2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing

Predicting the Performance of Parallel Computing Models using Queuing System

Shen Chao[1], *Tong Weiqin*[1], *Samina Kausar*[1]
[1]*School of Computer Engineering and Science*
Shanghai University
Shanghai, China
shenchao1010@aliyun.com

***Abstract*—Computing models provide the parallel and distributed algorithms for cloud. The ability to estimate the performance of parallel computing models for efficient resource scheduling is critical. Current techniques for predicting the performance are mostly based on analyzing and simulating. The behavior of parallel computing model directly leads to the diversity of mathematical model. Without a general prediction model, it is very hard to compare fairly different parallel computing models in several critical aspects, including computing capacity, resource configuration, scalability, fault tolerance and so on. In this paper, we design a mathematical model for predicting the performance by using queuing system. We make various computing models as a service system for shielding the diversity. The performance can be accurately estimated with the job waiting time and the job performing time. The heterogeneity of computing nodes may also be considered.**

***Keywords*-parallel computing model; performance prediction; queuing system; service time;**

I. INTRODUCTION

Cloud computing is a prevailing technology to perform massive-scale and complex computing. It facilitates the companies and institutes by diminishing the need of expensive resources. Parallel computing models that provide the parallel and distributed algorithms in the cloud, not only accelerate the processing of large amounts of data, but also facilitate the renting of Infrastructure-as-a-Service offerings. However significant time may be required for big data applications in configuring the cluster and performing the job on the parallel computing models. Furthermore, due to the complexity of factors that influence the performance of parallel computing models, the time for the data processing has non-linear relationship with the amount of resources. Consequently, there is an immense need for an efficient mathematical model that can accurately predict the performance of different parallel computing models and help users to identify and compare different infrastructure configurations for achieving optimal performance.

Currently, based on the existing computing models, such as MapReduce[1], BSP[2] (Bulk Synchronous Parallel model), much performance modeling research[3][4][5] has been carried out to deeply analyze the parallel computing framework. [3]described the default mathematical model for predicting the performance of BSP model. The performance of MapReduce applications that run on cloud has been accurately predicted in[4][5]. But, the techniques for predicting the performance are mostly based on analyzing and simulating the behavior of parallel computing models. The differences of behavior description for the same model or different models directly lead to the diversity of mathematical model.

However for big data analytics, a general performance prediction model is essential for the processing paradigm of different parallel computing models. It is very challenging to compare different parallel computing models in several critical aspects, including computing capacity, scalability, fault tolerance and framework functionality. Additionally, a general model can provide direction to design new parallel computing models and improve the performance of existing computing models.

In this paper, we make the performance of parallel computing models as the service capacity for big data applications, and use the computing nodes to present the service windows. Then we leverage queuing theory to design a general performance prediction model for different parallel computing models. The distinguished features of this general model are as follows: Firstly, it is suitable for several computing models due to its diversity. Secondly, the computing nodes may be physical computer, VM or process, the differences of computing capacity among the computing nodes are considered. Thirdly, having rare parameters, it is easy-to-use for the users.

The rest of the paper is organized as follows. Section II discusses the related work. Section III presents the general prediction model. In Section IV, we describe the detail of prediction model that consists of service waiting time and service performing time. In Section V, we present conclusion and future work.

II. RELATED WORK

Prediction of performance for computing models has attracted the remarkable research attention. There has been many performance modeling research focused on MapReduce[1], which is made as the de facto standard for large scale data analysis. In[4], the authors described the dataflow and cost information at the finest granularity of phases

978-1-4799-8006-2/15 $31.00 © 2015 IEEE
DOI 10.1109/CCGrid.2015.92

within the MapTask and ReduceTask for a complete job execution, furthermore, MapTask is divided into the read and map, collect and spill, and merge phases. ReduceTask is partitioned for shuffle, merge, reduce and write phases. [6]utilized vector-style cost model for calculation of job costs. [5][7]proposed machine-learning based performance prediction model for tuning the configuration parameters of MapReduce. But, these prediction models are all based on the default MapReduce, so the models are not suitable for the improving MapReduce. In[8], the authors proposed a detailed mathematical model of RDMA-enhanced MapReduce based on a number of cluster-wide and job-level configuration parameters, however this model included a lot of complex detail that is not easy for the user to use.

Besides MapReduce, [2]described the default mathematical model for predicting BSP performance, but the prediction model is only used in specific application. [9]analysed the behavior of MapReduce and BSP for performing big data application, and compared their performance, however, the comparison has been done without using a same mathematical model, that leads to unfair comparison result. In addition to, there is also some recent work on analyzing cloud services using queuing analogy [10][11][12]. In this paper, we leverage queuing theory to design a general performance prediction model for a number of parallel computing models that considers the drawbacks of existing prediction models.

III. THE PERFORMANCE PREDICTION MODEL

In this section, we firstly describe the behaviors of parallel computing models with a same paradigm, and then design a queuing system for the various computing models to predict performance.

We divide all the behaviors of computing models into four phases, which are input data, parallel computing, data communication and output result. Based on the four phases and queuing theory, we make the following assumptions:

1) Parallel computing model is as a service system.

2) Computing node is as service window.

3) A job of big data application is as a customer, and the service system used the scheduling of first come first service.

4) The job arriving is according to a Poisson process with the parameter $\lambda(\lambda>0)$

5) A job is divided into $h(h \geq 1)$ tasks. The number of tasks is according to probability distribution $\{h_n$,n =1, 2,...$\}$,and the random variable h_n is a finite positive integer-value.

6) The number of service windows is N, which also represent the number of computing nodes.

Based on the assumptions, the general mathematical model for performance prediction is built by the equations of job waiting time and job performing time in the service system. The detail of the performance model is introduced in the next sections.

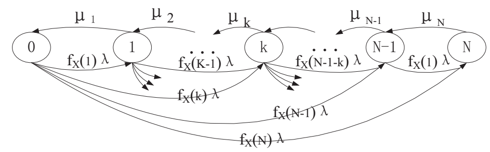

Figure 1. State transition of service system

IV. SERVICE TIME

We firstly describe the job wait time, and then give the detail of getting job perform time.

A. Job Waiting Time

Computing model provides the parallel algorithm for processing big data job, so in this service system, maybe many service windows provide service for one customer at the same time. We denote parameter X as the number of computing nodes which are occupied by one job.

The value of X is a discrete random variable. And the sample space S is $\{1, 2,...,N\}$. So probability mass function could be defined as:

$$f_X(x) = \begin{cases} P(X=x), & x \in S \\ 0, & x \notin S \end{cases} \quad (1)$$

Based on equation(1), make X as the Markov point, the state transition of service system is shown in Fig. 1.

In Fig.1, the state n(n=0, 1, 2,...,N) represent the number of non-idle computing nodes, μ_n represent the service rate of the system at state n. When the service system stays at state n, denote p_n as the steady state probability. And the state transition probability from i to j could be defined as:

$$q_{ij} = \begin{cases} 0, & i-1>j \quad or \quad i=j \\ f_X(j-i)\times\lambda, & i<j \\ \mu_n, & i-1=j \end{cases} \quad (2)$$

In(2) μ_n represents the service rate of the system at state n(0, 1, 2,...,N).

In Fig.1, the differential equation for the state probability p_k at the time t also can be described as:

$$\begin{cases} p'_k(t) = -(\sum_{j=k+1}^{N} q_{kj} + q_{k(k-1)}) \times p_k(t) \\ \quad + \sum_{i=0}^{k-1} q_{ik}p_k(t) + q_{(k+1)k}p_k(t), \quad N>k\geq 1 \\ p'_0(t) = -\sum_{j=1}^{N} q_{0j}p_0(t) + q_{10}p_0(t) \end{cases} \quad (3)$$

When $t \to \infty$, the service system can reach the steady state (at here we omit the process of proof). Equation(3) could be deduced to equation(4).

$$\begin{cases} (\sum_{j=k+1}^{N} q_{kj} + q_{k(k-1)})p_k = \sum_{i=0}^{k-1} q_{ik}p_k \\ \qquad\qquad\qquad +q_{(k+1)k}p_k, N > k \geq 1 \\ \sum_{j=1}^{N} q_{0j}p_0 = q_{10}p_0 \end{cases} \quad (4)$$

The following equations is deduced by equation(4):
For the state 0 :

$$\sum_{j=1}^{N} q_{0j}p_0 = q_{10}p_0 \quad (5)$$

For the state k:

$$\sum_{j=k+1}^{N} q_{kj}p_k = q_{(k+1)k}p_{k+1} + \sum_{j=0}^{k-1}(q_{jk}p_j), \quad (k = 1, 2, 3, ..., N-1) \quad (6)$$

According to the regularity, anther equation is built as follows:

$$\sum_{k=0}^{N} p_k = 1 \quad (7)$$

At the end, solve the equations (2)(5)(6)(7) to obtain the value of p_n(n=1, 2,...,N).

In this paper, we consider the computing node types conclude physical computer, VM and process. When the service system is at the state n(n=1, 2,...,N), the combinations of computing nodes are various, so the μ_n is not a fixed value. At here the value of μ_n is defined as following:

$$\mu_n = min(\mu_N - \mu_{N-1}, \mu_{N-1} - \mu_{N-2}, \mu_{N-2} - \mu_{N-3}, ..., \mu_2 - \mu_1) \times n, \quad (n = 1, 2, 3, ..., N) \quad (8)$$

So, the wait time of job W_q at the state n can be solved with the value of p_n and μ_n.

B. Job Performing Time

A job is divided into $h(h \geq 1)$ tasks, and the complete time of every task is different. We make $T_i(i = 1, 2, 3, ..., h)$ represent the task performing time. Thus the complete time of entire job T can be calculated as follow:

$$T = \begin{cases} max(T_1, T_2, T_3, ..., T_h), & h \leq N \\ \lceil \frac{h}{N} \rceil \times max(T_1, T_2, T_3, ..., T_h), & h > N \end{cases} \quad (9)$$

From equation(9) it is assume that the service system at one round can parallel perform N tasks. If the task number is larger than N, the job will complete in many rounds. In addition, there is a barrier between the rounds for avoiding deadlock.

For getting the T_i, one task is divided into three time phases, which are processing time, I/O time and communication time. The parameter of affecting the time is acquired by benchmark application.

V. CONCLUTION AND FUTURE WORK

In this paper, a general performance prediction model for parallel computing model has been designed. It is suitable for different computing models. The result of performance prediction model is presented by the job waiting time and the job performing time. This general prediction model not only solves the diversity of existing prediction models, but also provides an easy method for users to use.

In future, we will consider more aspects affecting on the performance, such as the resource competition between the jobs, the fluctuation of network bandwidth and so on. These factors will further improve the general prediction model.

REFERENCES

[1] Jeffrey DeanSanjay Ghemawat., "MapReduce: simplified data processing on large clusters," Communications of the ACM, Vol.51, No.1, 2008, pp.107-113.

[2] Leslie G. Valiant, "A bridging model for parallel computation," Communications of the ACM, Vol. 33, No.8, 1990, pp. 103-111.

[3] A. N. Yzelman, R. H. Bisseling, D. Roose, et al, "MulticoreBSP for C: A High-Performance Library for Shared-Memory Parallel Programming," International Journal of Parallel Programming, 2013, pp.1-24.

[4] H. Herodotou, "Hadoop Performance Models," Computer Science Department,Duke University, Tech. Rep. CS-2011-05.

[5] N. Yigitbasi, T. L. Willke, G. Liao, and D. Epema, "Towards Machine Learning-Based Auto-tuning of MapReduce," 21st International Symposium on Modelling, Analysis & Simulation of Computer and Telecommunication Systems, 2013, pp.11-20.

[6] X. Lin, Z. Meng, C. Xu, and M. Wang, "A Practical Performance Model for Hadoop MapReduce," IEEE International Conference on Cluster Computing Workshops, 2012,pp.231-239.

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

[8] Wasi-ur-Rahman, M., Xiaoyi Lu, Islam, N.S., Panda, D.K., "Performance Modeling for RDMA-Enhanced Hadoop MapReduce," 43rd International Conference on Parallel Processing, 2014,pp.50-59.

[9] M Felice Pace, "BSP vs MapReduce,". Procedia Computer Science, Vol.9, 2012,pp. 246-255.

[10] Hamzeh Khazaei, Jelena Misic, Vojislav B. Misic, "Performance analysis of cloud computing centers using M/G/m/m+r queuing system," IEEE Transactions on Parallel and Distributed Systems, 2012, pp.936C943.

[11] Khazaei H, ”Performance analysis of cloud computing centers,” In: Quality, reliability, security and robustness in heterogeneous networks, 2012,pp 251C264.

[12] Xiaodong Liu, Weiqin Tong, Xiaoli Zhi, Fu ZhiRen, Liao WenZhao, ”Performance analysis of cloud computing services considering resources sharing among virtual machines,” J Supercomput, 2014,pp357-374.