INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

EFFICIENT DATA PROCESSING IN PEER NETWORK USING CLOUD COMPUTING

SHILPA VASANTRAO PARALKAR[1], GAYATRI KABRA[2]

1. ME Scholar, Department of Computer Science and Engineering, KSIET, Balsond, Hingoli, SRTMU, Nanded.
2. Assistant Professor, Department of Computer Science and Engineering, KSIET, Balsond, Hingoli , SRTMU, Nanded.

Accepted Date: 05/03/2015; Published Date: 01/05/2015

Abstract: In same industry sector networks they used to share collaboration information which facilitates common interests based information sharing. As this method reduces costs and increases throughput thus by effectively managing data processing and sharing challenges they build their performance, income and security. Data intensive technique with p2p is a type of service sharing in corporate networks through peer to peer data processing platform using cloud. Already existing method integrates cloud computing, peer to peer technologies and database management system, and found an economical flexible and scalable data sharing services related to network applications. There are many different areas need to be included and concentrated the data is spilt up to be dealt whenever user includes current set of data. As it have without any loss of information, several split ups data should be properly fetched. Elimination of less efficient hadoop tool in corporate network Reduces total intercompany costs. In our proposed system efficient data processing in peer network is used for integrating data and enhances the model pay for efficient storage. Robustness of data, upgrading performance for size of data increase for prolonged storage use.

Keywords: Cloud Computing, Efficient Data

Corresponding Author: MS. SHILPA VASANTRAO PARALKAR

Access Online On:

www.ijpret.com

How to Cite This Article:

Shilpa Vasantrao Paralkar, IJPRET, 2015; Volume 3 (9): 1220-1229

PAPER-QR CODE

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.

What we will do in this project

1) We propose a cloud called data intensive technique with p2p, 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 provide 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 databases 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

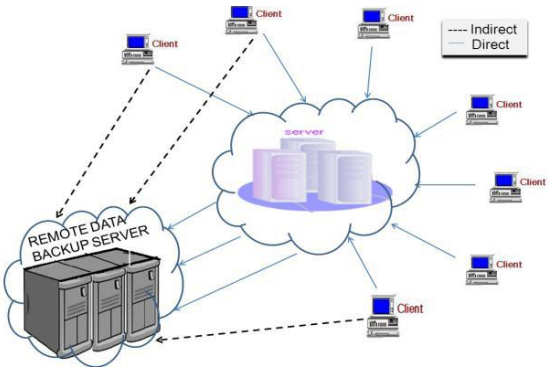

Fig 1: Showing cloud computing with P2P system.

2. RELATED LITERATURE

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 network. It describes both accurate and tentative result 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 analysis of large text corporate based on collections 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 complexity.

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 large data processing in peer network with query processing and P2P overlay includes platform independency. Bootstrap and normal peer structured software component executes on top of cloud structure [15] [16]. The data flow and individual components join and manages its service provider with single 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.

They have certificate authority certifies the normal peer for their identities [17] [18]. They use encryption and decryption scheme for data transmission between peer to peer which increase security [19]. Query processing inculcates balanced tree peer to peer overlay in distributed manner [20].

3. MAP REDUCE IMPLEMENTATION

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 data. MapReduce can effectively deal with machine failures and easily expand the number of system nodes since its simplicity. 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 subdivided into data subsets, and then processed on different node, which has achieved parallelism. In the MapReduce programming model, data is seen as a series of key value pairs like , as shown in Figure 1, the workflow of MapReduce consists of 3 phases: 1)Map 2)Shuffle, and 3)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) −→ list (k2, v2)

4. ELASTIC DATA EXERTION

With database engine by connecting Database server is created and manages. Data is stored in database engine by database server which registers in DB engine. DB server raise request to

server for new database engine. For elastic model non-scalable dB engine is needed. When clients are created by registering the server they upload data in pay as you go model.

In cloud computing services Elastic data is the data sharing services for delivering pay as you go query processing. Data storage purpose elastic data capable for flexible data model and easily manages the data model. In Distributed database storage join and leave nodes for elasticity. It is a difficult task in relational database. Previous data model are often inelastic for relational databases. In concern with data store modifications every row and columns are in different numbers.

In this type more elastic Data store. As DB engine is elastic model data of individual user will be stored in various places in database engine. According to the availability of data storage space in the database engine the information entered and store in the database engine in a distributed manner in the available space. From the database system for retrieving a data is it will be integrated into one and provided as one file to the user. In elastic data, query processing the controllable transaction throughput.

5. ADAPTIVE REPLICATION STRATEGY

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- quality of service. 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.

Architecture Of project

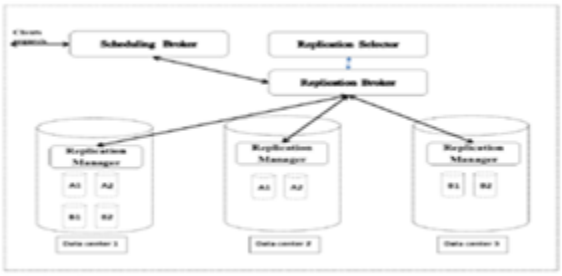

Fig 2: Architecture of project

Remote Data Backup server is a server which stores the main cloud's entire data as a whole and located at remote place (far away from cloud). And if the central repository lost its data, then it uses the information from the remote repository. The purpose is to help clients to collect information from remote repository either if network connectivity is not available or the main cloud is unable to provide the data to the clients. As shown in Fig 1, if clients found that data is not available on central repository, then clients are allowed to access the files from remote repository (i.e. indirectly).

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.

6. CONCLUSION

In same kind of industry sector their collaborative information is shared according to its common interests shared. Such data integrations reduce the costs and improvise the throughput source with cost effective feasible measures for efficient data processing services. Such data is implemented using efficient data processing in peer network technique which helps in reducing workloads which integrates peer to peer technology, database management system and cloud computing which involves platform as a service application. 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 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 .Elastic data sharing service is used for efficient query processing along with peer to peer services in cloud service

7. 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, "I Distance: 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.

20. 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.