International Journal For Technological Research In Engineering
Volume 4, Issue 7, March-2017 *ISSN (Online): 2347 - 4718*

IMPROVING NETWORK TRAFFIC PERFORMANCE IN MAP REDUCE FOR BIG DATA APPLICATIONS USING ONLINE ALGORITHM IN DYNAMIC MANNER

B.Adithya Ram[1], S.Vamshi Krishna[2]
[1,2]B.Tech, Department of CSE, Sreenidhi Institute of Science and Technology, Village Yamnampet, Mandal Ghatkesar, Dist Ranga Reddy, Telangana, India.

ABSTRACT: MapReduce may be a scheme for process and managing large scale information sets throughout a distributed cluster, which has been used for applications like document clustering, generating search indexes, access log analysis,and numerous alternative types of information analytic. In existingsystem, a hash operate is used to partition intermediatedata among reduce tasks. Throughout this project the systemproposed a decomposition-based distributed algorithmic rule todeal with the large-scale improvement drawback for bigdata application and an internet algorithm is in additiondesigned to regulate information partition and aggregation during adynamic manner. Network traffic value under each offlineand on-line cases is considerably reduced as incontestable by the intensive stimulation results by the varied proposalsconsidered and used.

I. INTRODUCTION

Big data could be a term that refers to information sets or combinations ofdata sets whose size (volume), quality (variability), andrate of growth (velocity) create them tough to be captured,managed, processed or analyzed by standardtechnologies and tools, like relative databases. HadoopMapReduce programming model is being employed for processBig Data that consists of information process functions: Mapand Reduce. Parallel Map tasks are run on input file that ispartitioned into fixed sized blocks and manufacture intermediateoutput as a group of <key, value> pairs. These pairs areshuffled across completely different reduce tasks supported < key, value>pairs. Every reduce task accepts only 1 key at a time andprocess information for that key and outputs the results as < key,value> pairs. The Hadoop MapReduce design consistsof one JobTracker (Master) and lots of TaskTrackers(Workers). The MapReduce on-line could be a changed version ofHadoop MapReduce that supports on-line Aggregation andreduces time interval. Traditional Map reduceimplementations happen the intermediate results ofmapper and don't permit pipelining between the map and therefore thereduce phases. This approach has the advantage of easyrecovery within the case of failures, however, reducers cannotstart execution tasks before all mapper have finished. Thislimitation lowers resource utilization and results in inefficientexecution for several applications. The most motivation ofMap reduce on-line is to beat these issues, byallowing pipelining between operators, whereas protectiveFault tolerance guarantees. Redis is an ASCII text file,networked, in-memory, key-value information store with electivedurability. It’s written in ANSIC.The name Redis means that Remote dictionary Server. In itsouter layer, the Redis information model could be a dictionary that mapskeys to values. one amongst the most variations between Redisand alternative structured storage systems is that Redis supportsnot only strings, however additionally abstract information varieties like lists ofstrings, sets of strings (collections of non-repeating unsortedelements), sorted sets of strings (collections of non-repeatingelements ordered by a number known as score),hashes wherever keys and values are strings. The sort of a pricedetermines what operations (called commands) are out therefor the worth itself. Redis supports high-level, atomic, serversideoperations like intersection, union, and distinctionbetween sets and sorting of lists, sets and sorted sets; Themain goal of the project work is to implement on-lineMapReduce and Redis on the highest of the Hadoop, which willimprove the performance of Hadoop for economical huge informationprocessing.

II. RELATED WORK

Most existing work focuses on MapReduce performanceimprovement by optimizing its information transmission. Blancaetalhave investigated the question of whether or not optimizingnetwork usage will result in higher systemperformance andfound that prime network utilization and low networkcongestion ought to be achieved at the same time for employment withgood performance. Palanisamyetal have bestowed Purlieus, aMapReduce resourceallocation system, to enhance theperformance of MapReduce jobs within the cloud by locating intermediate data to the local machines or close-by physicalmachines.This locality-awareness reduces network trafficinthe shuffle section generated within the cloud information center.However,little work has studied to optimize networkperformance ofthe shuffle method that generates largeamounts of knowledge trafficin MapReduce jobs. A criticalfactor to the networkperformance within the shuffle phaseis the intermediate informationpartition. The default schemeadopted by Hadoop is hashbasedpartition that will yield unbalanced loads amongreduce tasks because of itsunawareness of the information size associatedwith every key.To overcome this defect, Ibrahietalhavedeveloped a fairness-aware key partition approach thatkeeps track of the distribution of intermediatekeys’frequencies, and guarantees a good distribution amongreduce tasks. Meanwhile, Liyaetal have designedanalgorithm to schedule operations supported the keydistribution of intermediate key/value pairs to improvetheload balance. Larsetal have planned and evaluated 2effective load equalization approaches to dataskew handling forMapReduce-based entity resolution.Unfortunately, all abovework focuses on load balance eat reduce tasks, ignoring thenetwork traffic throughout the shuffle section.In addition to informationpartition, several efforts are created on local aggregation,in-mapper combining and in-network aggregation to reducenetwork traffic withinMapReduce jobs. Condieetal haveintroduced a combiner perform that reduces the number ofdata tobe shuffled and incorporated to reduce tasks. designer and dyer have planned an in-mapper combining scheme byexploitingthe fact that mapper will preserve state across the processof multiple input key/value pairs and defer emission ofintermediate information till all input records are processed.Both proposals are constrained to one map task, ignoringthe data aggregation opportunities from multiple map tasks.Costaetal have planned a MapReduce-like system todecrease the traffic by pushing aggregation from the stinginto thenetwork. However, it will be only applied to thenetwork topology with servers directly connected to differentservers,which is of restricted sensible use. Completely different fromexisting work, we have a tendency to investigate network traffic reductionwithin MapReduce jobs by together exploiting traffic-awareintermediate information partition and information aggregation amongmultiple map tasks.

III. FRAME WORK

Map phase and reduce phase like models are wide used toprocess “Big Data”. Applications supported suchmodels placeheavily data-dependency or communication on VirtualMachines; so network traffic becomes the bottleneckof jobs. The subsequent 3 sections of information exchange at intervalsthe execution technique of an application supported MapReduce model. This paper purpose to the provisioning a virtualcluster in line with the position relationship between VirtualMachines so on decrease the network traffic andimprove the performance of Map section and Map reducephase like applications instead of modifying the workscheduling ways or Virtual Machine configurations. Byoptimizing the design of virtual clusters, cloud userswill get a lot of economical platform with constant resourcerequest and price, and cloud suppliers can get a far betterresource utilization magnitude relation. The foremost contributions of thispaper unit summarized below throughout this system tomeasure the affinity by method the space of a virtualcluster the shorter the space, the nearer the virtual cluster.The shortest distance disadvantage is given to get theclosest virtual cluster to resolve the shortest distance drawbackby formulating it into variety applied math. Aheuristic Virtual Machine placement rule is suggests provisioning a virtual cluster. It’s designed for MapReduceapplications to boost the shuffle speed and stimulate theexecution. It’s additionally optimize, the virtual cluster from theglobal, i.e., provisioning virtual clusters for letter of invitation queuerather than one request. The online heuristic Virtual Machineplacement algorithmic rule and also the international optimization algorithmic ruleare compared by simulations. The previous has lower timecomplexity whereas the latter arrival shorter average distancefor multiple requests to research the performance of ourapproach through experiments. Within the experiment, describethe support completely different virtual cluster architectures to seedifferent MapReduce applications. 2 metrics ofapplication runtime and cluster compatibility show the efficiency of virtual cluster optimization. The subsequentfigure a pair of showed the straightforward Map reduce task exploitation key withaggregation.

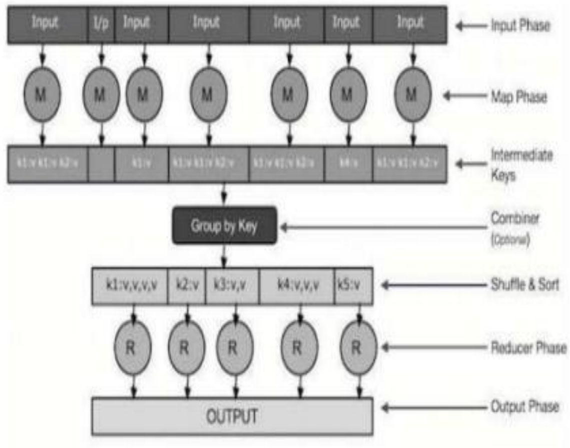
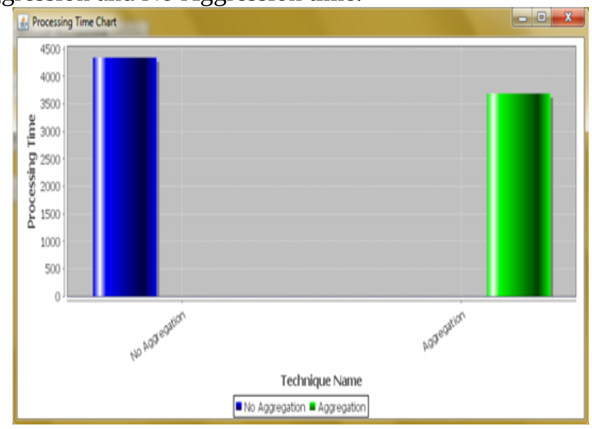


Figure: MapReduce Processing

The system projected a distributed algorithmic rule for bigdataapplications by moldering the initial large-scaleproblem into many sub problems which is able to besolved inparallel The system investigate network traffic reductionwithin Map reduce jobs by conjointly exploiting traffic-awareintermediate information partition and information aggregation amongmultiple map tasks. It offers computers as physical or a lot ofoften as virtual machines. A cluster of virtual machines,virtual cluster, is commonly requested as a platform for users to run parallel or distributed applications like Map reduce anddryad applications. Thus on get high turnout, fastresponse, load balance, low cost, and low value, several topicson virtual machines configuration, virtual machinesplacement, virtual machines consolidation, and virtualmachines migration are explored. The configuration of avirtual cluster includes a very important impact on the executionof applications running thereon as results of the physical nodeswhere virtual machines are situated are connected in manyways. For instance, some nodes are situated within the same rackwhereas others in many racks through a slow link. Map andreduce tasks might partly overlap below some cases butthe execution is to extend system turnout, and it isdifficult to estimate system parameter set a high accuracy forbig information applications. An internet algorithm to dynamicallyadjust information partition and aggregation throughout the execution ofmap and reduce tasks is thus intended. The essential arrange of thisalgorithm is to defer the migration operation until thecumulative traffic value exceeds a threshold. Another is on-linealgorithm that is in addition designed to take care of theinfo partition and aggregation in an exceedingly very dynamic manner. Thesimulation results finally incontestable advocate that ourproposals can considerably reduce network traffic price inboth offline and on-line cases.

IV. EXPERIMENTAL RESULTS

In this paper we have represented a map reduce frame workused for multiprocessing large amount of information. Althoughthere are several existing works that are focused onthe traffic reduction, they failed to had best within the map reduceparadigm. Those works focused in the main on the map andreduce sections rather than concentrating on shuffle phase.They failed to commit to reduce the information traffic for theimprovement of the network traffic and therefore the data cost.Although the present works focused on the trafficimprovement they used the massive range of keys within thesystem to form a method additional complicated. During this paper, wehave enforced an economical system for map reduces jobsby information partition and aggregation for the big dataapplications.In the below chart we are able to observe that distinction betweenthe length of each Aggression and No Aggression time.



V. CONCLUSION

In this system, we look into so we will reduce network traffic cost for a MapReduce job by planning a novel intermediate information partition scheme. What is more, we jointly consider the person placement drawback, wherever every person will reduce merged traffic from multiple map tasks. A decomposition-based distributed formula is planned to deal with the large-scale improvement downside for large information application and an online algorithm is additionally designed to regulate information partition and aggregation during a dynamic manner. The partition and aggregators facilitate to feature to distance aware routing for process the information for the big information applications. Placing the aggregators as near the nodes and therefore the client would also increase the network traffic reduction and successively helps to reduce the price of the information processing.

REFRENCES

[1] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," Commun. ACM, vol. 51, no. 1, pp. 107–113, 2008.
[2] W. Wang, K. Zhu, L. Ying, J. Tan, and L. Zhang, "Map task scheduling in mapreduce with data locality: Throughput and heavytraffic optimality," in Proc. IEEE INFOCOM, 2013, pp. 1609–1617.
[3] F. Chen, M. Kodialam, and T. Lakshman, "Joint scheduling of processing and shuffle phases in mapreduce systems," in Proc. IEEE INFOCOM, 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 Proc. IEEE Int. Conf. Cluster Comput., 2013, pp. 1–5.
[5] T. White, Hadoop: The Definitive Guide: The Definitive Guide. Sebastopol, CA, USA: O'Reilly Media, Inc, 2009.
[6] S. Chen and S. W. Schlosser, "Map-reduce meets wider varieties of applications," Intel Res., Pittsburgh, PA, USA, Tech. Rep. IRP-TR-08-05, 2008.
[7] H. Lv and H. Tang, "Machine learning methods and their application research," IEEE Int. Symp. Intel. Info. Process. Trusted Comput. (IPTC), pp. 108–110, Oct. 2011.
[8] S. Venkataraman, E. Bodzsar, I. Roy, A. AuYoung, and R. S. Schreiber, "Presto: Distributed machine learning and graph processing with sparse matrices," in Proc. 8th ACM Eur. Conf. Comput. Syst., 2013, pp. 197–210.
[9] A. Matsunaga, M. Tsugawa, and J. Fortes, "Cloudblast: Combining mapreduce and virtualization on distributed resources for bioinformatics applications," in Proc. IEEE 4th Int. Conf. eScience, 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 Proc. 4th Int. Workshop Data Intensive Comput. Clouds, 2013, pp. 1–5.

