INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATIONS AND ROBOTICS
ISSN 2320-7345

EFFICIENT STORAGE AND RETRIEVAL OF LARGE DATA SET

M.Vennila[1], V.R.Revathy Priya[2], M.Revathy[3], A.Priya [4]

[1]*B.E (CSE), ULTRA college of Engineering, Madurai, India*
[2]*B.E (CSE), ULTRA college of Engineering, Madurai, India*
[3]*B.E (CSE), ULTRA college of Engineering, Madurai, India*
[4]*Assistant Professor, ULTRA college of Engineering, Madurai, India*

Abstract: - In Data mining environment a large amount of data is been produced, that need to be analyzed and several patterns to be extracted to gain Knowledge. In the field of big data many challenges are faced in data mining. So, a proper architecture should be handled to gain knowledge on big data. However, the inter-company indulged in data sharing and unique challenges to a data management system including scalability & security. Data storage can be done in two ways such as scalable and elastic. While dealing with elastic data it needs to concentrate on many areas splitting up of data. In this proposed paper it includes a review necessary for handling such a large data set by fragmenting & defragmenting. It involves mainly on fetching up of data without any loss, also in the paper it enhances integration of data & Pay as you go model for efficient storage. It uses clustering of data in data set storage with K means algorithm. It is a reliable and efficient form of providing security which use cluster data.

Keywords: Elastic data, Map join reduce, Cloud Computing, K-Means, Data set, clustering

1. Introduction

In data mining the data to be mined varies from small data set to large data set i.e. big data [1]. Each and every network maintains its own site and selectively shares a portion of its business data with others. Each and every corporate network scouts for a right data sharing platform. Examples of such network can be freelancer in which job seekers apply for jobs. In this it hires freelancer [2]. The information of the seekers will be given and job availability should be shared.

From the existing process of sharing is done only between inter companies. It results in performance degradation where traditional data sharing is achieved by centralized data warehouse which extracts data from internal production systems [3]. First the network needs to improve the scalability to support various participants this system involves huge hardware/software investments and high maintenance cost. The network site should customize the access control to see which seekers can see which part of the data shared. In most of data mining flexibility is needed [4]. Data fusion is termed as binding together data from multiple sources to create new sight.

Time series Analysis measures and predicts one or more values at different variance of times. Visualization grows with complexity of data sets that needs better ways to display them in meaningful ways. Security and privacy issues will only grow as data get bigger [5]. Data Mining is the process of recovering patterns among many fields in the database. Big Data are the large amount of data being processed by the Data Mining environment. Browsing through a large data set would be difficult and time consuming, we have to follow certain procedures, a proper algorithm and method is needed to classify the data, find a suitable pattern among them. Due to Increase in the amount of data in the field of applications, environmental research and many others, it has become difficult to find, analyze patterns, associations within such large data. As a querying

process we use map reduce join technique [6]. This also provides robustness to data. In existing system it only focuses on the benefits of corporate networks. Faults may occur in existing system and hence we are using fault tolerance method to overcome the fault that occurs.

To form a corporate network, companies simply register their sites with the service providers that place instances in cloud and the export the data to those instances for sharing. Elastic data model adopts the pay-as-you-go business model popularized by cloud computing [7]. They pay for what they use in terms of node instance's storage capacity. The data which can be stored and retrieved between corporate networks gets transferred node to node. The main workload of corporate network is simple low-overhead queries. Such searching process involves typically only querying a very small number of launchers and can be processed in short time [8]. For this time consuming analytical tasks, we provide an interface for exporting data from node to Hadoop and allow user to analyze those data using Map reduce. The major contribution is that the system provides flexible and scalable solutions for corporate network applications.

2. Database Server Engine with Large Data Set

The various techniques used for storing a large data set are db server and db engine creation that makes multiple clients creation along with that. This implementation helps elastic data for flexible storage and retrieval of server engine data set along with storage and retrieval. Map join reduce will be applied for effective storage and retrieval in large data set [9]. They have fragmentation and clustering of data sets. These are the techniques used for storage and retrieval of large data set.

2.1 Db server and Db Engine Creation

Db server is created which manages all the db engine connected.The database will be stored in the db engine via db server.Db server registers the db engine [10]. Db server raise request to server for new db engine .Since it is elastic model need for db engine is non scalable.

2.2 Client Creation

Clients are created by registering them in the server.They will upload data in pay as you go model.They will retrieve their own data from the db server.

2.3 Elastic Data Implementation

The data store in the db engine will be in elastic model the data of the particular user will be stored in various places [11]. According to the space availability in the db engine the data of the user will be scattered and stored.When the data is retrieved then it will be integrated to provide as one file.

2.4 *Map Join And Reduce Implementation*

The data stored in db engines will be mapped by the db server.when the user query for his data then the mapping of data will take place and then the joining of data will be done.the n number of split ups will be integrated and reduced to one particular file [12].

3. Fragmentation Algorithm Usage

This algorithm is fragmentation of data set i.e.splitting up of data set and storing the datas in the available space. The fragmentation can be held both vertically or horizintally.The algorithm will improve data mining performance by dataware housing [13]. While storing the data it will check for the free space according to the need .If the available free space is sufficient for the storage of all the split up of a particular user then it will defragment the files in one place.Through this will improve performance for every particular period.
This can also be performed when the database is in idle state.The server can defragment all the db engines and like wise the individual db engine can also do [14]. While proceeding the split ups of data set the server will search for the free space for the datas to be stored it will map the data & search for the space .While performing the process of fragmentation & defragmentation the system performance will long last. Performance upgradation will take place. & scalability may increase. By implementing fragmentation process crashing of data or hanging of data while usage may get decreased.

There are several reasons that make the cloud computing a contender for traditional computing techniques. Reasons to be listed:unlimited scalability.There are more cloud resources . If a customer desires more resources, he/she can rent those capabilities and resources will be available to the customer almost instantly. Speed of deployment. Offering of full-fledged services by cloud providers can reduce deployment time compared to in-house deployment. Elasticity in which cloud computing a pay-per-use payment model is generally applied, meaning that you only pay for the resources you actually use. This model ensures that startup costs and costs due to over-provisioning are avoided, without the risk of missing service levels due to under provisioning [15].

Reliability. Theoretically, a cloud provider can achieve high reliability. This can be achieved not only by taking one copy of all but also by having for example multiple data centers (allowing for handling for example power outages). However, there have already been significant service outages in cloud computing, making this debatable [16]. Reduced costs. Thanks to economies of scale (things tend to get cheaper when scale increases) at the cloud provider, costs can be reduced, potentially allowing customers to reduce costs as well. Elasticity allows for reduced costs when usage is low, as the customer uses less resources and therefore pays less. Because certain expertise can be centralized at the cloud provider the customer does no longer have to have this experts while allowing the customer to potentially save money.

Now that the fields of both data warehousing and cloud computing have been surveyed, we will speculate about the combination to see whether or not there are promising possibilities for data storingwarehousing in the cloud [17]. We will do so by analyzing the following forementioned arguments against moving towards the cloud:its the slow speed of moving data towards the cloud, poor performance in the cloud, loss of control and costs issues. We will also specifically analyze the current possibilities for enabling elasticity. Compute nodes or between compute nodes in a fast. Funtionality: It also provides high performance relevant to the applications needd and the space needed. Data needs to move from persistent cloud storage towards compute nodes or between compute nodes in a fast way. Moving significant amounts of data (up to terabytes) can be needed when new nodes become active (in order to deal with a changing workloads for example) as well as during query processing.

This is a challenge because storage and network bandwidth in the cloud are generally not ast compared to traditional data warehousing systems. A possible (partial) solution is to use compression to reduce bandwidth usage while paying the price of higher CPU usage. The capabilities of virtual machines will have to be exploited by data warehousing systems in the cloud, resulting in low-level technical issues. While traditional data warehousing systems generally do not make use of virtual machines, data warehousing systems in the cloud will likely have to because VM's are generally the platform offered by cloud providers. Virtual machine optimizations are also discussed [18].

4. **Securing DB Server with K-Means Algorithm**

Flexibility :In order to achieve elasticity data warehousing systems in the cloud will have to be able to scale up and down automatically once workloads, amounts of users or data volumes increase or decrease. Partitioning can be use in order to distribute data across different nodes in the cloud, comparable to the situation in traditional (cluster) data warehousing systems [19]. Each node can receive a number of these partitions (or 'shards').In traditional system data warehousing partitioning is been alloted.. In a cloud a much more dynamic system is required.

Load balancing is required in order to be able to fairly spread workloads over nodes in the cloud storage. It can brought up many upcoming issues regarding the re-partitioning, re-replication and/or re distributing of data based on changing usage patterns. Privacy. Data mining systems in the cloud must be able to encrypt data locally to assure privacy. The are possible operations for data encryption where the data must be a valuable addition. It leads to discuss issues that encrypts requirement of different analysis of data cryptography.

Security. Data warehousing systems in the cloud must be secured by the data warehouse which includs all kind of communication to the data warehouse are accessible only to the original customer. Security threats have many kind of storage capabilities. Monitoring. Monitoring capabilities have to be available for customers and administrators in order to analyze the systems operations. Potential bottlenecks (e.g. 'killer-queries') must also be detected and cancelled when required. Monitoring capabilities are important in achieving for example elasticity and load balancing. It changes different patterns of detecting the ability to detect the decision where scaling is requiredwhich does not need any load balancing techniques.

Though fragmentation and defragmentation process is adhered some security problems may occur. To protect the next overcover we use k-means clustering algorithm. The clustering is an exploratory task of data mining. Most of the researches on privacy preservation in clustering are developed for k-means clustering. By, using this

algorithm it may provide bounteous security to the data set. This Algorithm is mostly used for statis tical data analysis. K-means clustering is used when different sites contain different attributes for different common entities. Every cluster of entity in have learning for attributes towards every sites for entity.

4.1 Building TheBlocks of Security

The processing servers then privately collaborate this may held without running the actual data to run the K-means algorithm over the secret shares. In order to achieve thie security we use chinese remainder theorem(CRT). In our setting of problem, we ask each of the collaborating users to compute the secret shares of their private data, and send them over to the processing. Shatter Function φ(x) - Compute and store the secret shares of the private data : is defined as the one that splits the data x into R parts, x1, x2, ..., xR, such that each share,xi, by itself does not reveal any information about x. The participating users pre-decidea set uses prime data set that will have different scale factor.

5. Conclusion

In this proposed approach we define data storage and data retrieval in large set of data sets in several kinds of patterns. To handle knowledge about splitting of data with fragmentation technique and store it in cluster form of database that deals with elastic data. They concentrate on enhancement of data fetching and integration of data in a clustered format using K- means algorithm. This algorithm of security blocks process the secret shares of private data that achieves security for actual data that runs using shatter function. Thus the data here will be stored by fragmenting the data and there it uses K means algorithm and the data cluster will be formed using them. Thus by applying this we achieve high security for large data sets.

REFERENCES

[1] K. Aberer, A. Datta, and M. Hauswirth, “Route Maintenance Overheads in DHT Overlays,” in 6th Workshop Distrib. Data Struct., 2004.
[2] A. Abouzeid, K. Bajda-Pawlikowski, D.J. Abadi, A. Rasin, and A. Silberschatz, “HadoopDB: An Architectural Hybrid of MapRe-duce and DBMS Technologies for Analytical Workloads,” Proc. VLDB Endowment, vol. 2, no. 1, pp. 922-933, 2009.
[3] C. Batini, M. Lenzerini, and S. Navathe, “A Comparative Analysis of Methodologies for Database Schema Integration,” ACM Com-puting Surveys, vol. 18, no. 4, pp. 323-364, 1986.
[4] D. Bermbach and S. Tai, “Eventual Consistency: How Soon is Eventual? An Evaluation of Amazon s3’s Consistency Behavior,” in Proc. 6th Workshop Middleware Serv. Oriented Comput. (MW4SOC ’11), pp. 1:1-1:6, NY, USA, 2011.
[5] B. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, and R. Sears, “Benchmarking Cloud Serving Systems with YCSB,” Proc. First ACM Symp. Cloud Computing, pp. 143-154, 2010.
[6] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels, “Dynamo: Amazon’s Highly Available Key-Value Store,” Proc. 21st ACM SIGOPS Symp. Operating Systems Princi-ples (SOSP ’07), pp. 205-220, 2007.
[7] J. Dittrich, J. Quian_e-Ruiz, A. Jindal, Y. Kargin, V. Setty, and J. Schad, “Hadoop++: Making a Yellow Elephant Run Like a Chee-tah (without it Even Noticing),” Proc. VLDB Endowment, vol. 3, no. 1/2, pp. 515-529, 2010.
[8] H. Garcia-Molina and W.J. Labio, “Efficient Snapshot Differential Algorithms for Data Warehousing,” technical report, Stanford Univ., 1996. Google Inc., “Cloud Computing-What is its Potential Value for Your Company?” White Paper, 2010.
[9] R. Huebsch, J.M. Hellerstein, N. Lanham, B.T. Loo, S. Shenker, and I. Stoica, “Querying the Internet with PIER,” Proc. 29th Int’l Conf. Very Large Data Bases, pp. 321-332, 2003.
[10] H.V. Jagadish, B.C. Ooi, K.-L. Tan, Q.H. Vu, and R. Zhang, “Speeding up Search in Peer-to-Peer Networks with a Multi-Way Tree Structure,” Proc. ACM SIGMOD Int’l Conf. Management of Data, 2006.
[11] H.V. Jagadish, B.C. Ooi, K.-L. Tan, C. Yu, and R. Zhang, “iDistance: An Adaptive B+-Tree Based Indexing Method for Nearest Neighbor Search,” ACM Trans. Database Systems, vol. 30, pp. 364-397, June 2005.
[12] H.V. Jagadish, B.C. Ooi, and Q.H. Vu, “BATON: A Balanced Tree Structure for Peer-to-Peer Networks,” Proc. 31st Int’l Conf. Very Large Data Bases (VLDB ’05), pp. 661-672, 2005.
[13] A. Lakshman and P. Malik, “Cassandra: Structured Storage Sys-tem on a P2P Network,” Proc. 28th ACM Symp. Principles of Distrib-uted Computing (PODC ’09), p. 5, 2009.

[14] W.S. Ng, B.C. Ooi, K.-L. Tan, and A. Zhou, "PeerDB: A P2P-Based System for Distributed Data Sharing," Proc. 19th Int'l Conf. Data Eng., pp. 633-644, 2003.
[15] Oracle Inc., "Achieving the Cloud Computing Vision," White Paper, 2010.
[16] V. Poosala and Y.E. Ioannidis, "Selectivity Estimation without the Attribute Value Independence Assumption," Proc. 23rd Int'l Conf. Very Large Data Bases (VLDB '97), pp. 486-495, 1997.
[17] M.O. Rabin, "Fingerprinting by Random Polynomials," Technical Report TR-15-81, Harvard Aiken Computational Laboratory, 1981.
[18] E. Rahm and P. Bernstein, "A Survey of Approaches to Automatic Schema Matching," The VLDB J., vol. 10, no. 4, pp. 334-350, 2001.
[19] P. Rodr_ıguez-Gianolli, M. Garzetti, L. Jiang, A. Kementsietsidis, I. Kiringa, M. Masud, R.J. Miller, and J. Mylopoulos, "Data Sharing in the Hyperion Peer Database System," Proc. Int'l Conf. Very Large Data Bases, pp. 1291-1294, 2005.

M.Revathy, currently studying B.E. computer science and engineering in ultra college of Engineering and Technology for women at Madurai.

V.R.Revathy priya, currently studying B.E. computer science and engineering in ultra college of Engineering and Technology for women at Madurai.

M.Vennila, currently studying B.E. computer science and engineering in ultra college of Engineering and Technology for women at Madurai.

A.Priya received her bachelor's degree from Bharathidasan University (2004), Madurai, and then did her Master Degree in computer science and engineering from Anna University, Trichy. She is currently working as an Assistant Professor in Ultra College of Engineering & Technology for Women, Madurai. She has 6 years of experience as lecturer and 5 years in IT field.