Traffic Control in Big Data applications using MapReduce functions

[1]Divya Sharma,[2]Madhuri Wade,[3]Aisha Sayyad ,[4]R. D. Chintamani
[1,2,3]Student, [4]Professor
Department of Information Technology,
SRES's Sanjivani College Of Engineering,Kopergaon-423603,Maharashtra,India

---

***Abstract* - The MapReduce programming typical make simpler large-scale data processing on service cluster by developing parallel map tasks and decrease jobs. Though several struggles have been thru to advance the performance of MapReduce jobs, they disregard the network traffic created in the shamble stage, which plays a serious role in performance improvement. Usually, a hash function is used to partition transitional data between reduce tasks, which, though, is not traffic-efficient since network topology and data scope related with every key are not taken into consideration. We study to decrease network traffic rate for a MapReduce job by planning a new transitional data partition system. Additionally, we equally contemplate the aggregator assignment problem, where every aggregator can reduce complex traffic from numerous map tasks. A decomposition-based distributed algorithm is suggested to deal with the large-scale optimization issue for big data application and an online algorithm is also intended to regulate data partition and aggregation in an active way.**

***Keywords* - Hadoop, HDFS, MapReduce, Big-Data**

---

1. INTRODUCTION

Big data is a word that states to data sets or groupings of data sets whose scope (volume), difficulty, and rate of growth (velocity) make them hard to be seized, coped, treated or examined by orthodox technologies and tools, such as relational databases. Hadoop MapReduce programming model is being used for handling Big Data, which contains of data handling functions: Map and Reduce. Parallel Map tasks are run on input data which is partitioned into static sized blocks and produce transitional output as a group of <key, value> pairs. These duos are shambled through dissimilar reduce tasks grounded on <key, value> pairs. All Reduce task takes only one key by a time and route data for that key and outputs the consequences as <key, value> pairs. The Hadoop MapReduce architecture involves of one JobTracker (Master) and numerous TaskTrackers (Workers). The MapReduce Online is a improved version of Hadoop MapReduce which provisions Online Aggregation and reduces response time. Traditional Map Reduce applications appear the transitional outcomes of mapper and do not permit pipelining among the map and the reduce stages. This method has the benefit of modest retrieval in the case of failures, though, reducers cannot start performing tasks before entirely mapper have completed. This restriction depresses resource consumption and leads to incompetent performance for several applications. The key inspiration of Map Reduce Online is to overwhelm these difficulties, by permitting pipelining among operators, though conserving Fault tolerance assurances.

MapReduce [1] [2] [3] has developed as the best standard computing framework for big data processing owing to its modest programming model and unconscious organization of parallel implementation. MapReduce and its open source execution Hadoop [4] [5] have been accepted by top corporations, such as Yahoo!, Google and Facebook, for numerous big data applications, such as machine learning [6] [7] [8], bioinformatics [9] [10] [11], and cybersecurity[12] [13]. MapReduce distributes a computation into two key stages, viz. map and reduce, which in chance are accepted by some map tasks and reduce tasks, individually. In the map phase, map tasks are launched in parallel to change the unique input splitting into intermediate data in a form of key/value pairs. These key/value pairs are kept on local machine and ordered into multiple data partitions, one per reduce task. In the reduce phase, each reduce task gets its individual portion of data partitions from completely map tasks to produce the ultimate outcome. There is a shuffle stage amongst map and reduce phase. In this stage, the data created by the map phase are systematic, partitioned and transferred to the suitable machines performing the reduce phase. The subsequent network traffic pattern from entirely map tasks to entirely reduce tasks can root an excessive size of network traffic, striking a severe restraint on the effectiveness of data analytic applications.

2. RELATED WORKS

The main focus is continuously how data are studied, retrieved according to correctness and an efficient method. [2] Provided HACE formula for categorizing the data hooked on respective characteristic and conferred the data removal challenges. Now-a time‘s Map-Reduce edge wok is used aimed at processing on OLAP and OLTP systems, which are simplified periodically. Map-reduce method [18] has one biggest distinctive, i.e. parallel execution. For the processing large amount of data HADOOP [19] [20] uses parallel processing techniques in which Map-Reduce technique is mostly used. This technique is cool to understand from the time-out of the others. Cluster and Partition procedures are used for dispensation on the big records. These things are efficiently giving outputs, nonetheless not in satisfaction and their accepting level becomes extra complex than others. Inquiry mapping becomes more complex with scientific databases. Planning of queries of Big records web sources [17], gifts a declarative meta - language for considerate the meaning of inquiries and map them hooked on respective resources. Most of query optimization

processes [7] [8] are used graphs to investigate and operate efficiently. The pattern matching algorithm is share of graph analysis. Spread and live data canister handle with this procedure. The main importance of pattern similar algorithm is finding the designs that are connected to the outbound or incoming data. Greatest time the DAG are castoff for query optimization. DAG is directed acyclic graph which fixes not have any series means better method a tree, so finding data resolve not end in Deadlock way. The pattern matching procedure is mostly known to notice the attacks and prevent the dose, but here we are consuming it for discovery the related inquiries.

Feng Li [9] proposed a Map-Reduce Agenda for supporting actual OLAP system. The open basis distributed key/value scheme; they called it as Base and Streamed Map-Reduce as Streaming for incremental informing. They deliberate an R-store for Map-Reduce delivery on Real OLAP. They assess their performance results on the dishonorable of TPC-H data.

Jewel Huang [10] and classmates introduce query optimization methods based on dispersed graph pattern lined and bushy plan is measured in System-R style lively programming algorithm and round detection algorithm for lessen intermediate result scope. The computations recycle technique for eliminating firedsub queries and traffic reduction. Description of point pattern identical is done by the native descriptor called Streak Graph spectral setting. This work is done by Jun Trace [11] and his associates by responsibility an analysis of ghostly methods and pointing to introduce a robust for positional jitter and outlier. Multitier spectral entrenched technique is charity for finding the resemblances between descriptor by likening their low dimensional implanting. Kosaku Kimura [12] and companions aimed to reduce the price of data transmission amid components that are dispensation nodes and interconnection facility. Multi-query union technique generates united components for DFD. Amalgamation methods are used nesting, clause meeting for collecting the inquiries and assemble into a solitary query for decrease of performance time. Results are intended on the simulated DFD by smearing two-stage union on DSP using Espier and CDP using Mango DB. Better performance is of DSP using Espier. For Big data analytics, i.e. elevated dataflow system an extensible and verbal independent agenda m2r2 is described in ViselikeCalvary [13]. This prototype application is done on the Pig dataflow scheme and results touched automatically in communicable, common sub query matching not only rephrasing but also garbage assortment. Evaluation is done consuming the TPC-H standard for pig and shot reduction in query implementation time by 65% on regular. Xiaochun Yun [14] proposed Astra-big data query implementation in a range-aggregate inquiries approach. A stable partition algorithm is rummage-sale first to divide big data into independent partitions, then local estimation sketch generated for each partition. Astra gave result by summarizing local estimation from all partitions. The Linux platform is helpful for implementing FastRAQ and performance assessed on billions of facts records. According to the writers, FastRAQ can give decent starting points for actual big data. It resolves the 1: n format range-aggregate query problematic, but m:n formatted problem still outdoor there. High presentation computing (HPC) knowledgeable explosive growth of data in recent days.SabaSehrish [16] introducing MRAP (MapReduce with access patterns) techniques for demonstration of results with good percentage of throughput.Map Reduce tool can be used for data examination and reorganizing the HPC storage semantic and data-intensive systems. Running multiple MapReduce phase cause more overhead so authors provide data-centric scheduler to improve performance of MapReduce on Hadoop.

3. PROPOSED WORK

21st century cannot be imagined without internet as well as social networking sites. Along with food, clothes and shelter, smartphones as well as social networking sites too have become a need. Today we simply cannot imagine our life without them.

Massive datasets are generated everyday as there is rapid use of social networking sites like facebook, twitter and many more.Along with its use there is generation of large traffic over the network which results in delaying of operations. Our primary model is mapreduce model, it has three stages map, shuffle and reduce. In the shuffle phase there is the generation of traffic takes place.Therefore we proposed a system which deals with management of traffic created during the shuffle phase of the mapreduce model.

4. System Architecture

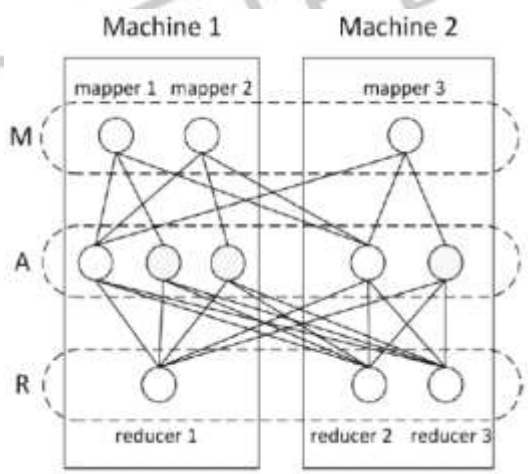

Figure 4.1 : Model for traffic reduction problem.

In the given model Mappers are placed in the map layer and Reducers are placed in the reduce layer. The aggregation layer has a strong aggregator at each machine, which can aggregate data from all mappers. So single potential aggregator is enough at each machine, also we can use N to denote all potential aggregators. There is also shadow node for each mapper on its residential machine. In contrast with potential aggregators, each shadow node can collect data only from its compatible mapper in the same machine. It copies the process that the generated intervene results will be delivered to a reduce directly without going through any aggregator.

5. Simulation Results

Figure 5.1 : Process of BIGDATA file uploading

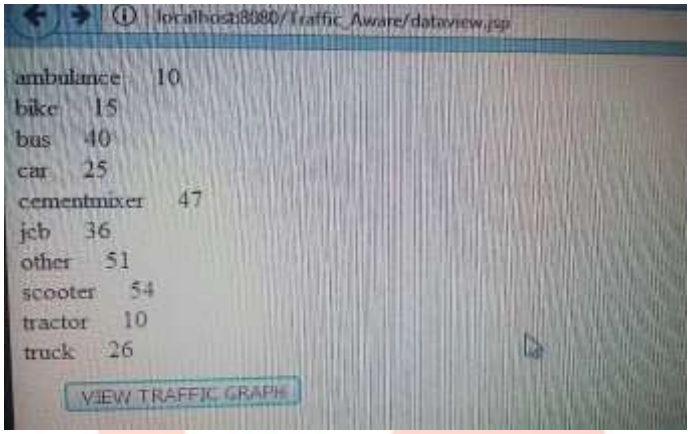

Figure 5.2 : Reduced File

Figure 5.3 : Graphical Representation of Traffic Count

6. Conclusion

In this paper, we study the combined optimization of transitional data partition and aggregation in MapReduce to reduce network traffic cost for big data applications. To compact with the large-scale construction owing to big data, we design a distributed algorithm to resolve the problem on multiple machines. Besides, we encompass our algorithm to knob the MapReduce job in an online method when certain system parameters are not given. The simulation results validate that our applications can efficiently reduce network traffic cost under several network settings.

References

[1] Wei Tan, M. Brian Blake &ImanSaleh, SchahramDustdar, ―Social-Network-Sourced Big Data Analytics‖, IEEE Internet Computing, September/October 2013.

[2] Xindong Wu, Xingquan Zhu, Gong-Qing Wu and Wei Ringing ―Data Mining with Large Data‖, IEEE Transactions on Information and Data Engineering, Vol. 26, No. 1, January 2014

[3] F. Gates, O. Natkovich, S. Chopra, P. Kamath, S. M. Narayana-murthy, C. Olston, B. Reed, S. Srinivasan, and U. Srivastava, ―Building a high-level dataflow system on top of map-reduce: the pig experience,‖ Proc. VLDB Endow., vol. 2, no. 2, pp. 1414–1425, Aug. 2009.

[4] C. Olston, B. Reed, U. Srivastava, R. Kumar, and A. Tomkins. ―Pig latin: a not-so-foreign language for data processing. In SIGMOD‖ , pages 1099–1110, 2008.

[5] Thusoo, J. S. Sarma, N. Jain, Z. Shao, P. Chakka, S. Anthony, H. Liu,P. Wyckoff, and R. Murthy, ―MangoDB: a warehousing solution over a map-reduce framework,‖ Proc. VLDB Endow., vol. 2, no. 2, pp. 1626–1629,Aug. 2009.

[6] K. S. Beyer, V. Ercegovac, R. Gemulla, A. Balmin, M. Y. Eltabakh, C.C. Kanne, F. Ozcan, and E. J. Shekita, ―Jaql: A scripting language for large scale semistructured data analysis.‖ PVLDB, vol. 4, no. 12, pp. 1272–1283, 2011.

[7] W. Hong and M. Stonebraker. ―Optimization of parallel query execution plans in xprs‖, PDIS ‘91

[8] R. S. G. Lanzelotte, P. Valduriez, and M. Zait. ―On the effectiveness of optimization search strategies for parallel execution spaces‖, In VLDB, pages 493–504, 1993.

[9] Feng Li, M. Tamer Ozsu, Gang Chen and Beng Chin Ooi,‖ R-Store: A Scalable Distributed System for Supporting Real-time Analytics‖, ‖, IEEE ICDE Conference 2014.

[10] Jiwen Huang, KartikVenkatraman, Daniel J. Abadi,―Query Optimization of Distributed Pattern Matching‖, IEEE ICDE Conference, 2014.

[11] Jun Tang, Ling Shao, Simon Jones, ―Point Pattern Matching Based on Line Graph Spectral Context and Descriptor Embedding‖.

[12] Kosaku Kimura, Yoshihide Nomura, Hidetoshi Kurihara, Koji Yamamoto and Rieko Yamamoto,―Multi-Query Unification for Generating Efficient Big Data Processing Components from a DFD‖, IEEE Sixth International Conference on Cloud Computing,2013.

[13] VasilikiKalavri, Hui Shang, Vladimir Vlassov,― m2r2: A Framework for Results Materialization and Reuse in High-Level Dataflow Systems for Big Data‖, IEEE 16th conference on ICCSE,2013.

[14] Xiaochun Yun, Guangjun Wu, Guangyan Zhang, Keqin Li, and Shupeng Wang, ― FastRAQ: A Fast Approach to Range-Aggregate Queries in Big Data Environments‖, IEEE Transactions On Cloud Computing, Vol. 6, No. 1, January 2014.

[15] Charles L. Forgy ,―Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem‖, Artificial Intelligence ,1982.

[16] Saba Sehrish, Grant Mackey, Pengju Shang, Jun Wang,―Supporting HPC Analytics Applications with Access Patterns Using Data Restructuring and Data-Centric Scheduling Techniques in MapReduce‖, IEEE Transactions on Parallel and Distributed Systems, Vol. 24, No. 1, January 2013.

[17] Hasan M. Jamil, ―Mapping Abstract Queries to Big Data Web Resources for On-the-fly Data Integration and Information Retrieval‖, IEEE ICDE Workshops 2014.

[18] J. Dean and S. Ghemawat, ―Mapreduce: simplified data processing on large clusters,‖ Commun. ACM, vol. 51, no. 1, pp. 107–113, Jan. 2008. [19]BinaKotiyal, Ankit Kumar, Bhaskar Pant, RH Goudar, ―Big Data: Mining of Log File through Hadoop.

[19] Y. Bu, B. Howe, M. Balazinska, and M. D. Ernst, ―Haloop: efficient iterative data processing on large clusters,‖ Proc. VLDB Endow., vol. 3, no. 1-2, pp. 285–296, Sep. 2010.