DIB: DATA INTEGRATION IN BIGDATA FOR EFFICIENT QUERY PROCESSING

P.Divya, K.Priya

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. Bestpeer++ 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.**

Index Terms — **BestPeer++, cloud computing platform, peer to peer, database management system, Data Integration in Bigdata (DIB), storage efficiency.**

I. INTRODUCTION

Various corporate companies conserve business data in their corporate network database and manage their own website providing strong security over distributed data. Organizations to attain its marketing solutions and strategies in a supply network all the other retailers, manufacturers and suppliers have good association with each other and sustain their integrity [1]. Internal productive systems shares data with centralized data warehouse of data sharing platform from various corporate companies. From these warehouse data is extracted from companies using subsequent querying technique. Thus they entail wide database system.

Usually companies determine their policies strictly will full customization and choose their partners and allot their share and rights for duties. Such access controls will have less flexibility in data warehouse [2] [3]. Occasionally for maximizing their company income they dynamically alter their business process, partners and their rights etc. Data warehouse technique precisely not designed for handling such dynamic process and participants.

For improving efficiency in query processing Data Integration in Bigdata (DIB) integrates cloud computing concept peer to peer technology and database warehouse[4] [5]. With distributed networks access control cloud computing based web interface used for controlling policies. P2P technology used for retrieving data in efficient manner in a network overlay. Query processing implied in data warehouse technique makes corporate networks low over head queries. Database can be indexed in table and have wide data range retrieval.

For scalable and economical solutions for corporate network applications we need to benchmark these applications with hadoop and analyze their data using map reduce technique [6].

In data service of elastic data the ultimate goal is to imply enterprise application towards peer to peer systems. Basically peer to peer concerts unstructured network and retrieval of information from matching columns in various tables [7]. For retrieving information mapping is used for giving queries which improvises the result quality and enhancing performance in corporate network applications. Network overlay tree structure and partial indexing scheme provides effective distribution of search services.

Bestpeer++ enhances its quality with distributed access control, various indexing techniques, query processing services in cloud data. Core and adapter is the different

***P.Divya**, M.E. CSE Krishnasamy College of Engineering and Technology, ,Affiliated to Anna University (Chennai), S.Kumarapuram, Cuddalore, Tamil Nadu – 607109, India.*

***K.Priya,** Assistant Professor , Krishnasamy College of Engineering and Technology, ,Affiliated to Anna University (Chennai), S.Kumarapuram, Cuddalore, Tamil Nadu – 607109, India.*

ISSN: 2278 – 1323

software components used in Bestpeer++. Core structure is a platform independent design shares data functionalities. Adapter have concrete elastic interface contains specific cloud services.

Corporate network applications provides platform for Bestpeer++ which delivers data sharing sevices with peer to peer data management platform. The total cost cut down for partnership in related companies and this also eliminates hadoop which is less efficient tool [8]. They all focuses only on corporate networks benefits and don't find solutions with data loss with integration of files. They need effective fault tolerance mechanism which when retrieving integrated files affects with great data loss.

Proceeding with prolonged use of data storage the size of data increases with performance degradation due to extended functionality and operations dealt with data retrieval and storage updates.

In this paper we impart Data Integration in Bigdata (DIB) which is used for efficient integration of data. This enhances efficient storage and enhances income source which provides data robustness. In data warehouse the important factor that affects the whole network is insecure data retrieval or storage [9]. As they increase their retrieval and data storage for extended time period they have to face performance degradation which leads to increase in data and makes some faults in performance.

As this algorithm handles with multiple applications at a time it does not meant for single application [10]. They provide long lasting performance even for prolonged usage of data and maintains best ever performance. It comes with long lasting usage of data with effective and sustained performance upgrade [11].

Thus in the following contents of the paper discuss with the advantage of other corporate secured networks over Data Integration in Bigdata (DIB). They explain how to incorporate peer to peer system along with data warehouse and maintain their platform integrity with cloud computing platform as a service moreover it discusses the pros and cons of the network performance [12]. Here we propose a new algorithm which maintains data integrity and provide improvised performance of network inculcating all the existing functionalities [13].

II. Related Work

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].

There are different other algorithms used for analysis and implementation of data warehouse potential gains. The ways to detect for warehouse concepts includes: applications altered from top of the source to warehouse. With relevant modifications systems log file is parsed. By comparing current source with the earlier one they detect different problems and infer result for those probabilities.

In a balanced overlay of peer to peer tree structure they support range and exact queries efficiently. The load at every load is equal in different levels of nodes between distinctions in tree structure. They also provide enough fault tolerance mechanism which permits sideways routing tables for repair efficiently. They mainly contributes on consequences that supports match and range queries which efficiently supports balanced tree structure in peer to peer overlay network. In other peer to peer systems they take n log steps to find joining or departure node for updating routing tables. Flexible load balancing in either adjacent or far away node schemes are provided. Between any pair of nodes a tree can provide adequate fault tolerance mechanism which efficiently recovers any kind of malfunctioning in a balanced tree structure.

III. Elastic Data exertion in DIB

Database server is created and manages connected with database engine. Database is stored in database engine via database server which already registers in db engine [20]. For new database engine db server raise request to server. Db engine is non scalable which is needed for elastic model. When clients are created by registering the server they upload data in pay as you go model. From the database server they retrieve their own data.

Elastic data is the data sharing services in cloud computing services for delivering pay as you go query processing. Elastic data for data storage capable for flexible data model and clustering tolerates and manages the data model easily.

Distributed database storage eases adding and removing nodes for clustering elasticity. It is a difficult and delicate task even for expert in relational database. Predefined data model are more often inelastic for relational databases. In concern with data store modifications every row and columns are in different numbers. Data model may change

ISSN: 2278 – 1323

over time even though this type data stores more elastic data in the schema.

Usually the data to be stored in the DB engine will be of elastic model. Thus the data of the particular 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 will be entered in a distributed manner and stored in the available space [21]. When the data is retrieved from the database system it will be integrated into one and will be provided to the user as one file.

On platform as a service application of cloud concept reliable and scalable storage of elastic data accustom dynamically. The scalability of different elastic storage system varies with system through put with different levels of consistency. Read consistency for system throughput scales linearly. Then the system response time demonstrates effective scaling property of load handling system storage nodes. This requires better consistency for query requesting which incurs high latency.

When dealing with elastic data in query processing the transaction throughput concurrency commits controllable transactions. Number of queries served by the node denominates system load distribution to measure load of a node. A high work load level helps for additional replicas and sent the overloaded primary copy. Ratio between heaviest loaded node and the lightest loaded node is divided by the maximum load imbalance. Various replication Data factors determine the access patterns effect for rating. In multi version concurrency scheme data conflicts update transactions with restart probability. Load adaptive replication and multi versioning optimistic concurrency control. Effective balance for skewed workload systems proposes load adaptive replication method.

IV. MAP JOIN AND REDUCE IMPLEMENTATION IN DIB

In the DB server data stored in DB engines will be mapped in the database server. When a user provides query for his desired data then the mapping of data will be undergone and then the joining of data will be done. This is known as map join to be done with database server. There are N times of split ups will be done by integrated and reduced to one particular file.

Large data sets needs joining of data by key which is not essential. Correlating the events by joining the timestamps for gaining insight with joining data together describes map reduce effectively. Map process sends a particular tuple which attributes map key values whose values are hashed as identifier of reduce process [22]. Optimizing the shares with a fixed number of reduce process. Map is given an algorithm for detection and fixing problems done mistakenly. To implement joins map reduce method joined with small dimension tables which joins analytic queries. In a web and social network queries involves graph paths.

Google map reduce for powerful application tool is a kind of open source function with Hadoop. Key value for data stored in one or more files allows map reduce in map function. Master controller allows instantiation of map function to operate at once with routed produced pairs of several reduce processes [23]. So the combination of other values associated with another function for reduce processes results for a key value.

V. DATA INTEGRATION IN BIGDATA (DIB) IMPLEMENTATION

In lieu with other data sharing services in the same manner as elastic data sharing services or comparatively the other method like map, reduce and join of data sharing techniques they are somewhat reliable and scalable with data. System throughput for above method is scaled linearly and accustoms dynamically [24]. The workload level comprises of workload distribution denominates concurrency with number of queries with controllable transactions.

In the concept of data warehousing Data Integration In Bigdata (DIB) helps improving performance of data mining technique. Before storing the data the algorithm first checks for free space availability according to the need of the query user. If in case the available free space is sufficient to accommodate all the split ups of particular user in a storage system then it will be defragmented into different files in one place. Even when the database is in idle state this algorithm implementation is effectively monitored.

For every particular time interval period their performance improves with implementing Data Integration in Bigdata (DIB). When a server is defragmented with all database engines likewise the individual database engine can also dealt with.

By introducing this new algorithm we can have more added advantage over the current corporate infused networks along with cloud services of platform as a service.

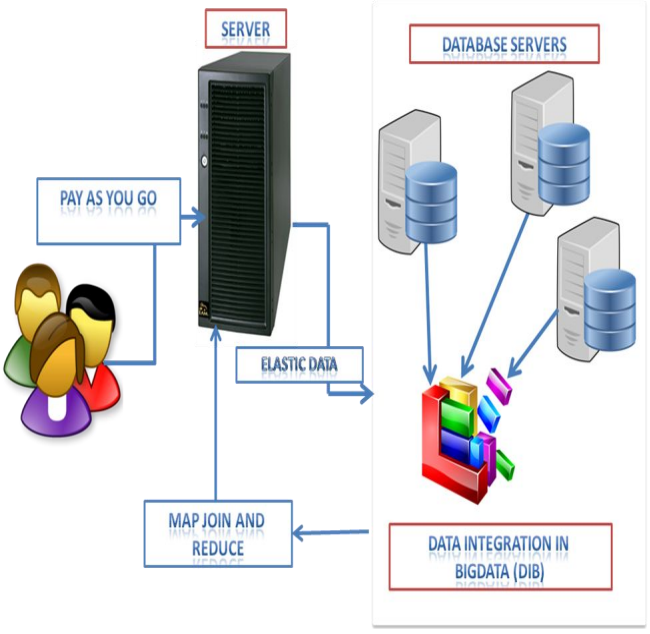

Fig.1 Implementation of Data Integration in Bigdata (DIB) in peer network.

ISSN: 2278 – 1323

It also maintains integrity within the peer networks and its access towards the server of database. The database keeps track of all data storage and retrieval in the split ups by maintaining separate database. Whenever a data is required the applied technique will defragment all the data and have number of split ups for different set of data. Thus from the Fig.1 we can clearly have the structure for how we inculcate pay as you go model in the server which sends elastic data integrity with database servers. Db engine will defragment data integration with bigdata from database servers. Even map join reduce along with DIB from DB server to main server. It gives more data integrity and quick query processing with data storage and retrieval with defragmented data.

Distributed parallel processing of data in huge amount from various database servers across the database engine rely on cost effective and maintain up to the mark standards for bigdata in which the hadoop concept of advanced database included. From structured as well as unstructured data format regardless of its original format can be efficiently handled by hadoop which can be stored in clusters.

As hadoop renders schema which is very expensive and it undergoes performance degradation according to the increase in database overflow. In DIB model we can have many split ups which can retrieve data in quicker and easier way [25] [26]. The main advantage over this model implementation is improvised performance and data integration done in very fast manner. Data storage and retrieval keeps on increase performance in query interaction with analysis in organization.

VI. CONCLUSION

In industry sector their collaborative information is shared according to its common interests shared. Such data integrations cut down the costs and improvise the income source with cost effective feasible measures for data sharing services. Such data is implemented using Data Integration in Bigdata (DIB) 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. Data Integration in Bigdata (DIB) involves data warehousing in query processing which is applied with cloud network service. Elastic data sharing service is used for efficient data services according to its query processing along with peer to peer services in cloud service. Elastic data controls concurrency control with optimistic multi versioning of load adaptive replications. Even Data Integration in Bigdata (DIB) involves map, join and reduce for join and reduce data which will be defragmented and integrated to one particular file effectively. In this point Data Integration in Bigdata (DIB) is inculcated by improving its performance of data warehousing. As it first checks for the free space availability according to the need of the query they can accommodate split up areas sufficiently. In the proposed model as we infer pay as you go from database the flexibility achieves through elastic data. All these are defragmented into different files in one particular place even when the database is in idle state.

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_iguez-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.
[20] Saepio Technologies Inc., "The Enterprise Marketing Manage-ment Strategy Guide," White Paper, 2010.
[21] I. Tatarinov, Z.G. Ives, J. Madhavan, A.Y. Halevy, D. Suciu, N.N. Dalvi, X. Dong, Y. Kadiyska, G. Miklau, and P. Mork, "The Piazza Peer Data Management Project," SIGMOD Record, vol. 32, no. 3, 47-52, 2003.
[22] A. Thusoo, J. Sarma, N. Jain, Z. Shao, P. Chakka, S. Anthony, H. Liu, P. Wyckoff, and R. Murthy, "HIVE: A Warehousing Solution over a Map-Reduce Framework," Proc. VLDB Endowment, vol. 2, no. 2, pp. 1626-1629, 2009.
[23] H.T. Vo, C. Chen, and B.C. Ooi, "Towards Elastic Transactional Cloud Storage with Range Query Support," Proc. VLDB Endow-ment, vol. 3, no. 1, pp. 506-517, 2010.

ISSN: 2278 – 1323

[24] S. Wu, S. Jiang, B.C. Ooi, and K.-L. Tan, "Distributed Online Aggregation," Proc. VLDB Endowment, vol. 2, no. 1, pp. 443-454, 2009.
[25] S. Wu, J. Li, B.C. Ooi, and K.-L. Tan, "Just-in-Time Query Retrieval over Partially Indexed Data on Structured P2P Overlays," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '08), 279-290, 2008.
[26] S. Wu, Q.H. Vu, J. Li, and K.-L. Tan, "Adaptive Multi-Join Query Processing in PDBMS," Proc. IEEE Int'l Conf. Data Eng. (ICDE '09), 1239-1242, 2009.

P.Divya , Completed her B.E. (CSE) degree in the year 2013.Currently she is pursuing M.E (CSE) at Krishnasamy College of Engineering & Technology, Cuddalore, Tamil Nadu, India. Her research areas are Image Processing, Cloud Computing, Data Mining and Big Data. She had attended many workshops also interested in attending seminars and conferences in various technologies.

K.Priya, Completed her B.E. degree from Madras University, M. Tech degree from SRM University Chennai. Currently she is working as a Assistant professor in Computer Science and Engineering at Krishnasamy College of Engineering & Technology, Cuddalore, Tamil Nadu, India. Her area of interest includes Image Processing, Network Security, and Compression Techniques. She had presented research papers in National/ International conferences.

ISSN: 2278 – 1323