ISSN 2278-3091

Volume 8, No.1.6, 2019

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse1681.62019.pdf

https://doi.org/10.30534/ijatcse/2019/1681.62019

MapReduce Intermediate Data Migration Time Modeling using Machine Learning Technique

Aisha Shabbin[1], Kamalnizam Abu Bakar[2], Raja Zahilah[3], Mary Aksa[4]
[1,2,3]School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia (UTM), Malaysia
[4]Department of Computer Science, COMSATS Institute of Information Technology, Pakistan

ABSTRACT

The world acclaimed platform that can efficiently deal with the gigantic amount of data is MapReduce. In order to effectively utilize any computational platform, information about the components affecting its performance is necessary. Encapsulating a factor or optimizing a crucial factor acts as a catalyst which can accelerate overall peformance of the platform. Some researchers provided some techniques to improve the overall performance of MapReduce by suitable selection and scheduling of processing units i.e. mappers. However, negligible attention has been paid towards the intermediate (shuffle) phase optimization for its effect on overall MapReduce performance. This paper aimed at modeling the data migration time within the intermediate phase of MapReduce by encountering the contributing factors with the help of machine learning technique. In addition to it, the contributing factors for their contribution toward the shuffle's phase time are tested both analytically and experimentally. This research provides the Shuffle phase time model that estimates data migration time in the intermediate phase of MapReduce. The data set has been collected by historical MapReduce job execution records. The model has been proposed over training data sets by employing the machine learning techniques termed as regression technique and validated over the different test data sets.

Key words : Big Data, Machine Learning, Hadoop MapReduce, Total job execution time, Predictive modeling

1. INTRODUCTION

Current era has depicted that for every couple of years, there is a burst of data. 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 [1]. 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.

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 [2]. There are many factors which can affect the Hadoop MapReduce performance and thus the overall job execution profile. To achieve a better performance, the careful consideration of the factors affecting the execution time of different phases of MapReduce is needed.

There are some factors explored by many researches for improving the total completion time of big data tasks over MapReduce Platform. Some researchers focused on the scheduling techniques to improve the overall job execution time [3], [4], [5] and [6]. Similarly, some researches focused on particular phase scheduling [7], [8] and [9]. Some research has been done to improve the fault tolerance [10] and [11]. Some tried to focus on the replication and reliability issues [12] and [13]. There are many factors contributing towards the total 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 for dealing with intermediate data. For huge data sets, the intermediate traffic in the shuffle phase is tremendous and it will take much extra time if will be processed with default parameter setting. If all the metadata will be forwarded to the reducers as it received from the mappers, there will be traffic congestion and it deteriorates the overall MapReduce performance severely. Encountering and introducing the factors which can improve the shuffle's phase execution time is necessary.

This study is focusing on the data traffic congestion of the shuffle phase of the MapReduce which deteriorates the overall performance of the MapReduce especially in case of huge data set at the input. This research problem has been focused over the contribution of data traffic partitions' effect for Shuffle's phase time of MapReduce. For the modeling and evaluation of the impact of these partitions upon the shuffle phase's time and on the overall performance of MapReduce, a machine learning technique called regression has been used. 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 [14]. Regression analysis has been providing the selection criterion for the inclusion or rejection of a variable or factor affecting the dependent variable. The general framework used for conveying the idea described is shown in Figure 1.

The organization of the paper is as follows: Section II provides the Preliminaries that is, Big data characteristics and Machine learning techniques classification. Section III is about the Problem insight. This section also enlightens the overview of the MapReduce Framework and the working of its different phases. Section IV provides the details of the model preparation. Section V provides the Experiments and evaluation. Section VI comprises the Results and discussion. Last section i.e. VII is of Conclusion.

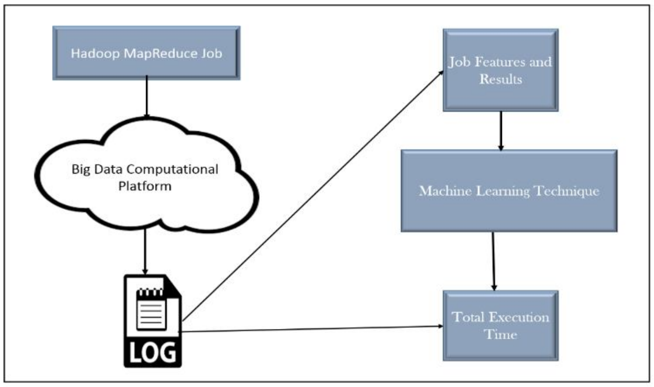


Figure 1: General framework

2. PRELIMANARIES

This section comprises the details of Big data characteristics and machine learning techniques with its classification.

2.1. Big Data and Its Characteristics

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 [15]. The increasing number of challenges of big data are due to its diverse nature which is categorized as its V's by [16]. The 5V's characteristics of big data as shown in the Figure 2 are volume, velocity, variety, veracity and value.

The information representation by different V's are as follows.

Volume illustrates about the size of the data that is Megabytes, Gigabyte or in Tera bytes. The predictions from [17] is that it will reach up to zettabytes in next couple of years.

Variety tells about the different formats of data as given below:

1) Structured: Nearly 20% of the whole world's data is in the structured form. For example, the date, tables and numbers.
2) Unstructured: Do not have specific structure. Approximately 80% of whole world's data is unstructured. For instance, images, texts, etc.
3) Semi-Structured: Lies in the middle of structure and unstructured. Presenting some specific scheme but not the highly structured one. Like Tweets, logs.

Velocity gives the information about the speed of the data. It indicates whether the data is coming in the form of batches or in streams.

Veracity gives the information about the quality and originality of the data sets.

Value is responsible for the information about the statics and hypothetical forms of data.

2.2. 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 [23]-[24]. The Figure3 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. In addition to it, by using regression, the modeling of the dependent variable can be performed by encountering the factors which qualify the contribution criteria. 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 a significant effect over the given dependent variable [18].

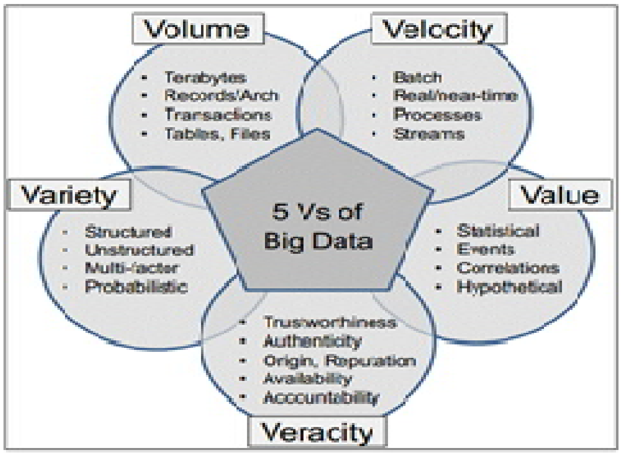


Figure 2: Big data characteristics [16]

3. PROBLEM INSIGHT

This section gives the details of the core problem due to which the partitioning of the data traffic has been carried out.

3.1. Hadoop MapReduce Overview

For massive and huge data set's processing in a distributed manner, MapReduce was proposed by Google. The default implementation of the MapReduce is Hadoop. MapReduce is emerging as a most promising solution for the big data problems. It exploits parallelism among the compute node for the processing of gigantic sets of data. MapReduce has main two phases as by the name of MapReduce that is. Map and Reduce. There is another intermediate phase called shuffle phase for transferring the intermediate data from Map side to the reduce part. MapReduce computes any tasks in the following way:

Map Phase: It is the first phase of the MapReduce in which the input data sets are divided into the chunks to process it in a parallel manner. This mapping of the tasks can be realized as a scheduling of the subtasks of a bigger tasks over the compute units. The output of the Map phase is in the form of the key value pairs.

Shuffle Phase: Shuffle phase is the intermediate phase of the Map Reduce. This phase holds the responsibility of transferring the output data of the mappers to the reduce units for its further processing. This intermediate phase also sorts the intermediate data.

Reduce Phase: This phase of the MapReduce gathers all the intermediate data and do the data reduction according to the requirement of the application or the program running over it. The extraction of the desired results are also carried out in this phase.

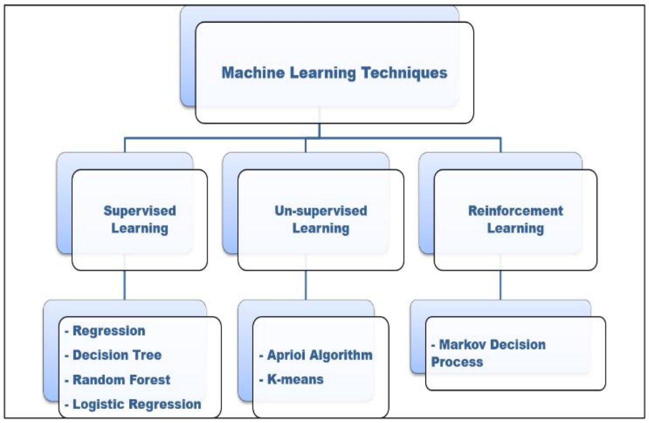


Figure 3: Machine learning techniques classification

MapReduce framework provides an extendable and reliable data processing method that significantly improved the performance of the clustering in massive data driven application .The general workflow of MapReduce is given in Figure 4. In default setting of Hadoop MapReduce, there are no partitions in the shuffle phase for the intermediate data. In the shuffle phase, the extra time will be taken especially if the parameters are not properly optimized. If all the metadata will be forwarded to the reduce units especially when the input data size is bigger, there will be much more Meta data and will create huge traffic and increases the total time of migration of the data from mappers to the reducers. The zoomed view of the traffic congestion in given in Figure 4. In order to handle the intermediate data traffic in an effective means for reducing the shuffle phase execution time of tasks, the number of partitions has been considered.

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, negligible attention has been paid towards the creation of partitioning the Meta data before sending towards the reducers. MapReduce computes a job into different phases and during its operation it follow the general workflow as shown Figure 4 for all types of jobs.

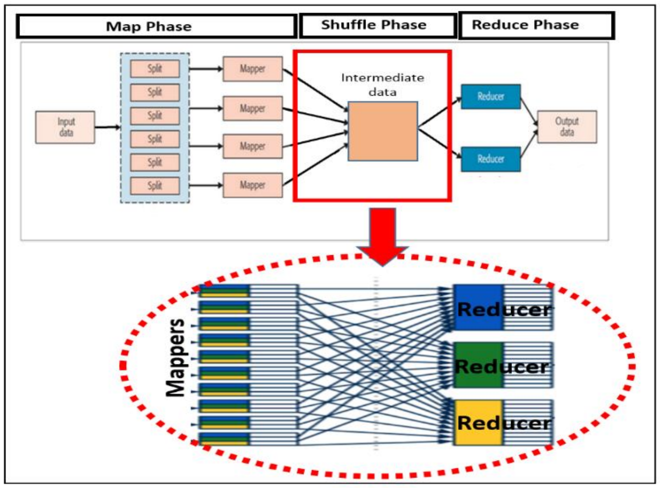


Figure 4: Traffic congestion zoomed view

4. MODEL PREPARATION

From the previous section of Problem insight, there is a research question arises that whether the number of partitions (independent variable) contribute to shuffle phase's execution time (dependent variable) or not? To examine this research question, a linear regression has be conducted to investigate whether or not independent variable has a significant effect over the dependent variable. The testing has been done through P-value test and behavior of shuffle time with partitioning effect has been observed. The following steps will be followed to model the contributing factor.

1) Split data into train and test sets.
2) Building a model on a training set.
3) Evaluate on test set and retrain.

The procedure for overall modeling and spitting of data set is shown by a block diagram in Figure 5.

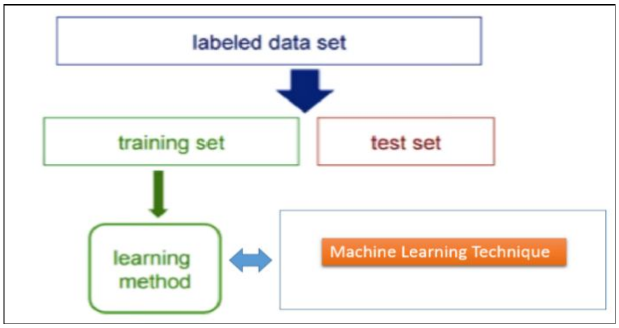


Figure 5: Modeling Overview

A linear regression is an appropriate analysis when the goal of research is to assess the extent of a relationship between the predictor variable on criterion variable. For the regression technique, the predictor variable is the independent variable which in this case is the number of partitions and the dependent variable can be computed by the Eq. 1. Before going into the depth, it is necessary to see why the model is necessary and why to be prepared by using the machine learning technique. As by enquiring the past data values, the future predictions can be made for better processing and effective resource provisioning. For making the predictions, the machine learning techniques have ventured their foot in wide range of applications. These techniques basically provides some mathematical models by finding the patterns and relationship among the components of the given data sets [19-22]. Shuffle execution time model is formed by using regression equation whose general form is given in Eq. 1.

$$O=\alpha \ +\beta u \qquad (1)$$

Where O is estimated dependent variable, α is constant or intercept, β is estimated slope and u is independent variable. For our case, u are the number of partitions and O is the shuffle phase execution time. Slope can be calculated for the above equation as:

$$\beta \ = r\ (S_O/\ S_U) \qquad (2)$$

Where r is the sample correlation coefficient and S_O and S_U are the standard deviations of the dependent variable O and the independent variable u respectively. Further, r can be calculated using the following equations.

$$r = \frac{C}{\sqrt{B.A}} \qquad (3)$$

$$C = \sum [(O - \bar{O})\ (u - \bar{u})] \quad (4)$$

$$A = \sum (O - \bar{O})\ 2 \qquad (5)$$

$$B=\sum (u - \bar{u})^2 \qquad (6)$$

Where, Ō and ū are the sample means of the dependent variable O and the independent variable u respectively. Substituting values from equations 4, 5 and 6 in equation 3, we get the value of r. Next, the standard deviations for both independent and dependent factors can be calculated by the following equations.

$$S_O = \sqrt{\frac{\Sigma(O - \bar{O})^2}{n - 1}} \qquad (7)$$

$$S_U = \sqrt{\frac{\sum(u-\bar{u})^2}{n-1}} \qquad (8)$$

Thus the figures calculated by equations 3, 7 and 8 after substitution in equation 2 will give us the value of slope. Slope provides the fundamental unit increase or decrease of the predictor or the contributing factor in a statistical models for the dependent variable. Further, y-intercept α can be calculated by the following Eq. 9.

$$\alpha = \bar{O} \ - \beta\ \bar{u} \qquad (9)$$

The 70 % of the data collected from the experimental results are used for the training and the 30% of the data are used for the testing purpose. The data used for the testing has not been used for training the model. The model is built in order to get the estimate of the shuffle's phase time in relevant to the number of partitions. To elaborate it further, to get a forecast of the shuffle phase time for altering the factor i.e. number of partitions. In this regard, the large number of simulations has been done for different test files by changing the number of partitions.

5. EXPERIMENTAL SETUP

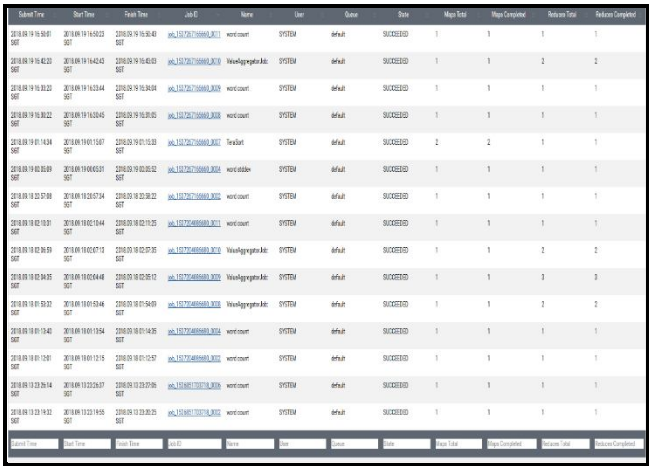
The implementation and evaluation of this work has been conducted in Hadoop version 2.7.1. Several jobs has been evaluated by making the partitions against different sizes of input data ranging from Bytes to Giga bytes. The number of partitions has been varied for each data size to see its effect on the shuffle's phase time. The tested file size's varied from Megabyte to Gigabytes. In addition to it, comparison analysis has been carried out in Microsoft Excel. A variety of examples as shown in Figure 6 have been run across different data sizes.

5.1. Validation and Testing

The purpose of the validation is to ensure the working of the created model. It is important to validate the model because the generalization of the model is necessary. Without doing for cross validation, we will be merely having the information which is tuned for specific data set. In science, theories are judged by its predictive performance. The model obtained from the regression equation has been tested through Regression model validation procedure. Typically, researches are using 70% of the given data for the testing purpose and the 30 % for the validation and testing purpose. The regression forecast model for the shuffle phase time has been tested for the 30% test data and in Figure7 the results has been presented.

Table 1: Symbols Description

Symbol	Description
O	Shuffle Execution Time
α	y-intercept/ constant
β	Slope
u	Dependent variable
r	Regression co-efficient
S_O	standard deviation for O
S_U	standard deviation for u
C	Intermediate variable for calculation
A	Intermediate variable for calculation
B	Intermediate variable for calculation
Ū	Mean value for the partitions
Ō	Mean value for the shuffle execution times
n	Total number of data points



Submit Time	Start Time	Finish Time	Job ID	Name	User	Queue	State	Maps Total	Maps Completed	Reduces Total	Reduces Completed
2018.09.19 16:50:01 SGT	2018.09.19 16:50:23 SGT	2018.09.19 16:50:43 SGT	job_1537267166660_0011	word count	SYSTEM	default	SUCCEEDED	1	1	1	1
2018.09.19 16:42:20 SGT	2018.09.19 16:42:43 SGT	2018.09.19 16:43:03 SGT	job_1537267166660_0010	ValueAggregatorJob:	SYSTEM	default	SUCCEEDED	1	1	2	2
2018.09.19 16:33:20 SGT	2018.09.19 16:33:44 SGT	2018.09.19 16:34:04 SGT	job_1537267166660_0009	word count	SYSTEM	default	SUCCEEDED	1	1	1	1
2018.09.19 16:30:22 SGT	2018.09.19 16:30:45 SGT	2018.09.19 16:31:05 SGT	job_1537267166660_0008	word count	SYSTEM	default	SUCCEEDED	1	1	1	1
2018.09.19 01:14:34 SGT	2018.09.19 01:15:07 SGT	2018.09.19 01:15:33 SGT	job_1537267166660_0007	TeraSort	SYSTEM	default	SUCCEEDED	2	2	1	1
2018.09.19 00:05:09 SGT	2018.09.19 00:05:31 SGT	2018.09.19 00:05:52 SGT	job_1537267166660_0004	word stddev	SYSTEM	default	SUCCEEDED	1	1	1	1
2018.09.18 20:57:08 SGT	2018.09.18 20:57:34 SGT	2018.09.18 20:58:22 SGT	job_1537267166660_0002	word count	SYSTEM	default	SUCCEEDED	1	1	1	1
2018.09.18 02:10:31 SGT	2018.09.18 02:10:44 SGT	2018.09.18 02:11:25 SGT	job_1537204086680_0011	word count	SYSTEM	default	SUCCEEDED	1	1	1	1
2018.09.18 02:06:59 SGT	2018.09.18 02:07:13 SGT	2018.09.18 02:07:35 SGT	job_1537204086680_0010	ValueAggregatorJob:	SYSTEM	default	SUCCEEDED	1	1	2	2
2018.09.18 02:04:35 SGT	2018.09.18 02:04:48 SGT	2018.09.18 02:05:12 SGT	job_1537204086680_0009	ValueAggregatorJob:	SYSTEM	default	SUCCEEDED	1	1	3	3
2018.09.18 01:53:32 SGT	2018.09.18 01:53:46 SGT	2018.09.18 01:54:09 SGT	job_1537204086680_0008	ValueAggregatorJob:	SYSTEM	default	SUCCEEDED	1	1	2	2
2018.09.18 01:13:40 SGT	2018.09.18 01:13:54 SGT	2018.09.18 01:14:35 SGT	job_1537204086680_0004	word count	SYSTEM	default	SUCCEEDED	1	1	1	1
2018.09.18 01:12:01 SGT	2018.09.18 01:12:15 SGT	2018.09.18 01:12:57 SGT	job_1537204086680_0002	word count	SYSTEM	default	SUCCEEDED	1	1	1	1
2018.09.13 23:26:14 SGT	2018.09.13 23:26:37 SGT	2018.09.13 23:27:06 SGT	job_1536851703718_0006	word count	SYSTEM	default	SUCCEEDED	1	1	1	1
2018.09.13 23:19:32 SGT	2018.09.13 23:19:55 SGT	2018.09.13 23:20:25 SGT	job_1536851703718_0002	word count	SYSTEM	default	SUCCEEDED	1	1	1	1

Figure 6: Simulation results

6. RESULTS AND ANALYSIS

It has been observed by making the partitions of the output data from the mappers has a great impact on the data traffic migration time. The results are extracted from the logs after running the various Hadoop MapReduce Jobs. There is a lot of information in the logs after executing the job i.e. containers, counters used, memory, total time elapsed, reduce time, map time and the shuffle time. For this study, particularly the information about the shuffle time and total execution time has been extracted out. Number of partitions are varied from 1 to 10. The input data size has been varied from Bytes to Gigabytes(GB).

For analytical analysis, the technique widely adopted by the research community i.e. regression analysis has been used. Regression analysis has been used for predictive analysis and for the qualification of a variable dependence. The regression analysis has been done over the simulations results to verify the effect of number of partitions on the shuffle phase time. The results of regression analysis are shown in Table 2. The prediction value i.e. the P-value of the regression has been used for the qualification criteria for a factor. If the P-value against a specific factor is greater than 0.05 then the factor has no significant contribution towards the testing experiment and that factor can be ignored even in analysis. The P-value for the number of partitions against the shuffle time is given in Table II which is less than 0.05. Thus it has been statistically proved that it will affect the shuffle time.

The core idea of making the partitions and modeling the shuffle time is to increase the overall performance of MapReduce. The proposed model mathematically models the data traffic migration time through the shuffle phase. The model has been built over the training sets of the data by employing the machine learning techniques termed as regression technique. It employs linear regression (LR) technique to estimate the shuffle phase time with varied number of partitions. The performance of the model has been evaluated on the test data set which is not employed to train the model. The data sets have been collected by historical MapReduce job execution records. For this, different job scenarios have been used with varying the input data size and number of partitions for each data size.

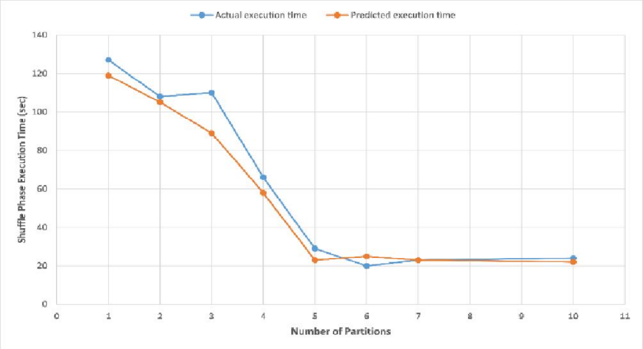


Figure 7: Comparison analysis between predicted and actual execution time

Table 2: Regression analysis results

	Coefficients	**Standard Error**	**t Stat**	**P-value**	**Lower 95%**	**Upper 95%**
Intercept	32.02298612	4.0313664	7.943457002	7.2553 4E-09	23.78983	40.2561348
Number of partitions	0.943005374	0.0802435	11.75178606	9.3901 7E-13	0.779126	1.106884626

7. CONCLUSION

MapReduce has been considered as the promising and acceptable platform for big data processing. MapReduce executes the tasks in three main phases: Map, Shuffle and Reduce. In this paper, the number of partitions for its effect on Shuffle phase execution time has been evaluated. Several simulations has been done for different number of partitions for each data size. While the data size has been varied from Megabytes to Gigabytes. It has been observed that the variation of the number of partitions impacts the shuffle phase execution time and thus the overall performance. Incorporating the partitions for Shuffle traffic has reduced the network traffic and shuffle time execution. In addition, Machine learning technique that is, Regression has been used for the qualifying criteria. The evaluation results evidently shows the dependence of the shuffle phase execution time on the number of partitions. The results are also validated through regression analysis P-value. From the Table II we can see that P-value is less than 0.05 which means number of partitions contributes as major to shuffle phase execution time. Consequently, it has a substantial effect on overall performance of the Hadoop MapReduce also. Thus, this paper has provided a model for the estimation of shuffle time encountering the number of partitions. The model has been made by using the machine learning technique called regression technique. The model has been validated and compared with the test data.

ACKNOWLEDGEMENTS

The authors would like to express their appreciation to the Universiti Teknologi Malaysia for the support under Vot No (Q.J130000.2545.19H80.

REFERENCES

1. P. Koutroumpis, A. Leiponen, and L. D. Thomas, “The (Unfulfilled) Potential of Data Marketplaces: The Research Institute of the Finnish Economyo,” 2017.
2. C. Delimitrou, and C. Kozyrakis, *Quasar:* “Resource-efficient and QoS-aware cluster management,” presented at the *ACM SIGPLAN Notices*, 2014. https://doi.org/10.1145/2541940.2541941
3. Y. Balagoni, and R. R. Rao, “Locality-Load-Prediction Aware Multi-Objective Scheduling in Heterogeneous Cloud Environment,” *Indian Journal of Science and Technology*, 10(9), 2017. https://doi.org/10.17485/ijst/2017/v10i9/106576
4. Q. Althebyan, Y. Jararweh, Q. Yaseen, O. AlQudah, and M. Al□Ayyoub, “Evaluating map reduce tasks scheduling algorithms over cloud computing infrastructure,” *Concurrency and Computation: Practice and Experience*, 27(18), 5686-5699, 2015. https://doi.org/10.1002/cpe.3595
5. Z. Guo, G. Fox, and M. Zhou, “Investigation of data locality in mapreduce,” presented at the *Cluster, Cloud and Grid Computing (CCGrid), 2012 12th IEEE/ACM International Symposium*, 2012. https://doi.org/10.1109/CCGrid.2012.42
6. Z. Tang, M. Liu, A. Ammar, and K. Li, “An optimized MapReduce workflow scheduling algorithm for heterogeneous computing,” *The Journal of Supercomputing*, 72(6), 2059-2079, 2016. https://doi.org/10.1007/s11227-014-1335-2
7. H. Ke, P. Li, S. Guo, and M. Guo, “On traffic-aware partition and aggregation in mapreduce for big data applications,” *IEEE Transactions on Parallel and Distributed Systems*, 27(3), 818-828, 2016. https://doi.org/10.1109/TPDS.2015.2419671
8. N. Tiwari, S. Sarkar, U. Bellur, and M. Indrawan, “Classification framework of MapReduce scheduling algorithms” *ACM Computing Surveys (CSUR)*, 47(3), 49, 2015. https://doi.org/10.1145/2693315
9. S. Neelakandan, S. Divyabharathi, S. Rahini, and G. Vijayalakshmi, “Large scale optimization to minimize network traffic using MapReduce in big data applications,” presented at the *Computation of Power, Energy Information and Commuincation (ICCPEIC), 2016 International Conference*, 2016.
10. H. Fu, H. Chen, Y. Zhu, and W. Yu, “FARMS: Efficient mapreduce speculation for failure recovery in short jobs,” *Parallel Computing*, 61, 68-82, 2017.
11. H. Xu, and W. C. Lau, “Optimization for speculative execution in big data processing clusters,” *IEEE Transactions on Parallel and Distributed Systems*, 28(2), 530-545, 2017.
12. A. Shabbir, K. Abu, and R. Zahilah, “Replication Effect over Hadoop MapReduce Performance using Regression Analysis,” *Int. J. Comput. Appl.*, vol. 181, no. 24, pp. 33–38, 2018. https://doi.org/10.5120/ijca2018918034
13. G. Ananthanarayanan, A. Ghodsi, S. Shenker, and I. Stoica, “Effective Straggler Mitigation: Attack of the Clones”, presented at the *NSDI*, 2013.
14. M. Khan, Y. Jin, M. Li, Y. Xiang, and C. Jiang, “Hadoop performance modeling for job estimation and resource provisioning,” *IEEE Transactions on Parallel and Distributed Systems*, 27(2), 441-454, 2016.
15. A. Shabbir, K. Abu, R. Zahilah, and M. Siraj, “Resource Management in Cloud Data Centers,” *Int. J. Adv. Comput. Sci.* Appl., vol. 9, no. 10, pp. 416–421, 2018. https://doi.org/10.14569/IJACSA.2018.091051
16. P. Géczy, “Big data characteristics,” *The Macrotheme Review*, 3(6), 94-104, 2014.
17. F. Gens, and I. Predictions, Team. IDC Predictions, 2015.
18. S. Solutions, “What is linear regression,” 2013.
19. Tabachnick and Fidell, “DSS - Introduction to Regression,” 1989. [Online]. Available: https://dss.princeton.edu/online_help/analysis/regression_intro.htm. [Accessed: 30-Jun-2019].
20. S. Mustafa, I. Elghandour, and M. A. Ismail, “A Machine Learning Approach for Predicting Execution Time of Spark Jobs,” *Alexandria Eng.* J., vol. 57, no. 4, pp. 3767–3778, Dec. 2018. https://doi.org/10.1016/j.aej.2018.03.006
21. “7 Regression Types and Techniques in Data Science.” [Online]. Available: https://www.analyticsvidhya.com/blog/2015/08/comprehensive-guide-regression/. [Accessed: 30-Jun-2019].
22. N. B. Rizvandi, J. Taheri, R. Moraveji, and A. Y. Zomaya, “On modelling and prediction of total CPU

usage for applications in mapreduce environments,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),* vol. 7439 LNCS, no. PART 1, pp. 414–427, 2012,. https://doi.org/10.1007/978-3-642-33078-0_30

23. M. A. Arasi and S. Babu, “Survey of Machine Learning Techniques in Medical Imaging,” *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 8, no. 5, pp. 2107-2116, 2019. https://doi.org/10.30534/ijatcse/2019/39852019
24. S. K. Trisal and A. Kaul, “Dynamic Behavior Extraction from Social Interactions Using Machine Learning and Study of Over Fitting Problem,” *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 8, no. 5, pp. 2205-2214, 2019. https://doi.org/10.30534/ijatcse/2019/54852019