ARTICLE IN PRESS

Future Generation Computer Systems ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

VM instance selection for deadline constraint job on agent-based interconnected cloud

Chih-Tien Fan [a,*], Yue-Shan Chang [b], Shyan-Ming Yuan [a]

[a] Institute of Computer Science and Engineering, National Chiao Tung University, 1001, University Road, Hsinchu City, Taiwan
[b] Department of Computer Science and Information Engineering, National Taipei University, 151, University Road, San Shia District, New Taipei City, Taiwan

HIGHLIGHTS

- Agent framework is used to create and manage the interconnected cloud.
- Using Rough Set Theory to predict job's execution time.
- Using deadline and execution time to select and adjust the computing resource.

ARTICLE INFO

Article history:
Received 30 June 2017
Received in revised form 27 March 2018
Accepted 7 April 2018
Available online xxxx

Keywords:
Interconnected cloud
Rough Set Theory
Intelligent agent

ABSTRACT

In recent years, users have been buying computing resources, such as VM instances, from the cloud resource providers. However, every cloud has its pricing models, solutions, and interfaces. Without a united interface, it is hard to manage across different clouds. To make things worse, it is even hard to select proper VM instances among multiple cloud resource providers when the resources are not sufficient. This paper proposes an interconnected cloud of selecting VM instances based on the job's deadline constraint. To make the system function, five agents are created for different purposes: a System Monitoring Agent, a Job Dispatching Agent, an Instance Group Managing Agent, an Instance Group Administrating Agent, and a Job Executing Agent. This paper also presents two decision algorithms, the Job Dispatching Algorithm and the Instance Group Invocation Algorithm, based on the Rough Set Theory, the jobs' deadline constraint, and the computing resource of the VM instances. The proposed system was deployed and evaluated on the real machines. The result shows not only the VM instances on different machines are interconnected, but also the algorithms successfully help manage the interconnected cloud's computing resources according to the submitted jobs with deadline constraints.

© 2018 Elsevier B.V. All rights reserved.

1. Introduction

The technology of cloud computing [1] has become more mature in the past few years. Cloud computing provides simple, flexible, and quicker ways to allow consumers to obtain required computing resources. Base on the deployment, there are three kinds of cloud model: the private cloud, the public cloud, and the interconnected cloud.

In general, the private cloud owner exploits virtualization technology, such as Xen [2], Hyper-V [3], or VMWare [4], to efficiently manage and utilize the resources. However, the private cloud owner needs a big budget and effort to run the platform. When the existing computing resource is not enough to serve the extra workload on demand [5], the owners are likely to buy additional computing resources from public cloud providers, such as Amazon Elastic Compute Cloud [6], GoGrid [7], or HiCloud [8]. The computing resource usually charges as the pay-as-you-go model. The interconnected cloud [9], or hybrid cloud [1], is necessary to federate multiple cloud environments [10]. The interconnected cloud can inherit the advantages from both the private cloud and the public cloud. An additional benefit is to help reduce hardware cost and improve dynamic resource provisioning [11,12]. To conclude, the interconnected cloud can host computing resource and share the load in private cloud at the same time. For example, if a user submits a job with a deadline constraint to a private cloud that cannot be completed before the deadline, the task can be dispatched and executed in the public cloud to meet the deadline constraint.

To deal with the problem that different Cloud providers have different kinds of Virtual Machine (VM) instance types, pricing

* Corresponding author.
E-mail addresses: admin@ctfan.idv.tw (C.T. Fan), ysc@mail.ntpu.edu.tw (Y.S. Chang), smyuan@cis.nctu.edu.tw (S.M. Yuan).

https://doi.org/10.1016/j.future.2018.04.017
0167-739X/© 2018 Elsevier B.V. All rights reserved.

Please cite this article in press as: C.T. Fan, et al., VM instance selection for deadline constraint job on agent-based interconnected cloud, Future Generation Computer Systems (2018), https://doi.org/10.1016/j.future.2018.04.017.

models, and management interfaces, a Cloud Broker can be used. A Cloud Broker provides a unified abstraction layer for the user and developers to manage all the computing resources under one interface. In addition, the Cloud Broker also manages the VM instances on different cloud providers and collects useful information, such as the specification of the VM instance, the pricing model, availability, etc., from every Cloud provider [1]. OpenStack [13] is one of the most well-known solutions to deploy a federated cloud. Some academic studies or projects such as Aneka [14–21], the Reservoir project [10,22], and other projects [23–28] also addresses this issue.

Most of the interconnected cloud systems only provide hypervisors that connecting each VM instance. Connecting the VM instances only makes the brawn (the computing ability) becomes powerful. Without the brain (the smart management), it is hard to show the elastic of the brain [24]. To give the brawn with a powerful brain, the agent technology [5,29,30] can be exploited. With the agents' knowledge, the amount of the computing resource of the interconnected cloud can be adjusted dynamically according to the status of the environment.

Another issue is that the brokers can only manage the instances on multiple clouds. All the decisions, such as adding some VM instances to the cloud, are decided by the user. Without proper knowledge, it is not easy to determine which VM instance should be activated. When the job has a deadline constraint, how to choose a suitable VM instance becomes a complicated task [10,25]. If the deadline constraint job is assigned to a VM instance that does not have enough computing resource, the constraint will be violated. However, if it is assigned to a VM instance with more computing resources, the cost of renting the VM instance will increase.

This paper proposes an agent-based interconnected cloud model with a system that can react according to the events and the interconnected cloud's environments. The overview of the model is shown in Fig. 1, which consists of three layers: the Federation Layer, the Cloud Infrastructure Layer, and the Data Center Layer. The bottom layer of the model is the Data Center Layer, where all the source data and the results are stored. All of the VM instances used in the interconnected cloud can access the data centers. In this paper, we assume that there is no concurrency problem of any files. That is, each job can only read data from the data centers; the results are written to different files. Tasks cannot update or delete any existing file in the data center. Another assumption is that the source data's transfer time is not considered. These assumptions help the paper focus on the proposed model.

The Cloud Infrastructure Layer is the second layer of the proposed model, where all the private and public clouds belong to. Each cloud consists of multiple VM instances with different capabilities (e.g., different numbers of core processor, RAM sizes, and Internet speeds). Users can control the VM instances through the VM Instance Hypervisor provided by the resource provider.

The last layer is the Federation Layer, which is located at the top of the interconnected cloud model. A centralized server is set up to manage the computing resource in the interconnected cloud. A knowledge base is also installed on the server to help the agents to make decisions. The server also provides an interface that allows administrators to monitor and submit jobs to the interconnected cloud.

When a user submits a job to the interconnected cloud via the server in the federation layer, the agents will redirect the job to one of the VM Instances to execute the job. The job may have a deadline; the user can assign a deadline to a job when submitting it. If the agents in the federation layer notice that some jobs are about to violate the deadline, the agent will start more VM instance. On the other hand, if there are too many idle VM instances, some of them will be closed. According to the knowledge base in the federation layer, the agents are capable of determining which VM instance to be started or to be stopped.

The contribution of the paper can be stated as follows.

1. An agent framework is used to create and to manage the interconnected cloud.
2. Using the technique of the Rough Set Theory [23,31–35] to estimate the execution of the job according to its condition attributes and the execution environment.
3. Allowing jobs with deadline constraints. If the job's deadline constraint is detected to be violated, additional VM instances will be started to scale out the computing resource of the interconnected cloud.

Most of the works share the same consumption: the algorithm knows the jobs' execution time when the jobs are submitted to the system. This is not suitable for the real cloud environment. Different sizes of jobs may need different durations of execution time. Even though the average execution time of the tasks may be a good solution, it would be better if more parameters can be put into consideration. In our proposed work, the data size of a job is defined by the user, and it is treated as an attribute when the execution time is estimated. When the job cannot find any similar historical jobs, the average execution time of the historical jobs can be treated as its estimated execution time.

The following sections of the paper are organized as follows. In Section 2, related works about the interconnected cloud and the deadline aware resource provisioning models are presented. Section 3 describes the design of the agents in this work. The section also presents the interactions between the agents. In Section 4, how to approximate a job's execution time by using the Rough Set Theory is shown. The main algorithms are also described in Section 4. Section 5 displays the implementation of the system. The evaluation of the proposed interconnected cloud system is shown in Section 6. The final section is the conclusion of this paper.

2. Related works

2.1. Interconnected cloud model

With the interconnected cloud mode, it allows users to federate multiple clouds and control them under one interface. Toosi et al. mentioned six motivations to enable the cloud interoperability [9]. The interconnected cloud allows users to use multiple cloud resources; therefore, the availability and the scalability of the cloud are better compared with those of the single cloud. Users can have more options with a reasonable cost. The other benefit from enabling the cloud interoperability is to avoid vendor lock-in problem.

Reservoir [10,22] is an interconnected cloud model developed under the RESERVOIR European research project. The primary goal of the model is to deal with the scalability problem that happened in the single provider cloud computing model. In Reservoir, a *service provider* is a user who needs the computing resource to run the service applications. *Infrastructure providers* operate the *Reservoir sites* that provide the computing resource. On each *Reservoir site*, a *service manager* provides the interface that allows the *service providers* to receive the services, to negotiate pricing, and to handle billing. The service manager of the Reservoir site also manages the *virtual execution environment* (*VEE*) and the service running in it. By unifying the interface, a service provider can find a proper Reservoir site to run the service without any difficulty. However, the *service manager* may be the bottleneck of each Reservoir site as the number of the service provider and *VEE* grow. In our work, the function of management and monitoring is separated. Each agent is designed for the specific purpose, and the agents can communicate with each other via the network. Agents can be deployed on different computers to avoid resource contention.

Aneka is an interconnected cloud model developed by The Cloud Computing and Distributed Systems Laboratory and

Download English Version:

https://daneshyari.com/en/article/6872936

Download Persian Version:

https://daneshyari.com/article/6872936

Daneshyari.com