IDL - International Digital Library Of
Technology & Research
Volume 1, Issue 2, Mar 2017

Available at: www.dbpublications.org

International e-Journal For Technology And Research-2017

# *On Traffic-Aware Partition and Aggregation in Map Reduce for Big Data Applications*

Ms. Varalakshmi M N [1], Mrs. Shashirekha H [2]

*Dept. of Computer Science*

1 MTech, Student– VTU PG Center, Mysuru, India
2 Guide, Assistant Professor– VTU PG Center, Mysuru, India

## Literature Survey

**ABSTRACT:**

The MapReduce programming model simplifies large-scale data processing on commodity cluster by exploiting parallel map tasks and reduces tasks. Although many efforts have been made to improve the performance of MapReduce jobs, they ignore the network traffic generated in the shuffle phase, which plays a critical role in performance enhancement. Traditionally, a hash function is used to partition intermediate data among reduce tasks, which, however, is not traffic-efficient because network topology and data size associated with each key are not taken into consideration. In this paper, we study to reduce network traffic cost for a MapReduce job by designing a novel intermediate data partition scheme. Furthermore, we 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 under both offline and online cases.

**1. On Traffic-Aware Partition and Aggregation in MapReduce for Big Data Applications. IEEE Transactions on Parallel and Distributed Systems ( Volume: 27, Issue: 3, March 1 2016**

"The MapReduce programming model simplifies large-scale data processing on commodity cluster by exploiting parallel map tasks and reduce tasks. Although many efforts have been made to improve the performance of MapReduce jobs, they ignore the network traffic generated in the shuffle phase, which plays a critical role in performance enhancement. Traditionally, a hash function is used to partition intermediate data among reduce tasks, which,

 

however, is not traffic-efficient because network topology and data size associated with each key are not taken into consideration. In this paper, we study to reduce network traffic cost for a MapReduce job by designing a novel intermediate data partition scheme. Furthermore, we 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 under both offline and online cases."

**2. Improving Network Traffic In Mapreduce For Big Data Applications. Electrical, Electronics, And Optimization Techniques (ICEEOT), International Conference On 2015**

"Improving the performance of network traffic in shuffle phase is important to improve the performance of MapReduce. The goal of enhancement of network traffic is achieved by using partition and aggregation. According to traditional method a hash function is used to partition intermediate data among reduce tasks but the traditional function is not efficient to handle network traffic. A novel intermediate data partition scheme is designed to reduce network traffic cost in MapReduce. The aggregator placement problem is considered, 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 applications. Also an online algorithm is designed to adjust data partition and aggregation in a dynamic manner. Network traffic cost under both offline and online cases is significantly reduced as demonstrated by the stimulation results by the various proposal considered and used."

**3. An Efficient Network Traffic-Aware Partition for Big Data Applications and Aggregation Techniques using Map-Reduce.**



"The objective of this system to reduce network traffic cost for a Map-Reduce job by designing a novel intermediate data partition scheme. 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. Map Reduce is a desirable parallel programming platform that is widely applied in data processing fields. Hadoop provides the cloud environment and is the most commonly used tool for analyzing big data. K-Means and DBSCAN are parallelized to analyze big data on cloud environment. The limitation of parallelized K-Means is that it is sensitive to noisy data, sensitive to initial condition and forms fixed

shape while DBSCAN has an issue of processing time and it is more complex than K-Means. The implementation of the technique will be on top of Hadoop which will help to sample HDFS blocks uniformly. We will evaluate this technique using real world datasets and applications and we will try to demonstrate the system's performance in terms of accuracy and time. The objective of the proposed technique is to significantly improve the performance of Hadoop Map Reduce for efficient Big Data processing."

**4. MAPREDUCE FOR BIG DATA PROCESSING BASED ON NETWORK TRAFFIC PERFORMANCE, International Journal of Computer Engineering and Applications, ICCSTAR-2016, Special Issue, May.16**

"After the Map phase and before the beginning of the Reduce phase is a handoff process, known as shuffle and sort. Here, data from the mapped tasks is prepared and moved to the nodes where the reducer tasks will be run. When the mapped task is complete, the results are sorted by key, partitioned if there are multiple reducers, and then written to disk. The MapReduce programming model simplifies large-scale data processing on commodity cluster by exploiting parallel map tasks and reduces tasks. Although many efforts have been made to improve the performance of MapReduce jobs, they ignore the network traffic generated in the shuffle phase, which plays a critical role in performance enhancement."

**5. Aggregation Methodology on Map Reduce for Big Data Applications by using Traffic-Aware Partition Algorithm. Vol. 4, Issue 2, February 2016. International Journal of Innovative Research in Computer and Communication Engineering**

"Data clustering is an important data mining technology that plays a crucial role in numerous scientific applications. However, it is challenging to cluster big data as the size of datasets has been growing rapidly to extra large scale in the real world. Meanwhile, MapReduce is a desirable parallel programming platform that is widely applied in data processing fields. Hadoop provides the cloud environment and is the most commonly used tool for analyzing big data. K-Means and DBSCAN are parallelized to analyze big data on cloud environment. The limitation of parallelized K-Means is that it is sensitive to noisy data, sensitive to initial condition and forms fixed shape while DBSCAN has an issue of processing time and it is more complex than K-Means. Comprehensive analysis of these existing techniques has been carried out and appropriate clustering algorithm is provided. A hybrid approach based on parallel K-Means and parallel DBSCAN is proposed to overcome the drawbacks of both these algorithms. This approach allows combining the benefits of both the clustering techniques. Further, the proposed technique is evaluated on the MapReduce framework of Hadoop Platform. The results show that the proposed approach is an improved version of parallel K-Means clustering algorithm. This algorithm also performs better than parallel DBSCAN while handling clusters of circularly distributed data points and slightly

 

overlapped clusters. The proposed hybrid approach is more efficient than DBSCAN-MR as it takes less computation time. Also it generates more accurate clusters than both K-Means MapReduce algorithm and DBSCAN MapReduce algorithm."

**6. MAPREDUCE FOR BIG DATA APPLICATIONS TO ENHANCE THE NETWORK TRAFFIC PERFORMANCE. International Journal For Technological Research In Engineering Volume 4, Issue 4, December-2016**

MapReduce is a scheme for processing and managing large scale data sets during a distributed cluster, which has been used for applications such as document clustering, generating search indexes, access log analysis, and numerous other forms of data analytic. In existing system, a hash function is used to partition intermediate data among reduce tasks. During this project the system proposed a decomposition-based distributed algorithm to deal with the large-scale optimization problem for large data application and an online algorithm is additionally designed to adjust data partition and aggregation in a dynamic manner. Network traffic price under both offline and on-line cases is significantly reduced as demonstrated by the extensive stimulation results by the various proposals considered and used".

**7. OPTIMIZATION OF MAP REDUCE APPLICATIONS USING PARTITION AND AGGREGATION IN BIG DATA APPLICATION. wjert, 2016, Vol. 2, Issue 3, 179 -189 Research Article**

"Cloud computing, rapidly emerging as a new computation concept, offers agile and scalable resource access in a utility-like fashion, particularly for the processing of big data. An important open problem here is too effectively progress the data, from various geographical locations more time, into a cloud for efficient processing. Big Data introduces to datasets whose sizes are beyond the capability of typical database software tools to capture, accumulate, maintain and examined. The application of Big Data differs across verticals since of the several challenges that bring about the various use cases. With the increasing amount of data and the availability of high performance and relatively low-cost hardware, database systems have been extended and parallelized to run on multiple hardware platforms to manage scalability. Recently, a new distributed data processing framework called Map Reduce was proposed whose fundamental idea is to simplify the parallel processing using a distributed computing platform that offers only two interfaces. To further reduce network traffic within a Map Reduce job, we consider to aggregate data with the same keys before sending them to remote reduce tasks."

**OBECTIVES:**

***Map-Reduce algorithm:*** MapReduce is a programming model and an associated implementation for processing and generating large data sets with a parallel, distributed algorithm on a cluster. A MapReduce program is composed of a Map() procedure (method) that performs filtering and sorting (such as sorting students by first name into queues, one queue for each name) and

 

a Reduce() method that performs a summary operation.

***Online Algorithm:*** In computer science, an online algorithm is one that can process its input piece-by-piece in a serial fashion, i.e., in the order that the input is fed to the algorithm, without having the entire input available from the start. In contrast, an offline algorithm is given the whole problem data from the beginning and is required to output an answer which solves the problem at hand.

***Branch and Bound Algorithm:*** Branch and bound (BB or B&B) is an algorithm design paradigm for discrete and combinatorial optimization problems, as well as general real valued problems. A branch-and-bound algorithm consists of a systematic enumeration of candidate solutions by means of state space search: the set of candidate solutions is thought of as forming a rooted tree with the full set at the root. The algorithm explores *branches* of this tree, which represent subsets of the solution set.

***Mixed Integer Linear Programming:*** Mixed integer linear programming (MILP) involves problems in which only some of the variables are constrained to be integers, while other variables are allowed to be non-integers.

***Mixed Integer Non-Linear Programming:*** It refers to mathematical programming with continuous and discrete variables and nonlinearities in the objective function and constraints. The use of MINLP is natural approach of formulating problems where it is necessary to simultaneously optimize the system structure and parameters.

## CONCLUSION

Applying clustering algorithm to distributed DBs. Dynamic outcomes from different data sets. To store weather forecasting datasets into DBs, apply clustering and get aggregated outcomes according to parameter. Used Parallel distributed clustering algorithm. Weather forecasting data sets are used for reduction and aggregation. Aggregation and reduction outcomes are dynamically.

## REFERENCES

1. J. Dean and S. Ghemawat, “Map reduce: simplified data processing on large clusters,” Communications of the ACM, 2008; 51(1): 107–113.

2. W. Wang, K. Zhu, L. Ying, J. Tan, and L. Zhang, “Map task scheduling in map reduce with data locality: Throughput and heavy-traffic optimality,” in INFOCOM, 2013 Proceedings IEEE. IEEE, 2013, pp. 1609–1617.

3. F. Chen, M. Kodialam, and T. Lakshman, “Joint scheduling of processing and shuffle phases in map reduce systems,” in INFOCOM, 2012 Proceedings IEEE. IEEE, 2012, pp. 1143–1151.

4. Y. Wang, W. Wang, C. Ma, and D. Meng, “Zput: A speedy data uploading approach for the hadoop distributed file system,” in Cluster Computing (CLUSTER), 2013 IEEE International Conference on. IEEE, 2013, pp. 1–5.

5. T. White, Hadoop: the definitive guide: the definitive guide. ” O’Reilly Media, Inc.”, 2009.

6. S. Chen and S. W. Schlosser, “Map-reduce meets wider varieties of applications,” Intel Research Pittsburgh, Tech. Rep. IRP-TR-08-05, 2008.

7. J. Rosen, N. Polyzotis, V. Borkar, Y. Bu, M. J. Carey, M. Weimer, T. Condie, and R. Ramakrishnan,

 

"Iterative mapreduce for large scale machine learning," arXiv preprint arXiv:1303.3517, 2013.

8. S. Venkataraman, E. Bodzsar, I. Roy, A. AuYoung, and R. S. Schreiber, "Presto: distributed machine learning and graph processing with sparse matrices," in Proceedings of the 8th ACM European Conference on Computer Systems. ACM, 2013, pp. 197–210.

9. A. Matsunaga, M. Tsugawa, and J. Fortes, "Cloudblast: Combining mapreduce and virtualization on distributed resources for bioinformatics applications," in eScience, 2008. eScience'08. IEEE Fourth International Conference on. IEEE, 2008, pp. 222–229.

10. J. Wang, D. Crawl, I. Altintas, K. Tzoumas, and V. Markl, "Comparison of distributed data-parallelization patterns for big data analysis: A bioinformatics case study," in Proceedings of the Fourth International Workshop on Data Intensive Computing in the Clouds (Data Cloud), 2013.

11. R. Liao, Y. Zhang, J. Guan, and S. Zhou, "Cloudnmf: A mapreduce implementation of nonnegative matrix factorization for largescale biological datasets," Genomics, proteomics & bioinformatics, 2014; 12(1): 48–51. 12. G. Mackey, S. Sehrish, J. Bent, J. Lopez, S. Habib, and J. Wang, "Introducing mapreduce to high end computing," in Petascale Data Storage Workshop, 2008. PDSW'08. 3rd. IEEE, 2008, pp. 1–6. 13. W. Yu, G. Xu, Z. Chen, and P. Moulema, "A cloud computing based architecture for cyber security situation awareness," in Communications and Network Security (CNS), 2013 IEEE Conference on. IEEE, 2013, pp. 488–492.

14. J. Zhang, H. Zhou, R. Chen, X. Fan, Z. Guo, H. Lin, J. Y. Li, W. Lin, J. Zhou, and L. Zhou, "Optimizing data shuffling in dataparallel computation by understanding userdefined functions," in Proceedings of the 7th Symposium on Networked Systems Design and Implementation (NSDI), San Jose, CA, USA, 2012.