Middle-East Journal of Scientific Research 24 (11): 3460-3463, 2016
ISSN 1990-9233
© IDOSI Publications, 2016
DOI: 10.5829/idosi.mejsr.2016.3460.3463

E-Learning : Distributed Processing of Large Data Sets with Parallel Algorithm

[1]L. Paul Jasmine Rani, [2]S. Indhumathi, [2]C.H. Jyothirmayi and [2]S. Sunitha

[1]Assistant Professor, Department of Computer Science and Engineering,
Rajalakshmi Institute of Technology, India
[2]UG Students, Department of Computer Science and Engineering,
Rajalakshmi Institute of Technology, India

Abstract: Data that are generated from variety of sources with massive volumes, high rates and different data structure are collectively known as Big Data. Big Data processing and analyzing is a challenge for the current systems because they were designed without knowing the Big Data requirements and most of them were built on centralized architecture, which is not suitable for Big Data processing because it results in high processing cost and low processing performance and quality. MapReduce framework which was built as a parallel distributed programming model is used to process such large-scale datasets effectively and efficiently. Big Data software analysis solutions were implemented on MapReduce framework, describing their datasets structures and how they were implemented with MongoDB as NoSQL Database. NoSQL encompasses a large variety of different database technologies, developed in response to the demands which were presented in building the modern applications. MongoDB stores the data using a flexible document data model. Documents contain one or more fields, including arrays, binary data and sub-documents. E-learning, has turned to be an important part of society today, comprising an array of digitization approaches, also the components and delivery methods. Big Data allows the eLearning Professionals to customize the learning experience to provide learners with more effective, engaging and informative eLearning courses and modules.

Key words: E-Learning · MapReduce · MONGODB · Bigdata

INTRODUCTION

Data collected from variety of sources with massive volumes, high rates and different data structure is known as Big Data. The large-scale data sets are processed efficiently by the MapReduce framework which is built as a parallel distributed programming model. Big Data usually includes data sets with sizes beyond the ability of commonly used software tools. It doesn't sample, it just obes and tracks what happens. It is often available in real-time. Big Data draws from text, images, audio, video plus it completes missing pieces through data fusion. The MapReduce algorithm that helps in sending the Map & Reduce tasks to the appropriate servers in a cluster. The MapReduce engine provides a platform for the parallel execution of algorithms written in Java. Although big data is often described as unstructured, the incoming data always has some structure. It does not have a fixed, predefined structure when written to HDFS[Hadoop Distributed File System]. Instead, MapReduce creates the desired structure as it reads the data for a particular job. The same data can also have many different structures imposed by different MapReduce jobs.

E-learning is electronic learning and typically this means using a computer to deliver part, or all of a course whether it's in a school, part of your mandatory business training or a full distance learning course. There are several benefits to e-learning whether you choose to use it on its own, or to enhance your existing in house training. Some of them are as follows: It is cost effective and saves time. Provides learning facilities 24/7, anywhere. It makes tracking of course progress a breeze and also it's discreet.

The storage of large amount of data should also be considered. The solid-state hybrid drive (SSHD) incorporates a small NAND flash memory into a hard drive, resulting in an integrated device with combined HDD (Hard Disk Drive) and SSD (Solid State Disk)

Corresponding Author: L. Paul Jasmine Rani, Assistant Professor, Department of Computer Science and Engineering, Rajalakshmi Institute of Technology, India.

storage. By identifying the data highly associated with the performance and buffering them in the SSD part, SSHD can deliver a better performance than the standard hard drive. For MapReduce data processing system, the adoption of SSHD requires us to completely redesign the core modules, including the data layout, query optimizer, indexing technique and compression algorithm.

Existing System: Present day organizations are confronted with the challenge and the opportunity of data growing at unprecedented rates.. “Big Data” is the most important concept in today's world. Big Data analytics has the potential to provide great insights and opportunities to most of the organizations in the areas of consumer behavior, marketing, fraud detection and customer service. Big Data uses the HDFS to store the data. But the HDFS is not the best platform because it is vulnerable by nature. Also HDFS is unsuitable for small data, has some stability issues. And also managing a complex application using HDFS is a difficult task. It does not split large data files. The size of the files and the number of replications is not configurable. While most of the organizations recognize the importance and benefits of Big Data analytics, there are challenges arising from the nature of Big Data and limitations of already existing technologies that need to be considered.

The existing e-learning websites use the traditional database to store the data. The traditional database include the XML database, Oracle database or the MySQL servers which help to store and transport the data. These traditional database help to store the structured data. These traditional database do not scale well to very large sizes of data. Then these databases do not do the unstructured data search well nor they do handle data in unexpected formats well.

Also using serial algorithm will have some problems like scalability, they take large time for processing large clusters of data, the processing time is also long. The paper says that [4] presents two new algorithms, GD2L and CAC, for managing the buffer pool and the SSD in a database Management system. Both algorithms are cost-based and the goal is to minimize the overall access time cost of the workload. Since data present in the SSD may be more efficient than that in the HDD, the system can identify the data only after system failure.

Proposed System: The project provides a e-learning environment which provides information about the different programming languages. The e-learning environment provided here helps the users to have a clear understanding on the concepts they learn.But the greatest challenge is meeting the storage level. Traditional model cannot have an access to a very large amount of scalable data. So, Centralized System came into view which could store and process a very large data. More over, Centralized System cause too much of bottleneck during multiple file processing. So, as a result of which Map Reduce algorithm was proposed, in which data would be stored in a common database and then retrieved by Map Reduce algorithm. A layout plan has been laid to maximize the performance in which discovering the query patterns is prior. Keyword based queries, data mining tasks and machine learning are the possible workloads.

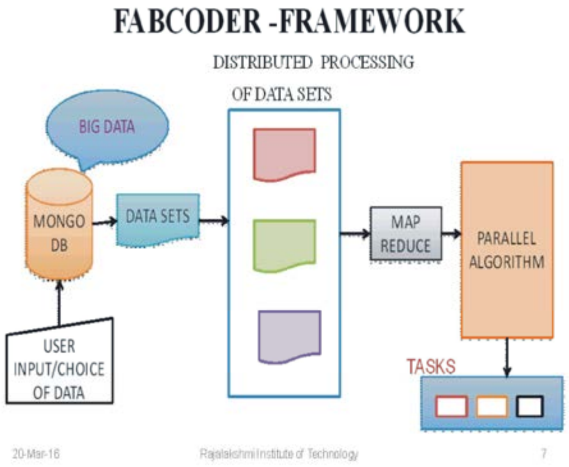

Fig. 1: FRAME WORK OF PROPOSED MODEL

The Figure 1 shows the framework design of the proposed model. In this the database used is the MongoDB which is used to store the unstructured data. This large amount of unstructured data which is to be processed is referred as bigdata. The large data here is divided into small datasets. These datasets are processed in a distributed manner. These processed data sets are given as input to the MapReduce algorithm. In this the data sets are mapped and reduced, then the parallel algorithm helps to process those data sets simultaneously.

Algorithm: MapReduce algorithm uses a parallel programming model for processing data on a distributed system. Vast amount of data are processed quickly and also can be scaled linearly. The algorithm is more effective as a mechanism for the processing of unstructured and semi-structured data. MapReduce job is the successive alternation of two phases, the Map phase and the Reduce phase. Each of the Map phase applies a transform function over each record in the input data and produces a set of records expressed as key-value pairs. The output from the Map phase is considered to be the input to the Reduce phase. The Map output records are sorted into key-value sets so that all records in a set have the same key value in the reduce phase. A reducer function is applied to all the records in a set and a set of output record is produced as key-value pairs. The Map phase is logically run in parallel over each record while the Reduce phase is run in parallel over all key values.

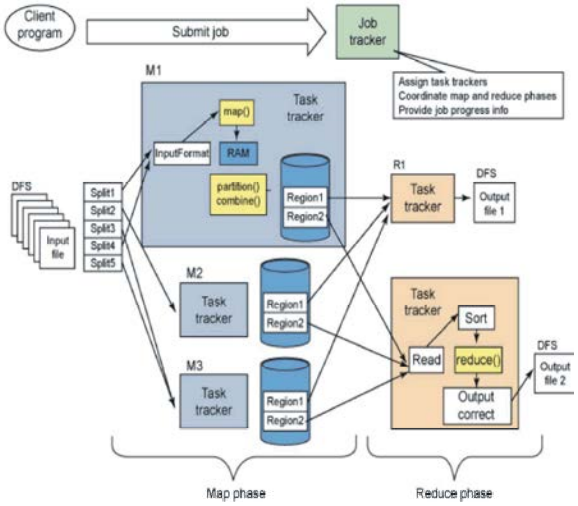

Fig. 2: Mapreduce Design

The MapReduce algorithm contains two important tasks, namely Map and Reduce.

The map task is done by means of Mapper Class
The reduce task is done by means of Reducer Class.

Mapper class takes the input, tokenizes it, maps and sorts it. The output of Mapper class is used as input by Reducer class, which in turn searches matching pairs and reduces them.

MapReduce implements various mathematical algorithms to divide a task into small parts and assign them to multiple systems. In technical terms, MapReduce algorithm helps in sending the Map & Reduce tasks to appropriate servers in a cluster.

These mathematical algorithms may include the following -

Sorting
Searching
Indexing
TF-IDF

Sorting: Sorting is one of the basic MapReduce algorithms to process and analyze data. MapReduce implements sorting algorithm to automatically sort the output key-value pairs from the mapper by their keys. Sorting methods are implemented in the mapper class itself. In the Shuffle and Sort phase, after tokenizing the values in the mapper class, the **Context** class (user-defined class) collects the matching valued keys as a collection. To collect similar key-value pairs (intermediate keys), the Mapper class takes the help of **Raw Comparator** class to sort the key-value pairs.The set of intermediate key-value pairs for a given Reducer is automatically sorted by Hadoop to form key-values (K2, {V2, V2, …}) before they are presented to the Reducer.

Searching: Searching plays an important role in MapReduce algorithm. It helps in the combiner phase (optional) and in the Reducer phase.

Indexing: Normally indexing is used to point to a particular data and its address. It performs batch indexing on the input files for a particular Mapper. The indexing technique that is normally used in MapReduce is known as **inverted index.** Search engines like Google and Bing use inverted indexing technique.

TF-IDF: TF-IDF is a text processing algorithm which is short for Term Frequency - Inverse Document Frequency. It is one of the common web analysis algorithms. Here, the term 'frequency' refers to the number of times a term appears in a document.

Experimantal Analysis: In this paper the result is that a e-learning page is developed which helps the user to learn about the different programming languages. In this the unstructured data is processed and retrieved as structured data. Processing of large datasets is done efficiently using the parallel algorithm. MapReducce algorithm helps to process those datasets and provide the desired information to the users.

We evaluate HM using TPC-H benchmark in Table 1 and the results show that with our new design, the hybrid system can provide a similar performance as the SSD-only system.

Table 1:

TYPE	TPC-H Q1	TPC-H Q14
HDD	175s	180s
SSD	56s	81s

The TPC BenchmarkH (TPC-H) is a decision support benchmark. It consists of a suite of business oriented ad-hoc queries and concurrent data modifications. The queries and the data populating the database have been chosen to have broad industry-wide relevance. This benchmark illustrates decision support systems that examine large volumes of data, execute queries with a high degree of complexity and give answers to critical business questions.

The raw performances of our HDDs and SSDs are shown

Table 2:

TEST	HDD	SSD
Sequential Read	15s	454s
Sequential Write	103s	250s
Random Seek	151s	121s

The cost of query optimization is actually determined by the number of columns in each select, predicate and join group. We show the query optimization cost for Q1 (7 columns),Q2 (11 columns) and Q3 (14 columns) in Figure. The most expensive query is Q3 involving a join of three tables. However, its optimization cost is still less than 1 second. Compared to the query process time, query optimization cost is negligible and finding a good plan can reduce the process time by an order of magnitude.

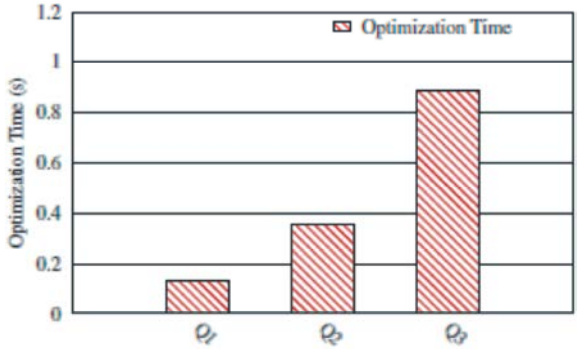

Figure: Cost of Optimization

CONCLUSION

This paper enables the use of e-learning, in which large datasets are processed. Here the important factor considered is the storage area. SSHD provides a better performance than the standard hard drive. Three most popular workloads on unstructured data are keyword based queries, data mining tasks and machine learning tasks. These workloads are also considered while processing the data. This also maximizes the performance of the system. Hence the maximum use of the system can be made by the users. Also in this it is found that unstructured data is processed efficiently and the retrieval of information is also found to be quick.

REFERENCES

1. Sai Wu, Gang Chen, Ke Chen, Feng Li and Lidan Shou, 2015. HM: A Column-Oriented MapReduce System on Hybrid Storage.
2. Abadi, D.J., P.A. Boncz and S. Harizopoulos, 2009. Column oriented database systems. PVLDB, 2(2): 1664-1665.
3. Agrawal, N., V. Prabhakaran, T. Wobber, J.D. Davis, M.S. Manasse and R. Panigrahy, 2008. Design tradeoffs for ssd performance. In USENIX Annual Technical Conference, pp: 57-70.
4. Canim, M., G.A. Mihaila, B. Bhattacharjee, K.A. Ross and C.A. Lang, 2009. An object placement advisor for db2 using solid state storage. Proc. VLDB Endow., 2(2): 1318-1329.
5. Dittrich, J., J.A. Quian´e-Ruiz, A. Jindal, Y. Kargin, V. Setty and J. Schad, 2010. Hadoop++: Making a yellow elephant run like a cheetah(without it even noticing). PVLDB, 3(1): 518-529.
6. Do, J., D. Zhang, J.M. Patel, D.J. DeWitt, J.F. Naughton and A. Halverson, 2011. Turbocharging dbms buffer pool using ssds. In SIGMOD, pp: 1113-1124.
7. Floratou, A., J.M. Patel, E.J. Shekita and S. Tata, 2011. Column-oriented storage techniques for mapreduce. PVLDB, 4(7): 419-429.
8. He, Y., R. Lee, Y. Huai, Z. Shao, N. Jain, X.Z. 0001 and Z. Xu, 2011. Rcfile: A fast and space-efficient data placement structure in mapreduce-based warehouse systems. In ICDE, pp: 1199-1208.
9. Jeon, H., K. El Maghraoui and G.B. Kandiraju, 2013. Investigating hybrid ssd ftl schemes for hadoop workloads. In Proceedings of the ACMInternational Conference on Computing Frontiers, pp: 20:1-20:10.
10. Jiang, D., B.C. Ooi, L. Shi and S. Wu, 2010. The performance of mapreduce:An in-depth study. PVLDB, 3(1-2): 472-483.
11. Jiang, D., B.C. Ooi, L. Shi and S. Wu, 2010. The performance of mapreduce:An in-depth study. PVLDB, 3(1): 472-483.
12. Kang, W.H., S.W. Lee and B. Moon, 2012. Flash-based extended cache for higher throughput and faster recovery. PVLDB, 5(11): 1615-1626.