reduce redundance, different distance functions are required in different pyramids.

The process of M-Index reduces multidimensional metadata to one-dimensional index is as shown in below:

(1)Normalization. Giving that pyramid technology is targeted to $\left[0,1\right]^d$ dimension, M-Index should transform metadata first by transforming the values of v_j $(0 \le j < d)$ into the range of $\left[0,1\right]$ as shown in formula (2):

$$v_j^{(1)} = \frac{v_j - v_{j\min}}{v_{j\max} - v_{j\min}}(0 \le j < d) \qquad \textbf{(2)}$$

$v_{j\min}$ represents the minimum value of v_j and $v_{j\max}$ represents the maximum value, $v_j^{(1)}$ represents the normalized value of v_j for expression.

(2) Code the pyramid the decision point V belongs to. Giving that different distance functions set by M-Index in different pyramids have different expressions, computing code i of pyramid the decision point V belongs to before computing distance is required, as shown in formula (3):

$$i = \begin{cases} j_{\max}, if\,(\mathrm{v_{jmax}} < 0.5), \\ j_{\max} + \mathrm{d}, if\,(\mathrm{v_{jmax}} \ge 0.5), \end{cases} \qquad \textbf{(3)}$$

$$j_{\max} = (\mathrm{j} \mid (\forall\,\mathrm{k}, 0 \le (\mathrm{j},\mathrm{k}) < \mathrm{d}, \mathrm{j} \ne \mathrm{k} : \left|0.5 - v_j\right| \ge \left|0.5 - v_k\right|)) \qquad \textbf{(4)}$$

j_{max} represents the one-dimensional vector with the largest difference of coordinate between point V and point P.

(3) Calculation of the difference of coordinate between point V and point P in $i\left(i \le d\right)$ or $i-d\left(i>d\right)$ dimension is as shown in formula (5):

$$hv = \left|0.5 - v_{ior}(i-d)\bmod d\right| \quad \textbf{(5)}$$

(4) Calculation of one-dimensional index of point $v_i d_y$ is as shown in formula (6):

$$id_v = i + h_v \quad \textbf{(6)}$$

Taking two dimension as example, after above calculation, two dimension is divided into 4 pyramids with point $P\left(0.5, 0.5\right)$ as vertex as shown in Figure 5. Hypothetically, the size of audio data is 2MB, duration is 60s, written as: $D = \left\{s = 2MB, t = 60s\right\}$; value interval is [0,4MB], [0,240s]. According to the above formulas, D-dimensional index is reduced to one-dimensional index id_y, $id_y = 1.25$.

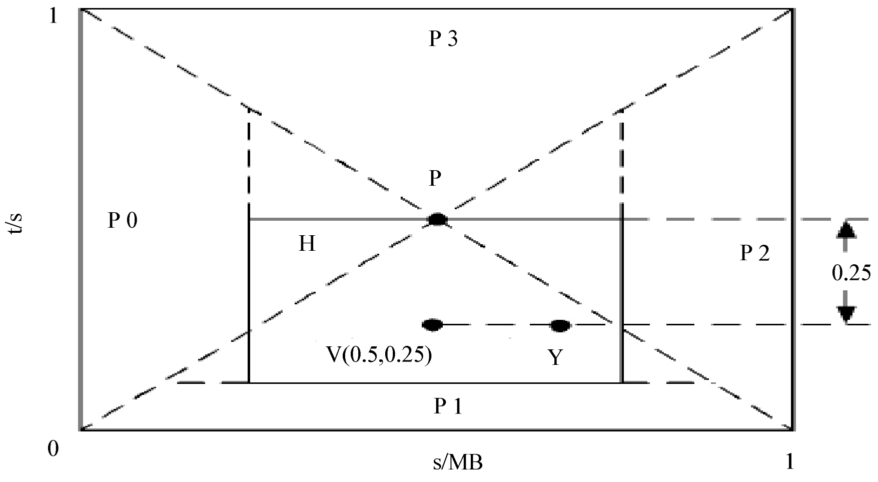

Figure 5 : Pyramid of two-dimensional space

Introduction of consistency Hash may break the order of values within one-dimensional indexing range $\left[id_{min},\ id_{max}\right]$, indexing system with primary key of id_y cannot realize support to interval query. Therefore, divide this interval $\left[id_{min},\ id_{max}\right]$ into n+1 subintervals, transmit interval query into a single value query according to the intersection of query interval and subinterval in data query while keeping load balance of data distribution, select common prefix as primary key to adjust the size of interval dynamically, use tree structure to manage common prefix of subinterval. For example, prefix binary-tree in Figure 6 is a prefix binary-tree with depth of 5 which needs further research.

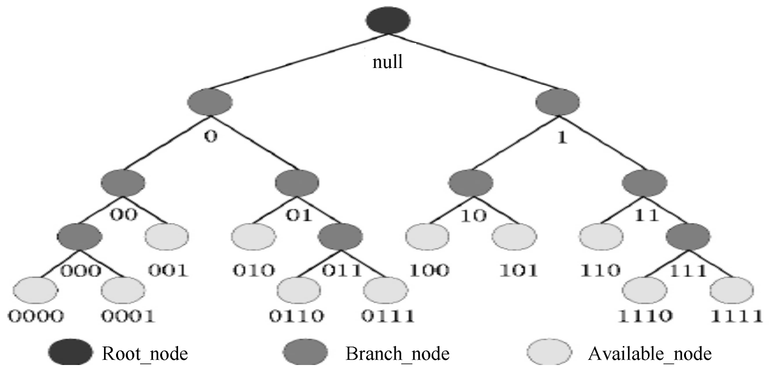

Figure 6 : Prefix binary-tree with depth of 5

CONCLUSION

Storage of mass data in cloud environment brings difficult to data indexing. With the development of cloud computing technology, multidimensional indexing technology in cloud environment is also improved. To tackle the problem that data indexing of distributed storage system in cloud environment doesn't support complex indexing with several programs, this paper proposes multidimensional indexing mechanism M-Index and introduces pyramid technology to reduce multidimensional data to one-dimensional indexing.

REFERENCE

[1] Chen Quan, Deng Qianni; Cloud computing and its key techniques [J], Journal of Computer Applications, **29(9)**, 56-57 **(2009)**.

[2] Zhang Jian; Analysis of concept and influence of cloud computing [J], Telecommunications Technology, **01**, 15-18 **(2009)**.

[3] P.Mell, T.Grance; The NIST definition of cloud computing, SP800-145[R], Gaithersburg: National Institute of Strabdards and Technology, **(2011)**.

[4] S.Ghemawat, H.Gobioff, S.Leung; The google file system[C]//Proc of the 19th ACM symp on operating systems principles, New YORK:ACM, 29-43 **(2003)**.

[5] F.Chang, J.Dean, S.Ghemawat et al.; Bigtable:A distributed storage ststem for structured data[J], ACM Trans on Computer syetems, **26(2)**, 1-26 **(2008)**.

[6] G.DeCandia, D.Hastorun, M.Jampani et ai.; Dynamo: Amazons highly available key-value store[J], //Proc of the 21st Acm Symp on Operating Systems Orinciples, New York:ACM, 205-220 **(2007)**.

[7] A.Lakshman, P.Malik; Cassandra: A decentralized structured storage system[J], ACM SIGOPS Pperating Systems Review, **44(2)**, 35-40 **(2010)**.

[8] D.Karger, E.Lehman, T.Leighton et al.; Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the World Wide Web[C]//Proc of 29th Annual ACM Symp on Theory of Computing, New York:ACM, 654-663 **(1997)**.