I J C T A, 9(17) 2016, pp. 8463-8472

A Review on Enhancing Map Reduce Performance with Data Locality in Heterogeneous Environment

K. Radha[*] and B. Thirumala Rao[*]

ABSTRACT

Map Reduce is an emerging tool to process the massive volume of data. Various techniques are proposed to improve the Map Reduce performance in heterogeneous environments. Some of the techniques are discussed in this paper, such as data prefetching mechanism, Data locality aware scheduling methods are proposed. To support the map tasks data locality, Next-K-Node Scheduling (NKS) procedure is proposed. Dynamic Data Placement policy (DDP) for map tasks of data locality to assign data blocks. This algorithm is based up on the distinct computing capacities of nodes to assign the data blocks, thereby improving the data locality and decreases the additional overhead to improve the Hadoop performance. Regarding the Grep, the DDP can improve up to 32.1% with the average improvement of 23.5 and regarding Word count, DDP can be improved up to 24.7% with the average improvement of 14.5%. Anyhow, its performance is increased because of additional network traffic. Data Locality attentiveness is focusing on the Reduce Tasks scheduling, which are near to the Map Tasks. CoGRS algorithm is used to increase, Intermediate data locality. Reduce tasks are scheduled to the neighborhood map tasks. For every reduce task, this algorithm computes by the respective centre of gravity associated map tasks. Fine partitioning techniques calculates the cost of partitioning data. Input data is splitted into fixed number of partitions. Dynamic Fragmentation calculates the workload for the nodes and also executes the Reduce tasks and splits the intermediate data locally. This paper presents various techniques to improve Map Reduce performance in Heterogeneous environments through Data Locality by partitioning the intermediate data at the Reducer side.

Keywords: Map Reduce, CoGRS, Data Locality, NKS, DDP

I. INTRODUCTION

Map Reduce has several features such as less expensive, reliability and scalability. Anyhow, due to the additional network traffic its performance is degraded [14]. Data is loaded is into HDFS [15]. Whenever user attaches the files to HDFS, they are segmented in the form of chunks with the ranges from 16MB to 64MB. Data Locality refers to the movement of computational tasks to the nodes. During this period, some of the intermediate data is produced. Therefore, to start the Reduce task which node is the best node it's not possible to decide. However, extra network traffic is produced during the transmission of intermediate data, if the Reduce tasks are placed at inappropriate machines. Map Reduce application contains three phases such as map, shuffle and reduce. Efficient scheduling of the reduce tasks for the resources is necessary for data-intensive applications whenever massive data is moved in between map and reduce tasks. To overcome this problem, most of the techniques are concentrating on data locality property for task scheduling. Data Locality issue causes due to low physical resource utilization in Non-Virtualized clusters and requires more power consumption [49]. For supporting the Data Locality and Better Cluster resource Utilization, Virtualized Clusters gives a possible solution [1]. Research on Reduce Task Scheduling is limited.

[*] Department of Computer Science and Engineering, KL University, Guntur, Andhra Pradesh, India, *E-mails: radha.klu13@gmail.com; thirumail@yahoo.com*

1.1. Network Traffic , Job Completion Time

LoNARS algorithm is used for Reduce Task Scheduling. This algorithm uses the factors into consideration such as network traffic and data locality. To reduce the delay in data access along with network traffic, Data Locality attentiveness is focusing on scheduling the reduce tasks, Ching-Hsien Hsu et al [2]. Network traffic evenly share the traffic on the entire network and reduces hot spots as shown in Table 1, Mingming Sun et al [46]. In comparison of other reduce task scheduling algorithms as shown in Table 1. LoNARS algorithms achieves 15% improvement in data shuffling time and Total job completion time is 4% improvement to save energy consumption, the amount of network traffic on switches is reduced by 15% [16]. To overcome the issues of big data processing map reduce programming framework is proposed. This framework splits the large datasets into smaller tasks and distributed them over a collection of computational nodes. Various solutions are proposed to optimize the data computation scheduling problem by sorting the data location in an order and computational tasks are always allocated to nodes [16]. As shown in Table 1, Ching-Hsien Hsu et al [15], Virtual Machine Mapping technique is proposed to Map Reduce performance in Heterogeneous Environment. It partitions the data dynamically before Map phase and in order to increase the resource usage in the Reduce phase; Virtual Machine Mapping technique is used.

Experimental Results are improving the Map Reduce performance of Data Locality, Energy Efficiency and Total Completion Time in terms of Execution Time. MRA++ is a new framework; during the data distribution, MRA++ considers the heterogeneity of nodes. To collect the information prior to data distribution, it establishes a training task. Anyhow there is delay in set up phase by the efficient algorithms gains 70% in performance in 10Mbps networks as depicted in Table 1, Julio C.S. Anjos et al [35]. As shown in Table 1, Miguel Liroz-Gistau et al [14], input data is divided into fixed number of partitions by the Fine Partitioning technique. To overcome this problem, the nodes are grouped according to the computational capabilities. Network latency and energy efficiency is improved in the heterogeneous environment. As shown in Table 1, Ching-Hsien Hsu et al [19], Virtual Machine Mapping technique is proposed to Map Reduce performance in Heterogeneous Environment. To maximize the resource utilization, it partitions the data dynamically before Map phase. Experimental Results are improving the Map Reduce performance Data Locality, Energy Efficiency and Total Completion Time in terms of Execution Time. MRA++ is a new framework; it takes the consideration into heterogeneous nodes' data during the data distribution. A training task is completed to collect the information prior to the data distribution. Anyhow there is delay inset up phase that gains more performance 70% in 10Mbps networks as depicted in Table 1, Julio C.S. Anjos et al [35].As shown in Table 1, Miguel Liroz-Gistau et al [22], Fine Partitioning technique is used to prevent the problem of increase in execution time.

1.2. Energy Efficiency, Network Latency

Nodes are grouped according to the computational capabilities .Network latency and energy efficiency is improved in the heterogeneous environment. As shown in Table 1, Miguel Liroz-Gistau et al [22], Dynamic Fragmentation divides the intermediate data locally in the execution time. It also reduces the data transfer and Network Traffic in the Map Reduce shuffle phase. CoGRS reduces the network traffic with average 38.6 in Amazon EC2 on private cloud cluster respectively. Job execution time is improved up to 23.8%as shown in Table 1, Miguel Liroz-Gistau et al [22], Data partitioning technique is proposed to minimize the Data transfer and Network Latency in the Heterogeneous Environment. As shown in Table 1, Miguel Liroz-Gistau et al [22], MR-Part technique is proposed to minimize the data transfer between the Mappers and Reducers in the intermediate stage. Reducing the data transfer in Map Reduce's intermediate stage is mandatory. If network is slower, largest impact is there in data locality on execution time. Job execution time and network latency is reduced in the heterogeneous environment. Job Scheduling is a critical issue in Map Reduce that affects the Hadoop framework performance. To optimize the data locality during the job scheduling a delay occurs.

Table 1
Comparison of Data Locality Related techniques with respective to Various Parameters

			Cost Estimation Parameters							
S. No.	*Authors*	*Technique*	*Execution Time*	*Network Latency*	*Data Transfer*	*Data Locality*	*Network Traffic*	*Energy Efficiency*	*Heterogeneity*	*Performance Improvement/Pros*
1	Ching-Hsien Hsu et al [2]	Virtual Machine Mapping	Y				Y	Y	Y	Y-To Map Reduce performance in Heterogeneous Environent
2	Julio C.S. Anjos et data al [25]	MRA++	Y					70%	Y	Yes-Collect the information prior to distribution
3	Miguel Liroz-Gistau et al. [11]	Fine Partitioning	Y	Y				Y	Y	Yes-Tobalancethe processing load in Reduce phase
4	Miguel Liroz-Gistau et al[11]	Dynamic Fragmentation	Y		Y		Y	Y	Y	Yes- reduces data transfer in Map Reduce shuffle phase
5	Miguel Liroz-Gistau et al[11]	Data Partitioning		Y	Y				Y	Yes-to minimize the Data transfer
6	Miguel Liroz-Gistau et al[11]	MR-Part	Y	Y	Y				Y	Yes-to minimize the data transfer in shuffle phase
8	Seo, S et al[15]	Data prefetching	Y	Y	Y	Y				Yes-minimizes the execution time
9	Mingming Sun et al [39]	HPSO	Y			Y			Y	Yes- improves the data locality for Map Reduce jobs
10	Mingming Sun et al [39]	Next-K-Node Scheduling				Y			Y	Yes-to improve the data locality

Delay scheduling algorithm performs well in Hadoop and records low execution time as depicted in Table 1, Krishan Kumar Sethi et al [56]. As shown in Table 1, Seo, S et al [42], to improve the overall performance in heterogeneous environment, Prefetching and pre-shuffling optimization schemes are proposed. HPMR is used to minimize the execution time, three different workloads and different test sets are taken from Yahoo. If the map tasks don't have input data it causes significant data access delay due to task scheduling policy and cluster resource competition. Data Locality is considerable parameter. It affects Map Reduce clusters performance. In-memory prefetching improves the data locality. To use the prefetching effectively, HPSO (High Performance Scheduling Optimizer), improves the data locality for Map Reduce jobs as depicted in the Table 1, Mingming Sun et al [32]. Map Reduce real time scheduling framework improves the Map Reduce performance in Heterogeneous Environment with the factors taken into

consideration such as Data Locality, Network Traffic, Energy Efficiency, Network Latency, Execution Time and Data Transfer as shown in Table 1, Yu-ChonKao et al [46]. When transferring the massive data from computational machines, it leads to low data localization. Because, HDFS is manages the data placement tasks. Two operations are used to transfer the data between compute units. Delay scheduling algorithm is used by Matei et al. [5] to improve the map tasks data locality. Map task scheduling improvement is completely different from shuffle section improvement. Intermediate phase improvement consists of the many parts to think about and therefore creating the matter harder. Then, reduce tasks receives the shuffle phase data which carries for future computations to generate the results. The task on which the data is stored locally is referred to as local task and the node is called as local node otherwise it is referred to as a remote task and node is called as remote node. Locality means running the tasks on local nodes. Some of the nodes may be highly congested and some other nodes may be idle. Hence, balance between the load balancing and data locality is necessary in Map Reduce [18]. Data locations affect on different data nodes. Preliminary results are showing that multiple replica copies reduce the network data transfer improve the performance. When the number of replica is greater than one [19], data nodes process more data replica on the local machine.

Partition problem in Reduce tasks for scientific applications is addressed [8]. Qutaibah Althebyan, Omar ALQudah, Yaser Jararweh et al 2015 proposed [21] a technique to enhance the overall system performance through Multi-Threading Based Map reduce task scheduler. This approach improves the cluster usage by partitioning the cluster into blocks. System scheduling is improved by the Multithreading scheduling. For each block data locality is improved with fair share. Due to the association among run time blocking for non pre-emptive data locality for non pre-emptive execution, Real Time scheduling trouble is complex. Data-Locality-Aware Map Reduce real-time scheduling framework is proposed to overcome the above mentioned problems and to guarantee the QoS for relative Map Reduce applications [47]. Reduce tasks scheduling near the map tasks that generate their input [23, 24]. Transferred data is reduced throughout the intermediate stage. Dynamically splitting the intermediate keys to balance the load among the network transfers and reduce tasks MR-Part technique minimizes the data transfer among the mappers and reduce tasks within the intermediate stage of map reduce. This technique partitions the input data to connected partitions for the next map reduce jobs. Repartitioning techniques are employed to improve the data locality reduce tasks. An algorithmic in [24], every key value pair includes a fairness locality score. Every key is processed severally during a greedy algorithmic rule. For every key, key frequencies organized in dropping order for the candidate nodes (frequencies nodes have the simplest data locality). To allot a partition for each intermediate key, some modifications are in map reduce framework like the partitioning operate is modified.

1.3. Load Balancing, Data Locality

Master node is allocating the shuffled keys to the reduce phase on data, to improve the performance of load balancing and data locality, [26, 46], Pre-shuffling scheme is proposed to shrink the data transfers in intermediate stage. Reduce tasks are allocated to reduce the network transmission between the nodes and racks [23]. To enhance the data locality in heterogeneous environments, A collection of VM Placement and Data placement techniques are proposed [24, 48]. Map reduce partitioning function allocates intermediate keys to reduce tasks and all these jobs are limited. Allocation of intermediate keys to the reducers at scheduling time problem is addressed in [25]. Good data locality result in high performance and reduces cross network traffic. Greedy approach is proposed to create a cluster of VMs. Network is shared in a heterogeneous environment. Hence, every map reduce job bandwidth will be different [34]. Where as in cloud environment multiple users run their map and reduce jobs simultaneously. In a heterogeneous environment, bandwidth availability for every map reduce job .Hence, the task allocation should be network aware [24]. The data is stored in physical machines and allocation of VMs also on the same node.This will enhance the data

locality. In the Cloud Computing, allocation of more number of VMs on same physical nodes is possible according to request of users. Various classes of Virtual Machine pools are created and allocated on request [22]. By determining the computational node locations on various constraints and to optimize the data access latencies an optimal algorithm is used [19].

To enhance the performance and to dynamically reduce or increases the computational capacity of nodes, A method called locality-aware task scheduling is proposed [30]. Alternatively, increasing the computational potential whilst executing tasks may not be an convenient venture and it's unimaginable in physical nodes that host extra quantity of VMs [31] highlighted the problems to be faced and the prior outcome which might be regarding the progress of data-intensive functions are distributed through cloud data core. LoNARS takes network traffic and data locality into consideration. Furthermore, to save the energy consumption, traffic on the switches is reduced by 15% [33]. To increase the intermediate data locality, CoGRS algorithm is proposed by Hammoud et al. it schedules the reduce tasks nearby map tasks. LARTS acquires the data locality. Poor system utilization is avoided, scheduling delay [34]. Throughout the data distribution, job control and task scheduling, MRA++ considers the shared nodes. In [36], performance problems are addressed in Map Reduce in shared environments, e.g. Amazon EC2.Zaharia et al found out that there are areas, some considerations over the simplification of the Map Reduce model.

1.4. Job Completion Time, Response Time

This might generate an additional variety of speculative tasks. To improve the map reduce applications performance in heterogeneous environments· [37] proposed allocation of data according to the capability of the machines [37] uses massive input data to differentiate the execution times, in their performance analysis experiments. In distinction, [36] outlines, SAMR (Self adaptive Map Reduce) is associate adaptive scheduler of LATE currently that changes the improvement parameters for every node. Virtual Machine mapping technique improves the performance of Map Reduce Total Completion Time in terms of Reduce Task Time, Map Task Time, and Job Completion Time as depicted in Table 1, Ching-Hsien Hsu et al [14]. As shown in Table 1, Miguel Liroz-Gistau et al [22], Fine Partitioning technique overcomes the problem of large data increase in (Reduce Task) execution time. MR-Part technique is proposed to reduce the Map Task Execution Time, Reduce Task Execution Time, Response Time and Job Completion Time, as shown in Table 1, Miguel Liroz-Gistau et al [22]. As shown in Table 1, Seo, S et al [26]. Three different workloads and different test sets from On Yahoo! Grid platform, HPMR minimizes the execution time up to 73%. Increasing the utilization of Map Reduce cluster is a challenging issue in state-of-the art Cloud cluster systems. Data transfer delay is customarily on the execution time of map phase, even as map phase dominates the execution time of Map Reduce jobs. HPSO is reinforcing efficiency for Map Reduce cluster. Data-Locality-Aware Map Reduce real Time Scheduling Framework improves the Execution Time in terms of Ma Task Execution Time, Reduce Task Execution Time,

Response Time and Job Completion time of in Heterogeneous as shown in Table 1, Yu-ChonKao et al [47]. Unlike, MRA++ doesn't need a history of executions throughout the set up phase, before the work execution. MRA++ emits sending the data to the nodes with low speed, which might later be differentiated by the system. [38] Solves the partitioning drawback of the reduce tasks in scientific applications that shows the features that the present Map Reduce systems were not designed. Data center network aware load balancing in shuffle introduce map reduce is examined for the primary time. Effective solutions are shown for each of them (network flow and load balancing), that yields an entire resolution towards close to the optimum data center aware load balancing [49].

1.5. Next –K-Node Scheduling Method

Table 1.1
Comparison of Execution Time Parameters for Data Locality

S. Np.	*Authors*	*Algorithm*	*Technique*	*Execution Time Parameters*			
				Map Task Execution Time	*Reduce Task Execution Time*	*Response Time*	*Job Completion Time*
1	Ching-Hsien Hsu et [2]	Data Repartitioning	Dynamic Data Partitioning	Y	Y-44%		Y-29%
2	Ching-Hsien Hsu et [2]	VM Mapper	Virtual Machine Mapping		Y-14-23%		Y-41%
3	Miguel liroz-Gistau et al[11]	Assigning Partitions to Reducers	Fine Partitioning	Y	Y		
4	Miguel Liroz-Gistau et al [11]	Meta Data Combination	MR-Part	Y	Y	Y	Y
5	Ibrahim, S et al[14]	LEEN			Y	Y	
7	Seo, S et al [15]	Data prefetching		Y	Y	Y	Y
8	Mingming Sun et al [39]		HPSO	Y	Y		
9	Yu-Chon Kao et al[4]	Data Locality aware task partition, Reduce Task	Data Locality Aware Real time Scheduling	Y	Y	Y	Y

The rest of the paper is organized as follows. Related Work on Procedures to Enhance the Efficiency of Map Reduce in Heterogeneous and Homogeneous Environments are explained and Concluding.

II. RELATED WORK ON PROCEDURES TO ENHANCE THE EFFICIENCY OF MAP REDUCE IN HETEROGENEOUS AND HOMOGENEOUS ENVIRONMENTS

Data prefetching is applied while with data processing; as a result data transfer is overlie with data processing. There via Map Reduce job execution time can be lowered with no trouble. Overhead of data transference influences, effectivity Map Reduce in heterogeneous environments. The input data of map responsibilities is transferred in small slices nonetheless of multi operate block. As a map task will get a slice of the input data, it begins process the received data. Data Prefetching mechanism is proposed to overlap the data transmission process with data processing procedure with the data processing system. The overhead of data transmission is hidden when the input data of map tasks is not local. Consequently complete efficiency of Map Reduce may also be extended. Data placement approach can help the Map Reduce invariably by means of rebalancing the data within the course of nodes prior to participate in the data intensive software in a heterogeneous Hadoop cluster [3]. Some researchers occupied with optimizing project scheduling algorithms to expand the data locality in Map Reduce [8]. They've worked on simplest to increase the data locality in Map Reduce and they also may just expand the load balance complexity. To strengthen the Map

slash efficiency in heterogeneous atmosphere, LATE scheduling algorithm is proposed [9]. To maintain the clash between locality and equity in shared Map

Reduce clusters M. Zaharia et al., proposed an algorithm [5]. Modified task scheduling and computing potential are classification of computing nodes in Map Reduce task scheduling algorithm for deadline constraints in Hadoop Platform [12]. Scheduling with the dignity of data locality in homogeneous cluster is studied in [11]. DARE is an allotted adaptive data replication algorithm [14] in heterogeneous computational nodes, these nodes will get additional data replications. Data placement algorithms are applied in Hadoop's HDFS reminiscent of to preliminary data placement algorithm and data redistribution algorithm. (I). Preliminary Data Placement algorithm [13] divides a significant data into a wide variety of equal number of fragments. Distribution of file fragments for the period of the nodes of the cluster is managed through data distribution server. (II). in Data Redistribution, input data fragments are disbursed through the preliminary data placement algorithm. The preliminary data placement algorithm interrupted due to the fact that of the following explanations comparable to (a) New data is delivered to the existing input file, (b) from the present input file data blocks are deleted, (c) for existing cluster new data computational nodes are offered. Data redistribution algorithm is applied, to deal with this dynamic abilities load-balancing concern. Founded on the computing ratios data redistribution algorithm, reorganizes the file fragments.

Based on the computing ability of every node in a heterogeneous Hadoop cluster, this proposed procedure dynamically balance and adapt the data stored in every node [5]. Data locality aware scheduling procedure is proposed to enhance efficiency of the Data locality of Map Reduce in Heterogeneous environment. Two factors have an effect on the map duties execution efficiency such as waiting time and transmission time. The overhead of the transferred data is very massive. [6] Developed a procedure to enhance the efficiency of the Map Reduce in heterogeneous environments is to reduce the data movement between slow and speedy nodes in heterogeneous clustering. HDFS that allotted and saved a significant data set during a couple of heterogeneous nodes in accordance to the computing ability of every node. Data locality is principal component which influences the performance of Hadoop in a heterogeneous environment when the data required for performing a task is nonlocal. Dynamic Data Placement (DDP) for map tasks of data locality to assign data blocks. This algorithm is established up on the specific computing capacities of nodes to assign the data blocks thereby making upgrades to the data locality and reduces the additional overhead to improve the Hadoop efficiency. Involving the Grep, the DDP can increase as much as 32.1% with the common development of 23.5 and regarding the word count, DDP can be improved up to 24.7% with the normal development of 14.5% [5].

III. CONCLUSION

LoNARS algorithm is used for Reduce Task Scheduling. This algorithm uses the factors into consideration such as network traffic and data locality. Dynamic Data Placement coverage (DDP) for map tasks of data locality to assign data blocks. This algorithm is established up on the distinct computing capacities of nodes to assign the data blocks, thereby making improvements to the data locality and decreases the extra overhead to improve the Hadoop efficiency. Involving the Grep, the DDP can enhance as much as 32.1% with the common development of 23.5 and regarding word count, DDP may also be elevated up to 24.7% with the typical improvement of 14.5%. Reducing the data transfer in Map Reduce's intermediate stage is mandatory. If network is slower, largest impact is there in data locality on execution time. Job execution time and network latency is reduced in the heterogeneous environment. Job Scheduling is a critical issue in Map Reduce that affects the Hadoop framework performance. To improve performance of the data locality of Map Reduce in Heterogeneous environment, Data locality aware scheduling technique is proposed.

REFERENCES

[1] Qutaibah Althebyan,yaser Jararweh,Qussai yaseen,"Evaluating map reduce tasks scheduling algorithms over cloud computing infrastructure", Concurrency and Computation: Practice and Experience,2015 Volume 27, (18), pp 5686–5699.

[2] Ching-Hsien Hsu, Kenn D. Slagter , Yeh-Ching Chung, "Locality and loading aware virtual machine mapping techniques for optimizing communications in Map Reduce applications", Future Generation Computer Systems, Vol.53,2015 pp.43 -54, Copyright © 2015 John Wiley & Sons, Ltd.

[3] Xia Tang , Lijun Wang and Zhiqiang Geng, "A Reduce Task Scheduler for Map Reduce with Minimum Transmission Cost Based on Sampling Evaluation", International Journal of Database Theory and Application, Vol.8, No.1 (2015), pp.1 -10.

[4] Engin Arslan, Mrigank Shekhar, Tevfik Kosar, "Locality and Network-Aware Reduce Task Scheduling for Data-Intensive Applications", DataCloud '14 Proceedings of the 5th International Workshop on Data-Intensive Computing in the Clouds Pages 17-24.

[5] Zaharia, M., Borthakur, D., Sen Sarma, J., Elmeleegy, K., Shenker, S., and Stoica, I. Delay scheduling: A simple technique for achieving locality and fairness in cluster scheduling. In Proceedings of the 5thEuropean Conference on Computer Systems (New York,NY, USA, 2010), EuroSys '10, ACM, pp. 265-278.

[6] Weina Wang, Kai Zhu, Lei Ying, Jian Tan, Li Zhang, "Map Task Scheduling in Map Reduce with Data Locality: Throughput and Heavy-Traffic Optimality", IEEE/ACM TRANSACTIONS ON NETWORKING, Digital Object Identifier 10.1109/TNET.2014.2362745, 1063-6692 © 2014 IEEE.

[7] Yixian Yang,"Improve I/O performance and Energy Efficiency in Hadoop Systems", https://etd.auburn.edu/handle/10415/3353,August 4, 2012.

[8] B. Gufler, N. Augsten, A. Reiser, A. Kemper, Handling data skew in MapReduce, in: F. Leymann, I. Ivanov, M. van Sinderen, B. Shishkov (Eds.), CLOSER, 2011,pp. 574–583.

[9] Qutaibah Althebyan , Omar ALQudah, Yaser Jararweh, "Multi-Threading Based Map Reduce Tasks scheduling", 2014 5th International Conference on Information and Communication Systems (ICICS), APRIL 2014, DOI: 10.1109/IACS.2014.6841943, pp. 1-7.

[10] Xia Tang 1, Lijun Wang,et al ,"A Reduce Task Scheduler for MapReduce with Minimum Transmission Cost Based on Sampling Evaluation", International Journal of Database Theory and Application Vol.8, No.1 (2015), pp.1 -10 http://dx.doi.org/10.14257/ijdta.2015.8.1.01.

[11] Miguel Liroz-Gistau, Reza Akbarinia, et al, "Data Partitioning for Minimizing Transferred Data in Map Reduce", Proceedings 6th International Conference, Globe 2013, Data Management in Cloud, Grid and P2P Systems Prague, Czech Republic, August 28-29, 2013.

[12] Hammoud, M., Rehman, M.S., Sakr, M.F.: Center-of-gravity reduce task scheduling to lower mapreduce network traffic. In: IEEE CLOUD. pp. 49–58. IEEE (2012).

[13] Palanisamy, B., Singh, A., Liu, L., Jain, B.: Purlieus: locality-aware resource allocation for mapreduce in a cloud. In: Conference on High Performance Computing Networking, Storage and Analysis, SC 2011, Seattle, WA, USA, November 12 -18,2011. p. 58 (2011).

[14] Ibrahim, S., Jin, H., Lu, L., Wu, S., He, B., Qi, L.: LEEN: Locality/fairness-aware key partitioning for mapreduce in the cloud. In: Cloud Computing, Second International Conference, CloudCom 2010, November 30 - December 3, 2010, Indianapolis,Indiana, USA, Proceedings. pp. 17–24 (2010).

[15] Seo, S., Jang, I., Woo, K., Kim, I., Kim, J.S., Maeng, S.: HPMR: Prefetching and pre-shuffling in shared mapreduce computation environment. In: CLUSTER. pp.1–8. IEEE (2009).

[16] T.P. Shabeera,S.D. Madhu Kumar, Optimising virtual machine allocation in MapReduce cloud for improved data locality, Int. J. Big Data Intelligence, Vol. 2, No. 1, 2015

[17] Xu, F., Liu, F., Zhu, D. and Jin, H. (2014) 'Boosting MapReduce with network -aware task assignment', *Proceedings of CloudComp 2013*, China, Published in LNICST Springer 2014, pp.79–89.

[18] Palanisamy, B., Singh, A., Liu, L. and Langston, B. (2013) 'Cura: a cost optimized model for MapReduce in a cloud', *IEEE 27th International Symposium on Parallel & Distributed Processing (IPDPS)* , pp.1275–1286, IEEE.

[19] Alicherry, M. and Lakshman, T.V. (2013) 'Optimizing data access latencies in cloud systems by intelligent virtual machine placement', *Proceedings IEEE INFOCOM*, IEEE, pp. 647–655.

[20] Park, J., Lee, D., Kim, B., Huh, J. and Maeng, S. (2012) 'Locality aware dynamic VM reconfiguration on MapReduce clouds', *Proceedings of the 21st international symposium on High- Performance Parallel and Distributed Computing*, pp.27–36, ACM.

[21] Martino, B.D., Aversa, R., Cretella, G., Esposito, A. and Ko³odziej, J. (2014) 'Big data (lost) in the cloud',*International Journal of Big Data Intelligence*, Vol. 1, No. 1,pp.3–17, Inderscience.

[22] J. Tan, S. Meng, X. Meng, and L. Zhang, "Improving reduce task data locality for sequential mapreduce jobs," in *2013 Proceedings of IEEE INFOCOM*, Turin, Italy, 2013.

[23] Engin Arslan et al, "Locality and network-aware reduce task scheduling for data-intensive applications", Proceedings of the 5th International Workshop on Data-Intensive Computing in the Clouds, pages 17-24.

[24] Mohammad Hammoud, Majd F. Sakr, "Locality-Aware Reduce Task Scheduling for MapReduce", CLOUDCOM'11 Proceedings of the 2011 IEEE Third International conference on CloudComputing and Technology and Science,pp:570-576.

[25] Julio C.S. Anjos, et al, "MRA++: Scheduling and data placement on Map Reduce for heterogeneous environments", Future Generation Computer Systems, 42 (2015), pp: 22–35.

[26] M. Zaharia, A. Konwinski, A.D. Joseph, Y. Katz, I. Stoica, Improving MapReduce performance in heterogeneous environments, in: OSDI, 2008, pp. 29–42.

[27] J. Xie, S. Yin, X. Ruan, Z. Ding, Y. Tian, J. Majors, A. Manzanares, X. Qin, Improving MapReduce performance through data placement in heterogeneous Hadoop clusters, in: IEEE International Symposium on Parallel and Distributed Processing, Workshops and Ph.D. Forum, IPDPSW, 2010,pp. 1–9. http://dx.doi.org/10.1109/IPDPSW.2010.5470880.

[28] B. Gufler, N. Augsten, A. Reiser, A. Kemper, Handling data skew in MapReduce, in: F. Leymann, I. Ivanov, M. van Sinderen, B. Shishkov (Eds.), CLOSER, 2011, pp. 574–583.

[29] Yanfang Le, Feng Wang, et al ,"On Datacenter-Network-Aware Load Balancing in Map Reduce", 2015 IEEE 8th International Conference on Cloud Computing, June 27 2015-July 2 2015, 485 – 492, 10.1109/CLOUD.2015.71.

[30] B. Gufler, N. Augsten, A. Reiser, and A. Kemper, "Load balancing in map reduce based on scalable cardinality estimates," in *28th IEEE ICDE*, Washington, DC, USA, 2012.

[31] Chuang Zuo, Qun Liao,et al, "Node Capability Modeling for Reduce Phase's Scheduling in Map Reduce Environment", Second International Conference, CloudCom-Asia 20152015Cloud Computing and Big Data Volume 9106 of the series Lecture Notes in Computer Science, , Huangshan, China, June 17-19,pp 217-231, 10.1007/978-3-319-28430-9_17.

[32] Nguyen, P., Simon, T., Halem, M., Chapman, D., Le, Q.: A hybrid scheduling algorithm for data intensive workloads in a Map Reduce environment. In: Proceedings of the 5th International Conference on Utility and Cloud Computing, Chicago, IL, USA, 5–8 November 2012.

[33] Zhang, X., Zhong, Z., Feng, S., Tu, B., Fan, J.: Improving data locality of Map reduce by scheduling in homogeneous computing environments. In: Proceedings of the 9th International Symposium on Parallel and Distributed Processing with Applications, Busan, Korea, 26–28 May 2011.

[34] Tang, Z., Zhou, J., Li, K., et al.: A Map Reduce task scheduling algorithm for deadline constraints. Cluster Comput. 16(4), 651–662 (2013).

[35] Abad, C.L., Lu, Y., Campbell, R.H.: DARE: adaptive data replication for efficient cluster scheduling. In: Proceedings of IEEE International Conference on Cluster Computing, Austin, TX, USA, 26–30 September 2011.

[36] Berliñska, J., Drozdowski, M.: Scheduling divisible Map Reduce computations. J. Parallel Distrib. Comput. 71, 450–459 (2011).

[37] Wei Shi, Yang Wang, et al , "Smart Shuffling in Map Reduce: A Solution to Balance Network Traffic and Workloads", Conference: 8th IEEE/ACM International Conference on Utility and Cloud Computing (UCC 2015), At St. Raphael Resort, Limas sol, Cyprus,Jan7,2016,pp 1-10.

[38] Krishan Kumar Sethi, Dharavath Ramesh," Delay Scheduling with Reduced Workload on Job Tracker in Hadoop", Proceedings of the 6th International Conference on Innovations in Bio-Inspired Computing and Applications (IBICA 2015) held in Kochi, India during December 16-18, 2015, pp 371-381, 10.1007/978-3-319-28031-8_32.

[39] Mingming Sun, Hang Zhuang, Changlong Li, Kun Lu, Xuehai Zhou, "Scheduling algorithm based on prefetching in Map Reduce clusters ",14th International Conference, ICA3PP 2014, Dalian, China, August 24-27, 2014. Proceedings, Part II, pp 82-95, 10.1007/978-3-319-11194-0_7, volume 8631, LNCS, Springer.

[40] Yu-ChonKao et al, "Data-locality-aware map reduce real-timeschedulingframework"The Journal of Systems and Software, Volume 112, February 2016, Pages 65–77, Elsevier.

[41] Xiaohong Zhang, "Improving Data Locality of Map Reduce by Scheduling in Homogeneous Computing Environments", Proceedings of Parallel and Distributed Processing with Applications (ISPA), 2011 IEEE 9th International Symposium on, 26-28 May 2011, pp 120 – 126.

[42] K. Shyamala, T. Sunitha Rani ,"An Analysis on Efficient Resource Allocation Mechanisms in Cloud Computing", Indian Journal of Science and Technology,2015 May, 8(9), Doi no:10.17485/ijst/2015/v8i9/50180.

[43] A. Souvik Pal, B. Prasant Kumar Pattnaik, “Classification of Virtualization Environment for Cloud Computing”, Indian Journal of Science and Technology,2013 Jan, 6(1), Doi no:10.17485/ijst/2013/v6i1/30572.

[44] Moon Suk Yeon, Byeong Soo Jeong,”Multi-Level Load Balancing Methods for Hierarchical Web Server Clusters”, Indian Journal of Science and Technology,2015 Sep, 8(21), Doi no:10.17485/ijst/2015/v8i21/78469.

[45] R. K. Nadesh, Meduri Jagadeesh, M. Aramudhan, “A Quantitative Study on the Performance of Cloud Data Centre: Dynamic Maintenance Schedules with Effective Resource Provisioning Schema”, Indian Journal of Science and Technology,2015 Sep, 8(23), Doi no:10.17485/ijst/2015/v8i23/51867.

[46] M. Lavanya, Aarthi Ravi, Ajay Aditya, Rukmani Samyuktha, V. Vaithiyanathan, S. Saravanan,”An Enhanced Load Balancing Scheduling Approach on Private Clouds”,Indian Journal of Science and Technology,2015 Dec, 8(35), Doi no:10.17485/ijst/2015/v8i35/86650.

[47] N Balaji, A. Umamakeshwari ,”Load Balancing in Virtualized Environment - A Survey”,Indian Journal of Science and Technology,2015 May, 8(S9), Doi no:10.17485/ijst/2015/v8iS9/65583.