International Journal of Research in Advent Technology (IJRAT) Special Issue
E-ISSN: 2321-9637
Sharadchandra Pawar college of Engineering, Dumbarwadi, Pune 410504, Organizes
National Conference "MOMENTUM-17", 04th& 05th February 2017
Available online at www.ijrat.org

Hadoop Performance Modeling for Job Optimization

Sneha Shegar[1]
1,2Student, SPCOE, Department Of Computer Engineering, Dumbarwadi, Otur

Email: snehashegar0712@gmail.com[1]

Abstract-MapReduce has turned into a noteworthy computing model for information serious applications. Hadoop, an open source execution of MapReduce, has been embraced by an undeniably developing client group. Cloud computing service suppliers, for example, Amazon EC2 Cloud offer the open doors for Hadoop clients to rent a specific measure of assets and pay for their utilization. Be that as it may, a key test is that cloud service suppliers don't have an asset provisioning component to fulfill client occupations with due date prerequisites. Right now, it is exclusively the client's duty to appraise the required measure of assets for running an occupation in the cloud. This paper introduces a Hadoop work execution demonstrate that precisely gauges work consummation time and further arrangements the required measure of assets for a vocation to be finished inside a due date. The proposed model expands on authentic employment execution records and utilizes Locally Weighted Linear Regression (LWLR) system to appraise the execution time of a vocation. Besides, it utilizes Lagrange Multipliers System for asset provisioning to fulfill occupations with due date prerequisites. The proposed model is at first assessed on an in-house Hadoop bunch and therefore assessed in the Amazon EC2 Cloud. Trial comes about demonstrate that the exactness of the proposed model in occupation execution estimation is in the scope of 94.97 and 95.51 percent, and employments are finished inside the required due dates taking after on the asset provisioning plan of the proposed model.

Index Terms-MapReduce; Cloud Computing; Locally Weighted Linear Regression (LWLR); Hadoop.

1. INTRODUCTION

MapReduce 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, Phoenix, Dryad 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.

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). 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).

Objectives:

1. Offer the opportunities for Hadoop users to lease a certain amount of resources and pay for their use.
2. Provisioning mechanism to satisfy user jobs.
3. Accurately estimates job completion time and further provisions the required amount of resources for a job to be completed within a deadline.
4. Providing accuracy for performance of system.

The rest of the paper has been organized as: section 2 highlights the related work along with their downsides, section 3 discusses the proposed work of system. section 4 followed by conclusion and references.

2. RELATED WORK

Hadoop performance modeling is an emerging topic that deals with job optimization, scheduling, estimation and resource provisioning. Recently this topic has received a great attention from the research community and a number of models have been proposed.Morton et al. proposed the parallax model [1] and later the ParaTimer model [2] that estimates the performance of the Pig parallel queries, which can be translated into series of MapReduce jobs. They use debug runs of the same query on input data samples to predict the relative progress of the map and reduce phases. This work is based on simplified suppositions that the durations of the map tasks and the reduce tasks are the same for a MapReduce application. However, in reality, the durations of the map tasks and the reduce tasks cannot be the same because the durations of these tasks are depended on a number of factors. More importantly, the durations of the reduce tasks in overlapping and non-overlapping stages are very different. Ganapathi et al. [3] employed a multivariate Kernel Canonical Correlation Analysis (KCCA) regression technique to predict the performance of Hive query. However, their intention was to show the applicability of KCCA technique in the context of MapReduce.

Kadirvel et al. [4] proposed Machine Learning (ML) techniques to predict the performance of Hadoop jobs. However, this work does not have a comprehensive mathematical model for job estimation. Lin et al. [5] proposed a cost vector which contains the cost of disk I/O, network traffic, computational complexity, CPU and internal sort. The cost vector is used to estimate the execution durations of the map and reduce tasks. It is challenging to accurately estimate the cost of these factors in a situation where multiple tasks compete for resources. Furthermore, this work is only evaluated to estimate the execution times of the map tasks and no estimations on reduce tasks are presented. The later work [6] considers resource contention and tasks failure situations. A simulator is employed to evaluate the effectiveness of the model. However, simulator base approaches are potentially error-prone because it is challenging to design an accurate simulator that can comprehensively simulate the internal dynamics of complex MapReduce applications.

Virajith et al. [7] proposed a system called Bazaar that predicts Hadoop job performance and provisions resources in term of VMs to satisfy user requirements. The work presented in [8] uses the Principle Component Analysis technique to optimize Hadoop jobs based on various configuration parameters. However, these models leave out both the overlapping and non-overlapping stages of the shuffle phase. There is body of work that focuses on optimal resource provisioning for Hadoop jobs. Tian et al. [9] proposed a cost model that estimates the performance of a job and provisions the resources for the job using a simple regression technique. Chen et al. [10] further improved the cost model and proposed CRESP which employs the brute-force search technique for provisioning the optimal cluster resources in term of map slots and reduce slots for Hadoop jobs. The proposed cost model is able to predict the performance of a job and provisions the resources needed. However, in the two models , the number of reduce tasks have to be equal to the number of reduce slots which means that these two models only consider a single wave of the reduce phase. It is arguable that a Hadoop job performs better when multiple waves of the reduce phase are used in comparison with the use of a single, especially in situations where a small amount of resources is available but processing a large dataset. Lama et al. [11] proposed AROMA, a system that automatically provisions the optimal resources and optimizes the configuration parameters of Hadoop for a job to achieve the service level objectives. AROMA uses clustering techniques to group the jobs with similar behaviors. AROMA uses Support Vector Machine to predict the performance of a Hadoop job and uses a pattern search technique to find the optimal set of resources for a job to achieve the required deadline with a minimum cost. However, AROMA cannot predict the performance of a Hadoop job whose resource utilization pattern is different from any previous ones. More importantly, AROMA does not provide a comprehensive mathematical model to estimate a job execution time as well as optimal configuration parameter values of Hadoop.

There are a few other sophisticated models that are similar to the improve HP model in the sense that they use the previous executed job profiles for performance prediction. Herodotou et al. proposed Starfish [12] which collects the past executed jobs profile information at a fine granularity for job estimation and automatic optimization. On the top of the Starfish, Herodotou et al. proposed Elasticiser [13] which provisions a Hadoop cluster resources in term of VMs. However, collecting detailed job profile information with a large set of metrics generates an extra overhead, especially for CPU-intensive applications. As a result, Starfish overestimate the execution time of a Hadoop job. Verma et al. [15] presented the ARIA model for job execution estimations and resource provisioning. The HP model [14] extends the ARIA mode by adding scaling factors to estimate the job execution time on larger datasets using a simple linear regression. The work presented in [16] divides the map phase and reduce phase into six generic sub-phases (i.e. read, collect, spill, merge, shuffle and write), and uses a regression technique to estimate the durations of these sub-phases. The estimated values are then used in the analytical model presented in to estimate the overall job execution time.

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.

3. PROPOSED WORK

The proposed system called Hadoop Performance Modeling for Job Optimization. In Proposed System it present improved HP model for Hadoop job execution estimation and resource provisioning. 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 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:

1. 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.
2. 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.
3. 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 out performs 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.

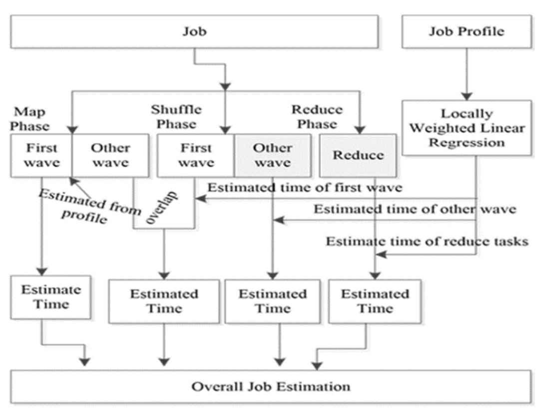

Fig. 1. System Architecture

The estimated values of both the shuffle phase and the reduce phase are used in the improved HP model to estimate the overall execution time of a Hadoop job when processing a new input dataset. Figure shows the overall architecture of the improved HP model, which summarizes the work of the improved HP model in job execution estimation. The boxes in gray represent the same work presented in the HP model. It is worth noting that the improved HP model works in an offline mode and estimates the execution time of a job based on the job profile.

4. PROPOSED ALGORITHM

Algorithm 1: Compute VM Load from data nodes

Input: ith Node input
Output: Idle or Normal Or Overloaded in percent

Compute Load (VM id) :
Weight Degree Inputs: The static parameter comprise the number of CPUs, the CPU dispensation speeds, the reminiscence size, etc. active parameters are the memory consumption ratio, the CPU exploitation ratio, the network bandwidth.
Procedure:

Step 1: Characterize a load limit set: F= {F1,F2. .Fm} with each Fire present the total number of the consideration.

Step 2: Calculate the load capacity as weight Load Degree(N)=(∑αi Fi) , Where, i=(1,...,m).

Step 3: Ordinary cloud partition degree from the node consignment degree statics as:
Load amount avg =∑(i=1..n)Load Degree(Ni)

Step 4: Three height node position are defined Load Degree(N)=0 for Inactive.
- 0 <Load Degree(N)<Load Degree(N) for overfull.
- Load Degree(N)high ≤ Load Degree(N)for overloaded.

Algorithm 2: Equally Spread Current Execution Throttled Load balancing Algorithm

Input: File form user as Fi.
Output: Equally distributed chunks on data servers

Step 1: Read Fi from data owner with size

Step 2: count total number of data nodes Ni

Step 3: for each(score=read each vm node and call to computenode(k))
Read when k==null
End for

Step 4: create data chunks base on server loads score.

Step 5: save all data on data nodes.

5. CONCLUSION

The improved HP model mathematically modeled three core phases i.e. map phase, shuffle phase and reduce phase included overlapping and non-overlapping stages of a Hadoop job. The proposed model employed LWLR to estimates execution duration of a job that takes into account a varied number of reduce tasks The LWLR model was validated through 10-fold cross-validation technique and its goodness of fit was assessed using R-Squared. In future for resources provisioning, the model applied Lagrange Multiplier technique to provision right amount of resources for a job to be completed within a given deadline. The improved HP model was more economical in resource provisioning than the HP model.

Acknowledgments

I express my sincere thanks to my project guide Prof. S. A. Kahate 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 project 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] K. Morton, A. Friesen, M. Balazinska, and D. Grossman, "Estimating the progress of MapReduce pipelines," in Data Engineering

(ICDE), 2010 IEEE 26th International Conference on, 2010, pp. 681–684.

[2] K. Morton, M. Balazinska, and D. Grossman, "ParaTimer: A Progress Indicator for MapReduce DAGs," in Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, 2010, pp. 507–518.

[3] A. Ganapathi, Y. Chen, A. Fox, R. Katz, and D. Patterson, "Statistics-driven workload modeling for the Cloud," in Data Engineering Workshops (ICDEW), 2010 IEEE 26th International Conference on, 2010, pp. 87–92.

[4] S. Kadirvel and J. A. B. Fortes, "Grey-Box Approach for Performance Prediction in Map-Reduce Based Platforms," in Computer Communications and Networks (ICCCN), 2012 21st International Conference on, 2012, pp. 1–9.

[5] X. Lin, Z. Meng, C. Xu, and M. Wang, "A Practical Performance Model for Hadoop MapReduce," in Cluster Computing Workshops (CLUSTER WORKSHOPS), 2012 IEEE International Conference on, 2012, pp. 231–239.

[6] X. Cui, X. Lin, C. Hu, R. Zhang, and C. Wang, "Modeling the Performance of MapReduce under Resource Contentions and Task Failures," in Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on, 2013, vol. 1, pp. 158–163.

[7] J. Virajith, B. Hitesh, C. Paolo, K. Thomas, and R. Antony,"Bazaar:EnablingPredictable Performance in Datacenters," Microsoft Reasearch, MSR-TR- 2012.

[8] H. Yang, Z. Luan, W. Li, D. Qian, and G. Guan, "Statistics-based Workload Modeling for MapReduce," in Parallel and Distributed Processing Symposium Workshops PhD Forum (IPDPSW), 2012 IEEE 26th International, 2012, pp. 2043–2051.

[9] F. Tian and K. Chen, "Towards Optimal Resource Provisioning for Running MapReduce Programs in Public Clouds," in 2011 IEEE 4th International Conference on Cloud Computing, 2011, pp. 155–162.

[10] 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.

[11] P. Lama and X. Zhou, "AROMA: Automated Resource Allocation and Configuration of Mapreduce Environment in the Cloud," in Proceedings of the 9th International Conference on Autonomic Computing, 2012, pp. 63–72.

[12] H. Herodotou, H. Lim, G. Luo, N. Borisov, L. Dong, F. B. Cetin, and S. Babu, "Starfish: A Self-tuning System for Big Data Analytics," in In CIDR, 2011, pp. 261–272.

[13] 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. 1–14.

[14] 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. 165–186.

[15] A. Verma, L. Cherkasova, and R. H. Campbell, "ARIA: automatic resource inference and allocation for MapReduce environments.," in 8th ACM International conference on autonomic computing, 2011, pp. 235–244.

[16] Z. Zhang, L. Cherkasova, and B. T. Loo, "Benchmarking Approach for Designing a Mapreduce Performance Model," in Proceedings of the 4th ACM/SPEC International Conference on Performance Engineering, 2013, pp. 253–258.

[17] Zhang, Cherkasova, and B. T. Loo, "Performance Modeling of MapReduce Jobs in Heterogeneous Cloud Environments," in Proceedings of the 2013 IEEE Sixth International Conference on Cloud Computing, 2013, pp. 839.