Future Generation Computer Systems 55 (2016) 74–86

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

A new technique for efficient live migration of multiple virtual machines

Gang Sun [a,*], Dan Liao [a,b], Vishal Anand [c], Dongcheng Zhao [a], Hongfang Yu [a]

[a] Key Lab of Optical Fiber Sensing and Communications (Ministry of Education), University of Electronic Science and Technology of China, Chengdu, China
[b] Institute of Electronic and Information Engineering in Dongguan, UESTC, China
[c] Department of Computer Science, The College at Brockport, State University of New York, USA

HIGHLIGHTS

- We propose an improved serial migration strategy for migrating multiple VMs.
- We present the *m* mixed migration strategy for improving the time efficiency.
- For a given tolerable downtime, we calculate *m* to minimize the migration time.
- We evaluate the effectiveness of the proposed approach by using queuing models.

ARTICLE INFO

Article history:
Received 15 May 2015
Received in revised form
8 August 2015
Accepted 6 September 2015
Available online 16 September 2015

Keywords:
Virtual machine
Live migration
Queuing model
Virtualization
Cloud computing

ABSTRACT

Datacenter virtualization technologies have attracted a lot of attention to enable various cloud computing services and to facilitate virtual machine (VM) migration. VM migration can help service providers to achieve the goals of saving energy, enhancing resource efficiency and quality of service (QoS). In order to ensure the QoS, the migration time and the downtime of VM should be considered while implementing the VM migration. Most researches focus on the issue of single VM migration by using the post-copy migration strategy or pre-copy migration strategy. However, there is few research that focuses on the problem of live migration for multiple VMs. Therefore, in this paper, we first propose an improved serial migration strategy and introduce the post-copy migration scheme into it. We then propose the *m* mixed migration strategy that is based on the improved serial migration strategy and the parallel migration strategy. Furthermore, we develop queuing models (i.e., the M/M/C/C and the M/M/C queuing models) to quantify performance metrics, such as the blocking ratio and average waiting time of each migration request. We evaluate the performance of the proposed migration strategy by conducting mathematical analysis, the numerical results show that our proposed strategy outperforms the existing approach.

1. Introduction

In recent years there has been an exponential growth in cloud computing. Today most businesses use some form of cloud computing (or cloud model) and services to access various kinds of shared resources or services to reduce their operating expenditures (OPEX) and capital expenditures (CAPEX). Hence, cloud computing has received a lot of attention from the industry as well as academia. The main enabling technology for cloud computing is virtualization. Virtualization provides a lot of flexibility by separating the physical computing and networking resources into virtual resources that can be managed and used independently in a variety of ways. Thus, "cloud providers" offer services to users by using virtualization technologies and using large datacenters that have huge amounts of computing, storage and bandwidth resources. Previous studies have only considered the case where virtual resources are provided by a single data center. In practice [1], service providers should have the ability to provision requested virtual resources across their distributed infrastructure (i.e., multiple datacenters) to achieve multiple goals including revenue maximization, operational costs reduction, energy efficiency, and green IT, or to simply satisfy geographic location constraints of users. Therefore, some recent studies [1,2] on cloud computing have focused more on multiple datacenters.

As the scale of datacenters has grown, more and more research has focused on saving energy and improving the resources

* Corresponding author.
E-mail address: gangsun@uestc.edu.cn (G. Sun).

http://dx.doi.org/10.1016/j.future.2015.09.005

utilization. In addition, cloud providers have to comply with the service level agreements (SLAs) that have been agreed upon with the customers or users. To maintain and/or improve the SLAs, researchers have proposed virtual machine (VM) migration technologies [3,4]. When there are failures or faults in the datacenter, we can migrate VMs to improve the reliability of the datacenter. A VM is a complete computer system that is simulated in software and has complete hardware system functions and runs in an isolated environment. The working of a VM is akin to the working of a real computer system, for example, we can install operating systems and applications in the VM, access network resources through the VM, and so on. Large computers or servers may run various different VMs providing different services. VM migration technologies enable the migration of VMs from one server to another, so as to guarantee quality of service (QoS) and maintain SLAs. Thus, VM migration technologies have received much attention in recent years. Most research consider the live migration strategy for migrating single VM. However, in some real applications, multiple cooperating VMs are necessary to provide certain services, we thus may need to consider migration of groups of cooperating VMs [5].

Therefore, we need to further study the multiple VM migration strategy to optimize the migration performance. In this paper, we study the problem of efficient live migration of multiple VMs. We first propose an improved serial migration strategy for multiple VMs migration. Then we present the *m* mixed migration strategy. In the *m* mixed migration strategy, we migrate *m* VMs by using the parallel migration strategy, while migrate the rest of VMs by using the improved serial migration strategy. In the parallel migration process [5], all VMs are migrated simultaneously, and these VMs together share the migration bandwidth resources. Whereas in the improved serial migration process, VMs are migrated one by one, and each VM occupies all of the migration bandwidth resources. Thus, the *m* mixed migration strategy is based on the improved serial migration strategy and the parallel migration strategy. This *m* mixed migration strategy aims to minimize the total migration time while satisfying the maximum downtime constraint that users agreed with the service providers in the SLAs. In order to evaluate the blocking ratio, we model the problem as the M/M/C/C queuing model. We use the M/M/C multi-server queuing model to evaluate the average waiting queue length and the average waiting time of the arriving migration requests. The average waiting queue length represents the number of requests waiting for service in the queue, and the average waiting time represents the queuing time.

The main contributions of this paper are as follows:

- In order to improve the performance of the serial migration strategy, we introduce the post-copy migration strategy into the serial migration strategy [5], and propose the improved serial migration strategy for migrating multiple VMs.
- In order to satisfy the maximum downtime constraint and minimize the total migration time, we present the *m* mixed migration strategy. The *m* mixed migration strategy divides the performance of the improved serial migration strategy and the parallel migration strategy into different grades. Therefore, the *m* mixed migration strategy can provide more choices to service providers and users.
- We use two classic queuing models to model our problem and calculate various performance metrics to evaluate the effectiveness of our approaches.

The rest of this paper is organized as follows. Section 2 discusses the related work. Section 3 presents the strategies for migrating multiple VMs in multiple datacenters. Section 4 calculates various performance metrics by using queuing models. The numerical results and discussions are given in Section 5. Section 6 shows a way to implement our proposed approaches. Section 7 concludes this paper.

2. Related work

2.1. VM migration mechanisms

VM migration mechanisms can be divided into two categories. One is the traditional offline migration mechanism. This migration mechanism has to first stop the currently running VMs, and then migrate the virtual machine's memory and status, and finally restart the virtual machine at the new destination server. The other is the online or live migration mechanism. In the offline migration process, the services being provided will be terminated while migrating the VM. Whereas the live migration mechanism can keep the services running without interruption during most of the migration process. There are two major performance parameters to describe and measure the performance of live migration [6]. Total migration time (T_{mig}) measures the total time for migrating all VM memory (the original memory and the dirty/modified memory) and the CPU status. Since the size of VM memory is usually much larger than that of CPU status, the migration time of CPU status is negligible. Hence the T_{mig} usually can be used to measure the total time for migrating the VM memory. The second metric is the downtime (T_{down}), which is the amount of time that the VM will be stopped or slowed down due to the migration. This migration mechanism can further be divided into: the pre-copy migration mechanism [7] and the post-copy migration mechanism [8].

The process of pre-copy strategy [5,7] can be divided into three phases: (i) Copy the VM memory from the source server to the destination server by iterating. This iterative process is described in [5]. During the memory copy process, the source VM will continue running. This phase is also known as the pre-copy phase. (ii) The system will stop the source VM and copy the CPU status and remaining dirty memory from the source VM to the destination VM. This is the stop-and-copy phase. (iii) Restart the VM on destination server, i.e., the restarting phase.

Similarly, the process of post-copy strategy [9,10] also includes three phases: (i) Stop the source VM and copy the CPU state to the destination VM; (ii) Restart the destination VM; and (iii) Copy the VM memory according to the demand. In the post-copy strategy, when the VM is restarted, the VM memory is empty. If the VM tries to access a memory page that has not yet been copied, this memory page needs to be brought from the source VM. However, most of the time, some memory pages will not be used, so we only need to copy the VM memory according to the demand.

2.2. Live migration for single VM

The migration of VMs has been typically done in an offline manner. Once a VM is migrated from one server to another server, then the VM operation will be resumed at the destination server. More recently, online migration technology has become available [11–13], which allows live migrate VMs.

According to the users' demands, service providers usually need to optimize total migration time (T_{mig}) or/and the downtime (T_{down}). For the purpose of this study, we will also focus on the pre-copy and post-copy strategies. These two strategies are being widely used in cloud-based systems. From the procedures, we note that the post-copy strategy has to migrate less memory than the pre-copy strategy does. This is because that ensuring the normal operation of the VM does not need all of the memory, and the post-copy strategy only has to copy the necessary memory on demand for the VM.

The work in [14], described an analytical performance model for single VM live migration. The model measures the VM blocking or rejection probability and total delay. In [15], the researchers tested the migration time of a single VM migration by means of experimentation. In [16], the researchers theoretically analyzed the necessary bandwidth resources to satisfy the constraint of the total migration time and the downtime in single VM live migration process.