International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 5, Issue 6, June 2018)

A Novel Approach of Data Partition and Aggregation in Big Data

Mohd Irshad Khan[1], Dr. Amit Sinhal[2], Dr. Rakesh Bhujade[3]
[1]M.Tech Scholar, Department of Information Technology, TIT, RGPV, Bhopal,India;
[2]Head of Department, Department of Information Technology, TIT, RGPV, Bhopal,India;
[3]Assistant Professors, Department of Information Technology, TIT, RGPV, Bhopal,India;

***Abstract* –Map Reduce and its open source implementation Hadoop have been adopted by leading companies, such as Yahoo!, Google and Facebook, for various big data applications, such as machine learning bioinformatics and cyber security. In the existing system, study the joint optimization of intermediate data partition and aggregation in MapReduce to minimize network traffic cost for big data applications. The existing system presents a three-layer model for this problem and formulates it as a mixed-integer nonlinear problem, which is then transferred into a linear form that can be solved by mathematical tools. In the proposed system proposed two algorithm distributed algorithm and online algorithm for reducing the network traffic cost and Mapreduce.**

***Keywords:* MapReduce, Network Traffic, Big Data, Online algorithm, Distributed algorithm, Hadoop**

I. INTRODUCTION

Big Data has been described by some as a "cultural movement" that allows us to discover how humans behave and interact with each other and the world. While the exact description is arguable, there is no doubt that Big Data has become one of the main driving forces for both science and industry in the current century. The analysis of large data sets has allowed businesses to understand their users' behavior and based on these findings, personalize experience, recommend relevant products, or display ads that might be of interest; it has enabled artificial intelligence to perform complex tasks that have long been thought to be impossible; and it has facilitated major scientific breakthroughs such as the discovery of the Higgs Boson[1].

Besides the new opportunities that Big Data provides, it also poses new challenges to the systems that store and process data. Traditional architectures based on monolithic mainframes are unable to cope with the rapidly increasing amounts. Their vertical scalability (scale up) model makes them expensive to scale as to store and process more data, more powerful hardware is required but data is growing faster than Moore's Law.

This has caused companies to build large-scale data centers based on off-the-shelf servers instead of small mainframe clusters. Rather than vertically, these data centers scale horizontally (scale out), which means that more storage and processing capacity can be added by adding more servers. Servers are based on commodity hardware and hence scaling becomes highly costeffective.

The adoption of MapReduce and similar high-level programming models was encouraged by another concurrent development: a new paradigm for software deployment known as cloud computing. The aim of cloud computing is to turn computing power into a common utility that is always available on demand and in abundant supply, much like electrical power delivered through the electricity grid. This long-held dream has recently become economically viable, with theconstruction and operation of extremely large-scale, commodity-computer data centers at low-cost locations [2] [3] [4].

To tackle the problem of high network usage, incurred by the traffic-oblivious partition scheme, in this worktake into account of both task locations and data size associated with each key in the project. By assigning keys with larger data size to reduce tasks closer to map tasks, network traffic can be significantly reduced. To further reduce network traffic within a MapReduce job, in this work consider to aggregate data with the same keys before sending them to remote reduce tasks. By aggregating the data of same keys before sending them reducers reduce network traffic as shown in the Fig. 1.

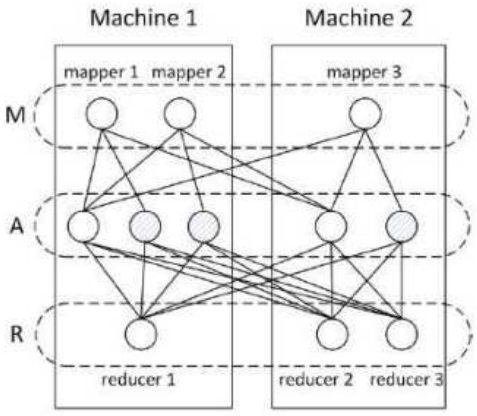

Fig.1 Three layer model for Traffic Aware optimization

II. NETWORK CONGESTION DURINGPARTITION/AGGREGATE QUERIES

The partition/aggregate pattern is challenging for the network because during the aggregation step, all worker nodes send their partial results to the few aggregator node(s) (see Figure 2a). As can be seen from the figure, this creates congestion at the master node if its incoming bandwidth is less than the combined outgoing bandwidth of the workers. If the

International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 5, Issue 6, June 2018)

network is oversubscribed, this congestion already happens earlier at the different layers of the data centre network. As a result, worker nodes have to reduce their sending rate and are not able to fully utilize the available outgoing bandwidth [5].

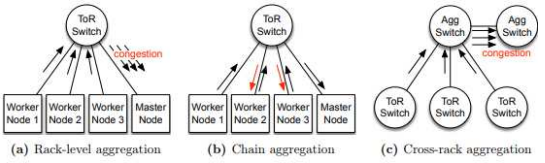

Fig.2 Different forms of edge-based aggregation

Edge-based aggregation helps to reduce network traffic during partition/aggregate queries. However, its main drawback is that its performance is still limited by the inbound bandwidth of the aggregator. For example, assuming 40 servers per rack and 1 Gbps edge network links, the maximum transmission rate per worker is only approximately 25 Mbps. d-ary trees can eliminate some of the shortcomings of rack-level aggregation but introduce new challenges. First, small values of d increase the depth of the tree. As data now has to traverse more hops before reaching the final aggregator, latency increases. Second, small values of d increase intra-rack bandwidth usage because now, incoming links of workers, as opposed to only outgoing links are used to move data. This can drastically reduce the performance of other network flows that cannot be aggregated [6].

A fundamental drawback of any edge-based aggregation approach is that it only applies to intra-rack traffic and not traffic in the core of the network. As shown in Figure 1.3c, if aggregation computation spans multiple racks, the links between aggregation switches can become a bottleneck, especially in the presence of oversubscription. A network-aware application could exploit the tree topology of modern data centre networks to execute additional partial aggregation steps at intermediate hops to further reduce the data.

III. RELATED WORK

A system similar to MapReduce uses CamCube's distinct properties for attaining high performance. Camdoop [7] running on cam-cube has higher performance than Camdoop running over a traditional switch (proven using a small prototype). In majority of the cases Camdoop running over a Camcube has a higher performance, even when current clusters attained full bisection bandwidth. This is because packets on the path are aggregated resulting in significantly decreasing the network traffic. In big data centers, one of the important frame works for data processing [8] that has emerged is MapReduce. MapReduce is an algorithm of consisting of three phases: Map, Reduce and Shuffle. As MapReduce systems are largely deployed, many projects to improve its performance have been in place and they look at one among three phases to enhance performance. The main storage component of the Hadoop framework is Hadoop Distributed File System (HDFS). HDFS is framed for processing and maintaining the huge datasets efficiently among cluster nodes. The computation infrastructure of Hadoop has to cooperate with MapReduce [9], the data has to be uploaded to HDFS from local file systems. However when data size is large, this upload procedure takes more time, causing delay for key tasks. This project named Zput mainly proposes a faster mechanism to upload data that uses the approach of metadata mapping and can greatly improve uploading.

Recent times have seen advancement of technologies with high throughput generating large amounts of data i.e. biological data requiring interpretation and analysis [10]. Of late, to decrease the data complexity and also to make it easier to interpret data, nonnegative matrix factorization (NMF) is used. Different fields in biological research use it. This project comes up with CloudNMF, which is an open source & distributed NMF implantation over a framework of MapReduce. Based on experimental studies, CloudNMF could deal with large data, is scalable [11].

IV. PROPOSED METHODOLOGY

In this work study to reduce network traffic cost for a Map Reduce job by designing a novel intermediate data partition scheme. Furthermore, jointly consider the aggregator placement problem, where each aggregator can reduce merged traffic from multiple map tasks. A decomposition-based distributed algorithm is proposed to deal with the large-scale optimization problem for big data application and an online algorithm is also designed to adjust data partition and aggregation in a dynamic manner. Finally, extensive simulation results demonstrate that our proposals can significantly reduce network traffic cost.

Algorithm:

1. Set t=1, and $v_j^p (j \in A, p \in P)$ to arbitrary nonnegative values
2. For t < T do
3. Distributively solve the subproblem SUB_DP and SUB_AP on multiple machines in a parallel manner;
4. Update the values of v_j^p with the gradient method and send the results to all subproblems;
5. T = 1 and $\hat{t}$ =1;
6. Solve the OPT_ONE_SHOT problem for t =1;
7. While t ≤ T do
8. If $\sum_{T=\hat{t}}^{t} \sum_{p \in P} C_t^p(T) > \gamma C_M(\hat{t})$ then
9. Solve the following optimization problem:

$$\min \sum_{p \in P} C^P(t)$$

For time slot t.

10. If the solution indicates a migration event then
11. Conduct migration according to the new solution;
12. $\hat{t}$ =t;

International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 5, Issue 6, June 2018)

13. Update $C_M(\hat{t})$
14. Splitting step takes input Datasets from Source and divides into smaller Sub-Datasets.
15. Mapping step takes those smaller Sub-Datasets and perform required action or computation on each Sub-Datasets
16. Shuffle Function Output = List of <key, List< value>> pairs
17. Reduce Function Output = List of <key, Value> Pairs
18. End

III.1. Distributed algorithm

Distributed algorithmic rule is planned for large information applications by decomposing the initial large-scale drawback into many sub problems and these sub problems may be resolved in parallel manner. Another is on-line algorithmic program that is additionally designed to affect the information partition and aggregation in a very dynamic manner. In this section, develop a distributed algorithm to solve the problem on multiple machines in a parallel manner. Our basic idea is to decompose the original large-scale problem into several distributive solvable sub problems that are coordinated by a high-level master problem.

III.2. Online algorithm

Until now, take the data size and data aggregation ratio as input of our algorithms. In order to get their values, in this work need to wait all mappers to finish before starting reduce tasks, or conduct estimation via profiling on a small set of data. In practice, map and reduce tasks may partially overlap in execution to increase system throughput, and it is difficult to estimate system parameters at a high accuracy for big data applications. These motivate us to design an online algorithm to dynamically adjust data partition and aggregation during the execution of map and reduce tasks. In this section, divide the execution of a Map Reduce job into several time slots with a length of several minutes or an hour.

III.3. Bayesian Algorithm

The Bayesian algorithm is a very simple learning algorithm. This is MapReduce Algorithm. It is also known as "Combine Function". Merging step combines all key-value pairs which have same keys (that is grouping key-value pairs by comparing "Key").Sorting step takes input from Merging step and sort all key-value pairs by using Keys. This step also returns <Key, List<Value>> output but with sorted key-value pairs.

V. RESULTS

To deal with the large-scale formulation due to big data, in this work design a distributed algorithm to solve the problem on multiple machines. Furthermore, extend our algorithm to handle the Map Reduce job in an online manner when some system parameters are not given. The simulation results demonstrate that our proposals can effectively reduce network traffic cost under various network settings. The simulation results are shown below:

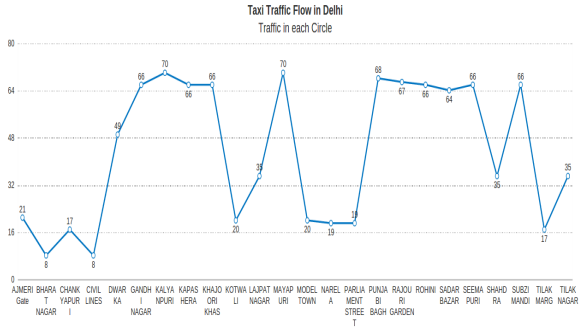

Fig.2 Line graph of traffic flow

Fig.2 shows Line graph of traffic flow. In this figure x axis show circles and y axis show number of taxies. In this figure circles shows the traffic. There is different traffic in different places shows in this figure.

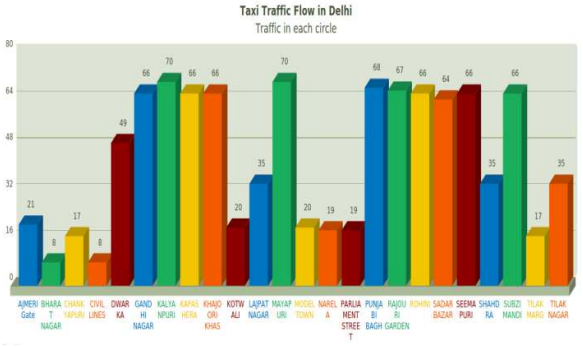

Fig.33D Bar graph of traffic flow

Fig.3 shows 3D Bar graph of traffic flow. In this figure x axis show circles and y axis show number of taxies. In this figure traffic shows in 3D bar chart in different places.

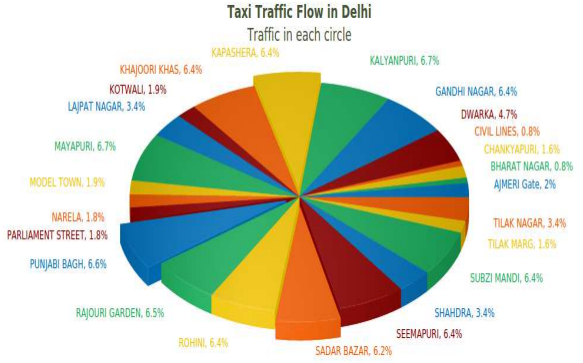

Fig.4 3D Pie graph of traffic flow

Figure 4 shows 3D Bar graph of traffic flow. In this figure x

International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 5, Issue 6, June 2018)

axis show circles and y axis show number of taxies. In this figure traffic shows in 3D bar chart in different places.

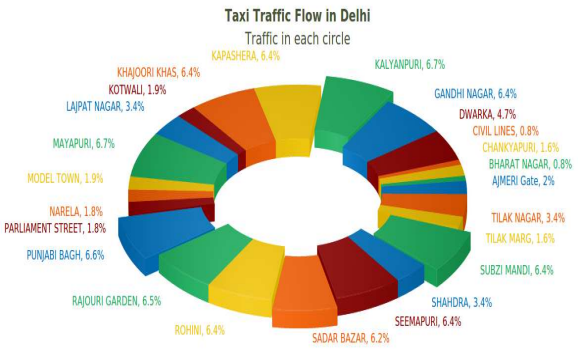

Fig.53D D" hunt graph of traffic flow

Fig.5 shows 3D D''hunt graph of traffic flow. In this figure traffic shows in percentage in different places. This is pie chart.

Table 1 Comparison of Accuracy

S.No.	Work	Data	Accuracy in %
1.	Previous	352.43/ 58.49	83.40%
2.	Proposed	1038 /24	97.68%

VI. CONCLUSION

The proposed MapReduce traffic-aware partition suffers from partition skew issue; where the output of map tasks is unevenly disseminated among reduce tasks. In above systemlook into so that can reduce network traffic cost for a MapReduce job by designing a novel intermediate data partition scheme. Then after, together consider the aggregator placement drawback, when aggregator will reduce merged traffic from multiple map tasks in data. Then decomposition-based distributed algorithmic rule is apply for optimized large data set and reduce map. The partition and aggregators help to add to distance aware routing for processing the data for the big data applications. Placing the aggregators as close to the nodes and the client would also add to the network traffic reduction and in turn helps to reduce the cost of the data processing.

REFERENCES

[1] Ke, Huan, Peng Li, Song Guo, and Minyi Guo. "On traffic-aware partition and aggregation in mapreduce for big data applications." IEEE Transactions on Parallel and Distributed Systems 27, no. 3 (2016): 818-828.

[2] Hsueh, Sue-Chen, Ming-Yen Lin, and Yi-Chun Chiu. "A load-balanced mapreduce algorithm for blocking-based entity-resolution with multiple keys." In Proceedings of the Twelfth Australasian Symposium on Parallel and Distributed Computing-Volume 152, 2014 pp. 3-9.

[3] Fan, Liya, Bo Gao, Xi Sun, Fa Zhang, and Zhiyong Liu. "Improving the load balance of mapreduce operations based on the key distribution of pairs." arXiv preprint arXiv: (2014) 14-35

[4] Ms. Varalakshmi M N , Mrs. Shashirekha H "On Traffic-Aware Partition and Aggregation in Map Reduce for Big Data Applications", International Digital Library For Education & Research Volume 1, Issue 3, Mar 2017.

[5] Reddy, Y. Dinesh, and A. Pio Sajin. "An efficient traffic-aware partition and aggregation for big data applications using map-reduce." Indian Journal of Science and Technology 9.10 (2016).

[6] Murali, S. "ON TRAFFIC-AWARE PARTITION AND AGGREGATION IN MAP REDUCE FOR BIG DATA APPLICATIONS."International Journal of Pure and Applied MathematicsVolume 117 No. 7 2017.

[7] Shweta Hegade ,Prof. Raghib Nasri "A Study of Traffic Aware Partition and Aggregation in MapReduce for Big Data Applications", International Journal of Innovative Research in Computer and Communication Engineering Vol. 5, Issue 5, May 2017.

[8] R. Dhanalakshmi , S.Mohamed Jakkariya , S. Mangaiarkarasi "Aggregation Methodology on Map Reduce for Big Data Applications by using Traffic-Aware Partition Algorithm", International Journal of Innovative Research in Computer and Communication Engineering Vol. 4, Issue 2, February 2016.

[9] Mewada, Khyati M., Amit Sinhal, and Bhupendra Verma. "Adaptive neuro-fuzzy inference system (ANFIS) based software evaluation." International Journal of Computer Science Issues (IJCSI) 10, no. 5 (2013): 244.

[10] Rajeshwari Adrakatti "Mapreduce For Big Data Processing Based On Network Traffic Performance", Special Issue, May.16 pp 287-292 ICCSTAR-2016.

[11] Mounika, Khambampati, And Nittala Swapna Suhasini. "An Efficient Network Traffic-Aware Partition for Big Data Applications and Aggregation Techniques using Map-Reduce."Vol.08,Issue.10, August-16, Pages:1951-1956: (2016).