Future Generation Computer Systems 37 (2014) 478–487

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Integrating QoS awareness with virtualization in cloud computing systems for delay-sensitive applications

Jenn-Wei Lin [a,*], Chien-Hung Chen [a], Chi-Yi Lin [b]

[a] Department of Computer Science and Information Engineering, Fu Jen Catholic University, Taiwan, ROC
[b] Department of Computer Science and Information Engineering, Tamkang University, Taiwan, ROC

HIGHLIGHTS

- We investigate the QoS-aware virtual machine placement (QAVMP) problem.
- We formulate the QAVMP problem as an Integer Linear Programming (ILP) model.
- We propose a polynomial-time heuristic algorithm to efficiently solve the QAVMP problem.
- A bipartite graph is used to model all possible placement relationships of virtual machines.
- The proposed heuristic algorithm can maximize the profit of cloud provider.

ARTICLE INFO

Article history:
Received 27 December 2012
Received in revised form
3 October 2013
Accepted 13 December 2013
Available online 9 January 2014

Keywords:
Cloud computing
Virtualization
Quality of service
Technique integration
Heuristic algorithm

ABSTRACT

Cloud computing provides scalable computing and storage resources over the Internet. These scalable resources can be dynamically organized as many virtual machines (VMs) to run user applications based on a pay-per-use basis. The required resources of a VM are sliced from a physical machine (PM) in the cloud computing system. A PM may hold one or more VMs. When a cloud provider would like to create a number of VMs, the main concerned issue is the VM placement problem, such that how to place these VMs at appropriate PMs to provision their required resources of VMs. However, if two or more VMs are placed at the same PM, there exists certain degree of interference between these VMs due to sharing non-sliceable resources, e.g. I/O resources. This phenomenon is called as the VM interference. The VM interference will affect the performance of applications running in VMs, especially the delay-sensitive applications. The delay-sensitive applications have quality of service (QoS) requirements in their data access delays. This paper investigates how to integrate QoS awareness with virtualization in cloud computing systems, such as the QoS-aware VM placement (QAVMP) problem. In addition to fully exploiting the resources of PMs, the QAVMP problem considers the QoS requirements of user applications and the VM interference reduction. Therefore, in the QAVMP problem, there are following three factors: resource utilization, application QoS, and VM interference. We first formulate the QAVMP problem as an Integer Linear Programming (ILP) model by integrating the three factors as the profit of cloud provider. Due to the computation complexity of the ILP model, we propose a polynomial-time heuristic algorithm to efficiently solve the QAVMP problem. In the heuristic algorithm, a bipartite graph is modeled to represent all the possible placement relationships between VMs and PMs. Then, the VMs are gradually placed at their preferable PMs to maximize the profit of cloud provider as much as possible. Finally, simulation experiments are performed to demonstrate the effectiveness of the proposed heuristic algorithm by comparing with other VM placement algorithms.

1. Introduction

Cloud computing provides scalable computing and storage resources via the Internet. User can use these infrastructure resources (computing and storage resources) based on a pay-per-use basis. This delivery model of *infrastructure as a service (IaaS)* has been provided by several cloud providers, such as Amazon Elastic Compute Cloud (EC2) [1], Google Compute Engine (GCE) [2], and GoGrid [3], etc. In the IaaS delivery model, the key technology is how to efficiently virtualize the computing and storage resources of physical machines (PMs) to provision a large number of virtual machines (VMs). Each customer (user) can rent a VM from the cloud provider to execute his/her application. Amazon EC2

* Corresponding author. Tel.: +886 229053855.
E-mail address: jwlin@csie.fju.edu.tw (J.-W. Lin).

0167-739X/$ – see front matter © 2014 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.future.2013.12.034

provides different instance types of VMs to meet the computing needs of users [4]. Several virtualization technologies (e.g. Xen [5], VMware [6], KVM [7]) have been used in cloud computing systems. Using the virtualization technologies, more than one VM can be created in the same PM, each of which acquires its required resources by slicing a portion of resources from the PM. For example, in Amazon EC2, assume that a small instance VM i and a large instance VM j are created on the same PM p. The PM p will be virtualized to form a machine with 1 EC2 Compute Unit, 1.7 GB memory, 160 GB instance storage for VM i and a machine with 4 EC2 Compute Units, 7.5 GB memory, 850 GB instance storage for VM j. However, the existing virtualization technologies cannot slice all hardware resources of a PM. Some types of hardware resources are non-sliceable, which are called the non-sliceable resources. For example, in [8], it clearly indicated that the disk I/O, network I/O, and L2 cache are the non-sliceable resources. In [9], the authors also claimed that I/O virtualization is a big challenge, and there is no ideal solution. It is common that two or more VMs on the same PM will contend the non-sliceable resources. In a VM, the application cannot be executed in a fully isolated computing environment. It is inevitable that if VMs i and j are created in the same PM, the application running on VM i will affect the performance of the application running on VM j. It means that, for the VMs on the same PM, there exists performance interference among the VMs. We use the term *VM interference* to describe this phenomenon. For two VMs on different PMs, their performance may be also interfered with each other if their corresponding applications are dependent with each other. This type of VM interference is not discussed in this paper since the issue is not introduced due to the contention of non-sliceable resources of a PM.

For a network I/O (delay-sensitive) application, its QoS requirement is usually defined how much time is taken to process a network I/O request. If the process time of the network I/O request is equal to or less than a pre-specified time requirement in the service level agreement (SLA). The QoS requirement of the application is satisfied. From the view point of a VM, the QoS requirement of its running application can be preliminary met by allocating appropriate resource to the VM. However, if multiple VMs are created on the same PM, each VM may incur certain degree of performance degradation due to the VM interference. In such situation, the VM is difficult to guarantee that its computing environments can continuously meet the QoS requirement of its running application. Even if there are few VMs on the same PM, it is also an inappropriate placement to allocate two or more VMs with high-QoS applications at the same PM. From the cloud provider perspective, if the created VMs cannot provide the computing environments to satisfy the QoS requirements of running applications, it will pay penalties due to violating the service level agreements (SLAs) with users.

In this paper, we investigate how to integrate QoS awareness with virtualization for efficiently performing delay-sensitive applications in cloud computing systems. Specifically, we called it the *QoS-aware virtual machine placement (QAVMP)* problem. In cloud computing, most of applications are with the data-intensive feature to frequently read and write data. The data is also stored based on the distributed manner. Therefore, a data access will involve one or more network I/O operations which is processed by the non-sliceable resource: network interface card. With involving the non-sliceable resource, the data access of one applications will be affected by other applications due to the VM interference. For a delay-sensitive application, if its data access delay is larger than its expected data response time stated in the service level agreement (SLA), the QoS requirement of this application will be violated. Compared to previous VM placement strategies [10–19], our QAVMP problem considers the QoS requirements of applications and VM interference in addition to the resource utilization of PMs. With the three concerned factors in the QAVMP problem, its optimal solution is difficult to be found. By integrating the three concerned factors into the profit metric of a cloud provider, we can formulate the QAVMP problem as an integer linear programming (ILP) model to find its optimal solution. However, long computation time will be required to find the optimal solution. To seek a time-efficient solution to the QAVMP problem, we also propose a heuristic algorithm with polynomial time to solve the problem. In the proposed heuristic algorithm, a bipartite graph is first used to model all the possible placement relationships between VMs and PMs. Based on the bipartite graph, each VM is gradually placed at its preferable PM to maximize the profit metric of the cloud provider as much as possible. Overall, the main contributions of this paper are summarized as follows.

- Unlike prior VM placement strategies in [10–19], our QAVMP problem additionally takes the VM interference effects and Qos requirements of applications into the VM placement. The proposed algorithm can reduce the VM interference and avoid violating the QoS requirements of applications after the VM placement.
- The proposed QAVMP algorithm also considers the dynamical VM creation requests. The VM creation requests arrive dynamically without any knowledge of future requests. Before creating a number of new VMs on a PM, the PM may possibly hold several existing VMs. The VM interference also exists among the existing VMs and the new VMs.
- We use an ILP model to formulate the optimal solution of the QAVMP problem.
- We propose a heuristic placement algorithm to efficiently solve the QAVMP problem in polynomial time.

The rest of the paper is organized as follows. In Section 2, we introduce the preliminaries of this paper. In Section 3, we propose an ILP model to solve the QAVMP problem optimally. In addition, the heuristic algorithm of the QAVMP problem is also presented. In Section 4, we conduct simulation experiments to evaluate the performance of the proposed heuristic algorithm. Finally, Section 5 concludes the paper.

2. Preliminaries

In this section, we give brief introduction to virtualization techniques and describe the system model used in our paper. Furthermore, we also review the previous VM placement schemes.

2.1. Virtualization techniques

Virtualization techniques can make hardware resources of a physical machine (PM) to be shared with multiple operating systems. It respectively provides a virtual environment for each operating system in the PM. Therefore, an operating system in the virtual environment can be viewed as an independent virtual machine (VM). There are two types of virtualization techniques: full virtualization and para-virtualization. The full virtualization emulates a complete hardware environment to be compatible with any operating systems. This approach is very popular, because it does not have to modify the original operating systems. However, it increases some overhead due to the virtualization of the hardware devices. Para-virtualization is another approach that can improve the efficiency of hardware virtualization. Using para-virtualization, the operating systems are aware of working in a virtual environment. In other words, the operating systems must be modified for virtualization. Regardless of adopting which one of the virtualization techniques, it requires a monitor to handle resource allocation for the VMs.

2.2. Xen hypervisor

Xen hypervisor is an open source virtual machine monitor [20,21]. As shown in Fig. 1, Xen architecture consists of Xen hypervisor and guest domains. The Xen hypervisor can protect guest