International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611

Analyzing Big Data Problem Using Hadoop and Cloud Technology

Hemlata S. Urade

Assistant Professor, Department of Computer Science and Engineering, Rashtrasant Tukdoji Maharaj University, Nagpur, Maharashtra, India

Abstract: ***In any kind of industry sector networks they used to share collaboration information which facilitates common interests based information sharing. As this method decreases costs and increases incomes thus by sharing and processing data management challenges they develop their performance and security. We have developed a data intensive technique which is a method of service sharing in corporate networks through cloud based peer to peer data management platform. Already existing method Data Integration in Bigdata (DIB) integrates database management system, cloud computing and peer to peer technologies and found a flexible and scalable data sharing service network applications. There are many different areas need to be included and concentrated the split up of data to be dealt whenever user includes new set of data. As it have several split ups data should be properly fetched without any loss of information. Corporate network eliminates less efficient hadoop tool thus reduced total intercompany costs. In our proposed system Data Integration in Bigdata (DIB) is used for integrating data and enhances the model pay for efficient storage. Robustness of data, performance upgrade for size of data increase for prolonged storage use. The benchmarking results show that our method outperforms HadoopDB, a recently proposed large-scale data processing system, in performance when both systems are employed to handle typical corporate network workloads. The benchmarking results also demonstrate that our method achieves near linear scalability for throughput with respect to the number of peer nodes***

Keywords: Peer-to-peer systems, cloud computing, Hadoop, Big Data, Data Management, query processing, index

1. Introduction

We are living in an age when an explosive amount of data is being generated every day. Data from sensors, mobile devices, social networking websites, scientific data & enterprises – all are contributing to this huge explosion in data. This sudden bombardment can be grasped by the fact that we have created a vast volume of data in the last two years. Big Data- as these large chunks of data is generally called- has become one of the hottest research trends today. Research suggests that tapping the potential of this data can benefit businesses, scientific disciplines and the public sector –contributing to their economic gains as well as development in every sphere. The need is to develop efficient systems that can exploit this potential to the maximum, keeping in mind the current challenges associated with its analysis, structure, scale, timeliness and privacy. There has been a shift in the architecture of data-processing systems today, from the centralized architecture to the distributed architecture. Enterprises face the challenge of processing these huge chunks of data, and have found that none of the existing centralized architectures can efficiently handle this huge volume of data. These are thus utilizing distributed architectures to harness this data.

The main aim of this project is that the companies with same sector should be able to share data within each other securely and efficiently.

The main focus is to add data from different companies (peers) at cloud and efficiently ,securely retrieve the data from cloud and share that with different companies. as the user(peers) grow there should be no effect in sharing the data in cloud.

Previously data sharing is achieved by building a centralized data warehouse, which periodically extracts data from the internal production systems (e.g., ERP) of each company for subsequent querying. Unfortunately, such warehousing solutions are very difficult to build and need more cost.

What we will do in this project

1) We propose a cloud in which different companies (peers) will store data .we will connect in companies network on P2P basis.
2) This cloud will be web based so it will be available any time any were online.
3) Companies need to login into the cloud system to upload there data.
4) After that one key will be send to registered user mail id. For each new user who will do registration on cloud new key will be generated and send to his registered email id.
5) When user upload data on cloud he has to provide the key which was send on his mail, this is done for security purpose so that different companies can upload there data on same cloud securely.
6) Only the registered user having the key will be able to upload data, since they need to provide key before uploading the data.
7) Then while uploading the data ,data will first encrypted and store it on server for scalability
8) The uploaded data of different companies will be shown on the cloud.
9) The companies who want to share the data with different companies, have to provide email id of the companies to whom data need to share.
10) When the email id of the companies is provided, then one key will be send to the mail id of the companies to whom data need to be share.
11) Only after the company provides the key, the data will be decrypted and share with the company.

12) This is done for security in cloud, such that only authorized companies (user) can share data within itself. and other user in cloud cannot access their data.
13) We can also limit the company to access data from cloud by providing specific date, such that within that data range companies can access that data.
14) For encryption and decryption of data, we will provide hybrid cryptography (which will be combination of existing two cryptography techniques), previously only one cryptography technique was used to encrypt and decrypt the data. due to this security will increase such that no one can hack the data in cloud.
15) Previously in cloud only one database was used to store data and if that database was down whole cloud stops working and no data was accessed by user.
16) In this project we will replicate data from one database to various database in cloud, such that if one database is down other database will be available and user can share data from that database. so uninterrupted service will available .such that workload will be managed efficiently.

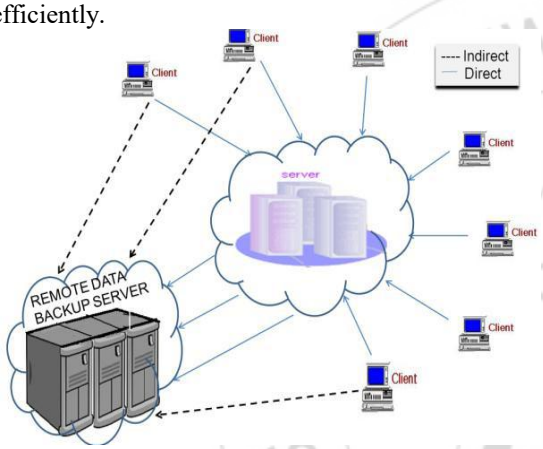


Figure 1: Showing Big Data with P2P system

MapReduce Programming Model MapReduce is a software framework proposed by Google, which is a basis computational model of current cloud computing platform. Its main function is to handle massive amounts of data. Because of its simplicity, MapReduce can effectivelydeal with machine failures and easily expand the number of system nodes. MapReduce provides a distributed approach to process massive data distributed on a large -scale computer clusters. The input data is stored in the distributed file system (HDFS), MapReduce adopts a divide and conquer method to evenly divided the inputted large data sets into small data sets, and then processed on different node, which has achieved parallelism. In the MapReduce programming model, data is seen as a series of keyvalue pairs like, as shown in Figure 1, the workflow of MapReduce consists of three phases: Map, Shuffle, and Reduce. Users simply write map and reduce functions.

In the Map phase, a map task corresponds to a node in the cluster, as the other word, multiple map tasks are be running in parallel at the same time in a cluster. Each map call is given a key-value pair (k1,v1) and produces a list of (k2,v2) pairs. The output of the map calls is transferred to the reduce nodes (shuffle phase). All the intermediate records with the same intermediate key (k2) are sent to the same reducer node. At each reduce node, the received intermediate records are sorted and grouped (all the intermediate records with the same key form a single group). Each group is processed in a single reduce call. The data processing [4-6] can be summarized as follows:

Map (k1, v1) $-\rightarrow$ list(k2, v2)
Reduce (k2, list(v2)) $-\rightarrow$ list(k3, v3).

2. Literature Survey

Distributed Data Mining in Peer-to-Peer Networks (P2P) [1] offers an overview of the distributed data mining applications and algorithms for peer-to-peer environments. It describes both exact and approximate distributed data-mining algorithms that work in a decentralized manner. It illustrates these approaches for the problem of computing and monitoring clusters in the data residing at the different nodes of a peer-to-peer network.

This paper focuses on an emerging branch of distributed data mining called peer-to-peer data mining. It also offers a sample of exact and approximate P2P algorithms for clustering in such distributed environments.

Architecture for data mining in distributed environments [2] describes system architecture for scalable and portable distributed data mining applications. This approach presents a document metaphor called \emph{Living Documents} for accessing and searching for digital documents in modern distributed information systems. The paper describes a corpus linguistic analysis of large text corpora based on collocations with the aim of extracting semantic relations from unstructured text.

Distributed Data Mining of Large Classifier Ensembles [3] presents a new classifier combination strategy that scales up efficiently and achieves both high predictive accuracy and tractability of problems with high complexity. It induces a global model by learning from the averages of the local classifiers output. The effective combination of large number of classifiers is achieved this way.

Map-Reduce for Machine Learning on Multi core [4] discuss the ways to develop a broadly applicable parallel programming paradigm that is applicable to different learning algorithms. By taking advantage of the summation form in a map-reduce framework, this paper tries to parallelize a wide range of machine learning algorithms and achieve a significant speedup on a dual processor cores.

The core of Data Integration in Bigdata (DIB) with query processing and P2P overlay includes platform independency Bootstrap and normal peer structured software component executes on top of cloud structure [14] [15]. The data flow and individual components launch and maintains its service provider with single strap normal peer instance. The entry point of all networks is bootstrap peer has several responsibilities for various administration. By scheduling different administration purpose they monitor and manages normal peer. For corporate network applications the metadata

of central warehouse bootstrap shares global schema, participates in normal peer list and defines roles of data.
They have certificate authority certifies the normal peer for their identities [16] [17]. They use encryption scheme for data transmission between normal peers to increase security which employs data encryption and decryption [18]. Data Integration in Bigdata (DIB) implies normal peers as best instances. Data retrieval requests by users manage and serve business owned by normal peers. Normal peer holds centralized server for locating high throughput requirements.
Query processing inculcates balanced tree peer to peer overlay in distributed manner [19].

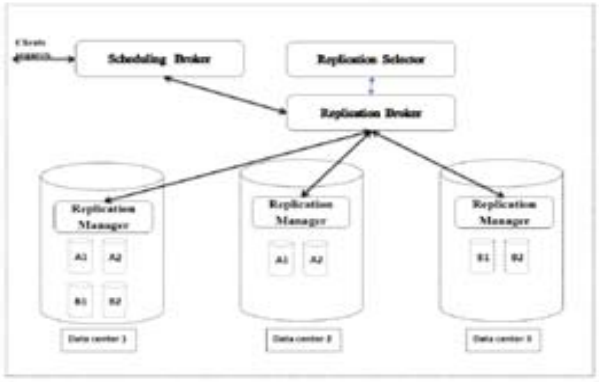
3. Proposed Methodologies

We propose an adaptive replication strategy in a cloud environment that adaptively copes with the following issues:

- What to replicate to improve the non-functional QoS. The select process is mainly depends on analyzing the history of the data requests using a lightweight time-series prediction algorithm. Using the predicted data request, we can identify what data files need replication to improve the system reliability.
- The number of replicas for each selected data.
- The position of the new replicas on the available data centers.
- The overhead of replication strategy on the Cloud infrastructure. This is the most important factor of the proposed adaptive replication strategy where the Cloud has a large number of data centers as well as a large-scale data.
- Hence, the adaptive replication strategy should be lightweight strategy.

The proposed adaptive replication strategy is originally motivated by the fact that the recently most accessed data files will be accessed again in the near future according to the collected prediction statistics of the files access pattern. A replication factor is calculated based on a data block and the availability of each existing replica passes a predetermined threshold, the replication operation will be triggered. A new replica will be created on a new node which achieves a better new replication factor. The number of new replicas will be determined adaptively based on enhancing the availability of each file heuristically. However, we employ a lightweight time-series algorithm for predicting the future requests of data files. The replication decision is primarily based on the provided predictions. The heuristic proposed for the dynamic replication strategy is computationally cheap, and can handle large scale resources and data in a reasonable time.

4. Architecture Of project



The Remote backup services should cover the following issues:
1) Privacy and ownership.
2) Relocation of servers to the cloud.
3) Data security.
4) Reliability.
5) Cost effectiveness.
6) Appropriate Timing.

1) Privacy and ownership
Different clients access the cloud with their different login or after any authentication process. They are freely allowed to upload their private and essential data on the cloud. Hence, the privacy and ownership of data should be maintained; Owner of the data should only be able to access his private data and perform read, write or any other operation. Remote Server must maintain this Privacy and ownership.

2) Relocation of Server
For data recovery there must be relocation of server to the cloud. The Relocation of server means to transfer main server's data to another server; however the new of location is unknown to the client. The clients get the data in same way as before without any intimation of relocation of main server, such that it provides the location transparency of relocated server to the clients and other third party while data is been shifted to remote server.

3) Data Security
The client's data is stored at central repository with complete protection. Such a security should be followed in its remote repository as well. In remote repository, the data should be fully protected such that no access and harm can be made to the remote cloud's data either intentionally or unintentionally by third party or any other client.

4) Reliability
The remote cloud must possess the reliability characteristics. Because in cloud computing the main cloud stores the complete data and each client is dependent on the main cloud for each and every little amount of data; therefore the cloud and remote backup cloud must play a trustworthy role. That means, both the server must be able to provide the data to the client immediately whenever they required either from main cloud or remote server.

5) Cost effectiveness
The cost for implementation of remote server and its recovery & back-up technique also play an important role while creating the structure for main cloud and its correspondent remote cloud. The cost for establishing the remote setup and for implementing its technique must be minimum such that small business can afford such system and large business can spend minimum cost as possible.

6) Appropriate Timing
The process of data recovery takes some time for retrieval of data from remote repository as this remote repository is far away from the main cloud and its clients. Therefore, the time taken for such a retrieval must be minimum as possible such that the client can get the data as soon as possible without concerning the fact that remote repository is how far away from the client.

5. Conclusion & Future Scope

We propose a cloud computing architecture based on P2P which provide a pure distributed data storage environment without any central entity for controlling the whole processing. The advantage of this is architecture is that it prevents the bottleneck problem that arises in most of the client server communications The proposed system does its operation based on the performance of the system. It does the monitoring operation to find out the best chunk servers within the P2P network. It does this operation in order to perform efficient resource utilization and load balancing of the servers.

When disaster occurred then all companies faced big los s of data and also financial then after many recovery mechanisms are introduced. As cloud nomenclature has a PaaS, IaaS, and SaaS as services which provide their service to cloud users in terms of infrastructure, software and platform as their requirement; so user can use cloud without any difficulty. By implementing DRaaS in cloud one can get recovered from data loss when he experiences a system failure or by natural disasters. So by implementing DRaaS in business continuity they can overcome their data loss.

A new method for managing access control based on the cloud server's internal clock. Our technique does not rely on the cloud to reliably propagate re-encryption commands to all servers to ensure access control correctness. We showed that our solutions remain secure without perfect clock synchronization so long as we can bound the time difference between the servers and the data owner.

In future in for Big Data processing in place of Hadoop we can use Apache Spark. Spark is 100% faster than Hadoop since it works on memory rather than Disk. Also Spark contains more in Build APIS which we need to separately install for Hadoop.

References

[1] K. Aberer, A. Datta, and M. Hauswirth, "Route Maintenance Overheads in DHT Overlays," in 6th Workshop Distrib. Data Struct., 2004.
[2] Francesco Maria Aymerich, Gianni Fenu, Simone Surcis. An Approach to a Cloud Computing Network. 978-424426249/08/$25.00 ©2008 IEEE conference.
[3] Boss G, Malladi P, Quan D, Legregni L, Hall H. Cloud computing. IBM White Paper, 2007.
[4] Ghemawat S, Gobioff H, Leung ST. The Google file system. In: Proc. of the 19th ACM Symp. On Operating Systems Principles. New York: ACM Press, 2003. 29_43.
[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 Principles (SOSP '07), pp. 205-220, 2007.
[7] J. Dittrich, J. Quiane-Ruiz, A. Jindal, Y. Kargin, V. Setty, and J. Schad, "Hadoop++: Making a Yellow Elephant Run Like a Cheetah (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.
[9] Google Inc., "Cloud Computing-What is its Potential Value for Your Company?" White Paper, 2010.
[10] 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.
[11] 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.
[12] 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.
[13] 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.
[14] A. Lakshman and P. Malik, "Cassandra: Structured Storage System on a P2P Network," Proc. 28th ACM Symp. Principles of Distributed Computing (PODC '09), p. 5, 2009.
[15] 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.
[16] Oracle Inc., "Achieving the Cloud Computing Vision," White Paper, 2010.

[17] 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.
[18] M.O. Rabin, "Fingerprinting by Random Polynomials," Technical Report TR-15-81, Harvard Aiken Computational Laboratory, 1981.
[19] E. Rahm and P. Bernstein, "A Survey of Approaches to Automatic Schema Matching," The VLDB J., vol. 10, no. 4, pp. 334-350, 2001.

Author Profile

Hemlata Urade received the B.E and M.tech . degrees Computer Science and Engineering from Rajiv Gandhi College of Engineering and Research Technology in 2010 and 2012, respectively. Her thesis work is on Evolutionary algorithms where she can put idea for problem solving method using multiobjective optimization. Her Research area is Evolutionary algorithm, Big data, Operating System etc. At presently she is working as assistant professor in computer Science and Engineering department.