Performance Modeling of MapReduce Jobs for Resource Provisioning

Sneha Shegar[1], Prof. K. S. Kore[2]

[1]Student, ME Computer, [2]Assistant Professor
SPCOE, Department Of Computer Engineering, Otur

***Abstract*: MapReduce has become a major computing model for data intensive applications. Hadoop, an open source implementation of MapReduce, has been adopted by an increasingly growing user community. Cloud computing service Providers such as Amazon EC2 Cloud offer the opportunities for Hadoop users to lease a certain amount of resources and pay for their use. However, a key challenge is that cloud service providers do not have a resource provisioning mechanism to satisfy user jobs with deadline requirements. Currently, it is solely the user's responsibility to estimate the required amount of resources for running a job in the cloud. This paper presents a Hadoop job performance model that accurately estimates job completion time and further provisions the required amount of resources for a job to be completed within a deadline. The proposed model builds on historical job execution records and employs Locally Weighted Linear Regression (LWLR) technique to estimate the execution time of a job. Furthermore, it employs Lagrange Multipliers technique for resource provisioning to satisfy jobs with deadline requirements. The proposed model is initially evaluated on an in-house Hadoop cluster and subsequently evaluated in the Amazon EC2 Cloud.**
Experimental results show that the accuracy of the proposed model in job execution estimation is in the range of 94.97 and 95.51 percent, and jobs are completed within the required deadlines following on the resource provisioning scheme of the proposed model.

***Keywords*: Cloud computing, Hadoop MapReduce, Performance Modeling, Job Estimation, Resource Provisioning.**

1. Introduction

MANY organizations are continuously collecting massive amounts of datasets from various sources such as the World Wide Web, sensor networks and social networks. The ability to perform scalable and timely analytics on these unstructured datasets is a high priority task for many enterprises. It has become difficult for traditional network storage and database systems to process these continuously growing datasets. MapReduce [1], originally developed by Google, has become a major computing model in support of data intensive applications. It is a highly scalable, fault tolerant and data parallel model that automatically distributes the data and parallelizes the computation across a cluster of computers. Among its implementations such as Mars [2] and Hadoop, Hadoop has received a wide uptake by the community due to its open source nature. One feature of Hadoop MapReduce is its support of public cloud computing that enables the organizations to utilize cloud services in a pay-as-you-go manner. This facility is beneficial to small and medium size organizations where the setup of a large scale and complex private cloud is not feasible due to financial constraints. Hence, executing Hadoop MapReduce applications in a cloud environment for big data analytics has become a realistic option for both the industrial practitioners and academic researchers. For example, Amazon has designed Elastic MapReduce (EMR) that enables users to run Hadoop applications across its Elastic Cloud Computing (EC2) nodes. The EC2 Cloud makes it easier for users to set up and run Hadoop applications on a large-scale virtual cluster. To use the EC2 Cloud, users have to configure the required amount of resources (virtual nodes) for their applications. However, the EC2 Cloud in its current form does not support Hadoop jobs with deadline requirements. It is purely the user's responsibility to estimate the amount of resources to complete their jobs which is a highly challenging task. Hence, Hadoop performance modeling has become a necessity in estimating the right amount of resources for user jobs with deadline requirements. It should be pointed out that modeling Hadoop performance is challenging because Hadoop jobs normally involve multiple processing phases including three core phases (i.e., map phase, shuffle phase and reduce phase) [3]. Moreover, the first wave of the shuffle phase is normally processed in parallel with the map phase (i.e., overlapping stage) and the other waves of the shuffle phase are processed after the map phase is completed (i.e., non-overlapping stage).
To effectively manage cloud resources, several Hadoop performance models have been proposed. However, these models do not consider the Overlapping and non-overlapping stages of the shuffle phase which leads to an inaccurate estimation of job execution [4].
Recently, a number of sophisticated Hadoop performance models are proposed . Starfish collects a running Hadoop job profile at a fine granularity with detailed information for job estimation and optimization. On the top of Starfish, Elasticiser is proposed for resource provisioning in terms of virtual machines. However, collecting the detailed execution profile of a Hadoop job incurs a high overhead which leads to an overestimated job execution time. The HP model considers both the overlapping and non-overlapping stages and uses simple linear regression for job estimation. This model also estimates the amount of resources for jobs with deadline requirements [5]. CRESP estimates job execution and sup-ports resource provisioning in terms of map and reduce slots. However, both the HP model and CRESP ignore the impact of the number of reduce tasks on job performance. The HP model is restricted to a constant number of reduce tasks, whereas CRESP only considers a single wave of the reduce phase. In CRESP, the number of reduce tasks has to be equal to number of reduce slots. It is unrealistic to config-ure either the same number of reduce tasks or the single wave of the reduce phase for all the jobs. It can be argued that in practice, the number of reduce tasks varies depending on

the size of the input dataset, the type of a Hadoop application (e.g., CPU intensive, or disk I/O intensive) and user requirements. Furthermore, for the reduce phase, using multiple waves generates better performance than using a single wave especially when Hadoop processes a large data-set on a small amount of resources [6],[9]. While a single wave reduces the task setup overhead, multiple waves improve the utilization of the disk I/O.

Building on the HP model, this paper presents an improved HP model for Hadoop job execution estimation and resource provisioning. The major contributions of this paper are as follows:
The improved HP work mathematically models all the three core phases of a Hadoop job. In contrast, the HP work does not mathematically model the non-overlapping shuffle phase in the first wave.

- The improved HP model employs locally weighted linear regression (LWLR) technique to estimate the execution time of a Hadoop job with a varied num-ber of reduce tasks. In contrast, the HP model employs a simple linear regress technique for job execution estimation which restricts to a constant number of reduce tasks [1].
- Based on job execution estimation, the improved HP model employs Lagrange Multiplier technique to provision the amount of resources for a Hadoop job to complete within a given deadline.

The performance of the improved HP model is initially evaluated on an in-house Hadoop cluster and subsequently on Amazon EC2 Cloud. The evaluation results show that the improved HP model outperforms both the HP model and Starfish in job execution estimation with an accuracy of level in the range of 94.97 and 95.51 percent. For resource provisioning, four job scenarios are considered with a var-ied number of map slots and reduce slots. The experimental results show that the improved HP model is more economi-cal in resource provisioning than the HP model.

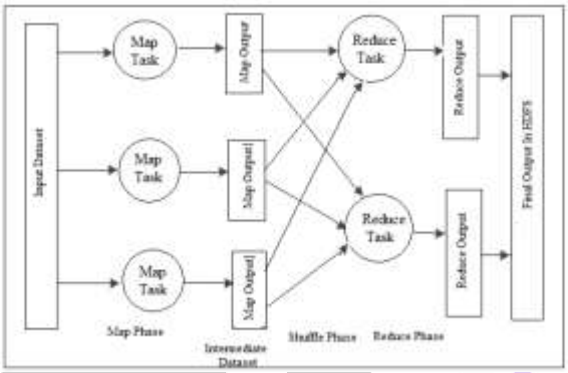

Figure 1. Hadoop Job Execution Flow

The remainder of paper is organized as follows. Section 2 present related work of propose scheme. Section 3 gives brief description of proposed work Section 4 models job phases in Hadoop Section 5 further enhances experimental result of propose scheme. Section 6 present screenshots of result analysis of proposed scheme. Finally, Section 7 concludes the paper and points out some future work.

2. Related Work

In this section, the reference are collected from all conferences, sites, articles, books from the internet which helps to implement the project. For development of this project we referred some of the base papers, ideas which helps in development, testing and deployment phase. For good understanding of the advanced authentication system there are some work on the IEEE international journal that we have referenced are:

Starfish is MADDER (MAD-Magnetism, Agility, Depth) and Self tuning system for big data analytics on big data. The main goal of starfish is to enable Hadoop user and applications to get good performance automatically throughout the data lifecycle without manually understanding the need of the Hadoop user and manually allocate the needs by tuning various knobs [7]. STARFISH collects detailed information of Hadoop job at a very fine granularity for automation of job estimation which is a burden.
The general problem in forming cluster is to determine resource and configure it according to Map Reduce Job in order to meet the desired constraints such as cost, deadline, execution time for given big data analytics referred as cluster sizing problem [10]. In this paper, Elastisizer a system added on the top of Hadoop stack to automate the allocation of resources to meet the desired requirements of massive data analytics by the Elastisizer will provide reliable answers to cluster sizing queries in an automated fashion by using detailed information of Hadoop job profile. In Elastisizer, needs detailed information of job profile which increase the overhead of high execution time.
In this paper, the proposed Hadoop performance model considers both overlapping and non overlapping stages. It uses scaling factor to automatically scale up or scale down the resource allocation within the specified timeframe[8],[11]. Though this Hadoop performance model increase the possibility of automatic resource al- location but it suffers simple linear regression. The HP model is restricted to a constant number of reduce tasks. [12][13] CRESP provides automatic estimation of job execution time and resource provisioning in an optimal manner. In general, the multiple waves generate better performance than single wave of the reduce phase.

In CRESP, the number of reduce slots should be equal to the number of reduce slots and it consider only single wave of the reduce phase. In general, traditional Hadoop cluster environment assumed as homogeneous clusters. But now there is a explosion of heterogeneous Hadoop cluster due to the evolvement of more parallel processing of data intensive application by many companies. Due to heterogeneity of resources in Hadoop cluster, there is an inefficiency of Hadoop job completion time as well as results in the bottleneck of resources.

In [14] proposed bound-based performance modeling of Map Reduce jobs completion times in heterogeneous Hadoop cluster. But they does not explore the properties of Hadoop job profiles as well as does not analysis the breakdown of Hadoop job execution on different types of heterogeneous nodes in the Hadoop cluster.

In this paper [15], they proposed a framework called ARIA (Automatic Resource Inference and Allocation) for MapReduce environments to solve the problem of automatic resource allocation. It contains three inter-related componenets. First, job profilewhich summarizes performance characteristics of the underlying application during the map and reduce stage. Second , MapReduce performance model for the given job and its SLO(Service Level Objective) to estimate the amount of resource required to complete the job within deadline. Finally will be SLO scheduler which is based on EDF(Earliest Deadline First) that determines job ordering and allocating resources. The major disadvantage in this framework was that they did not considered node failures. A proposed a framework called AROMA (Automated Resource Allocation and Configuration of Map Reduce Environment). It was developed to automate resource provisioning in heterogeneous cloud and manipulating configuration of Hadoop parameters for achieving goals but minimizing the incurring cost. The major disadvantage in this system were configuration ineffective and does not support multi-stage job workflows.

3. Proposed Work

In this paper, we propose a scheme called Hadoop Performance Modeling For Job Optimization. In Proposed System we present improved HP model for Hadoop job execution estimation and resource provisioning. The improved HP work mathematically models all the three core phases of aHadoop job. In contrast, the HP work does not mathematically model the non-overlapping shuffle phase in the first wave [1]. The improved HP model employs Locally Weighted Linear Regression (LWLR) technique to estimate the execution time of a Hadoop job with the varied number of reduce tasks. In contrast, the HP model employs a simple linear regress technique for job execution estimation which restricts to a constant number of reduce tasks. Based on the job execution estimation, the improved HP model employs Langrage Multiplier technique to provision the amount of resources for Hadoop job to complete within a given deadline.
The major contributions of this system are as follows:

- The improved HP work mathematically models all the three core phases of a Hadoop job. In contrast, the HP work does not mathematically model the nonoverlapping shuffle phase in the first wave.
- The improved HP model employs Locally Weighted Linear Regression (LWLR) technique to estimate the execution time of a Hadoop job with a varied number of reduce tasks. In contrast, the HP model employs a simple linear regress technique for job execution estimation which restricts to a constant number of reduce tasks.
- Based on job execution estimation, the improved HP model employs Lagrange Multiplier technique to provision the amount of resources for a Hadoop job to complete within a given deadline.

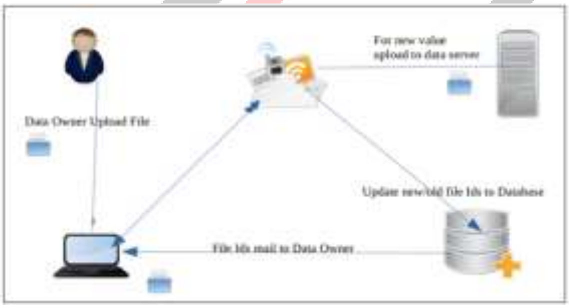

Figure 2. System Architecture

The performance of the improved HP model is initially evaluated on an in-house Hadoop cluster and subsequently on Amazon EC2 Cloud. The evaluation results show that the improved HP model outperforms both the HP model and Starfish in job execution estimation with an accuracy of level in the range of 94.97 percent and 95.51 percent. For resource provisioning, 4 job scenarios are considered with a varied number of map slots and reduce slots. The experimental results show that the improved HP model is more economical in resource provisioning than the HP model.

Advantages Of Proposed System:

1. VMs are configured with a large number of both map slots and reduce slots.
2. The accuracy of the proposed model in job execution estimation is very high.
3. System presents a Hadoop job performance model that accurately estimates job completion time within a deadline.

4. Modeling Map Phase:

In this phase, a Hadoop job reads an input dataset from Hadoop distributed file system (HDFS), splits the input dataset into data chunks based on a specified size and then passes the data chunks to a user-define map function. The map function processes the data chunks and produces a map output. The map output is called intermediate data. The average map output and the total map phase execution time can be computed.
It should be pointed out that the aforementioned models are limited to the case that they only consider a constant number of the reduce tasks. As a result, the impact of the number of reduce tasks on the performance of a Hadoop job is ignored. The improved HP model considers a varied number of reduce tasks and employs a sophisticated LWLR technique to estimate the overall execution time of a Hadoop job.

5. Experimental Result:

The presentation of the enhanced HP form was primarily calculated resting on an in house Hadoop cluster furthermore consequently on top of Amazon Elastic cloud. Following figure shows the recital of the enhanced HP form within work inference of consecutively the Sort appliance resting on the ElasticCloud.

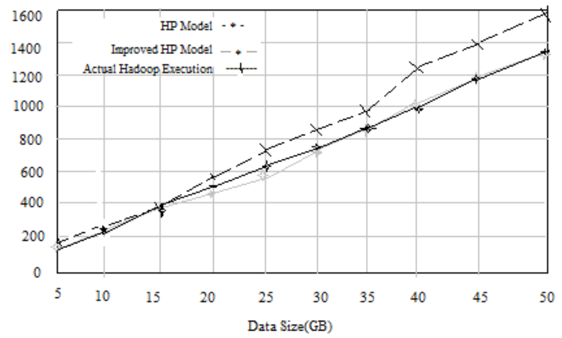

Figure 3. Performance of proposed model in job estimation on the EC2 Cloud

Our investigational consequences demonstrate so as
to the precision of propose method in trade implementation execution having range 94.97 percent plus 95.51percent, as well as works be finished inside the given time limit subsequent on the resource provisioning method of the projected form.

Datasize(GB)	Mapper	Reducer
1	2	2
2	3	3
3	5	5
4	6	6

Table 1: Experimental Result

6. Snapshots:

Figure 1. User Login

Figure 2. Login Successfull

Figure 3. Upload File

Figure 4. Browse File

Figure 5. File Uploaded

Figure 6. File Uploaded Successfully

Figure 7. File Details

Figure 8. Save File

7. Conclusion

Running a MapReduce Hadoop job on a public cloud such as Amazon EC2 necessitates a performance model to esti-mate the job execution time and further to provision a certain amount of resources for the job to complete within a given deadline. This paper has presented an improved HP model to achieve this goal taking into account multiple waves of the shuffle phase of a Hadoop job. The improved HP model was initially evaluated on an in-house Hadoop cluster and subsequently evaluated on the EC2 Cloud. The experimental results showed that the improved HP model outperforms both Starfish and the HP model in job execu-tion estimation. Similar to the HP model, the improved HP model provisions resources for Hadoop jobs with deadline requirements. However, the improved HP model is more economical in resource provisioning than the HP model.

Both models over-provision resources for user jobs with large deadlines in the cases where VMs are configured with a large number of both map slots and reduce slots. One future work would be to consider dynamic overhead of the VMs involved in running the user jobs to minimize resource over-provisioning. Currently the improved HP model only considers individual Hadoop jobs without logical depen dencies. Another future work will be to model multiple Hadoop jobs with execution conditions.

8. Acknowledgement

I express my sincere thanks to my project guide Prof. K. S. Kore who always being with presence & constant, constructive criticism to made this paper. I would also like to thank all the staff of computer department for their valuable guidance, suggestion and support through the paper work, who has given co-operation for the project with personal attention. Above all I express our deepest gratitude to all of them for their kind-hearted support which helped us a lot during paper work. At the last I thankful to my friends, colleagues for the inspirational help provided to me through a paper work.

References

[1] Mukhtaj Khan, Yong Jin, Maozhen Li, Yang Xiang, Changjun Jiang, "Hadoop Performance Modelling for Job Estimation and Resource Provisioning," IEEE Transaction on Parallel and Distributed Systems, Volume: 27, Issue: 2, Feb. 1, 2016.

[2] J. Dean and S. Ghemawat, “MapReduce: Simplified data processing on large clusters,” Commun. ACM, vol. 51, no. 1, pp. 107–113, 2008.

[3] R. L€ammel, “Google’s MapReduce programming model—Revis-ited,” Sci. Comput. Programm., vol. 70, no. 1, pp. 1–30, 2008.

[4] B. He, W. Fang, Q. Luo, N. K. Govindaraju, and T. Wang, “Mars: A MapReduce framework on graphics processors,” in Proc. 17th Int. Conf. Parallel Archit. Compilation Techn., 2008, p. 260.

[5] K. Taura, T. Endo, K. Kaneda, and A. Yonezawa, “Phoenix: A par-allel programming model for accommodating dynamically join-ing/leaving resources,” ACM SIGPLAN Notices, vol. 38, no. 10, pp. 216–229, 2003.

[6] M. Isard, M. Budiu, Y. Yu, A. Birrell, and D. Fetterly, “Dryad: Dis-tributed data-parallel programs from sequential building blocks,” ACM SIGOPS Oper. Syst. Rev., vol. 41, no. 3, pp. 59–72, Mar. 2007.

[7] Apache Hadoop [Online]. Available: http://hadoop.apache.org:/(Last accessed: 1 March 2015).

[8] D. Jiang, B. C. Ooi, L. Shi, and S. Wu, “The performance of Map-Reduce: An in-depth study,” Proc. VLDB Endowment, vol. 3, nos. 1/2, pp. 472–483, Sep. 2010.

[9] U. Kang, C. E. Tsourakakis, and C. Faloutsos, "PEGASUS: Mining Peta-scale Graphs," Knowl. Inf. Syst., vol. 27, no. 2, pp. 303–325, May 2011.
[10] H. Herodotou, F. Dong, and S. Babu, "No One (Cluster) Size Fits All: Automatic Cluster Sizing for Data-intensive Analytics," in Proceedings of the 2nd ACM Symposium on Cloud Computing (SOCC 11), 2011, pp. 114.
[11] A. Verma, L. Cherkasova, and R. H. Campbell, Resource provisioning framework for mapreduce jobs with performance goals, in Proceedings of the 12th ACM/IFIP/USENIX international conference on Middleware, 2011, pp. 165186.
[12] K. Chen, J. Powers, S. Guo, and F. Tian, CRESP: Towards Optimal Resource Provisioning for MapReduce Computing in Public Clouds, IEEE Transcation Parallel Distrib. Syst., vol. 25, no. 6, pp. 1403 1412, 2014.
[13] H. Herodotou, Hadoop Performance Models, 2011. [Online].Available:http://www.cs.duke.edu/starfish/files/hadoopmodels.pdf. [Accessed: 22-Oct-2013].
[14]E. Capriolo, D. Wampler, and J. Rutherglen, Programming Hive, O'Reilley, 2012.
[15] Z. Zhang, L. Cherkasova, and B. T. Loo, "Performance Modeling of MapReduce Jobs in Heterogeneous Cloud Environments," in Proceedings of the 2013 IEEE.