International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org
Volume 4, Issue 4, July - August 2015 ISSN 2278-6856

A SURVEY: DATA SHARING APPROACH USING PARALLEL PROCESSING TECHNIQUES

[1]Suruchi Shrivastava , [2]S . Sathappan

[1]M. Tech Student , [2]Associate professor
LNCT Bhopal

ABSTRACT

A various techniques and algorithms were proposed in the area of data sharing techniques which is efficient, fast and reliable in the entire possible domain. Technology got much solution by increasing time and years ahead, the work which is going to determine in this and here we are going to research and survey various different techniques which were applied in data sharing effectively. Our paper investigate the latest solution for efficient data sharing is peer++ data sharing scheme which used best approaches among data mining, cloud computing area and provide a hybrid technique. Which is Best Peer++ technique which provides data sharing effectively and efficiently over the corporate networks? In this paper we survey about the various technique and try to differentiate the problems occurred in them so that we can get accurate and more enhanced solution in the data sharing scheme over different network and usability.

Keyword – Data sharing, cloud computing, query manipulation, data processing.

I.NTRODUCTION

Industry today often looking for a reliable infrastructure , which outperform store their data and compute it fast as compare to the current scenario where the demand is increasing regularly.Examples of such corporate networks include supply chain networks where organizations such as suppliers, product manufacturing companies, and retailers collaborate with each other to achieve their very own business goals including planning production-line, making acquisition strategies and different marketing production based companies who are strike in the market to sell their goods.From a technical perspective, the key for the success of a corporate network is choosing the right data sharing platform,

a system which enables the shared data (stored and maintained by different companies) network-wide visible and supports efficient analytical queries over those data. Traditionally, 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 a warehousing solution has some deficiencies in real deployment.

Peer to peer Network

Peer-to-Peer networks involve millions of machines connected in a network. It is a decentralized and distributed network architecture where the nodes in the networks (known as peers) serve as well as consume resources. It is one of the oldest distributed computing platforms in existence. Typically, Message Passing Interface (MPI) is the communication scheme used in such a setup to communicate and exchange the data between peers. Each node can store the data instances and the scale out is practically unlimited (can be millions of nodes). The major bottleneck in such a setup arises in the communication between different nodes. Broadcasting messages in a peer-to-peer network is cheaper but the aggregation of data/results is much more expensive. In addition, the messages are sent over the network in the form of a spanning tree with an arbitrary node as the root where the broadcasting is initiated. MPI, which is the standard software communication paradigm used in this network, has been in use for several years and is well-established and thoroughly debugged. One of the main features of MPI includes the state preserving process i.e., processes can live as long as the system runs and there is no need to read the same data again and again as in the case of other frameworks such as MapReduce (explained in section “Apache hadoop”). All the parameters can be preserved locally. Hence, unlike MapReduce, MPI is well suited for iterative processing. Another feature of MPI is the hierarchical master/slave paradigm. When MPI is deployed in the master–slave model, the slave machine can become the master for other processes. This can be extremely useful for dynamic resource allocation where the slaves have large amounts of data to process. MPI is available for many programming languages. It includes methods to send and receive messages and data. Some other methods available with MPI are ‘Broadcast’, which is used to broadcast the data or messages over all the nodes and ‘Barrier’, which is another method that can put a barrier and allows all the processes to synchronize and reach up to a certain point before proceeding further.

Although MPI appears to be perfect for developing algorithms for big data analytics, it has some major drawbacks. One of the primary drawbacks is the fault intolerance since MPI has no mechanism to handle faults. When used on top of peer-to-peer networks, which is a completely unreliable hardware, a single node failure can cause the entire system to shut down. Users have to implement some kind of fault tolerance mechanism within the program to avoid such unfortunate situations. With other frameworks such as Hadoop (that are robust to fault tolerance) becoming widely popular, MPI is not being widely used anymore.

II.LITERATURE REVIEW

Gang Chen, Tianlei Hu, Dawei Jiang, Peng Lu, Kian-Lee Tan, Hoang Tam Vo, and Sai Wu in Paper "Best Peer++: A Peer-to-Peer BasedLarge-Scale Data Processing Platform"

They have presented a scheme bestpeer++ which performance when both systems are employed to handle typical corporate network workloads. The benchmarking results also demonstrate that BestPeer++ achieves near linear scalability for throughput with respect to the number of peer nodes. According to them The total cost of ownership is therefore substantially reduced since companies do not have to buy any hardware/software in advance. Instead, they pay for what they use in terms of BestPeer++ instance's hours and storage capacity.

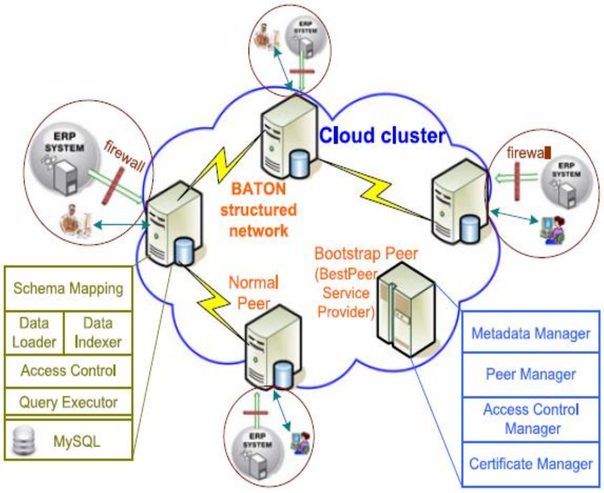

Figure-1 (Amazon network deployed using BestPeer++)

They stated that BestPeer++ system that provides economical, flexible and scalable solutions for corporate network applications. They show that for simple, low-overhead queries, the performance of BestPeer++ is significantly better than HadoopDB. They have shown the differentiate between Hadoop DB and BestPeer++ in which BestPeer++ Technique perform best than others. The benchmark conducted on Amazon EC2 cloud platform shows that our system can efficiently handle typical workloads in a corporate network and can deliver near linear query throughput as the number of normal peers grows. Therefore, BestPeer++ is a promising solution for efficient data sharing within corporate networks.

Azza Abouzeid1 , Kamil Bajda-Pawlikowski1 , Daniel Abadi1 , Avi Silberschatz1 , Alexander Rasin in paper "HadoopDB: An Architectural Hybrid of MapReduce and DBMS Technologies for Analytical Workloads"

The author demonstrated and discussed the heavy data sharing approach using Map Reduce concept of HadoopDB, where they explore the feasibility of building a hybrid system that takes the best features from both technologies; the prototype we built approaches parallel databases in performance and efficiency, yet still yields the scalability, fault tolerance, and flexibility of MapReduce-based systems. They have showed the superior performance of parallel databases relative to Hadoop. While this previous work focused only on performance in an ideal setting.they describe the design of a hybrid system that is designed to yield the advantages of both parallel databases and MapReduce. This system can also be used to allow single-node databases to run in a shared-nothing environment.

HadoopDB is therefore a hybrid of the parallel DBMS and Hadoop approaches to data analysis, achieving the performance and efficiency of parallel databases, yet still yielding the scalability, fault tolerance, and flexibility of MapReduce-based systems. The ability of HadoopDB to directly incorporate Hadoop and open source DBMS software (without code modification) makes HadoopDB particularly flexible and extensible for performing data analysis at the large scales expected of future workloads.

H. V. JAGADISH1 , Beng Chin OOI2,4 , Martin RINARD3,4 , and Quang Hieu VU "BATON: A Balanced Tree Structure for Peer-to-Peer Networks".

The author proposed a balanced tree structure overlay on a peer-to-peer network capable of supporting both exact queries and range queries efficiently. In spite of the tree structure causing distinctions to be made between nodes at different levels in the tree, we show that the load at each node is approximately equal. In spite of the tree structure providing precisely one path between any pair of nodes, we show that sideways routing tables maintained at each node provide sufficient fault tolerance to permit efficient repair. Specifically, in a network with N nodes, we guarantee that both exact queries and range queries can be answered in O(logN) steps and also that update operations (to both data and network) have an amortized cost of O(logN).

They obtain excellent fault tolerance, and also to get good load distribution without having to overload nodes near the root of the tree. they have shown how this tree structure can naturally be used to support an index structure for range queries.

Table- Comparison for three different Algorithm.

Serial	Author	Algorithm	Dis-Advantage
1	**Gang Chen, Tianlei Hu**	BestPeer++	Efficient classification required.
2	**Azza Abouzeid1 , Kamil Bajda-Pawlikowski 1**	HadoopDB	Slow performance.
3	**H. V. JAGADISH1 , Beng Chin OOI2**	Balance Tree	NIL

III.PROPOSED WORK CAN BE DONE

Upon discussing the work which are already done in the field we have analyzed the different approaches and further on we can extend the work of Best Peer++ in the tree structure based approach which is an efficient scheme for the classification area which provide the maximum output while we work with the classification technique, with the help of which we are going to implement and perform the result using few parameters such as performance, fault toleration , execution time.

IV.CONCLUSION

In this paper we have discussed various factors and procedure which is outperforms in the peer to peer work sharing approaches. Various ERP application to share data and information been used the different technique to get minimum processing time to process their data, here we investigate the approaches and their drawback with its current technique. Several details on each of these hardware platforms along with some of the popular software frameworks such as Hadoop and Spark are also provided. A thorough comparison between different platforms based on some of the important characteristics (such as scalability and real-time processing) has also been made through star based ratings. The technique and its iterative nature, compute-intensive calculations and aggregating local results in a parallel setting makes it an ideal choice to better understand the various big data platforms. It should be noted that many of the analytical algorithms share these characteristics as well. This article provides the readers with a comprehensive review of different platforms which can potentially aid them in making the right decisions in choosing the platforms based on their data/computational requirements. Further on we can work on improving the results with the same parameter we can use tree based approach to solve the existing problems

REFERENCES

[1] Gang Chen, Tianlei Hu, Dawei Jiang, Peng Lu, KianLee Tan, Hoang Tam Vo, and Sai Wu, “Extended BestPeer: A Peer-to-Peer Based Large-Scale Data Processing Platform”,VOL. 26,NO. 6, JUNE 2014.

[2] Azza Abouzeid1 , Kamil Bajda-Pawlikowski1 , Daniel Abadi1 , Avi Silberschatz1 , Alexander Rasin in paper “HadoopDB: An Architectural Hybrid of MapReduce and DBMS Technologies for Analytical Workloads.

[3] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield. Xen and the art of virtualization. In Proc. of SOSP, 2003.

[4] R. Chaiken, B. Jenkins, P.-A. Larson, B. Ramsey, D. Shakib, S. Weaver, and J. Zhou. Scope: Easy and efficient parallel processing of massive data sets. In Proc. of VLDB, 2008.

[5] Agneeswaran VS, Tonpay P, Tiwary J (2013) Paradigms for realizing machine learning algorithms. Big Data 1(4):207–214

[6] Zaharia M, Chowdhury M, Franklin MJ, Shenker S, Spark SI (2010) Cluster Computing with Working Sets. In: Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing., pp 10–10.

[7] Lee K-H, Lee Y-J, Choi H, Chung YD, Moon B (2012) Parallel data processing with MapReduce: a survey. ACM SIGMOD Record 40(4):11–20.

[8] H. V. JAGADISH1 , Beng Chin OOI2,4 , Martin RINARD3,4 , and Quang Hieu VU “BATON: A Balanced Tree Structure for Peer-to-Peer Networks”.