Indian Journal of Science and Technology, *Vol 9(10), DOI: 10.17485/ijst/2016/v9i10/88981, March 2016*

ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

An Efficient Traffic-Aware Partition and Aggregation for Big Data Applications using Map-Reduce

Y. Dinesh Reddy[1*] and A. Pio Sajin[2]

[1]Department of Computer Science and Engineering, Sathyabama University, Chennai – 600119, Tamil Nadu, India; yvdineshreddy@gmail.com
[2]Faculty of computing, Sathyabama University, Chennai – 600119, Tamil Nadu, India; piosajin19@gmail.com

Abstract

Objective: The objective of this system to reduce network traffic cost for a Map-Reduce job by designing a novel intermediate data partition scheme. **Methods/Analysis:** The Map-Reduce model streamlines the large scale information handling on commodities group by abusing parallel map and reduces assignments. Even though numerous endeavors have been made to increase the execution of Map-Reduce works, they disregard the network activity produced in the mix stage, which assumes a basic part in execution upgrade. **Findings:** Generally, a hash capacity is utilized to segment middle of the road information among decrease assignments, which, nonetheless, is not movement effective in light of the fact that network topology and its data size connected with every key are not thought seriously about. Reexamine to lessen system movement cost for a Map-Reduce work by planning a novel moderate information segment plan. **Applications/Improvement:** Decomposition based dynamic algorithm and hc algorithm is proposed to manage the huge scale optimization issue for enormous information application is likewise intended to change information parcel and conglomeration in a dynamic way. Finally, broad reproduction results show that the proposed recommendations can altogether decrease network movement cost under offline cases.

Keywords: Big Bata, Data Aggregation, Dynamic Decomposition-based Distributed K- means Algorithm, hc Algorithm, Traffic Minimization

1. Introduction

Big data is an evolving term that describes combinations of large amount of structured, semi-structured and unstructured data that has the potential to be extracted for information[1]. The structured data are easily organized whereas unstructured data are not organized in pre-defined manner. International Data Corporation (IDC) defined big data as volume, variety, velocity, variability, veracity and complexity. The features of big data include high-performance and inexpensive processing power, data integration and quality capabilities, unstructured text management. Relational database management system is difficult to work with big data. Hadoop is used to overcome this problem. Hadoop is a java based programming model which living big data. Hadoop having map-reduce and hadoop distributed file system (storage). The Hadoop Distributed File System (HDFS) consists of two nodes, data node and name node. The advantages of hadoop are that it is cost effective, flexible, fast, scalability, autonomous. But to compute and analyze this huge data is not an easy task. Many challenges arise while processing these terabytes of data. The data produced by the map phase are ordered, partitioned and transferred to the proper machines executing the reduce phase is represented in the Figure 1. The resulting network traffic pattern from all map tasks to all reduce tasks can cause a great bulk of network data flow enforce a difficult sate on the efficiency of data analytic applications. This work proposed an efficient Traffic Aware separation and Aggregation to minimize network sequence of operation cost for Big Data applications using Map-Reduce. It proposes a three-layer

**Author for correspondence*

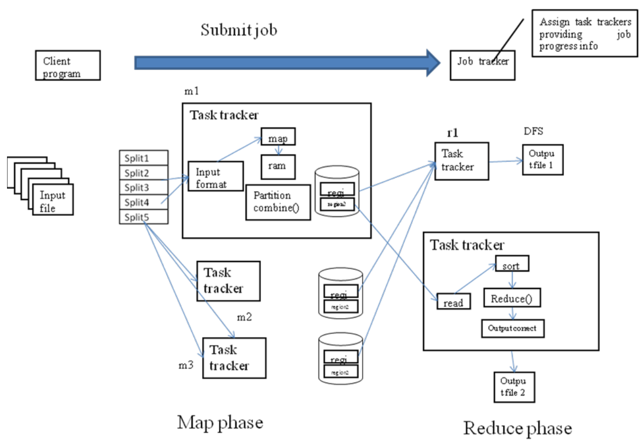

Figure 1. Architecture diagram.

model for this trouble and devises it as a mixed-integer nonlinear crisis, which is then transferred into a linear appearance with the intention of can be solved by mathematical tools. To compact with the major formulation due to big data, it intended a dynamic algorithm to solve the problem on multiple machines. Expanded dynamic decomposition-based distributed K- means algorithm to grip over the Map- Reduce job in an efficient way when a number of system parameter are not given. At last, our extensive simulation to evaluate our projected algorithm under offline cases. The experiment result shows that our proposal can successfully decrease network traffic cost under different complex settings.

That the Data processing is a censorious impact on data synchronous performance large bodies of researches have been promoted to address this challenge. The users have to specify map function that processes a key/value pair to give a set of in-between key/value pairs, and a reduce function that merges all intermediate values associated with the same intermediate key is employed[2]. The run-time system takes care of the details of partitioning the input data, scheduling and processing. This user function allows the programmers to work without any experience in parallel and distributed system and easily utilize the resources of a large distributed system which is the merit of this system. It is also said that this method is highly scalable and large computation of data is easily parallelized.

Popularity in the map reduce clusters can be gathered by collecting the information about the jobs that has been submitted for execution. The history of the usage is recorded in order to prioritize the maximum used skew content. The metric used to capture the popularity of the files is the number of concurrent access. The larger files are said to contribute the maximum accesses, so the reduction in the contention leads to the greater improvement in the performance of the cluster. To address the content of the file, Scarlett introduced the replication of the content at the smallest coarseness[3]. The reproduction is done based on popularity that is predicted. The file replication factor is determined by two approaches namely the priority approach and the round robin approach. After replication the desired numbers of replicas of the blocks are placed in as many as distinct racks and machines as possible with uniformity of the load given.

The Skew Tune is an approach that is used in automatic skew mitigation in user defined Map Reduce programs. Skew Tuning system acts as a replacement for existing Map reduce[4]. It minimizes the input from the user by identifying the task that has greatest processing time. The data that are unprocessed are repartitioned. This enables the full utilization of the nodes in the cluster as well as the preservation of the order of input such that the original output is reconstructed. Skew Tuning also evaluates the effectiveness of certain applications and makes it transparently compatible with the code that exists already. Even the complex workflow context and the advanced Map Reduce algorithm transparency is maintained. The experimentation shows that this process of Skew Tune can improve the Hadoop datasets to a four times larger one.

Big data is heavily increased on Map-Reduce for large scale data processing[5]. There are many map reduce I/O bound are raised in a very impressive manner which take down the performance, so reducing the bounds is the efficient task. In order to decrease the map reduce I/O bounds, this paper implemented, Themis an efficient I/O map reduce. Themis is a map reduce concept that reads and writes data records to place exactly twice in the disk, which is the minimum requirement for data sets to fit in memory. This helps to reduce the I/O bounds of Map-Reduce. Themis also performs various map reduce tasks, which includes log analysis, DNA read sequence alignment, and Page Rank.

In the concept explained that, Data intensive cluster computing systems like Hadoop and Dryad become more popular because there is a need to share cluster between users. But there is some job among map reduce and data locality[6]. To overcome this problem delay scheduling algorithm is introduced. This algorithm is used in various scheduling policies. When the job scheduled according to fairness cannot launch any task. By using this delay scheduling algorithm, the task will wait for small amount of time for other jobs to finish. 100% locality is achieved by using this delay scheduling algorithm and also increase the response time for small jobs.

In data-parallel computing frameworks, user has some limitation in running short tasks. To improve the performance of data parallel processing cluster computing has been used. If the cluster is mostly utilized then an interactive job may need to wait for other jobs to complete[7]. To avoid this tiny task is implemented in cluster to break large job into many smaller jobs. So time taken to perform the task is limited. It will improve the scalability in cluster framework.

In these, paper referred that, Map reduces is the programming language to process vast amount of data in large cluster. Map and reduce are the two functions with simple interface to perform parallel implementation of many tasks, but some operations like joins are not supported by this function. To this job a new framework called map reduce merge is implemented[8]. The data which are already partitioned and sorted is integrated expeditiously. Many join algorithm also enforced by using this Map Reduce Merge.

In his paper has referred that Map reduce gives a scheduling model for huge data processing. This map reduce is admired by many functional style which helps to write their code independently and it is divided automatically into many maps and also distributed over many machines. Hadoop is the open source technology helps in implementation of map reduce and schedules the map task. This leads to avoid the network traffic and decrease the performance. But hadoop agenda task without view data neighborhood which down the performance. To overcome this problem LARTS is used to improve the performance efficiently. LARTS is described as a locality aware reduces task scheduler. It helps in analyzing all data location, sizes and combines all the required data for computation parallel and increase the performance[9]. So LARTS improves node-local, rack- local and off-rack traffic and increase the performance efficiently.

Map reduce has become every popular nowadays. In map reduce, while shuffling, data skew problems are experienced. Though hash partitioning is used default in hadoop, it works well and good only if all the keys are equally present and they are uniformly stored in the data nodes. This blind partitioning leads to network congestion, performance degradation and inequity in the reducers input. So LEEN method gives a better way by tracking the intermediate key frequency and their distribution. The keys are sorted out in based on their locality and fairness value. The nodes are sorted out in descending order for each key value based on the frequency that each key has[10]. This partitioning is done by fairness score value. The solution overcomes the partitioning by implementing the locality concept in reducer and achieved a 40% increase in the performance while applying.

Many people focus on shuffling the data, but recently the network traffic is emerging in hadoop. Map reduce is used to avoid failures during partitioning. Many researchers have done a inquiry and establish that around 23% is used as non-local map tasks. To overcome this problem they proposed the algorithm called maestro. Maestro reduces maps up to 34%. Maestro schedules task in two waves, first wave states that when the task is started, it is used to fill each node and to avoid issues like empty node. It fills according to higher capacity. The second is runtime scheduling which helps in mapping the task to the replicas. These two waves results in high performance. So the Maestro design is used to improve the efficiency and to map non local map task[11].

The Map reduce is a framework provided by hadoop to manage large distributed data. A modified map reduce is used to deflect the fault tolerance and decrease the output of map task[12]. In this pipeline concept is introduced to interconnect machines, this leads to reduce completion time and extends the program utilization for all jobs. HOP (Hadoop Online Prototype) is an extended version of hadoop map reduces which supports online collection and continuous queries. HOP provides a log of all jobs to the user. HOP allows writing programs and supporting processes such as event monitoring and stream process, this leads to retain fault tolerance and run unmodified programs.

2. Materials and Methods

A decomposition based dynamic and hc algorithm for job distribution is proposed to manage the huge scale optimization issue for enormous information processing application in a dynamic way. Expanded dynamic decomposition-based distributed k-means algorithm to grip over the map-reduce job in an efficient way when a number of system parameter are not given.

2.1 DDD k-means Algorithm Aggregation

Dynamic decomposition-based distributed k-means algorithm for big data applications by decomposing the original large-scale problem into several sub problems that can be solved in parallel It is a sub-type of parallel algorithm, typically executed at the same time, with divide parts of the algorithm being run simultaneously on individual processors, and having small information about what the other parts of the algorithm are doing. One of the better challenges[13] in developing and implementing distributed algorithms is successfully organize the conduct of the individual parts of the algorithm in the face of processor loser and irresponsible communications links.

2.2 Healthcare Monitoring Algorithm

Heartbeat is a signal point that it is active. A data node sends heartbeat to Name node and task tracker will direct its heart beat[14] to job tracker. If the Name node does not receive heart beat then they will determine that place is some job in data node is unable to perform the assigned task is represented in the Figure 2. HM is the process of notice damage in structures. The goal of HM is to improve the safety and reliability of infrastructure by find damage before it reaches a censorious state. To reach this goal, technology is being developed to replace soft visual inspection and time based sustainment procedures with more quantifiable and control damage appraisal processes. These processes are implemented with the intent of achieving more cost effective condition based sustainment.

2.3 Traffic Minimization

It formulates the network traffic minimization problem. To facilitate our analysis and construct an auxiliary graph with a three-layer structure. The given placement of mapper's and reducers applies in the map layer and the reduce layer, respectively. In the aggregation layer, it creates a potential aggregator at each machine, which can aggregate data from all mapper's. Since a single potential aggregator is sufficient at each machine, it also use N to denote all potential aggregators. In addition, it creates a shadow node for each mapper's on its residential machine. In contrast with potential aggregators, each shadow node can receive data only from its corresponding mapper's in the same machine. It mimics the process that the generated intermediate results will be delivered to reduce directly without going through any aggregator.

2.4 Data Aggregation

Our visual displays are designed so that neighboring data values are represent to spatially adjacent engraved elements. When there are overly many factor to suitable within a given space, the elements are collective with their closest having the nature of space neighbor. By limiting collection to spatially distal objects, it will be much casual to infer the contents of a whole amount and subsequently interpret it. When the level of network aggregation change, its visual representation must be naturally change. Such changes can sometimes make it difficult to maintain visual setting from one level of assemblage to the next. This interactive display is crafted so that context is naturally saved across multiple levels. This is carrying through by encoding the data with a graphical holding that remains constant across conglomeration levels.

2.5 Modules

2.5.1 Partitioning the Data

The data partition is mainly depends upon number of Map task we are going to include in LBA data cluster. If there are too many map tasks, multiple tasks will contend for the same slot and there will be wasting times for slot re-allocation and task format is represented in the Figure 3. Divider-List is an array list that's each array contain the blocks of the same partition, and there are Split-Mem-Num arrays in this list. Then we choose the proper replica for every block which we will describe the detailed steps of this in next subsection. Then we cluster the replicas into partitions based on its locality.

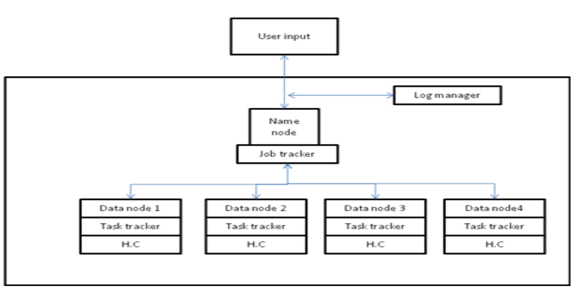

Figure 2. Flow diagram of healthcare architecture.

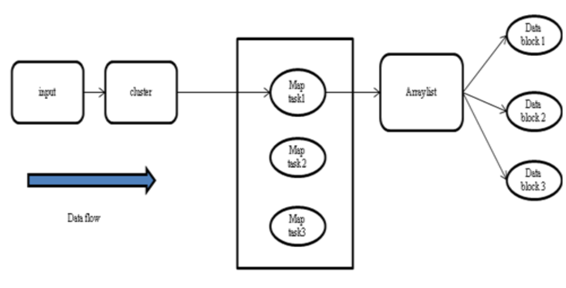

Figure 3. Partitioning the data.

We do this by checking the location of a replica whether existed in the Split-List, if it is then add it to the location partition, if not then insert a new list into Split-List of this new location. If there are too many map tasks, multiple tasks will contend for the same slot and there will be wasting times for slot re-allocation and task format. Split-List is an array list that's each array contain the blocks of the same partition, and there are Split-Mem-Num arrays in this list.

2.5.2 Log Maintenance

Server can make various other log files to help you see what action is taking place. However, you must dispose of these files in a regular manner to control that the logs do not take up too much disk space.

A maintenance log is a document that records who did what, when, and why. Maintenance logs are highly useful for troubleshooting repeat or confuse jobs, as they supply a record of all work execute on the system and may clear up hard-to-spot interactions between apparently uncorrelated symptoms.

3. Results and Discussion

Here we compare some other design alternatives with Hadoop.

Comparison between HDFS CLUSTER and MAP REDUCE:

HDFS CLUSTER: Combination of namenode and datanode. It is the name given to whole configuration of Master and Slave where the data is stored.

MAPREDUCE: It is the programming model. Which is used to retrieve the analyzing the data.

The implementation is the working of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the team's and tracking the team's progress throughout the development activity. For the proper working of the project the following configuration should be maintained. The operating system should be of Windows 95, till 8. The application Server is Tomcat 5.0/6.X. Since the Front used is Hadoop, the configuration should be strictly followed. The scripting language used must be java scripting. Java Database connectivity is the Database connectivity used in hadoop. Only if the following configuration is satisfied, then the partitioning works better. Now, we presume that you are aware of R, what it is, how to install it, what its key features are, and why you may want to use it. Now we need to know the limitations of R (this is a better introduction to Hadoop). Before processing the data; R needs to load the data into Random Access Memory (RAM). So, the data needs to be smaller than the available machine memory. For data that is larger than the machine memory, we consider it as Big Data (only in our case as there are many other definitions of Big Data)[15].

To avoid this Big Data issue, we need to scale the hardware configuration; however, this is a temporary solution. To get this solved, we need to get a Hadoop cluster that is able to store it and perform parallel computation across a large computer cluster. Hadoop is the most popular solution. Hadoop is an open source Java framework, which is the top level project handled by the Apache software foundation. Hadoop is inspired by the Google file system and Map Reduce, mainly designed for operating on Big Data by distributed processing. Hadoop mainly supports Linux operating systems. To run this on Windows, we need to use VMware to host Ubuntu within the Windows OS.

The users/clients are register to login page. Client is an application which is used to interact both the namenode and datanode, it means interact job tracker and task tracker. Whatever the user information is visible on the console of eclipse J2EE.Server configuration of UbutunLinux, we set the server ip address is 127.0.0.1 for creating the servers i.e. (server 1, server 2, server 3, server 4, server 5), hat servers are saved in the root directory Locations as HDFS mount folder in the system. The client wants to browse and upload the txt file. In console application it shows the Health care monitoring starts under that files are stored in the path mnt/hdfs/servers1/filename1.txt to mnt/hdfs/servers4/filename5.txt.the uploaded files are stored into the root directory as mount/hdfs. Mongo DB is document oriented database and is created by open source product develop and supported by "LOGEN". MongoDB is an open-source document database that provides high performance, high availability, and automatic scaling. Mongodb is available under general public license and commercial license. In ubuntulinux operating system open the terminal and to connect the mongodb. While seeing the uploaded files type query as db.UploadDetails.find (); then it shows the uploaded list in backend database. Let us consider that we blocked any one of the server in the File menu list of server configuration, it means that blocked of one server txt file information as the same data present in the another server/data node (i.e.) fault tolerant it means same text file data present in another files. Remaining servers are open, using query as db.ServerConfig.find (); After that we download the uploaded txt file information, then the decryption process started shown in console bar or log maintenance. For decrypting process we are using AES (Advanced Encryption Standard) technique. All the available files are display to download in File menu. The retrieved file is stored in the folder has "workspace" in the form user understandable format. Easy to process the datasets in large scale parallel data processing.i.e.the processing time and retrieving time is very less and fault tolerant is increased. The illustrations of experiment result shows that our proposal can successfully decrease network traffic cost under different complex settings. The client first asks for the data that is needed name node. The metadata does the clock operation into the data node. The data node maintains the blocks in the form of rack. The replicas are also written in another data node where similar data blocks are maintained. Thus, if, the client requests for any data, the above mentioned operations are performed and then data is fetched.

4. Conclusion and Future Work

This work proposed an efficient Traffic Aware separation and Aggregation to minimize network traffic cost for Big Data applications using Map-Reduce. It proposes a three-layer model for this trouble and devise it as a mixed-integer nonlinear crisis, which is then transferred into a linear appearance with the intention of can be solved by mathematical tools. To compact with the major formulation due to big data, it intended a dynamic algorithm to solve the problem on multiple machines. Expanded dynamic decomposition-based distributed K- means algorithm to grip over the Map-Reduce job in an efficient way when a number of system parameter are not given. At last, our extensive simulation to evaluate our projected algorithm under offline cases. The experiment result shows that our proposal can successfully decrease network traffic cost under different complex settings. I learned java 2 standard edition and Hadoop which are very use full to develop this application.

The future scope of the works should be made on complex data partitioning in the database where more intelligent methods have to be employed. This includes analyzing computation cost, skew record etc. So that optimization of data partition is done in map reduce.

5. Acknowledgement

I express my sincere gratitude to our honourable chancellor Col. Dr. Jeppiaar, M.A., B.L., Ph.D., Sathyabama University.

An endeavor over a long period can be successful only with the advice and supports of many well wishers. We take this opportunity to express our gratitude and appreciation to all of them. We are extremely thankful to our beloved Dr. T. Sasipraba Organizing Secretary GCMT-2015 at Sathyabama University Chennai, India. Who took keen interest and encouraged us in every effort throughout this event.

We owe our gratitude to our Directors Dr. Marie Johnson, B.E., M.B.A., M.Phil., Ph.D., and Dr. Mariazeena Johnson, B.E., M.B.A., M.Phil., Ph.D., for permitting us to use the facilities available to accomplish the project successfully.

We express our heartfelt thanks to Dr. S. Murugan, M.E., Ph.D., Head of Dept – CSE for his kind attention and valuable guidance to us throughout this course.

We express our profound respect and gratitude to our project guide Mr. A. Pio Sajin, M.E., Assistant Professor, faculty of computing, Sathyabama University for his valuable support and guidance in completing the project successfully.

Y. Dinesh Reddy received B.Tech Degree in computer science and engineering from Chaitanya Bharathi Institute of Technology in 2014(kadapa district, A.P).Present persuing M.E (CSE) in Sathyabama` University. Her current research interest in Big Data.

6. References

1. Justin SS, Koundinya RVP, Sashidhar K, Bharathi CR. A survey on big data and its research challenges. ARPN Journal of Engineering and Applied Sciences. 2015; 10(8):3343–47.
2. Deans J, Ghemawat S. Map Reduce: A flexible data processing tool. Communications of the ACM-Amir Pnueli: Ahead of his time. 2010; 53(1):72–7.
3. Ananthanarayanan G, Agarwal S, Kndula S, Greenberg A, Stoica I, Harlan D, Andharrise E. Scarlett; Coping with skewed content popularity in map-reduce clusters. In Proc. EuroSys'11. 2011. p.1–14.
4. Xu Y, Kostamaa P, Zhou X, Chen L, Kwon YC. Handling data skew in parallel joins in shared-nothing systems. In Proceedings of the SIGMOD International Conference on Management of Data, SIGMOD'08. 2008. p.1043–52.
5. Isard M, Budiu M, Yu Y, Birrell A, Fetterly D, Rasmussen A. Dryad: Distributed data-parallel programs from sequential

building blocks. In Proceedings of the 2nd ACM European Conference on Computer Systems (EuroSys '07). Lisbon, Portugal, 2007. p.59–72.

6. Szalay A, Bunn J, Gray J, Foster I, Raicu I. The importance of data locality in distributed computing applications. NSF Workflow Workshop. 2006. p.1.
7. Bu Y, Howe B, Balazinska M, Ernst MD. HaLoop: Efficient Iterative Data Processing on Large Clusters. In Proceeding of the VLDB Endowerment. 2010; 30(1). p.1–12.
8. Yu Y, Gunda PK, Isard M. Distributed aggregation for data-parallel computing: interfaces and implementations. In Proceedings of the 22nd SIGOSP Symposium on operating System principles, SOSP'09. 2009. p.247–60.
9. Yu Y, Isard M, Fetterly D, et al. Dryad LINQ. A system for general-purpose distributed data-parallel computing using a high-level language. In Proc. of the 8th OSDI Symposium; 2008.
10. Costa P, Donnelly A, Rowstron AI, O'Shea G. Camdoop: Exploiting in-network aggregation for big data applications. In ACM SIGCOMM and ACM SIGOPS, NSD'12. 2012. p.1–14.
11. Ousterhout K, Wendell P, Zaharia M. Stoica I. Sparrow: Scalable scheduling for sub-second parallel jobs. University of California: Berkeley; 2013.
12. Candea G, Kawamoto S, Fujiki Y, Friedman G, Fox A. Microreboot-A Technique for Map Reduce Cheap Recovery. In 6th Symposium on Operating Systems Design and Implementation, OSDI'04. 2004. p.31–44.
13. Hindman B, Konwinski A, Zaharia M, Stoica I. A common substrate for cluster computing. In Workshop on Hot Topics in Cloud Computing (Hot Cloud). 2009. p.1–5.
14. Yasodha P, Ananthanarayanan NR. Analyzing big data to build knowledge based system for early detection of ovarian cancer. Indian Journal of Science and Technology. July 2015; 8(14):1–7.
15. Noh K-S, Lee D-S. Bigdata platform design and implementation model. Indian Journal of Science and Technology. Aug 2015; 8(18):1–8.