Regression Models to Predict the Resource Usage of MapReduce Application

Yangyuan Li, Tien Van DO

Abstract: MapReduce applications are used to process big data. Therefore, the prediction of the resource usage of MapReduce applications is crucially needed. In this paper, we construct multiple linear regression models to predict the resource usage parameters of MapReduce applications.

Keywords: MapReduce application; resource usage parameters; multiple linear regression

Introduction

Map/Reduce[1] is the computational programming model for processing big data. The execution of MapReduce applications may need different resource requirements. Therefore, it is important to understand the behaviour of the resource usage of MapReduce applications. L. Bautista Villalpando et al. [2] modelled the relationship between performance measurements of big data application and the quality concepts of software engineering. Issa, J A et al. [3] proposed an estimation model based on Amdahl's law regression [4] methods to estimate performance and total processing time versus different input sizes for a given processor architecture. He intended to explore the relationship between processing time and input size of data. Glushkova. et al. [5] built a new performance model for Hadoop 2.x, which use the queuing network model to capture the execution flow of a MapReduce job and take architectural changes into account. These models only concerned the performance analysis with the given resource and did not mention the factors of the allocated resource which decrease the performance of Hadoop platform. A resource reuse optimisation mechanism for MapReduce short jobs was developed by Shi et al. [6], which effectively shortened the execution time of these jobs and significantly improved the resource utilisation of a cluster. Nghiem. et al. [7] proposed a novel algorithm for optimal resource provisioning to get the exact amount of task resources, which represented the best trade-off point between performance and energy efficiency for MapReduce jobs. Bakratsas. et al. [8] evaluated the performance of three algorithms when solid state drives and hard disk drives are used to store the real social network data. However, none of previous works characterizes the resource usage of MapReduce applications.
In this paper, we establish regression models to predict the resource usage of three MapReduce applications (Wordcount, Pi and Terasort). Wordcount calculates the number of occurrence of words and the matches to a regex in a text file. The Pi application estimates the value of the Pi number, and Terasort application sorts the generated data from Teragen. The resource usage parameters (the total percentage of time spent of CPU processing job, the total memory usage, the total KB read/second from hard disk and the total KB write/second to hard disks with time resolution 1 s) of three applications are measured in the following configuration:

- Bare metal servers with an Intel CoreTM i5-4670 CPU 3.40GHz 4 cores, 16GB Kingston HyperX Black DDR3 1600MHz RAM and 250GB 7200RPM hard drive.
- Hadoop version 2.7.3 and MapReduce v2 in Ubuntu server 16.04.3 LTS, kernel 4.4.0-62-generic the block size is set to 512MB.

The workload for Wordcount is a text file of 100 GB. The workload of Terasort is 60 GB data generated from Teragen. Application Pi is executed with 2000 map tasks in 10000000 times.

Correlation Matrix

The correlation scatter matrix of Terasort application is depicted in Figure 1. All the correlations between usage parameters and their corresponding lag series show the strong positive linear relevance. The correlation between the read rate and the write rate shows the moderate negative linear relevance. Meanwhile, the correlations between read rate and lagged write rate and between the write rate and the lagged read rate exhibits the similar results. The weak negative relevance is observed in the correlation between the memory usage and the read rate.

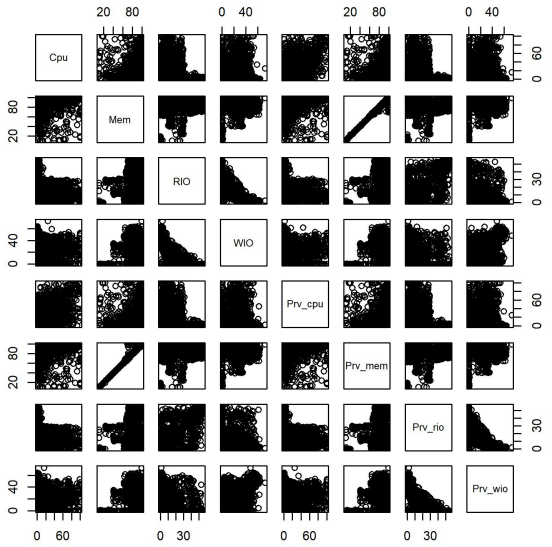

Figure 3: Scatter Matrix of resource usage parameters of Terasort

Linear Regression Models

Multiple linear regression methods are used to model resource usage parameters of MapReduce application. The entire procedure is divided into three parts in order: one is data collection; the second is filter predictors and the third is to fit model. Firstly, data collection is implemented by running Collectl, a light-weight performance monitoring tool, in parallel with the execution of applications. Typically, such kind of collected data is time series data. However, autocorrelation is usually a common characteristic of time series data. Thus, we extract the lagged usage variables which possess the largest autocorrelation coefficient for each parameter, add them to predictor dataset as well. Secondly, multicollinearity problem is taken into account and removed to subject to the important assumption of independence assumption of predictors. Furthermore, the best-subset method and ten-fold cross-validation are applied to filter predictors for regression models. Finally, the least squares methods are used to estimate the regression coefficients which is shown in Table 1.
Table 1 presents the multiple linear regression models and the corresponding residual standard error (RSE) and R^2 for three MapReduce application and regression models. Note that the Z-score (standardized coefficient), a statistical measure, is used to drop irrelevant variable from the set of predictors.

In Table 1, the estimated coefficient shows the strength of linear dependency and its sign represents the dependent direction. Except for the dependency between response and itself previous usage parameter, others dependency exposes the resource bottleneck of the corresponding application in Hadoop MapReduce environment. Meanwhile, according to the estimated coefficient, the optimized suggestion could be given for improving associated usage parameters. For example, read rate, denoted by RIO, in the model $\widehat{CPU} \sim 22.37 + 0.4 \times RIO + 0.6 \times Prv\_cpu$ of Wordcount application has an estimated coefficient 0.4. It means that the average CPU usage might increase 4% when the average read rate increase 10MB/S as well as the corresponding previous CPU usage keeps the fixed value.

Conclusion and Future

We have applied linear regression models to predict the resource usage parameters of MapReduce applications. The regression models could be beneficial to cloud operators in the assignment of MapReduce tasks in the cloud computing platform. In future, we will study the relationship between the stability of model and sampling time, the influence coming from the block size and scale of workload as well.

Name of application	Regression Model	RSE	R^2
Wordcount	$\widehat{CPU} \sim 22.37 + 0.4 \times RIO + 0.6 \times Prv\_cpu$	13.41	39.5%
	$\widehat{MEM} \sim 0.37 + 0.996 \times Prv\_mem$	0.36	99.9%
	$\widehat{RIO} \sim 11 - 6.12 \times WIO + 0.6 \times Prv\_rio$	3.70	38.9%
	$\widehat{WIO} \sim 0.04 + 0.41 \times Prv\_wio$	0.12	17.1%
Terasort	$\widehat{CPU} \sim 2.18 + 0.69 \times Prv\_cpu$	9.41	47.8%
	$\widehat{MEM} \sim 0.77 + 0.99 \times Prv\_mem$	0.89	99%
	$\widehat{RIO} \sim 1.25 - 0.24 \times WIO + 0.96 \times Prv\_rio + 0.27 \times Prv\_wio - 0.01 \times WIO : Prv\_rio$	3.19	90.4%
	$\widehat{WIO} \sim 3.01 + 1.11 \times Prv\_rio - 1.16 \times RIO + 0.93 \times Prv\_wio - 0.01 \times RIO : Prv\_wio$	6.39	83.7%
Pi	$\widehat{CPU} \sim 1.25 + 0.98 \times Prv\_cpu$	7	96.8%
	$\widehat{MEM} \sim 6.85 + 0.07 \times CPU + 0.31 \times Prv\_mem$	1.14	92.8%
	$\widehat{RIO} \sim 0.004 + 0.72 \times Prv\_rio$	0.13	54.2%
	$\widehat{WIO} \sim 0.10 + 0.36 \times Prv\_wio$	0.28	13%

Table 1: List of regression models

References

[1] V. K. Vavilapalli,A. C. Murthy,C. Douglas,S. Agarwal,M. Konar,R. Evans,T. Graves,J. Lowe,H. Shah,S. Seth,B. Saha,C. Curino, O. O'Malley, S. Radia, B. Reed, and E. Baldeschwieler. Yet Another Resource Negotiator, *Apache Hadoop YARN* , in Proceedings of the 4th Annual Symposium on Cloud Computing,p.5:1–5:16, 2013.

[2] L. Bautista Villalpando, A. April, and A. Abran. Performance analysis model for big data applications in cloud computing, *Journal of Cloud Computing: Advances, Systems and Applications*, vol.3, no. 1, pp. 19-38, 2014.

[3] J. A. Issa. Performance Evaluation and Estimation Model Using Regression Method for Hadoop WordCount, IEEE Access,vol. 3, pp. 2784-2793, 2015.

[4] D. P. Rodgers. Improvements in multiprocessor system design, *ACM SIGARCH Computer Architecture News*, vol. 13, no. 3, pp. 225-231, 1985.

[5] D. Glushkova, P. Jovanovic, and A. Abelló. Mapreduce performance model for Hadoop 2.x, *Information Systems*, 2017.

[6] Y. Shi, K. Zhang, L. Cui, L. Liu, Y. Zheng, S. Zhang, and H. Yu. MapReduce short jobs optimization based on resource reuse, *Microprocessors and Microsystems*,vol. 47, no. Part A, pp. 178-187, 2016.

[7] P. P. Nghiem and S. M. Figueira. Towards efficient resource provisioning in MapReduce, *Journal of Parallel and Distributed Computing*,vol. 95, no. Supplement C, pp. 29-41, 2016.

[8] M. Bakratsas, P. Basaras, D. Katsaros, and L. Tassiulas. Hadoop MapReduce Performance on SSDs for Analyzing Social Networks, *Big Data Research*, vol. 1, pp. 1-10, 2017.