Chapter 10
Comparison of Job Scheduling Policies in Cloud Computing

Yang Cao, CheulWoo Ro and JianWei Yin

Abstract Cloud Computing, as the new computing paradigm, provides cost-effective IT operations. In order to efficiently utilize the tremendous capabilities of the Cloud, efficient virtual machines (VMs) allocation and job scheduling mechanism is required. This paper presents an adaptive job scheduling and VM allocation method with threshold. Several scheduling policies are applied. The aim is to achieve effective resource utilization as well as saving users' cost. SimPy is used to build the simulation model.

Keywords Adaptive VM allocation · Scheduling policy · SimPy

10.1 Introduction

As a new prevailing paradigm, Cloud Computing manifests huge potential for more flexible, readily-scalable IT operations with reduction of infrastructure and management cost [1]. Another remarkable feature of Cloud Computing is the on-

This article is a periodic research result of the project on China-Korea Cooperative Study on Key-frame Matching-based Video Motion Retargeting, granted by Liaoning Natural Foundation, Project Number: 2012216031.

Y. Cao · J. Yin
Information Technology College, Eastern Liaoning University,
Dandong 118003, China
e-mail: ldxylccy@gmail.com

J. Yin
e-mail: ldxyyjw@126.com

C. Ro (✉)
Computer Engineering Department, Silla University, Busan 617736, Korea
e-mail: cwro@silla.ac.kr

demand provisioning of resources and pay-as-you-go mode it provides to consumers [2].

Cloud Computing facilitates applications by providing virtualized resources that can be provisioned dynamically. Scheduling in Cloud is to select the best suitable resources for job execution, by taking into consideration some static and dynamic parameters and restrictions of jobs. From the users' perspective, efficient scheduling may be based on parameters like application completion time or application execution cost etc. while from the service providers' perspective, efficient scheduling means that resources are utilized efficiently and to their best capacity so that resource potential is gotten fully excavated. In order to efficiently utilize the tremendous capabilities of the Cloud, user jobs' scheduling and VM allocation methods are required.

In our former work [3], a threshold-based dynamic VMs allocation method is presented. It preloads/revokes VMs when the current workload exceeds the predefined threshold values thus reduces the task waiting time as well as saves user's leasing cost. This paper takes more actual situation into account and supposes each user application as one job which contains different number of tasks with various execution time. Several job scheduling policies are applied and compared. The aim of this paper is to provide an effective VMs allocation method as well as job scheduling policy which can keep the trade off between Cloud service providers and the users. Python-based simulation package—SimPy is adopted to build the model.

The other parts of the paper are organized as follows. Corresponding research work is elaborated in Sect. 10.2. Section 10.3 describes the system modeling and scheduling policies applied. In Sect. 10.4 the measure of interests as well as the performance analysis is elaborated. The conclusion is made in Sect. 10.5.

10.2 Related Work

In cloud computing systems, software is migrating from the desktop to the "clouds", delivering various services to users at any time and anywhere according to their demands. Therefore resource management should be at a finer granularity (at VM layer) and more agile [4]. VMs in a Cloud environment can be selected in various ways, such as random, sequential, round robin etc. VMs allocation method should take into account the current state of each VM in the Cloud environment to minimize the operational cost [5]. The jobs scheduling policies can follow first-come-first-serve (FCFS), shortest-job-first (SJF), largest-job-first (LJF), or priority based [6]. Scheduling algorithm selects job to be executed and the corresponding VM where the job will be allocated to.

Hongbo Liu et al. in their paper [7] propose a novel security constraint model to solve the scheduling problem for workflow applications with security constraints in distributed data intensive computing environments. They also introduce several meta-heuristic adaptations to the particle swarm optimization algorithm with the aim to deal with the formulation of efficient schedules.

Dinesh et al. [8] present a model combining Berger model with Neural Network which would overcome the disadvantage of Berger Model. In this new model, the submitted jobs are classified based on different parameters like bandwidth, memory, Completion time and Resources Utilization, and the classified user tasks are passed to the neural network.

In [9], a family of 14 Cloud scheduling heuristics based on the remaining allocation times of Cloud resources is proposed. The scheduling heuristics consist of two phases: task ordering, where tasks are ordered prior to execution (when possible), and task mapping, where tasks are mapped to available (unoccupied) Cloud resources.

In [10], a real-time Cloud service framework is proposed where each real-time service request is modeled as RT-VM in resource brokers. They investigate power-aware provisioning of virtual machines for real-time services and present several schemes to reduce power consumption by hard real-time services and power-aware profitable provisioning of soft real-time services.

10.3 System Modeling

10.3.1 Modeling Description

Figure 10.1 shows the system architecture which consists of a Job Scheduler (JS), a VMs pool as well as a single cluster of VMs connected them together. The VM pool contains total of 300 distributed VMs; The JS acts on scheduling jobs (user applications) and allocating them to corresponding VMs. The VM pool waits for requests from JS, and dynamically provides extra VMs or revokes excessive VMs when the current workload exceeds some thresholds limitation. Initially, the cluster is empty because there are no VMs leased by the system. The system has the ability to lease new VMs up to a total number of Vmax = 300.

Suppose that the processing ability of each VM is the same (CVM = 10 jobs), each job consists of different number of tasks with different execution time. The tasks in the same job need to be executed concurrently. If there not enough VMs for those tasks running simultaneously, the job will be put in the waiting queue. Suppose each VM has a fixed booting time (BTvm = 10) and after the initialization process the VM can be scheduled to the cluster and execute user jobs. Simulation in Python (SimPy) is adopted to build the simulation model to get analytical data [11].

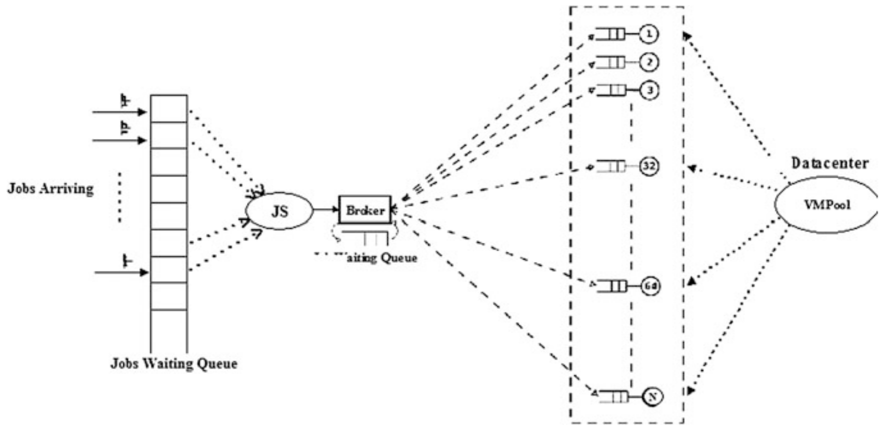

Fig. 10.1 System model

10.3.2 Scheduling Policies

The job scheduling follows several scheduling policies:

FCFS: when job arrives, it follows first-come-first-serve policy to be scheduled and allocated VMs.

SJNF: when job arrives, JS compared the size of the job (number of tasks) to others, the smallest job gets executed first.

SJEF: when job arrives, JS compared the size of the job (total execution time needed by tasks) to others, the smallest job gets executed first.

LJNF: is on the opposite to SJNF which means largest job (number of tasks) gets served first.

LJEF: is on the opposite to SJEF which means largest job (tasks execution time) gets served first.

10.4 Performance Analysis

10.4.1 Measures of Interest

Mean Waiting Time (MWT)

Jobs waiting time Wj of a job j is the time interval between the arrival and the beginning of execution. Its average is defined as:

$MWT = \frac{\sum_{j=1}^{N} Wj}{N}$, Where N is the total number of jobs arrived.

VM Mean Leased Time (MLT)

VMs leased time Vli is the total usage time of a VM i by jobs. Its average is defined as:

$MLT = \frac{\sum_{i=1}^{M} Vli}{M}$, Where M is the total number of VM leased.

Cost-Utilization Efficiency (CUE)

CUE is defined to represent Cost-Utilization Efficiency which is evaluated by combining MLT with the MWT. Metric w is the weight to take use of MLT. Here we let w = 10 to balance MLT and MWT. The formula is as the following:

$$CUE = MLT/w + MWT$$

10.4.2 Input Data

Suppose the VMs pool contains distributed VMs (Vmax = 300), each VM needs 10 s booting time to initialize and become active for use, and each VM has the same capacity to processing CVM = 10 jobs at most; each job has different number of tasks with various execution time Jdur, and the arriving of jobs follows some Exponential-analog distributions which are defined functions in the algorithm. The value arriving rate λ is changed to get different random number of arriving jobs to emulate the uncertainty of realistic environment.

In [3], Different thresholds are tried to get the optimal CUE values changes the optimal one is obtained when setting one threshold with 0.5. So the following experiments are based on this value with one threshold.

10.4.3 Numerical Results

Figures 10.2–10.4 shows the simulation results of MWT, MLT and CUE with different jobs arriving rate λ under different scheduling policies, respectively.

With the increasing of jobs arriving rate, VMs' MLT increases with some extent of decreasing jobs' MWT. Among those five scheduling policies, SJF policies are superior to other policies and get better CUE value (including SJNF and SJEF). LJF policies (LJNF and LJEF) gets higher jobs MWT thus influence the total CUE.

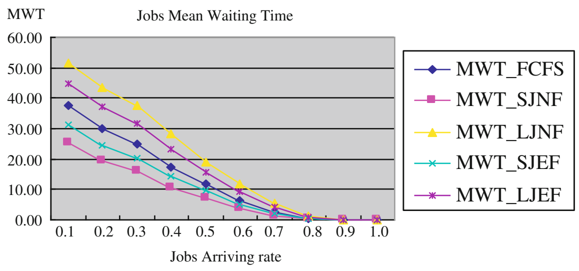

Fig. 10.2 Mean waiting time of jobs

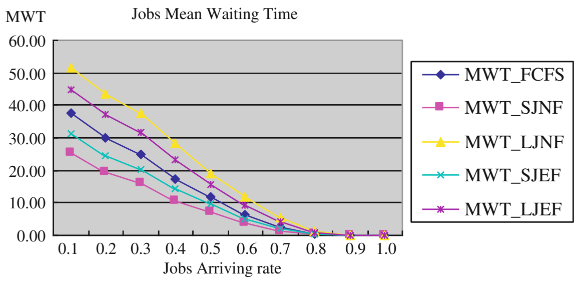

Fig. 10.3 Mean VMs leased time

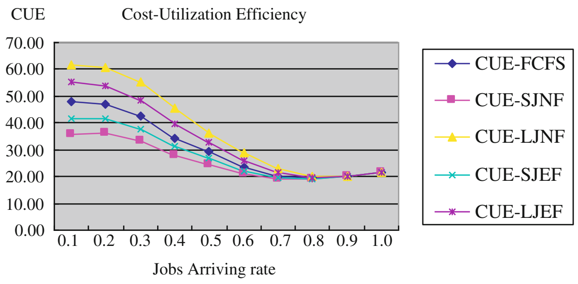

Fig. 10.4 Cost-utilization efficiency

10.5 Conclusion

In this paper, more realistic situations of VMs management in Cloud Computing are taken into account, such as random arriving jobs, each job has different number of tasks with various execution time. Dynamic jobs scheduling and VMs allocation method are presented with threshold limitation as well as trying different scheduling policies. Experiment results show that SJF policy is much fitter to this kind of situation and can have better system performance to achieve higher QoS.

References

1. Vijindra R, Shenai S (2012) Survey on scheduling issues in cloud computing. J Pro Eng 38:2881–2888
2. Mell P, Grance T (2011) The NIST definition of cloud computing. National Institute of Science and Technology (NIST) Special Publication, U.S. Dept. of Commerce, USA, pp 1–7
3. Cao Y, Ro CW (2012) Adaptive scheduling for QoS-based virtual machine management in cloud computing. Intern J Contents 8(4):7–11
4. You X, Wan J, Xu X, Jiang C, Zhang W, Zhang J (2011) ARAS-M: automatic resource allocation strategy based on market mechanism in cloud computing. J Comp 6:1287–1296
5. Patel P, Singh AKr (2012) A survey on resource allocation algorithms in cloud computing environment. J Gold Rese Thou 2:1–9
6. Lucas-Simarro JL, Moreno-Vozmediano R, Montero RS, Llorente IM (2013) Scheduling strategies for optimal service deployment across multiple clouds. J Fut Gene Comp Syst. Available online 28 Jan, 29(6):1431–1441

7. Liu H, Abraham A, Snanel V, McLoone S (2012) Swarm scheduling approaches for workflow applications with security constraints in distributed data-intensive computing environments. J Inf Sci 192:228–243
8. Dinesh K, Poornima G, Kiruthika K (2012) Efficient resources allocation for different jobs in cloud. J Com Appl 56:30–35
9. Octavio J, Garcia G, Sim KM (2012) A family of heuristics for agent-based ELASTIC cloud bag-of-tasks concurrent scheduling. J Fut Gene Comp Syst. Available online 7 Feb
10. Kim KH, Beloglazov A, Buyya R (2011) Power-aware provisioning of virtual machines for real-time cloud services. J Con Comp 23:1491–1505
11. Matloff NS, Introduction to discrete-event simulation and the SimPy language. http://heather.cs.ucdavis.edu/~matloff/156/PLN/DESimIntro.pdf