Resource Management in Cloud Data Centers

Aisha Shabbir[1], Kamalrulnizam Abu Bakar[2], Raja Zahilah Raja Mohd. Radzi[3]
School of Computing
University Technology Malaysia
Johor, Malaysia

Muhammad Siraj[4]
Department of Information Engineering
& Computer Science
University of Trento
Trento, Italy

***Abstract*—Vast sums of big data is a consequence of the data from different diversity. Conventional data computational frameworks and platforms are incapable to compute complex big data sets and process it at a fast pace. Cloud data centers having massive virtual and physical resources and computing platforms can provide support to big data processing. In addition, most well-known framework, MapReduce in conjunction with cloud data centers provide a fundamental support to scale up and speed up the big data classification, investigation and processing of the huge volumes, massive and complex big data sets. Inappropriate handling of cloud data center resources will not yield significant results which will eventually leads to the overall system's poor utilization. This research aims at analyzing and optimizing the number of compute nodes following MapReduce framework at computational resources in cloud data center by focusing upon the key issue of computational overhead due to inappropriate parameters selection and reducing overall execution time. The evaluation has been carried out experimentally by varying the number of compute nodes that is, map and reduce units. The results shows evidently that appropriate handling of compute nodes have a significant effect on the overall performance of the cloud data center in terms of total execution time.**

Keywords*—Big data; cloud data center; MapReduce; resource utilization

I. Introduction

Data sets that are so huge or complex that conventional data processing techniques are incapable to deal with them are called big data. The key sources of big data production are digital applications, social media, transactions, emails, sensors data and migration of almost every manual entity towards automation. The increasing number of challenges of big data are due to its diverse nature which is categorized by its V's [1]. As Big data is growing enormously so is its processing requirements. Consequently, it calls for requirements of huge computational infrastructure, in order to successfully analyze and process large amount of data. This is a two pronged challenge, on one hand the amount of data is constantly increasing and its allocation on suitable set of available resources and on the other hand need to yield the output in less time with minimum cost. To deal with ever growing data sets, the giants like Google, IBM, Microsoft and Amazon have ventured their concentration in cloud computing. They have offered various services based on cloud computing [2]. Cloud computing is a solution to perform large scale complex computing. It abolished the need for expensive hardware, software and devoted space. Cloud computing offers its users a platform which grants resources, services and applications. Mainly the cloud computing offers three different services; platform as a service (PaaS), software as a service (SaaS) and infrastructure as a service (IaaS). For the users, these services are easily accessible on pay per-Use-Demand [3].

Several frameworks have been proposed for processing big data. Some of widely used frameworks are Hadoop MapReduce, Dryad, Spark, Dremel and Pregel [4]. The most well-known framework is MapReduce. MapReduce is proposed by Google to simplify massively distributed parallel processing so that very large and complex datasets can be processed and analyzed efficiently. It is designed on the principle of exploiting the parallelism among the processing units. Popular implementation of the MapReduce framework is Hadoop and is used typically in conjunction with cloud computing, for executing various Big Data applications, including web analytics applications, scientific applications, data mining applications and enterprise data-processing applications [5].

The cloud data centers comprise of several compute nodes servers and storage nodes. Inappropriate handling of cloud data center resources will result in underutilization of the resources, high latency and computational costs. Thus, it would yeild the overall degradation of the system 'performance [6]. Existing researches are revolving around the improvement of resource management of cloud data centers by focusing more on scheduling of tasks on the relevant processers [7-9]. Some researchers tried to alleviate the communication cost of the data movement within the cloud data center [10-11]. While some researchers aiming at energy conservation for resources of the cloud data center [12-13]. Minimal attention has been given towards the optimization factor and some factors of framework used on servers has been explored [14].

However, there is lack of significant research upon the optimization of the resources of the of the cloud data center. In addition, the proper selection and distribution of compute nodes are not considered by research community. The important challenge is effective utilization of resources, with trifling computational cost, while skillfully allocating various assets of the data center to diverse tasks. This research focuses on the appropriate handling of the compute nodes of the cloud data center.

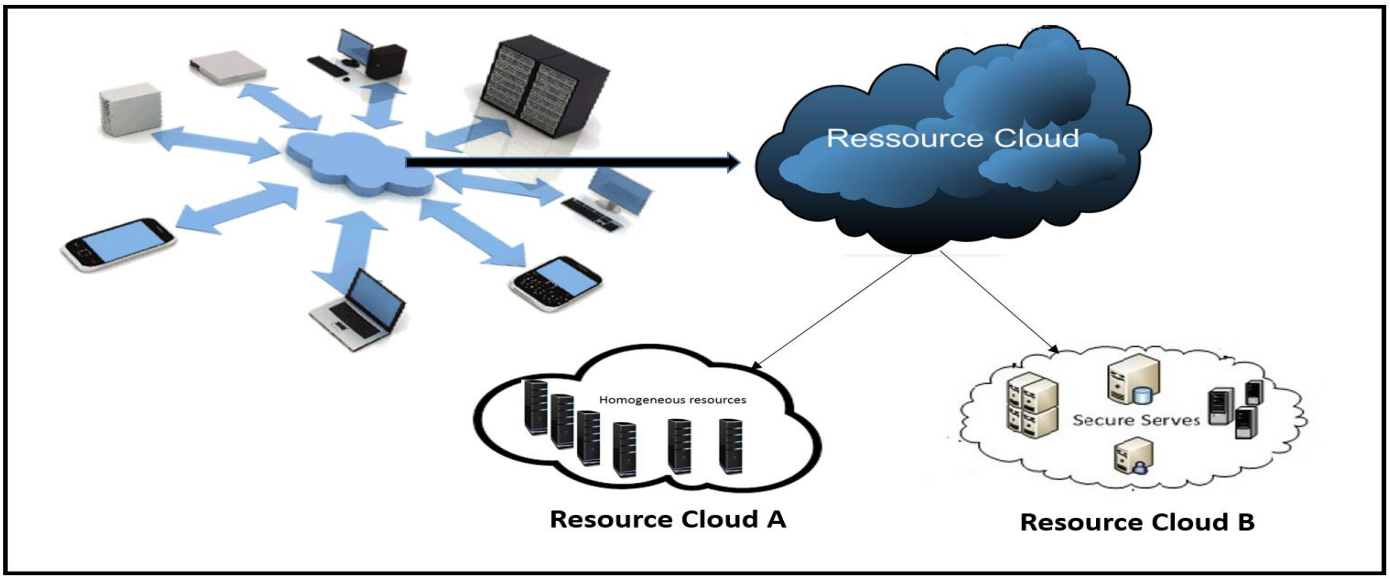

Fig. 1. Overview of Framework.

For illustration of the problem focused in this research study, a general framework is shown in Fig.1. When client submit their requests and the whole data size for processing from all requests is huge. Then, on the submission of this data, it will be divided further for efficient processing among the physical and virtual resources of the Cloud data center. As shown in Fig.1 the data is splitted among different resource clouds and for its data execution on the processing nodes, it will be further chunked and assigned to suitable processing machines according to the submitted task requirement. Thus, splitting the data among the compute nodes and proper parameter handling of the compute resources is necessary.

The organization of the paper is as follows: Section II describes the Preliminaries that is giving the overview of Big Data and its characteristics, enlightens the MapReduce Framework and Cloud Computing details. Section III comprises the Problem formulation. In Section IV, presents the Experimental Setup with the configuration details. Section V is about the Results and discussions. Last section that is, Section VI is of Conclusion and future work and following then are the acknowledgements and references of this study.

II. Preliminaries

A. Big Data

This vast sum of big data is a consequence of the data from different diversity that is, digital applications, scientific data, business transactions, social networking, emails and sensors data. The challenges of big data have been risen because of the 5V's characteristics of big data i.e. how to store, process, merge, manage and govern the different formats of data [15]. The brief detail of the V's are as follows.

- Volume shows the size of big data i.e. gigabyte, terabyte or zettabyte.
- Velocity presents the speed of data movement i.e. batch processing, stream processing or the real time processing.
- Variety represents the formats of the big data i.e. structured, unstructured and semi-structured.
- Veracity shows the quality of the big data, originality and authencity of the big data.
- Value presents the information extraction of big data, statics and hypothetical forms.

B. MapReduce Framework

MapReduce is proposed by Google to simplify massively distributed parallel processing so that very large and complex datasets can be processed and analyzed efficiently. Popular implementation of the MapReduce programming framework is Hadoop and is used typically in conjunction with cloud computing for executing various Big Data applications, including web analytics applications, scientific applications, data mining applications, and enterprise data-processing applications [16-17]. MapReduce is considered the most prominent and effective framework for the big data problems that allow the processing of gigantic data over many underlying distributed nodes. The MapReduce is composed of two basic components i.e., mappers and reducers. The basic concept is to design the map function to generate a set of the intermediate key-value pairs. The reducer is then used to merge all intermedia values associated with the intermediate key. The key feature of the MapReduce framework is that it invokes the parallelism among the computing nodes. The workflow of the MapReduce is shown in Fig.2. MapReduce computes a job in a way that it takes the big data sets for processing and chunked the huge data sets into small chunks and process it over the Map units. The output of the mappers is in the form of key-value pairs. This output has been forwarded to the reducers for further processing and the final out has been collected after the Reduce phase.

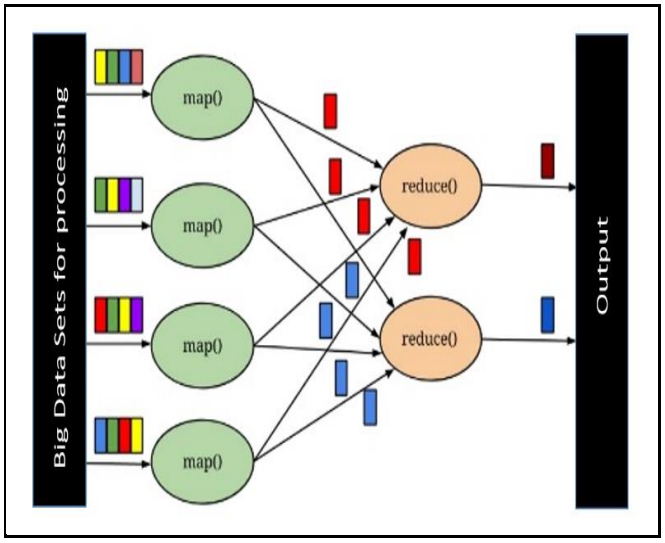

Fig. 2. MapReduce Workflow.

C. Cloud Computing

A newly emerged technology which allows the computations by providing such a platform which allows flexibility in efficient deployment, operation, and management environments. The core model of cloud computing is based on the distributed computing on massive scale, it provide services to the users in computing, networking and storage and the users can use them at their ease. For the organizations, cloud computing offers reliability, availability and scalability. The businesses can benefit from the cloud computing by reducing their cost of investing in business information systems and it can help them in increasing the resource utilization. With the passage of time, the Cloud services from providers like Google, Amazon and Microsoft have been better developed thus resulted in more and more companies diverting towards cloud computing platform. The main purpose of cloud computing model is provision of services, applications and sharing of resources to its users [18].

Big data and Cloud data centers mingled, as big data tasks needs cloud data centers' support for their processing [19]. Cloud data centers provides platforms with physical servers, applications, operating systems and virtual servers either virtually or physically. Cloud server provider can have highest level of resources as well. Traditionally the data centers have the applications which are overworked to deal with the heavy workload [20]. Modern cloud computing infrastructure has the ability to process user's requests at a much faster pace and this ability of analyzing and efficiently processing variety of data has attracted many large enterprises. Low latency results in online services leads to improved user satisfaction and revenue. If a single cloud data center has to process different tasks from different clients or users, then providing the efficient service to all tasks in minimum time with less hardware and computational cost is necessary.

III. Problem Formulation

The main aim of this study is to analyze and optimize the available resources of the cloud data center for the processing of big data sets. Let us consider a cloud data center with its compute resources as shown in Fig. 1. When tasks are submitted from different users to the computational platform for processing. The whole size of the input data tasks could be of certain value. It will be then forwarded to the further sub computational units as shown in Fig.1. Existing studies focused on the improvement of resource management of cloud data centers by encountering assignment of tasks to suitable processing units. However there is deficiency of critical research for the optimization of the resources of the cloud data center. In addition, the proper selection and distribution of compute nodes are underexplored. This research aims at exploring the potential of parameters optimization and suitable distribution of compute nodes for cloud data centers for big data sets processing.

As discussed earlier, MapReduce is becoming one of the most suitable and efficient framework adopted by many large enterprises. It works on principle of parallelism in order to reduce overall execution time and thus improving the performance of the computational platform. MapReduce computes a job mainly into two phases as by the name map and reduce. During the map phase, it splits the input and process it on the given set of nodes. The output of the map phase is in the form of key-value pairs. These key-value sets are stored on nearby machine and then forwarded to the reduce units. The workflow of MapReduce with its default component selection is shown in Fig. 3.

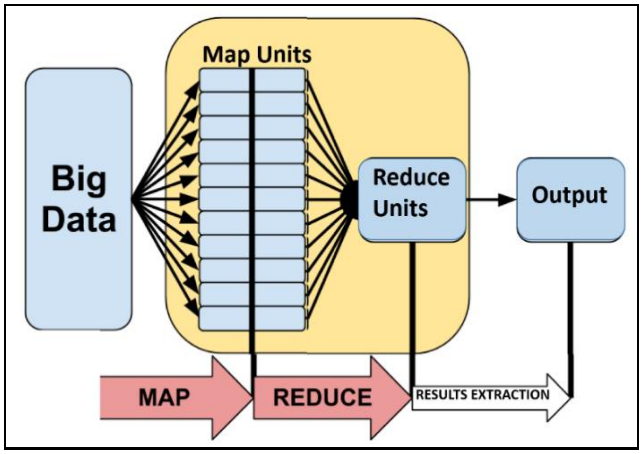

Fig. 3. MapReduce with its Default Number of Reducers.

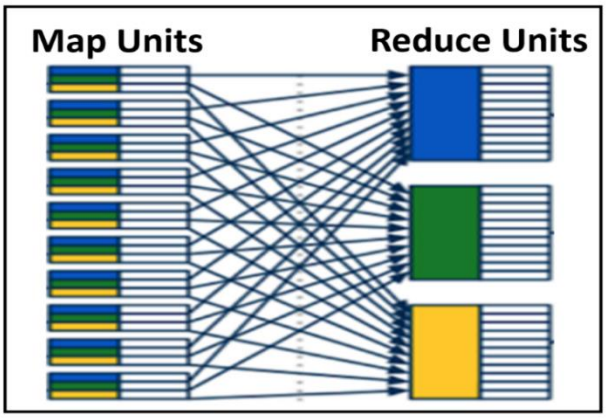

Fig. 4. Proposed Idea for Reduce Units.

In this progression, if the mappers output are divided and forwarded to reducers more than the default value, the overall traffic could be reduced. The overview of division of the mappers output among multiple reducers is presented in Fig. 4.

IV. Experiment Setup

For the implementation of this research study the following described simulation setup has been established. For its installation, first of all, the oracle virtual box was installed in order to take the advantage of creation of multiple machines. The machines are arranged for the master slave architecture. Next to it, is the installation of the Ubuntu setup over the virtual machines. It could be achieved either by installing the Ubuntu on one of the virtual machine and by using cloning feature of oracle virtual box to make the clones of that particular machine or by installing Ubuntu one by one on each of the virtual machines. After this, the Hadoop installation has been carried out. But there are some pre-requisites for the Hadoop installation i.e. the installation of the java either the java development kit jdk or java runtime environment i.e. jre.

The simulations has been done in Hadoop. Comparison analysis has been done in Microsoft Excel. The Hadoop version 1.x allows maximum of the 64 MB block size. The user is able to select the split size for the given data. The Hadoop version 2.x allows maximum of the 128 MB block size. This size can be varied by the user between 64 to 128 MB for Hadoop version 2.x. For the Hdoop 2.x case, the data split size of 128MB was selected for different inputs. The details of the input data size and the split size used are given in the following Table I.

TABLE I. Input Data and Split sizes

Input data size (MB)	Split size1 (MB)	Split size2 (MB)
100	64	128
250	64	128
510	64	128
1180	64	128
2020	64	128
5100	64	128
10480	64	128
15730	64	128

A. Configuration Details

For effective utilization of resources, assignment of the tasks to suitable resources is necessary. In addition, proper tuning of the parameters are equally important because inappropriate tuning of the parameters of the resources lead to degradation of the system overall performance. The parameters can be tuned by accessing the configuration files of the computational platform. For Hadoop MapReduce there are many configuration files, when user come across the configuration setting. The interactive files which need to be updated for proposed work setting are the following.

1) hadoop-env.sh
2) core-site.xml
3) hdfs-site.xml
4) mapred-site.xml

V. Results and Discussion

The input data size has been varied from Megabytes (MB) to Gigabytes (GB). The split size was taken as 64 MB and 128 MB. It has been observed that the variation of the split size with the input data size has effect on the number of mappers required for processing as shown in Fig.5. The variation of number of nodes from Fig.5 depicts that choosing the maximum split size will result in the less number of nodes for the input data i.e. split size with value 128 MB gives less than the case of 64 MB split size with same data size. Consequently the variation of mappers will affect the number of the reducers required for efficient processing.

Fig. 6 is depicting that the changing the number of reducers affect the overall execution time also. As observed by Fig. 6 that by elevation in the number of mappers that is with the increase in input data size results in lower execution time if the number of reducers are less. However, the same number of mappers with increasing the number of reducers to a certain value shows significant improvement in the total execution time.

It has been seen that changing the number of reducers along with variation of the number of mappers show a behavior of reducing the execution time at certain value However, the number of reducers providing the best execution time across the given number of mappers is different for different number of mappers. Fig.6 clearly shows that there is no fixed number of reducers for providing the best execution time but varies in number with variation among the number of mapper units. The default number of reducers are one for Hadoop MapReduce. However, for effective utilization of available resources, the number of mappers can be optimized by choosing the maximum split size available.

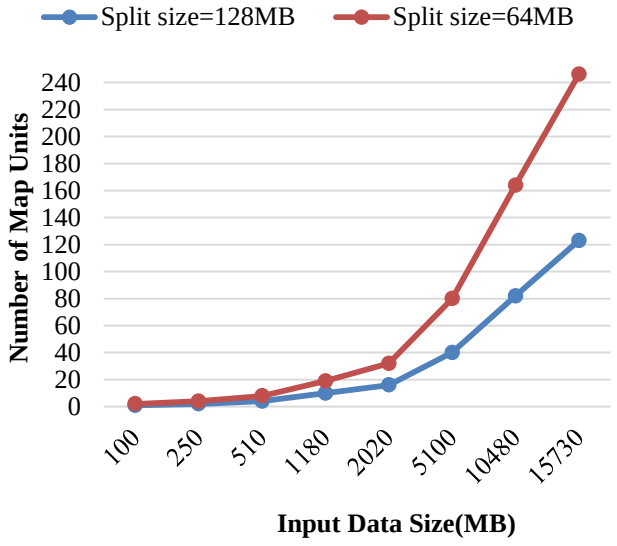

Fig. 5. Number of Mappers vs Different Input Data Size.

VI. Conclusion and Future Work

This research focused on the analysis and optimization of the number of compute nodes for cloud data centers by focusing upon the key issue of computational overhead due to inappropriate parameters selection and reducing overall execution time. For the computational platforms which are following the MapReduce framework, it provides a bases to handle larger type of data sets. As illustrated by the name of MapReduce, it computes a certain job into two phases map and reduce. In the map phase, the input data tasks are assigned to the processing units i.e. mappers. Next to it, in the reduce stage, the intermediate data generated by mapper are sorted and reduced to get the final output. Between the map and reduce phase, there is a migration of data from mappers to reducers. The default value for reducer is one for MapReduce framework. The default selection may lead to higher execution time with large network overhead. For increasing the number of reducers could possibly reduce some traffic and thus improves total execution time. However, for some special cases, number of reducers that is, the reduce phase can be skipped. In that case, mapper's output will be considered as final output. Otherwise, for the follow-up for the reduce phase the mapper's output has been stored to local disk or local storage and then passed to the reducers. The careful consideration is needed before assigning the tasks because the latency will increase if extra storing and fetching of input and output data blocks are manipulated unnecessary.

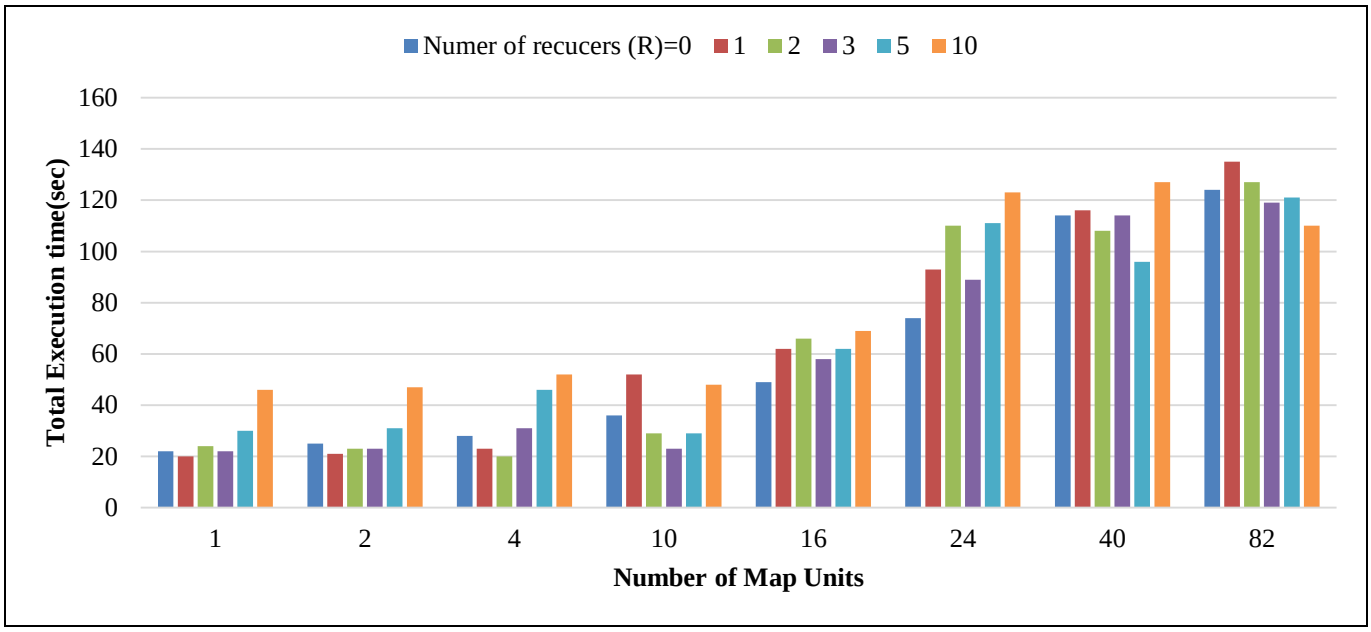

Fig. 6. Total Execution Time vs Number of Mappers with Different Number of Reducers.

Job ID	Name	User	Queue	State	Maps Total	Maps Completed	Reduces Total	Reduces Completed
job_1537633705227_0008	ValueAggregatorJob:	SYSTEM	default	SUCCEEDED	10	10	2	2
job_1537633705227_0007	ValueAggregatorJob:	SYSTEM	default	SUCCEEDED	123	123	0	0
job_1537633705227_0006	ValueAggregatorJob:	SYSTEM	default	SUCCEEDED	123	123	10	10
job_1537633705227_0005	ValueAggregatorJob:	SYSTEM	default	SUCCEEDED	123	123	5	5
job_1537633705227_0004	ValueAggregatorJob:	SYSTEM	default	SUCCEEDED	123	123	3	3
job_1537633705227_0003	ValueAggregatorJob:	SYSTEM	default	SUCCEEDED	123	123	2	2
job_1537633705227_0002	ValueAggregatorJob:	SYSTEM	default	SUCCEEDED	123	123	1	1
job_1537616406108_0029	ValueAggregatorJob:	SYSTEM	default	SUCCEEDED	82	82	0	0
job_1537616406108_0028	ValueAggregatorJob:	SYSTEM	default	SUCCEEDED	82	82	1	1
job_1537616406108_0027	ValueAggregatorJob:	SYSTEM	default	SUCCEEDED	82	82	2	2
job_1537616406108_0026	ValueAggregatorJob:	SYSTEM	default	SUCCEEDED	82	82	3	3
job_1537616406108_0025	ValueAggregatorJob:	SYSTEM	default	SUCCEEDED	82	82	5	5
job_1537616406108_0024	ValueAggregatorJob:	SYSTEM	default	SUCCEEDED	82	82	10	10
job_1537616406108_0023	ValueAggregatorJob:	SYSTEM	default	SUCCEEDED	10	10	1	1
job_1537616406108_0022	ValueAggregatorJob:	SYSTEM	default	SUCCEEDED	4	4	1	1
job_1537616406108_0021	ValueAggregatorJob:	SYSTEM	default	SUCCEEDED	4	4	0	0
job_1537616406108_0020	ValueAggregatorJob:	SYSTEM	default	SUCCEEDED	4	4	2	2
job_1537616406108_0019	ValueAggregatorJob:	SYSTEM	default	SUCCEEDED	4	4	3	3

Fig. 7. Job Completion with Various Numbers of Mappers and Reducers.

The input data size is varied from Megabytes (MB) to Gigabytes (GB). The split size was varied as 64 MB and 128 MB. The variation of number of nodes from Fig.5 depicts that choosing the maximum split size will result in the less number of nodes for the input data i.e. split size with value 128 MB gives less than the case of 64 MB split size with same data size. Fig. 7 shows that the simulations results with different number mapper and reduce units. In addition, the retrieving, storing and again sending the data to the Data nodes during the intermediate phase of MapReduce itself requires time, computation cost and may produce extra traffic. Moreover, the number of reducers providing the best execution time across the given number of mappers is different for different number of mappers. Fig. 6 shows that there is no fixed number of reducers for providing the best execution time but varies in number with variation among the number of mapper units. Thus, for the future work of this research will follows the modeling of the calculation of compute nodes according to the Cloud data center's capacity.

ACKNOWLEDGMENT

I am thankful to University Teknologi Malaysia (UTM) for providing me good research environment, tools, technical support and facilities to accomplish this research work. I pay my gratitude to my supervisor Prof. Dr. Kamalrulnizam Abu Bakar and co-supervisor Dr. Raja Zahilah Raja Mohd. Radzi for their guidance for the research. I am indebted of my senior Tasneem Darwish for her timely contributions and guidance.

REFERENCES

[1] Géczy, P. (2014). Big data characteristics. The Macrotheme Review, 3(6), 94-104.

[2] Chris Nolter, "IBM Rides the Cloud to New Heights," IBM Ratings Report. July 19, 2018.

[3] Bansal, N., Maurya, A., Kumar, T., Singh, M., & Bansal, S. (2015). Cost performance of QoS Driven task scheduling in cloud computing. Procedia Computer Science, 57, 126-130.

[4] Agneeswaran, V. S. (2014). Big data analytics beyond hadoop: real-time applications with storm, spark, and more hadoop alternatives: FT Press.

[5] Bechini, A., Marcelloni, F., and Segatori, A. (2016). A MapReduce solution for associative classification of big data. Information Sciences, 332, 33-55.

[6] Benifa, J. B. (2017). Performance Improvement of MapReduce for Heterogeneous Clusters Based on Efficient Locality and Replica Aware Scheduling Strategy. Wireless Personal Communications,1-25.

[7] Chen, Q., Zhang, D., Guo, M., Deng, Q., and Guo, S. (2010). Samr: A self-adaptive mapreduce scheduling algorithm in heterogeneous environment. Paper presented at the Computer and Information Technology, 2010 IEEE 10th International Conference on, 2736-2743.

[8] Tang, Z., Liu, M., Ammar, A., Li, K., and Li, K. (2016). An optimized MapReduce workflow scheduling algorithm for heterogeneous computing. The Journal of Supercomputing, 72(6), 2059-2079.

[9] Tiwari, N., Sarkar, S., Bellur, U., and Indrawan, M. (2015). Classification framework of MapReduce scheduling algorithms. ACM Computing Surveys (CSUR), 47(3), 49.

[10] Sowmya, T. S. R. (2016). Cost minimization for big data processing in geo-distributed data centers. Asia-Pacific Journal of Convergent Research Interchange, 2(4), 33-41.

[11] Cavallo, M., Cusmà, L., Di Modica, G., Polito, C., & Tomarchio, O. (2016). A Hadoop based Framework to Process Geo-distributed Big Data. Paper presented at the CLOSER (1).

[12] Beloglazov, A., & Buyya, R. (2012). Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Future generation computer systems, 28(5),755-768.

[13] Beloglazov, A., & Buyya, R. (2010). Energy efficient resource management in virtualized cloud data centers. Paper presented at the Proceedings of the 2010 10th IEEE/ACM international conference on cluster, cloud and grid computing.

[14] Aisha Shabbir, Kamalrulnizam Abu Bakar and Raja Zahilah Raja Mohd. Radzi. Replication Effect over Hadoop MapReduce Performance using Regression Analysis. International Journal of Computer Applications 181(24):33-38, October 2018.

[15] Koutroumpis, P., Leiponen, A., and Thomas, L. D. (2017). The (Unfulfilled) Potential of Data Marketplaces: The Research Institute of the Finnish Economyo. Document Number)

[16] Bechini, A., Marcelloni and Segatori, (2016). A MapReduce solution for associative classification of big data. Information Sciences, 332, 33-55.

[17] Aisha Shabbir, Kamalrulnizam Abu Bakar, "Big Data Processing Techniques and Platforms: A Comprehensive Survey", 7th International Graduate Conference on Engineering, Science and Humanities, IGCESH-Proceedings (2018) 698.

[18] Thaman, J., & Singh, M. (2016). Current perspective in task scheduling techniques in cloud computing: A review. International Journal in Foundations of Computer Science & Technology, 6(1), 65-85.

[19] Uta, A., & Obaseki, H. (2018). A Performance Study of Big Data Workloads in Cloud Datacenters with Network Variability. Paper presented at the Companion of the 2018 ACM/SPEC International Conference on Performance Engineering.

[20] Dong, Z., Liu, N., & Rojas-Cessa, R. (2015). Greedy scheduling of tasks with time constraints for energy-efficient cloud-computing data centers. Journal of Cloud Computing, 4(1), 5.