Proceedings of the International Conference on Industrial Engineering and Operations Management
Bangkok, Thailand, March 5-7, 2019

Big Data Tasks Execution Time Analysis Using Machine Learning Techniques

Aisha Shabbir, Kamalrulnizam Abu Bakar and Raja Zahilah Raja Mohd. Radzi
School of Computing
University Technology Malaysia, Johor, Malaysia
saisha3@graduate.utm.my, knizam@utm.my, zahilah@utm.my

Muhammad Siraj
Department of Information Engineering & Computer Science
University of Trento, Trento, Italy
muhammad.siraj@alumni.unitn.it

Abstract

Big data and its analysis are in the focus of current era. The volume of data production is tremendous and a significant part of delivered data is not utilized because of the limited assets to store and process them efficiently. The world acclaimed platform that can efficiently deal with the gigantic amount of data in a cost effective manner is Hadoop MapReduce. In order to effectively utilize any computational platform, information about the components affecting its performance is necessary. Similarly, Hadoop MapReduce's performance can be enhanced by identifying those factors that can affect its performance. Some researchers provided some schemes for improving total task completion time of big data tasks on Hadoop MapReduce by suitable selection and scheduling of processing units i.e. mappers. However, reducers are still underexplored for its effect on the total execution time. This paper aimed at evaluation of reducer's impact on total execution time of big data tasks on Hadoop MapReduce by employing machine learning techniques. The evaluation has been carried out both analytically and experimentally by changing different number of reducers across various types and length of tasks. The results clearly depicts the dependence of total MapReduce task execution time on the number of reducers.

Keywords

Big Data, Machine learning, Hadoop MapReduce, total execution time.

1. Introduction

The present era has witnessed a dramatic change in scientific frontiers. A plethora of developments have occurred in digital technologies like, social media and networks, financial transactions, sensor's data business and financial dealings and person to person communications via digital platforms. These developments resulted in the generation of massive amount of data termed as "Big Data". The data can be in various forms like, pictures, text, xml, sound, social context, video and so on (Koutroumpis, Leiponen, & Thomas, 2017). The challenge here is the storage, processing and analysis of tremendously growing amount of data by utilizing traditional databases and conventional tools and schemes. This challenge has aroused the need for processing, storing and investigating the large bulk of data in almost all fields with the help of smart and efficient platforms and techniques. In addition, shrinking time to analyze the growing amount of data and the information about time estimation for tasks execution over the computational resources is the biggest challenge faced by both researchers and industrialists.

© IEOM Society International

Hadoop is an indispensable component of the big data. Hadoop comprises three main sub frameworks: Hadoop Common, Hadoop Distributed File System (HDFS), and Hadoop MapReduce (Fahad et al., 2014). Hadoop Common offers the utility functions, including remote procedure call (RPC) facilities and object serialization libraries that are leveraged by the HDFS and MapReduce frameworks. HDFS is an implementation of a distributed file system that is based on Google's distributed file system, named GFS (Google File System). MapReduce is initially established by Google and it is designed for processing big data by exploiting the parallelism among a cluster of machines. Such parallelization enables compute frameworks to cope with growth in datasets being faster than Moore's law. The real implementation of MapReduce for huge scale data sets usually takes place on more than one machine or on a number of machines (Delimitrou & Kozyrakis, 2014) and (Delimitrou & Kozyrakis, 2014). There are many factors which can affect the Hadoop MapReduce performance and thus the overall MapReduce jobs execution time.

To achieve a better performance, the careful consideration of the factors affecting on execution time of big data jobs that is MapReduce jobs is needed. There are some factors explored by many researches for improving the total completion time of Big data tasks on Hadoop. Some researchers focused on the scheduling techniques to improve the overall job execution time (Balagoni & Rao, 2017), (Althebyan, Jararweh, Yaseen, AlQudah, & Al‐Ayyoub, 2015), (Guo, Fox, & Zhou, 2012) and (Tang, Liu, Ammar, Li, & Li, 2016). Similarly, some researches focused on particular phase scheduling (Ke, Li, Guo, & Guo, 2016), (Tiwari, Sarkar, Bellur, & Indrawan, 2015) and (Neelakandan, Divyabharathi, Rahini, & Vijayalakshmi, 2016). Some research has been done to improve the fault tolerance (Fu, Chen, Zhu, & Yu, 2017) and (Xu & Lau, 2017). Some tried to focus on the reliability issues (Ananthanarayanan, Ghodsi, Shenker, & Stoica, 2013). There are many factors contributing towards the total task execution time. This study is focusing on the analysis of the underexplored factor that is, reducers towards the overall task execution time. Hadoop MapReduce accomplish job processing on huge data tasks in two main phases i.e. Map and Reduce. There is another phase that is, shuffle phase also for dealing with intermediate data. Furthermore, the default number of reducer is one in Hadoop MapReduce. In the shuffle phase, the extra time will be taken especially if the parameters are not properly optimized. If all the metadata will be given to a single reducer, the total execution time will be more. Especially if the input data size is bigger, there will be much more Meta data and will create huge traffic, the results can be degraded with default values. Thus, suitable selection of this parameter is important.

Machine learning techniques has been emerged as one of the promising solutions for making the predictions in the research community. Regression analysis is one of the machine learning technique. Researchers used this technique for the predictive analysis (Khan, Jin, Li, Xiang, & Jiang, 2016). Regression analysis has been providing the selection criterion for the inclusion or rejection of a variable or factor affecting the dependent variable. This research problem has been focused over the contribution and effect of reducers for a Big data task execution time running on Hadoop MapReduce. For the evaluation of the impact of reducers upon total execution times of MapReduce jobs regression analysis has been used. The general framework used for conveying the idea described is shown in Fig. 1.

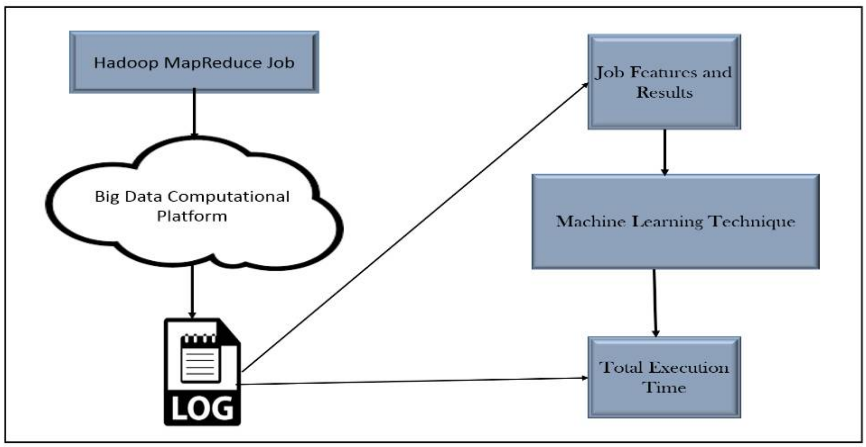

Figure 1. Framework

The organization of the paper is as follows: Section II describes the Preliminaries that is, giving the overview of Big data issues and Hadoop MapReduce. This section also enlightens the taxonomy of the machine learning techniques. Section III is about the Problem insight. Section IV provides the Experiments and evaluation. Section V comprises the Results interpretation. Last section i.e. VI is of Conclusion and following to it are the acknowledgements and references of this study.

© IEOM Society International

2. Preliminaries

This section provides the details about Big data issues, an overview of Hadoop MapReduce with some of its architectural features and the Machine learning techniques.

2.1. Big Data Issues

a) Data Storage Issues:
The amount of information has burst out each time and thus need for invention of a new storage method. For handling huge volume storage, Big Data storage companies such as IBM, EMC Amazon utilizing the tools like Apache Drill, SAMOA, NoSQL, IKANOW, Hadoop and Horton Works (Franková, Drahošová, & Balco, 2016) and (Lomotey & Deters, 2014).

b) Data Management Issues:
The data generation sources are different and thus the data also both by means of format and in terms of collection. People contribute digital information in a way which are suitable for them like archives, illustrations, pictures, audio and video messages, models, programming practices, and so forth with or without satisfactory metadata depicting where, who, what, when, why and how it was gathered. However, the collected information is promptly accessible for investigation and examination. Furthermore, data and its provenance will become a serious issue. As indicated by Gartner, Big Data challenge involves more than just managing volumes of data mentioned in his article (Saraladevi, Pazhaniraja, Paul, Basha, & Dhavachelvan, 2015).

c) Big Data Processing Issue:
To understand the Big data processing issue, let us consider an example for which Exabyte of data has to be processed at a time. Divide the data into the 8 blocks i.e. 1 Exabyte would be equal to 1k petabyte and the processor uses 100 instructions per block at 5 gigahertz. Thus the processing time for end-to-end will be 20 nanoseconds. So, 1K petabytes processing will need approximately 635 years' time for end-to-end processing. Along these lines, viable processing of Exabyte's of information would require broad parallel handling with a specific end goal to give convenient and significant data.

d) Security Issues:
It is challenging to manage a large data set in secure means. Further, public and private database and inefficient tools comprise many threats. Unexpected spillage of information, and inadequacy of public and private policy makes hackers/programmers to gather their assets at whatever point required. The security issues occurs for distributed systems when huge measure of private information put away in a database which is not encoded and encrypted.

2.2. Hadoop MapReduce Overview

Efficiently processing and analyzing huge volume and variety of data has become the major source for innovation for compute intensive and data-intensive applications. Hadoop is an indispensable component of the big data. It is an open source platform that uses the MapReduce model as a backbone. Hadoop MapReduce accomplish job processing on huge datasets by supposing that large dataset storage is distributed over a large number of machines. The computation is done in two main phases i.e. Map and Reduce. There is another phase i.e. shuffle phase also for dealing with intermediate data. MapReduce is proposed by Google to simplify massively distributed parallel processing so that very large and complex datasets can be processed and analyzed efficiently. There are two versions of Hadoop i.e. Hadoop 1.x and Hadoop 2.x. The two most important components that are the foundation to Hadoop framework i.e. HDFS and MapReduce.

a) Hadoop Distributed File System (HDFS):
HDFS component is divided into two sub-components:

- Name Node

Name node is considered as master node. It is used to store meta data about data nodes i.e. how many blocks are stored in data nodes, which data nodes have data, slave node details and data nodes locations.

© IEOM Society International

- Data Node

Data nodes serve as slave nodes. They used to store application actual data. It stores data in data slots of size 64 MB to 128 MB.

b) MapReduce:
MapReduce component is divided into two sub-components:

- Job Tracker

Job tracker is used to assign MapReduce tasks to task trackers in the cluster of nodes. Sometimes, it reassigns same tasks to other task trackers as previous task trackers are failed or shutdown scenarios. Job tracker maintains all the task trackers status like running, failed and recovered.

- Task Tracker

Task tracker executes the tasks which are assigned by job tracker and sends the status of those tasks to job tracker.

2.3. Machine Learning Techniques

Machine learning techniques has produced a lot of buzz due to its applicability across a wide range of areas and applications. Basically, machine learning is a collection of various methods that are specifically suited to each of its respondents coming from a diverse sets and business. Based on the working of machine learning techniques, it can be broadly classified in three categories that is, supervised, and unsupervised and reinforcement learning algorithms. The Fig.2 shows the grouping of the different algorithms under the specific learning scheme.

Regression analysis comes under both statistical and machine learning techniques. Regression analysis has been used for the qualification and disqualification of a variable for the particular dependent/task. There are many configuration parameters for the Hadoop MapReduce job. One the configuration parameter is the replication factor. Whether the Hadoop MapReduce job really depends upon the replication or not. The qualification for dependence of the variable depends upon the prediction value of the regression analysis results. Predictive values normally called as P-values. If a P-value against the variable is less than 0.05 then the variable has no effect over the given task (Solutions, 2013).

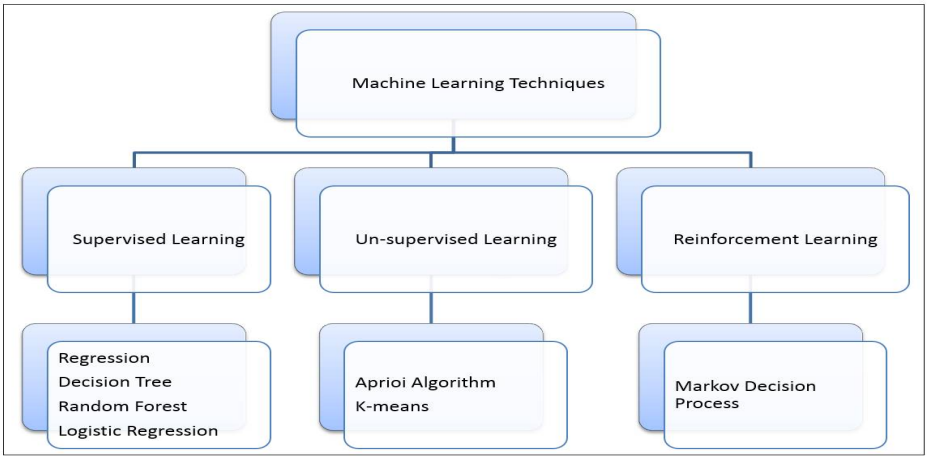

Figure 2. Machine learning techniques taxonomy

3. Problem Insight

Tasks having big data sets that are so huge or complex and conventional data processing techniques are incapable to deal with them. Nowadays, for handling huge volumes of data sets, Big Data companies such as IBM, EMC Amazon utilizing the tools like Apache Drill, SAMOA, NoSQL, IKANOW, and Hadoop MapReduce. At the present time,

© IEOM Society International

Hadoop has been 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 (Saraladevi et al., 2015).
In order to effectively utilize any computational platform, information about the components affecting its performance is necessary. Similarly, Hadoop MapReduce's performance can be enhanced by identifying those factors that can affect its performance. Some researchers put forward some schemes for improving total task completion time of big data tasks on Hadoop MapReduce by suitable selection and scheduling of processing units that is mappers. However, the reducers are still underexplored for its effect on the total execution time.
MapReduce framework was initially proposed by Google to simplify massively distributed parallel processing so that very large and complex datasets can be processed and analyzed efficiently. MapReduce generally executes a job in two phases. During the map section, it divides the data input and run it on the given set of nodes. The mappers produces the output as a key and value pairs. These pairs are passed to the reducers to reduce it for the final result. Thus, in the reduce phase, the output of mappers are treated as input generally termed as intermediate data. There exists a merge and sort section named as shuffle between the map and reduce phase. In this shuffle phase, the data added to the mappers are divided and exchanged to the ideal machines executing the reduce section services.
The general workflow of MapReduce is given in Fig.3. In default setting of Hadoop MapReduce, the number of reducers are one. Furthermore, the default replication value is three. In the shuffle phase, the extra time will be taken especially if the parameters are not properly optimized. If all the metadata will be given to a single reduce, the total execution time will be more. Especially if the input data size is bigger, there will be much more Meta data and will create huge traffic, the results can be degraded with default values. Further, this traffic eventually decreases the overall cluster's performance. Furthermore, some researchers put forward some schemes for improving total task completion time of big data tasks on Hadoop MapReduce by suitable scheduling of tasks on processing units that is mappers. However, less attention has been paid towards the selection and effect of reducer's side. MapReduce computes a job into different phases and during its operation it follow the general workflow as shown Fig. 3 for all types of jobs.

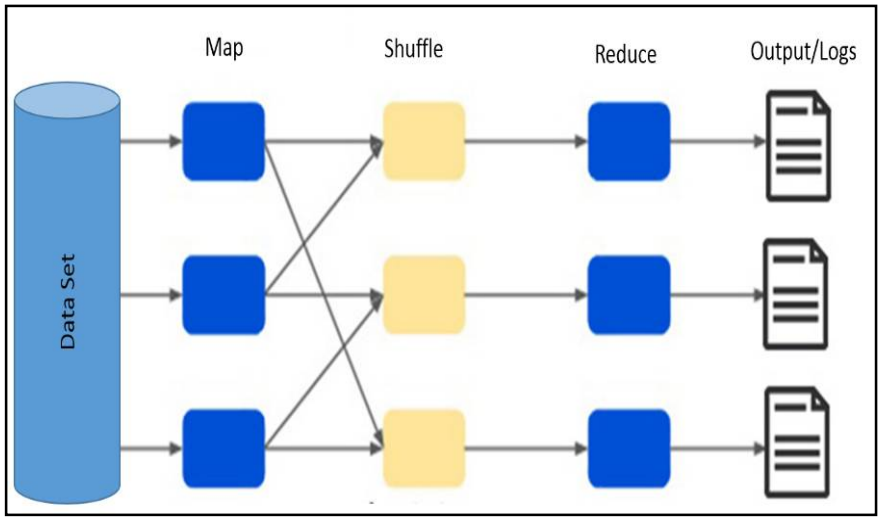

Figure 3. MapReduce workflow

4. Experiment And Evaluation

The implementation and evaluation of this work has been conducted in Hadoop version 2.7.1. Several MapReduce jobs has been evaluated against different number of reducers and different sizes of input data ranging from Bytes to Giga bytes. 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 second 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.

© IEOM Society International

Table 1. Input files sizes with abbreviation used for graphics

Input data file	Reference abbrevation for Graphs	File Size
File 1	F1	3.14MB
File 2	F2	130.38MB
File 3	F3	521.53MB
File 4	F4	1.01GB
File 5	F5	1.78GB
File 6	F6	2.03GB
File 7	F7	3.1GB
File 8	F8	4.07GB
File 9	F9	6.11GB

Hadoop MapReduce job has many configuration factors including replication factors and number of reducers. The configuration settings for replication has been set to one as shown in Fig. 4. The different numbers of reducers has been chosen for analysis as shown in Fig. 5. Machine learning technique i.e. Regression analysis has been chosen for analytical analysis and implemented in the Microsoft Excel. The total task execution time has been used as a performance evaluation metric.

name	value
dfs.replication	1
dfs.namenode.name.dir	file:/Syncfusion/BigData/3.2.0.20/BigDataSDK/Metadata/data/dfs/namenode
dfs.datanode.data.dir	file:/Syncfusion/BigData/3.2.0.20/BigDataSDK/Metadata/data/dfs/datanode
dfs.permissions	false
dfs.http.address	localhost:50070
dfs.webhdfs.enabled	true
dfs.datanode.du.reserved	1073741824
dfs.datanode.data.dir.perm	777
dfs.namenode.rpc-bind-host	0.0.0.0

Figure 4. Configuration details

© IEOM Society International

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

Figure 5. Different jobs successful completion along with different number of mappers and reducers

5. **Results Interpretation**

It has been identified that the variation of the number of reducers has a great impact on the total job completion time. As shown in Figure 6 that by increasing the number of reducers with varying the input data sizes has follows a trend line which shows a dip for each input or task. But the value of the reducer that is, the number of reducers showing the best execution time for each input data size is different. It is obvious from Fig. 6 that the fixed number of reducers does not provide the appropriate total execution time for each data set or task as the default number of reducers are one for Hadoop MapReduce. However, it has been clear from the simulation results shown in Fig. 6 also that changing the number of reducers affect the total execution time.

© IEOM Society International

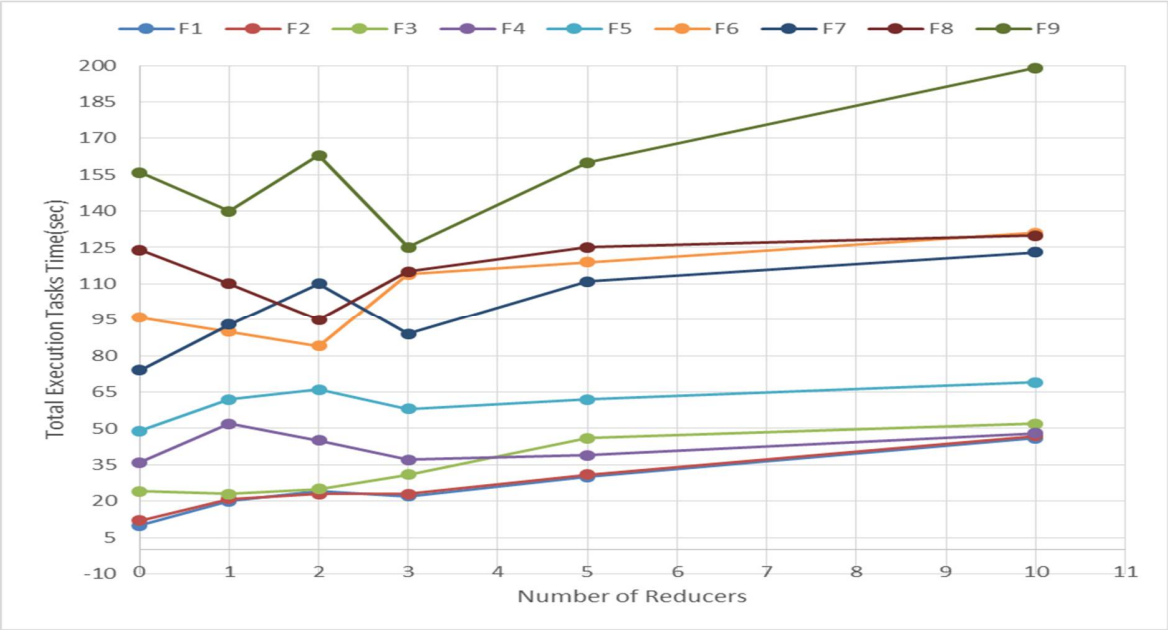

Figure 6. Task total execution time vs different no. of reducers

Table 2. Regression analysis Results

	Coefficients	Standard Error	t Stat	P-value	Lower 95%	Upper 95%	Lower 95.0%	Upper 95.0%
Intercept	-14.94597784	17.84331945	-0.837623172	0.406669233	-50.8843	20.99231	-50.8843	20.99231234
Number of reducers	2.171755725	3.253469094	0.667520011	0.507849017	-4.38107	8.724579	-4.38107	8.724578854

6. Conclusion

Hadoop MapReduce has been considered as the promising and acceptable platform for big data tasks processing. Hadoop MapReduce has many configuration parameters and their optimal tuning can significantly increase total task completion time of big data tasks on Hadoop MapReduce. In this paper, an underexplored factor that is the number of reducers is evaluated for its effect on overall task execution time of Big data tasks in Hadoop MapReduce. Several simulations has been done by different number of reducers and also by varying the input data sizes. It has been observed that the variation of the number of reducers influences the total tasks execution time. The results from the Fig. 6 depicts that changing the number of reducers has a significantly effect on total completion time for particular values. As shown in Figure 6 that by increasing the number of reducers with varying the input data sizes has follows a trend line which shows a dip for each input or task. But the value of the reducer that is, the number of reducers showing the best execution time for each input data size is different. In addition, Machine learning technique that is, Regression has been used for the qualifying criteria. The evaluation results evidently shows the dependence of the total execution time on the number of reducers. The results are also validated through regression analysis p-value. From the Table II we can see that p-value is greater than 0.05 which means replication factor contributes as major to total completion time. Consequently, it has a substantial effect on overall performance of the Hadoop MapReduce also.

© IEOM Society International

Acknowledgements

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

References

Althebyan, Q., Jararweh, Y., Yaseen, Q., AlQudah, O., & Al-Ayyoub, M. (2015). Evaluating map reduce tasks scheduling algorithms over cloud computing infrastructure. *Concurrency and Computation: Practice and Experience, 27*(18), 5686-5699.

Ananthanarayanan, G., Ghodsi, A., Shenker, S., & Stoica, I. (2013). *Effective Straggler Mitigation: Attack of the Clones*. Paper presented at the NSDI.

Balagoni, Y., & Rao, R. R. (2017). Locality-Load-Prediction Aware Multi-Objective Task Scheduling in the Heterogeneous Cloud Environment. *Indian Journal of Science and Technology, 10*(9).

Delimitrou, C., & Kozyrakis, C. (2014). *Quasar: resource-efficient and QoS-aware cluster management*. Paper presented at the ACM SIGPLAN Notices.

Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Khalil, I., Zomaya, A. Y., . . . Bouras, A. (2014). A survey of clustering algorithms for big data: Taxonomy and empirical analysis. *IEEE transactions on emerging topics in computing, 2*(3), 267-279.

Franková, P., Drahošová, M., & Balco, P. (2016). Agile project management approach and its use in big data management. *Procedia Computer Science, 83*, 576-583.

Fu, H., Chen, H., Zhu, Y., & Yu, W. (2017). FARMS: Efficient mapreduce speculation for failure recovery in short jobs. *Parallel Computing, 61*, 68-82.

Guo, Z., Fox, G., & Zhou, M. (2012). *Investigation of data locality in mapreduce*. Paper presented at the Cluster, Cloud and Grid Computing (CCGrid), 2012 12th IEEE/ACM International Symposium on.

Ke, H., Li, P., Guo, S., & Guo, M. (2016). On traffic-aware partition and aggregation in mapreduce for big data applications. *IEEE Transactions on Parallel and Distributed Systems, 27*(3), 818-828.

Khan, M., Jin, Y., Li, M., Xiang, Y., & Jiang, C. (2016). Hadoop performance modeling for job estimation and resource provisioning. *IEEE Transactions on Parallel and Distributed Systems, 27*(2), 441-454.

Koutroumpis, P., Leiponen, A., & Thomas, L. D. (2017). *The (Unfulfilled) Potential of Data Marketplaces*. Retrieved from

Lomotey, R. K., & Deters, R. (2014). *Towards knowledge discovery in big data*. Paper presented at the 2014 IEEE 8th International Symposium on Service Oriented System Engineering (SOSE).

Neelakandan, S., Divyabharathi, S., Rahini, S., & Vijayalakshmi, G. (2016). *Large scale optimization to minimize network traffic using MapReduce in big data applications.* Paper presented at the Computation of Power, Energy Information and Commuincation (ICCPEIC), 2016 International Conference on.

Saraladevi, B., Pazhaniraja, N., Paul, P. V., Basha, M. S., & Dhavachelvan, P. (2015). Big Data and Hadoop-A study in security perspective. *Procedia Computer Science, 50*, 596-601.

Solutions, S. (2013). What is linear regression. *Retrieved on August, 25*, 2017.

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

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

Xu, H., & Lau, W. C. (2017). Optimization for speculative execution in big data processing clusters. *IEEE Transactions on Parallel and Distributed Systems, 28*(2), 530-545.

© IEOM Society International

Biographies

Aisha Shabbir received her B.Sc. degree in Math Applied, Math Pure & Physics from the Hazara University, Pakistan, in 2008, M.Sc. degree in Electronics, from Quaid-i-Azam University, Pakistan, in 2010. She did MS in Electronic Engineering from Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Pakistan in 2013. Currently, she is doing PhD in Computer Science at Universiti Teknologi Malaysia (UTM), Malaysia, She is Student member IEEE and member of Pervasive Computing Research Group (PCRG). Her research interest includes data aggregation in networks, partioning and scheduling algorithms for big data and cloud computing.

Kamalrulnizam Abu Bakar received his B.Sc. degree in Computer Science from the Universiti Teknologi Malaysia, Malaysia, in 1996, M.Sc. degree in Computer Communications and Networks, from Leeds Metropolitan University, United Kingdom, in 1998, and his PhD. degree in Computer Science from Aston University, United Kingdom, in 2004. He is a Professor in Computer Science at Universiti Teknologi Malaysia, Malaysia, and member PCRG. His research interest includes mobile and wireless computing, ad hoc and sensor networks, big data and cloud computing. He is a member of the ACM, Internet Society (ISOC) and International Association of Engineering (IAENG). He involves in many research projects and is a referee for several scientific journals and conferences.

Raja Zahilah Raja Mohd. Radzi received her Dr. Eng in Electrical & Information System, from Osaka Prefecture University, Japan on 2012. She obtained M.Eng. in Electronics and Telecommunications and a B.Eng. in Computer, both from Faculty of Electrical Engineering, Universiti Teknologi Malaysia. Currently, she is a Senior Lecturer at Faculty of Computing, UTM, Malaysia. She is a member of the PCRG, IEEE Society (Communication Society & Photonics Society) and an overseas member of The Institute of Electronics, Information and Communication Engineers, Japan. Her research interests include IoT, Big Data and Cloud Computing, Software Define Networking and Embedded System.

Muhammad Siraj received his Bachelors' Degree in Electronic Engineering, from Politecnico Di Torino, Italy in 2013. He obtained the MS in Telecommunication Engineering Degree from University of Trento, Trento, Italy in 2018. Currently, he is a working as test Engineer at GLI Company in Italy. His research interests include IoT, Big Data and Cloud Computing, Wireless networks.

© IEOM Society International