Design and Analysis of Large Data Processing Techniques

Madhavi Vaidya
Asst Professor
VES College, Mumbai
Affiliated to Univ of Mumbai

Shrinivas Deshpande, Ph.D
Associate Professor, HVPM
Affiliated To SGBAU, Amravati

Vilas Thakare, Ph.D
Head,
Dept of Computer Sc
SGBAU, Amravati

ABSTRACT

As massive data acquisition and storage becomes increasingly affordable, a large number of enterprises are employing statisticians to make the sophisticated data analysis. Particularly, information extraction is done when the data is unstructured or semi-structured in nature. There are emerging efforts taken by both academia and industry on pushing information extraction inside parallel DBMSs. This leads to solving an significant and important issue on what can be a better choice for large scale data processing and analytics. To address this issue, we highlight the comparison and analysis of the three techniques which are nothing but the Parallel DBMS, MapReduce and Bulk Synchronous Processing in this paper.

General Terms

Reliable, Fault Tolerance, Indexing

Keywords

Parallel, MapReduce, Hadoop ,BSP, Distributed

1. INTRODUCTION

DFS provides a scalable, consistentl storage solution which cost effective in nature. Distributed File systems (DFS) have been widely used by search engines to store the vast amount of data. This data can be collected from the Internet. Search engine companies are built parallel computing platforms on top of DFS. Large-scale data analysis is run in parallel on data stored in DFS.

The purpose of this paper is to consider the choice to make the analysis of the large data and the trade-offs between the options is studied. The author's idea is to make the comparative analysis of using MapReduce for the processing of large data generated from the various devices which are mobile in nature. Even how MapReduce is faster and efficient in making the analysis of this data generated is compared along with the databases used. The BSP is another technique which is adopted by Google adopting as a major technology for graph analytics at massive scale via MapReduce.

Today, new technologies made possible to realize value from Big Data generated. For example, behavioral trends that improve campaigns, pricing and stock can be tracked by stock using user's web clicks.

Utilities can capture household energy usage levels to predict outages and to intent more efficient energy consumption. Governments and even Google has stated detecting and tracking the emergence of disease outbreaks via social media signals. Oil and gas companies obtain the output of sensors in their drilling equipment to make more efficient and safer drilling decisions. [1] The phenomenal growth of internet based applications and web services in last decade have brought a change in the mindset of researchers. There is an improvement in the traditional technique to store and analyze voluminous data. The organizations are ready to acquire solutions which are highly reliable. [2]

Google proposed MapReduce, which is a programming model. It is an associated implementation for large-scale data processing in distributed cluster. Using an index of the web as documents requires continuously transforming a large repository of existing documents as recent documents arrive. Storage of data processing tasks can't be done easily by databases. Google's indexing system stores tens of petabytes of data and processes billions of updates per day on thousands of machines. In such cases, MapReduce plays an important role [3].

In most organizations, data is always growing, changing, and manipulated and therefore time to analyze data significantly increases. To process the large and diverse data sets, graph data structures can be processed by Hadoop and MapReduce[4]. The Map Reduce model is applied to large batch-oriented computation, which is connected primarily to job completion in proportion to time. The Map Reduce framework by Google and open-source Hadoop's system emphasize the usage through a batch-processing implementation strategy: the whole output of each map and reduce stage is materialized to stable storage before it can be consumed by the next stage. This materialization allows for a simple that is critical in large deployment, which have a high probability of slowdowns or failures at worker nodes[5]. MapReduce proposed by Google is a programming model and an associated implementation for large-scale data processing in distributed cluster.[6]

There are many more approaches for handling the big data. Making the analysis of all such approaches is the task ahead. Here, we argue that using Map Reduce systems we can perform tasks that are best suited in parallel to solve the computing problems which works on extract-transform-load (ETL) system. The technology press has been focusing on the revolution of cloud computing systems that requires harnessing large number of processors. [7]. There are some approaches studied in this paper for making the analysis of the big data. This paper is divided into the various sections such as for explaining the architecture of Hadoop Distributed File Systems, Parallel Database Management Systems, Bulk Synchronous Processing. Then in the later sections, the architectural comparison is done on the basis of some parameters. In the final section, the analysis and discussion is stated. Finally the conclusion is drawn on the basis of study of all the three approaches.

1.1 Architecture of Hadoop Distributed File System

HDFS is the file system component of Hadoop (Refer Figure 1). The UNIX file system has faithfulness to standards was sacrificed in favor of improved performance for the applications at hand, the interface to HDFS is patterned based on it.

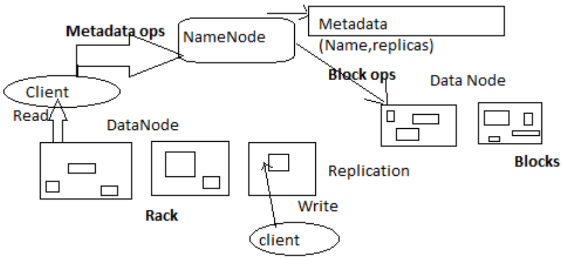


Figure 1: HDFS Architecture, Source: Apache Hadoop Website

One method of processing massive datasets, called MapReduce, has been drawing a great interest. This framework was developed and widely used by Google [8], while its open source implementation Hadoop [9] is currently used by more than 100 companies worldwide including eBay, IBM, Yahoo!, Facebook, and Twitter, with a lot of universities. [10]. Hadoop installation consists of a single master node and many Worker Nodes. The master is called as the Job Tracker, is responsible for accepting jobs from clients, dividing those jobs into tasks, and assigning those tasks are executed by worker nodes. Each worker node runs a Task Tracker process that manages the execution of the tasks currently assigned to that node[5]. Key-value pair forms the basic data structure in Map Reduce. The map function takes the input record and then generates intermediate key and value pairs. The reduce function takes an intermediate key and a set of values to generate a smaller set of values.

1.1.1 Architecture of Map Reduce Framework

MapReduce is simple and efficient for computing when the aggregation of data is done. It works on input-map-shuffle and sort. Finally reduce and then output phases are executed. The major advantages of it is,

- Simple and Easy to use – MapReduce model is simple but expressive. With MapReduce,a programmer defines job with only Map and Reduce functions. Here, physical distribution of the jobs across nodes is not specified.

- Flexible- MapReduce does not have any dependency on data model and schema. With MapReduce, a programmer can deal with unstructured data easily. This facility is not given by DBMS. MapReduce can work on petabytes of data on thousand of machines.

- Independent of the Storage – MapReduce is basically independent from underlying storage layers. It can work on Big Table and others [6]. Many projects at Google store data in Big Table which have different demands from Big Table, both in terms of data size (from URLs to web pages to satellite imagery) and latency requirements (from backend bulk processing to real-time data serving). Despite these varied demands, Big Table has successfully provided a flexible, high-performance solution for all of the following Google products like Google Earth, Google Finance.

- Fault Tolerance – MapReduce is highly fault tolerant, continues working in spite of failures per analysis job at Google[7]. The failed Map task can be repeated correctly by loading the replica.

2. ARCHITECTURE OF PARALLEL DATABASE MANAGEMENT SYSTEM

Database systems capable of running on clusters of shared nothing nodes are in existence since late 1980s. Those systems support standard relational tables and SQL. A parallel DBMS uses a cluster of nodes. It interconnects to compute expensive tasks in parallel in high speeds. In the horizontal partitioning, the tuples of a relational table are distributed across the nodes of the cluster. They support standard relational tables and SQL. There are two types of partitioning techniques, Hash and Round Robin Partitioning. In Hash Partitioning, a hash function is applied to every row of a relation and the values area distributed to the nodes. In Round-Robin Partitioning, that particular relation's records in blocks are distributed across the nodes of the cluster. There are many commercial implementations of parallel database management systems are available, including Teradata, Netezza, Vertica and DB2[11].

3. ARCHITECTURE OF BULK SYNCHRONOUS PROCESSING MODEL

The traditional von Neumann model is about the sequential algorithms has been in use, for a long time, as the main model for designing and reasoning. It has also served as a reference model for hardware design. In the context of parallel algorithm design, no such model exists. Valiant [12] introduced the BSP model to better reflect the hardware design features of mainstream parallel computers. The BSP model allows for efficient parallel algorithm design without any over-specification requiring the use of a large number of parameters. Each processor can communicate directly with every other processor, providing complete control over how the data is distributed between the processors in every round.
MapReduce and BSP handle parallel algorithms design in a coarse-grained fashion along with interleaving phases of computations and communication. Both can be used to design algorithms running on clusters of low-end systems connected with point-to-point communication, and both make use of synchronization between rounds.

Goodrich et al.[13] depicted a simulation for any BSP algorithm on their MapReduce model. The memory size M of each processor is limited to [N/P] , where N is total memory size of the BSP system and P is the number of BSP processors. The simulation on this basis is elaborated in [14], which works on the map and reduce phases.

4. ARCHITECTURAL COMPARISON

In this section, the aspects of the system architectures are considered for processing large amounts of data in a distributed environment. This comparison has been done on the basis of some parameters such as Fault tolerance, Flexibility, Data Distribution, processing of semi-structured data and sensor data analytics. MapReduce works on "filtering then group by aggregation" query processing if we compare it with a DBMS.

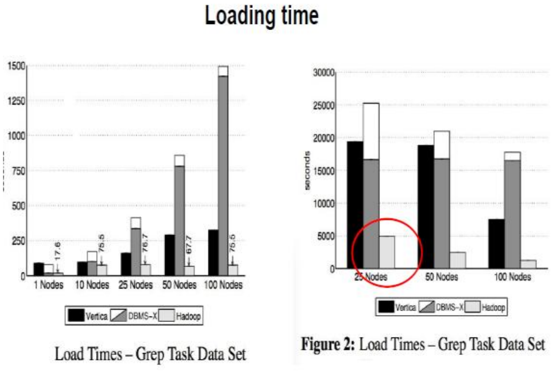
Fault Tolerance – MapReduce model is more proficient at handling node failures during the execution of MapReduce computation. It is highly fault tolerant. In a MapReduce system, if the data node fails then the MapReduce scheduler can automatically restart the task on alternate node. Here, there is a difference in the parallel DBMSs, which have transactions that are restarted in the event of the failure. DBMSs does not save the intermediate results. During a long running query, if a single node fails in a DBMS, the entire query must be completely restarted.

Flexibility – Though SQL is accepted at large but it is always criticized for its insufficient expressive ability. The programming environments allow developers to benefit from the robustness of DBMS technologies without the burden of writing complex SQL.

Proponents of MapReduce model argue that SQL does not facilitate the desired generality that MapReduce provides. MapReduce does not have any dependency on data model and schema. With MapReduce a programmer can deal with irregular data very easily than the programmers can do with DBMS. But almost all DBMS products provide support for user-defined functions, stored procedures and aggregate functions in SQL.

Data Distribution - In order to access larger data set, the user will have to load the data into the DBMS. Till the user plans to run many analysis, it is preferable to simply point the DBMS at data on the local disk without a load phase. As such there is no valid reason for DBMSs cannot deal with in case when bringing the computation is done to where data is located. The parallel database system can take up a longer time for loading of the data. Hadoop achieved load throughputs of up to three times faster than Vertica(DBMS) and almost 20 times faster than DBMS-X (Refer graph in 2) [11].

Although the whole task to load data into memory and refrain the execution of parallel DBMSs take much longer time than the MapReduce Systems.



Load Times – Grep Task Data Set

Figure 2: Load Times – Grep Task Data Set

Figure 2 : Trade-offs between Parallel Database and MapReduce[11].

Parallel DBMSs use the knowledge of data distribution and location to their advantage. Map Reduce process the data on-the-fly by loading it into a distributed file system. A MapReduce programmer has to execute the tasks manually. For processing a collections of documents in two parts; the Map function scans the documents, creates histogram of frequently occurring words and then passed to a Reduce function that groups files by their site of origin. In contrast, the SQL view and select queries perform a similar computation.

Processing of Semi-structured Data – MapReduce does not have any dependency on the data model and schema. With MapReduce, a programmer can deal with irregular or unstructured data more easily than they do with DBMS. Unlike a DBMS, a MapReduce systems do not require users to define a schema for their data.

There are versatile data sources which deserve to be taken into consideration which are processed by MapReduce. In addition to transactional databases, it is data from the web, be it blog contents or click streams which can help unveil valuable information, not to forget the content from social media which have evolved to the most commonly used communication platforms[15].

Sensor Data Analytics - When the data from the different sources is collected then it can be multi-media files like video, photo and audio, from which important conclusions for the business can be drawn. For example, on the roads, the data from the car's black boxes is collected if the vehicles met some accidents. There are huge text files including endless logs from IT systems, notes and e-mails which contain indicators that businesses are keen on. One more thing is very important to understand that the vast number of sensors built into smartphones, vehicles, buildings, robot systems, appliances, smart grids and whatever devices collecting data in a diversity which was unbelievable in the past. These sensors represent the basis for the ever evolving and frequently quoted Internet of things. All this data can be analyzed by the MapReduce. To address this issue, MapReduce has been used for large scale information extraction. [16,17]on the other hand, due to the rapid data volume increasing in recent years at customer sites, some data such as web logs, call details, sensor data and RFID data can not be managed by Teradata partially because of its expensiveness to load large volumes of data to a RDBMS[18].

In this section, the parameters elaborated on MapReduce are applicable to Bulk Synchronous Processing as this technique is implemented via MapReduce

5. ANALYSIS & DISCUSSION

Dean and Ghemawat[8] compared the two aspects, database and the MapReduce on the performance. First of all, they stated that for many of the tasks, the loading phase of the parallel DBMSs takes about five to fifty times the time needed to run the job in Hadoop. If the data is processed only once or twice, a Hadoop job would clearly end before the data is even loaded into the parallel DBMSs[8]. A specialty of MapReduce is to start computations over the data on-the-fly. The data has only to be copied in the distributed file system and the processing starts. The possibility of starting a task over a huge dataset without a long loading phase is one of the key features.

In many ways, MapReduce can be seen as a complement to an RDBMS. MapReduce is a good fit for problems that need to analyze the whole data set in a batch fashion, especially for an ad hoc analysis [19].

Attributes	**Traditional RDBMS**	**MapReduce**
Data Size	Gigabytes	Petabytes or Terabytes
Access	Interactive	Batch
Updates	Read and Write many times	Write Once, Read Many Times
Structure	Static Schema	Dynamic Schema
Integrity	High	Low
Scaling	NonLinear	Linear

Source : Hadoop: The Definitive Guide, RDBMS is compared to MapReduce [19]

Map Reduce proponents would always advocate that Map Reduce is best at executing the data which is on the fly i.e. dynamic in nature. Most database systems can not deal with in-situ data i.e. which is present in the file system. In short, there is an extensive research done on the management of massive sensor data generated by manufacturing devices which are mobile in nature. This need a framework supporting distributed storage of semi- or un-structured temporal data and efficient parallel processing of ultra large data sets. The research presents the frameworks[20] supporting massive sensor data management based on Hadoop technology. A solution is provided to organize the massive sensor data effectively and realize parallel processing efficiently, which will compensate for the limitations of traditional relation-database. MapReduce framework was used to solve another spatial data problem by Google [21], where the study on the problem of road alignment by combining satellite and vector data is done. This work concentrates on the complexities of the problem. Here, [22] it has been analyzed the processing of sensor data after it has been received from Wireless Sensor Networks and stored on servers. A data model structuring the sensor information stored allowing a wide range of analysis by MapReduce framework[22].

Parallel DBMSs are good at efficient queying of large data sets. MapReduce Style systems work extremely well at complex analytics and execute Extract Transform Load task which is the feature of Data Warehousing. Neither is good at what the other does well. So the authors would like to put forward that these two systems can complement each other by working together assuming BSP is implemented on MapReduce. HadoopDB is such a[23] hybrid system which aims at combining the best features from both Hadoop and RDBMS. The basic idea of HadoopDB is to connect multiple single node database systems (PostgreSQL) using Hadoop as the task coordinator and network communication layer. Greenplum and Aster Data allow users to write MapReduce type of functions over data stored in their parallel database products[24].

6. CONCLUSION

Parallel databases scored high on performance and MapReduce scored high on flexibility in handling unstructured data. Both systems offer a querying language: Pig Latin in MapReduce systems and SQL for parallel databases[25]. There are some other techniques like Bulk Synchronous Parallel Computing engine(BSP) on top of Hadoop as given earlier and MPI (YARN) which are for processing graph algorithms on huge datasets.

MapReduce related research continues to be an active and attract interests from both industry and academia. It is of great interest for the parallel DBMS vendors since all the elaborated techniques like MapReduce ,Parallel DBMS and BSP, which works on top of MapReduce use cluster of nodes and scale out technology for large scale data analysis. An interesting research topic for future work will be by using MapReduce for analysis of large data generated from the mobile platforms.

7. REFERENCES

[1] A Text from mongoDB official website, “Big Data:Examples and Guidelines for the Enterprise Decision Maker”, May 2013

[2] Feng Wang,Bo Dong,Jie Qiu,Xinhui Li,Jie Yang,Ying Li, Hadoop High Availability through Metadata Replication, CloudDB'09 Proceedings of the First International Workshop on Cloud data management, ACM , Pages 37-44, 2009

[3] Daniel Peng and Frank Dabek, Large-scale Incremental Processing Using Distributed Transactions and Notifications, Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation, USENIX , 2010

[4] An article written by Michael Walker , www.analyticbridge.com/profiles/blogs/percolator-dremel-and-pregel-alternatives-to-hadoop, August 12, 2012

[5] Tyson Condie, Neil Conway, Peter Alvaro, Joseph M.Hellerstein, Khaled Elemeleegy, Russel Sears, Map Reduce Online, Proceedings in NSDI'10 Proceedings of the 7th USENIX conference on Networked systems design and implementation, pages 1-14, Oct 9 2009

[6] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, Robert E. Gruber,Bigtable: A Distributed Storage System for Structured Data, ACM Transactions on Computer Systems (TOCS), Volume 26 Issue 2, Article No.4 , Pages1-14, June 2008

[7] Michael Stonebraker, Daniel Abadi, David J. DeWitt, Sam Madden, Erik Paulson, Andrew Pavlo, Alexander Rasin, Map Reduce and Parallel DBMSs Friends or Foes, communications of the ACM, Vol. 53 No. 1, Pages 64-71, January 2010

[8] J. Dean, S. Ghemawat, MapReduce: Simplified Data Processing on Large Clusters, ACM Symposium on Operating Systems Design & Implementation - Volume 6, Pages 137–150, 2004

[9] Apache Software Foundation, Hadoop MapReduce, http://hadoop.apache.org/mapreduce, March 2012

[10] Apache Software Foundation, Hadoop Wiki: PoweredBy,http://wiki.apache.org/hadoop/PoweredBMarch 2012

[11] Michael Stonebraker, Daniel Abadi, J. Dewitt, Sam Madden,Erik Paulson, Andrew Pavlo, Alexander Rasin, MapReduce and Parallel DBMSs: friends or foes?”, Communications of the ACM, Volume 53 Issue 1,Pages 64-71, January 2010

[12] L. G. Valiant, A Bridging Model for Parallel Computation, Communications of the ACM, Pages 103–111, 1990

[13] M. T. Goodrich, N. Sitchinava, Q. Zhang, Sorting, Searching and Simulation in the MapReduce Framework, ArXiv e-prints, Pages 1-11, January 2011

[14]Kaushik Chandrasekaran, "Analysis of Different Parallel Programming Models", Indiana University

[15] Kyo-Hang Lee, Hyunsak Choi, Mongki Moon, Parallel Data Processing with MapReduce: A Survey, SIGMOD Record, Vol. 40, No. 4, Pages 11-20, , December 2011

[16] J. Lin and C. Dyer, Data-Intensive Text Processing with MapReduce. Syn. Lec. on Human Lang. Tech.-10

[17] G. Weikum, J. Hoffart, N. Nisakashole, M. Spaniol, F. Suchanek, M. Yosef, Big data methods for Computational Linguistics, IEEE Data Eng. Bulletin, 2012

[18] Yu Xu,Pekka Kostamaa,Like Gao, Integrating Hadoop and Parallel DBMS, ACM SIGMOD'10 ACM SIGMOD International Conference on Management of Data Pages 969-974, 2010

[19] Book on "Hadoop: The Definitive Guide" by Tom White by O'Reilly Publication, 2010

[20] Shahfik Amasha, Distributed-Data-Analysis-Using-Map-Reduce, Singapore University

[21] Xiaqing Wu, Rodrigo Carceroni, Hui Fang, Steve Zelinka, Andrew Kirmse, Automatic Alignment of Large-Scale Aerial raster's to Road-Maps, 15th annual ACM international symposium on Advances in Geographic Information Systems, Article No. 17, 2007

[22] Christine Jardak, Janne Riihijärvi, Frank Oldewurtel, and Petri Mähönen, Parallel Processing of Data from Very Large-Scale Wireless Sensor Networks, HPDC '10 Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing, Pages 787-794, 2010

[23] A. Abouzeid, K. Bajda-Pawlikowski, D. Abadi,A. Silberschatz, and A. Rasin. Hadoopdb: an architectural hybrid of mapreduce and dbms technologies for analytical workloads. Proc. VLDB Endow, Pages 922–933, 2009

[24]J. N. Hoover. Start-ups bring google's parallel processing to data warehousing. 2008

[25] Thesis of Miriam Lawrence Mchome, Comparison study between MapReduce(MR) and Parallel Data Management Systems in Large Scale Data Analysis, 2011