International Journal of Control and Automation
Vol.7, No.1 (2014), pp.243-250
http://dx.doi.org/10.14257/ijca.2014.7.1.21

Scheduling Analysis of Failure-aware VM in Cloud System

Yang Cao[1], CheulWoo Ro[2] and JianWei Yin[3]

[1,3] *Information Technology College, Eastern Liaoning University, Dandong, China*
[2] *Corresponding Author* *Computer Engineering Department, Silla University, Busan, Korea*

ldxylccy@gmail.com, cwro@silla.ac.kr, ldxyyjw@126.com

Abstract

Clouds focus on flexibility and on-demand provisioning of virtualized resources, which allow users to scale up and down the usage of resources according to their needs. These characteristics give rise to more challenges for resource allocation in cloud system. The optimization of virtual machines (VMs) allocation and the minimization of time and infrastructure cost are still needed to be improved. Meanwhile, failure-aware VMs rescheduling and allocation is another key point to be considered. This paper focuses on the resource failures and presents efficient VMs scheduling mechanisms and policies which try to keep the trade off between system performance and QoS.

Keywords: *VM Scheduling; Failure-Aware; Cloud; SimPy*

1. Introduction

When the concept of Cloud arises with the aim to delivery high quality and transparent services to users, it opens out a new era of computing. Cloud computing is a model providing convenient and on-demand network access to a shared pool of configurable computing resources (storage, applications, networks, servers, services, *etc.*), which can be rapidly delivered and released with minimal management effort or service provider interaction [1]. It has pervaded all kinds of industry fields and has changed our daily life with its simplicity and the affordable price for seamless access to both computational and storage resources. The services that Cloud provides include online backup services, social networking services, personal data services as well as online applications, such as those offered through Microsoft Online Services, and hardware services, such as redundant servers, mirrored websites, and Internet-based clusters. Continuous development of virtualization has deeply accelerated interest in Cloud.

Virtualization Technology (VT) is the technological foundation of Cloud supporting many diverse Clouds which integrate everything into one. It enables virtualization of many factors, such as hardware, software, operating systems, storage, and IT resources, separating them from the physical platform [2]. VT also supports faster job migration within different virtual machines running on the same hardware, resulting in IT which is easier to implement with lower cost to own and manage [3]. VMs can achieve optimal hardware resource utilization since it can be easily moved, copied, and reassigned between host servers. In cloud system, VMs scheduling and allocation means the process of assign available resources to cloud applications over the Internet. VMs scheduling is a crucial part in cloud system. Its availability and efficiency will directly influence the whole system performance [4]. Therefore, much attention has been paid on developing virtualization solutions such as VMware Server, VirtualBox, Xen and KVM, resulting in the emergence of new capabilities like VM checkpoint/restart, VM pause/unpause, and VM live migration, *etc.* [5].

ISSN: 2005-4297 IJCA

The increased functionality and complexity of the Cloud systems accompanies inevitably the resource failures. Failures occurrence has become a challenge which may have negative impacts on system performance in cloud system. In other words, resource failures can lead to degrading performance, terminating user applications execution, data corruption and loss, and deviating Service Level Agreements (SLAs) [6]. Hence, it is crucial for Cloud system to consider failure-aware VM scheduling disciplines which can eliminate these negative influences. As Cloud provides elastic resource demand and release, the adjustments to achieve the elasticity are important to VM scheduling policies [7]. Due to the dynamical features of available resources provided by providers and the resource requirements presented by users, the efficiency of service provisioning and usage is another challenge. In this paper, we focus on the failure-aware VM scheduling mechanisms in cloud system. The aim is to control VM reallocation and tasks migration so as to enhance VM utilization (resource utilization) and seek better trade off between system performance and QoS.

The remainder of the paper is organized as follows. Section 2 elaborates the scheduling policies in cloud system. Section 3 describes the system modeling and several VM scheduling policies. Simulation and results are shown in Section 4 and the conclusion is made in Section 5.

2. Scheduling Policies

In cloud system, a scheduling policy is an algorithm used to determine when a VM is allocated to execute user applications/jobs. Jobs scheduling policies determine the order in which to run jobs from the queue, and determine how cluster resources are allocated to these jobs. In general, the scheduling work is fulfilled by a Job Scheduler.

The Job Scheduler focuses mainly on:

- Throughput - The total number of jobs that complete their execution per time unit.
- VMs Utilization - Percentage of time that VMs are occupied.
- Turnaround Time - Time interval from job submission until its completion. It includes actual processing time plus waiting time.
- Response Time - Time interval taken from when a request was submitted until the first response is engendered.
- Fairness / Waiting Time - Appropriate processing times according to each job's priority. It is the time for which the job remains in the waiting queue.

In practice, these goals often conflict with each other, such as throughput versus response time, and the tradeoff between minimizing mean response time while maintaining fairness is an important design constraint in many applications. So the main purpose of a scheduler is to minimize VMs starvation and keep fairness between user jobs. In other words, the job of the scheduler is to make a compromised decision based on the user's needs and objectives.

The common and representative scheduling policies include: First in first out (FIFO), Shortest Job First (SJF), Static/Fixed Priority Scheduling (SPS), Round Robin (RR), (variations of RR such as Weighted Round Robin (WRR), Deficit Round Robin (DRR), Hierarchical Round Robin (HRR)), Earliest Deadline First (EDF), and Multilevel Feedback Queue scheduling, *etc.* [8].

Service time, waiting time, turnaround time and throughput are some of the metrics used to compare scheduling policies. Any good scheduling discipline should, on the one hand,

maximize resource utilization and throughput, and minimize turnaround time, waiting time and response time, on the other hand.

Jobs scheduling in cloud system is to select the best suitable VMs for user jobs execution with the consideration of some static and dynamic parameters and restrictions of jobs. Especially when resource failures occur, reactive schemes to recover after failure may result in extra cost and negative system performance. From the users' perspective, efficient scheduling may be based on parameters like job completion time or job execution cost. While from the service providers' perspective, efficient scheduling means that VMs are utilized efficiently and to excavate their fully potential. In order to efficiently utilize the tremendous capabilities of the Cloud, efficient jobs scheduling and VMs reallocating schemes with consideration of VM failures are required.

3. System Model

3.1. Modeling Description

Figure 1 shows the system architecture that consists of a VM pool (with total of 100 distributed VMs) which connects Data Center and Job Scheduler together. The Data Center is responsible for scheduling VM in/out from the VMs pool; Job Scheduler schedules user applications (jobs) and allocates them to corresponding VMs. Each user job consists of different number of tasks with various execution time (priorities). The tasks in the same job need to be executed concurrently. If there are not enough VMs for those tasks running simultaneously, the job will be put in the waiting queue. Data Center waits for requests from Job Scheduler and provides extra VMs or revokes excessive VMs based on the requests change. Initially, there are no VMs leased. When the workload increases, the system leases new VMs up to a total number of Vmax = 100. According to the changing system workload, the system leases or withdraws VMs dynamically. However, not every job request will be accepted by the Job Scheduler in case when there are no sufficient VMs for the request, the Job Scheduler will place this request job on the appropriate waiting queue.

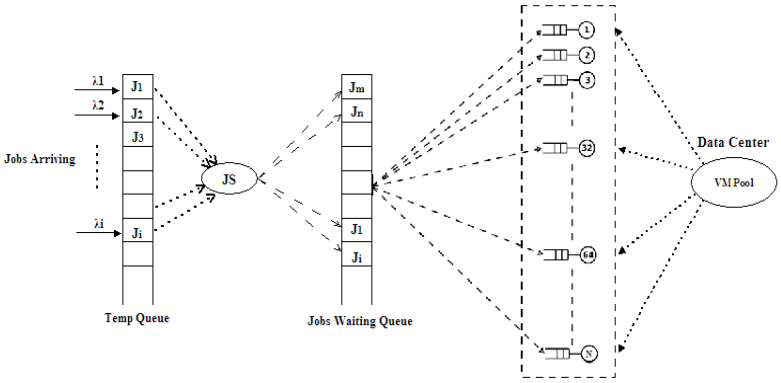

Figure 1. System Architecture

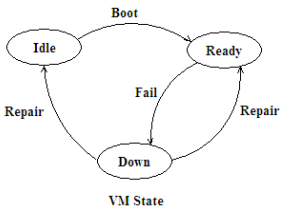

Figure 2. VM State

Each VM has three states as shown in Figure 2: Ready, Idle and Down. "Ready" state means the VM is now ready for use. It is scheduled and allocated to corresponding tasks. When a VM is idle for some time (not executing any tasks for a while), it will be changed to "Idle" state and regained back to the VM Pool. In our former work [9], 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. When new requests arrive and need an "Idle" VM, the VM experiences a boot stage and the state is changed to "Ready". Here booting means that the VM needs some time to get configured and prepared for using. "Down" state means that failure occurs on one of current working VMs, and that VM can't execute its tasks anymore. Through some duration the VM gets repaired. Here we consider two cases. The first case is that after repairing, the VM state changes to "Ready" again and continue to execute its tasks. The other case is that after repairing, the VM is set "Idle" state and back in the VM Pool, and the tasks it runs are migrated to other VM to be executed.

3.2. VMs Rescheduling Schemes

A failure is defined as an event in which the physical resource such as one server fails. It refers to any anomaly caused by hardware or software errors that make service unavailable which leads to VM unavailability. Under this situation, three kinds of VMs rescheduling schemes are taken into account.

Scheme 1 (S1): the first scheme is that when the state of a VM turns into "Down", the snapshot of current tasks running on this VM is captured and restored when that VM get repaired after some time, and the tasks continue to be executed on this VM.

Scheme 2 (S2): the second scheme is that those tasks are migrated to another VM and get execution from the beginning. This migration should be fast and transparent so that users may not perceive the service delay.

Scheme 3 (S3): the third scheme is that when the VM down, the tasks it runs are kept their executing state, migrate to another VM and continue to be executed.

Two normally used jobs scheduling policies are adopted and compared as follows:

FIFO: when job arrives, it follows first-in-first-out policy to be scheduled and allocated to VMs.

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

4. System Analysis

4.1. System Parameters

Suppose the VMs pool contains distributed VMs (V_{max} =100), each VM needs 10s booting time (VM_{BT} = 10sec) to initialize and become active for use, and each VM has the same capacity to processing C_{VM} =10 jobs at most; each job has different number of tasks with various execution time J_{dur}, 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. The time for a VM turns down and gets repaired from "Down" state adopts randint() function which is a random function in Python returning a random int. Here we give the range (1, 100). Python-based simulation package—SimPy (Simulation in Python) is adopted to build the simulation model to get analytical data [10].

4.2. Measures of Interest

- Cloud System Capability Utilization (CSCU)

CSCU is the using rate of cloud system's processing capability to execute tasks. It is calculated as dividing the total VM "Ready" time (SV_r) by all the VMs during system running time.

$CSCU = \frac{\sum_{i=1}^{M} SVr}{S * K}$, Where K is the total number of VMs and S is the total system running time.

- VM Mean Leased Time (VMLT)

VMs leased time V_{li} is the total usage time of a VM i by jobs. Its average is defined as:

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

- Mean Jobs Waiting Time (MJWT)

Jobs waiting time W_j 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.

4.3. Simulation Results

Figures 3~5 shows the simulation results of CSCU, VMLT and MJWT with different jobs arriving rate λ under different scheduling policies, respectively.

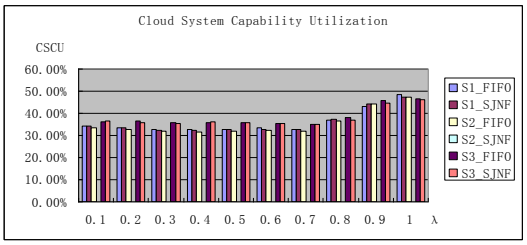

Figure 3. Cloud System Capability Utilization

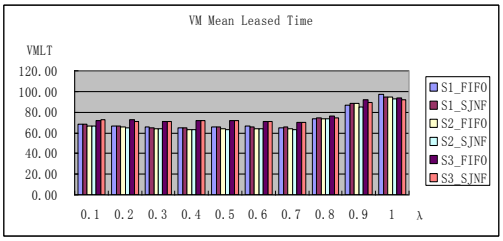

Figure 4. VMs Mean Leased Time

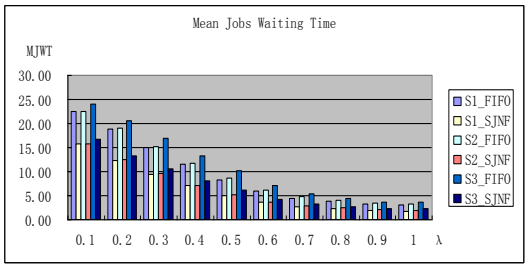

Figure 5. Mean Jobs Waiting Time

With the increasing of jobs arriving rate, all VMs rescheduling schemes with two jobs scheduling policies can have increasing capacity utilization of cloud system. Scheme 2 with two policies shows better VMLT results than Scheme 1 and 3 because when a VM fails, this scheme can migrate tasks quickly to other VMs to continue their execution, while Scheme 3 needs to run those tasks from the beginning which consume extra VM time. Schemes with SJNF policy show better MJWT and Scheme 1 is a stroke above Scheme 2 on this measurement. The reason is that although Scheme 2 and 3 can save time for running jobs, the jobs in the waiting queue are affect by this kind of migration as extra VMs resources are occupied.

5. Conclusion

VM and Job scheduling strategies and policies play important roles in cloud system, especially when considering resources failure situation. In that case, how to dynamically reschedule VMs and allocate jobs to appropriate VMs are the key points. VM rescheduling and allocating is crucial not only to the service providers, but also to users. This paper probes several VM rescheduling schemes with two job scheduling policies when facing with resource failure problems. The aim is to seek better way for efficient utilization of cloud system capability, at the same time, saving user costs.

Acknowledgements

This work was supported by Scientific Research Fund of Liaoning Provincial Education Department (No: L2013498), Liaoning Provincial Science and Technology Plan Project (No: 2012216031).

References

[1] P. Mell and T. Grance, “The NIST Definition of Cloud Computing”, National Institute of Science and Technology (NIST) Special Publication, **(2011)**, pp. 1-3.
[2] A. M. Sampaio and J. G. Barbosa, “Dynamic Power- and Failure-Aware Cloud Resources Allocation for Sets of Independent Tasks”, 2013 IEEE International Conference on Cloud Engineering, IEEE Computer Society, **(2013)**.
[3] M. Cafaro and G. Aloisio, “Grids, Clouds, and Virtualization”, Series: Computer Communications and Networks, Springer, **(2010)**.
[4] S. Tilak and D. Patil, “A Survey of Various Scheduling Algorithms in Cloud Environment”, International Journal of Engineering Inventions, vol. 1, no. 2, **(2012)**, pp. 36-39.
[5] K. Ye, X. Jiang, S. Chen, D. Huang and B. Wang, “Analyzing and Modeling the Performance in Xen-Based Virtual Cluster Environment”, Proc. 12th IEEE International Conference on High Performance Computing and Communications (HPCC), doi: 10.1109/HPCC.2010.79, **(2010)** September, pp. 273-280.
[6] B. Javadi, J. Abawajy and R. Buyya, “Failure-aware resource provisioning for hybrid Cloud infrastructure”, J. Parallel Distrib. Comput., vol. 72, **(2012)**, pp. 1318-1331.
[7] S. Tilak and D. Patil, “A Survey of Various Scheduling Algorithms in Cloud Environment”, International Journal of Engineering Inventions, vol. 1, no. 2, **(2012)**, pp. 36-39.

[8] P. Patel and A. K. Singh, "A Survey on Resource Allocation Algorithms in Cloud Computing Environment", Golden Research Thoughts, vol. 2, no. 4, **(2012)**, pp. 1-9.
[9] C. Yang and C. W. Ro, "Adaptive Scheduling for QoS-based Virtual Machine Management in Cloud Computing", International Journal of Contents, vol. 8, no. 4, **(2012)**, pp. 7-11.
[10] https://bitbucket.org/simpy/simpy/, **(2013)**.

Authors

Yang Cao

She works at Eastern Liaoning University in China. She received the PhD degree from Silla University in February 2013. Her main research interests include cloud computing, modeling and analysis of communication systems.

Cheulwoo Ro

He received his B.S., M.S. and the Ph.D. degrees from Sogang, Dongguk, Sogang University, Seoul, Korea in 1980, 1982, and 1995, respectively. From 1982 to 1991, he joined ETRI (Electronics and Telecommunications Research Institute) as a senior research member. Since 1991, He has been a professor of Silla University. His research interests include performance analysis of wireless networks, Petri Nets modeling, and embedded network system.

Jianwei Yin

He works at Eastern Liaoning University in China. He received the M.S degree from Northeastern University in June 2010. His main research interests include cloud computing, systems development and design.