Future Generation Computer Systems 78 (2018) 191–210

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

# An autonomic resource provisioning approach for service-based cloud applications: A hybrid approach

Mostafa Ghobaei-Arani[a], Sam Jabbehdari[b,*], Mohammad Ali Pourmina[a]

[a] Department of computer engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran
[b] Department of computer engineering, North Tehran Branch, Islamic Azad University, Tehran, Iran

## HIGHLIGHTS

- We designed a framework for autonomic resource provisioning to cloud services.
- We customized an autonomic resource provisioning approach based on the control MAPE loop.
- We enhanced the performance of the planning phase by using the RL-based agent.
- We conducted a series of experiments under real-world workload traces for different metrics.

## ARTICLE INFO

*Article history:*
Received 24 March 2016
Received in revised form
22 November 2016
Accepted 12 February 2017
Available online 15 February 2017

*Keywords:*
Cloud computing
Cloud services
Resource provisioning
Autonomic computing
Reinforcement learning

## ABSTRACT

In cloud computing environment, resources can be dynamically provisioned on deman for cloud services The amount of the resources to be provisioned is determined during runtime according to the workload changes. Deciding the right amount of resources required to run the cloud services is not trivial, and it depends on the current workload of the cloud services. Therefore, it is necessary to predict the future demands to automatically provision resources in order to deal with fluctuating demands of the cloud services. In this paper, we propose a hybrid resource provisioning approach for cloud services that is based on a combination of the concept of the autonomic computing and the reinforcement learning (RL). Also, we present a framework for autonomic resource provisioning which is inspired by the cloud layer model. Finally, we evaluate the effectiveness of our approach under two real world workload traces. The experimental results show that the proposed approach reduces the total cost by up to 50%, and increases the resource utilization by up to 12% compared with the other approaches.

## 1. Introduction

Cloud computing is one of the most popular technologies in the businesses, educational institutions, governments, and the research community that has become an integral part of many users are heavily dependent on cloud-based applications for their day-to-day activities in both professional and personal life [1,2]. In the simplest terms, cloud computing means storing and accessing data and programs that are delivered as services to the end users over the internet [3,4]. Service-oriented architecture (SOA) [5] is an enabling technology for cloud computing, and this architectural style helps to deliver separately application functionalities in form of a service as the main building blocks of applications and system development. The *cloud services* are the result of integrating cloud computing and SOA, and are typically comprise one or more functionalities that are offered by cloud providers and are can be delivered to the end users over the internet [6,7]. A *cloud application* is a software product running on the cloud environment, and it can be used with a web browser connected to the internet. Typically, each cloud application is composed of one or more cloud services that together perform the function of an application. Usually, cloud services are categorized into three basic service models includes: infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS). In this paper, we use the term *cloud service* instead of the phrase *cloud software service* offered by SaaS providers. In cloud ecosystem,

[*] Corresponding author.
*E-mail addresses:* m.ghobaei@srbiau.ac.ir (M. Ghobaei-Arani), s_jabbehdari@iau-tnb.ac.ir (S. Jabbehdari), pourmina@srbiau.ac.ir (M.A. Pourmina).

http://dx.doi.org/10.1016/j.future.2017.02.022
0167-739X/© 2017 Elsevier B.V. All rights reserved.

the end users submit the requests for utilizing the cloud services offered by a SaaS provider that owns a cloud application, and to host its cloud application, it rents resources from an IaaS provider such as Amazon EC2 [8].

One of the unique characteristics of cloud computing is its *elasticity*, which enables SaaS providers to adapt workload changes of their cloud services by provisioning and deprovisioning resources automatically such that at each point in time, the available resources match the current demand as closely as possible [9]. The SaaS providers can acquire and release resources for running their cloud services on demand and only pays for the resources that are actually used based on a pay-per-use model [10, 11].

Deciding the right amount of resources for the cloud service during its execution time is not trivial, and it depends on the current workload of the cloud service. As users have irregular access to cloud services offered by a SaaS provider, the cloud services will experience workload fluctuations. These fluctuating workloads may lead to undesirable states that are referred to as *over-provisioning* or *under-provisioning* states. The *over-provisioning* state can exist when the more resources than demands of a cloud application are provisioned. This is correct from the point of view of service level agreements (SLAs); however, it incurs an unnecessary cost to the user and SaaS provider. On the other hand, the *under-provisioning* state can exist when the fewer resources than demands of a cloud application are provisioned. This problem causes SLA violations, which lead to lose of revenue and users [12,13]. Therefore, an effective elasticity mechanism must be able to estimate the needed resources properly to satisfy a given SLA based on the current workload of a cloud application.

To deal with the mentioned above resource provisioning problems, dynamic resource provisioning is utilized. The dynamic resource provisioning is an effective approach which its fundamental idea is to provision the resources based on the workload changes of the cloud application. Its objective is to automate the dynamic provisioning of resources by minimizing the cost of renting resources from an IaaS provider and meeting the SLA of the cloud application. The main objective of the SaaS provider is to maximize its profit during the execution of its cloud application, and this can be achieved by minimizing the payment of using resources from the IaaS provider, as well as the penalties cost caused by SLA violations that have to be paid to users.

In this paper, we propose a hybrid resource provisioning approach for cloud applications based on a combination of the concept of the autonomic computing and the reinforcement learning (RL). To achieve autonomic computing, IBM has proposed a reference model for autonomic control loops [14,15], which is called the control MAPE (Monitor, Analysis, Plan, Execute) loop. The control MAPE loop is similar to the general agent model proposed by Russel and Norvig [16], in which an intelligent agent perceives its environment using sensors, and uses these perceptions to determine actions to be executed in the environment. The proposed approach follows the control MAPE loop, which consists of four phases: monitoring (M), analysis (A), planning (P), and execution (E). First, in the monitoring phase, a monitoring component gathers the information about the resources and cloud application state, and this information is processed to estimate future resource utilization and demands at the analysis phase. In the planning phase, a suitable resource modification action (e.g., scale in or scale out) is determined, and finally, the modification actions determined in the planning phase are performed in the execution phase. The control MAPE loop is regularly executed and manages the virtual machines (VMs) that are allocated to each cloud service at specific time intervals. We apply RL [17,18] as a decision-maker that uses the predicted results of an analysis phase in order to obtain the optimal action to remove or add VMs in the planning phase. RL is an adaptive self-learning system that improves its performance through repeated interactions with the cloud environment. The main contributions of this research can be summarized as follows:

- We designed a framework for autonomic resource provisioning which is inspired by the cloud layer model and PaaS layer, it is responsible for resource provisioning to cloud services based on the control MAPE loop.
- We customized an autonomic resource provisioning approach for cloud services offered by a SaaS provider, which is covered all the phases defined within the control MAPE loop.
- We enhanced the performance of the planning phase of the control MAPE loop by using the RL-based method as a decision-maker.
- We conducted a series of experiments to evaluate the performance of proposed approach under real-world workload traces for different metrics.

The rest of this paper is organized as follows: In Section 2, we focus on a survey of related work. Section 3 provides the necessary background. In Section 4, we formulate the problem and describe the proposed solution. In Section 5, we present an evaluation and discuss the experimental results, and in Section 6, we conclude the paper and present future works.

## 2. Related works

The dynamic resource provisioning mechanisms are achieved by scaling in/out the resources (i.e., removing or adding a VM) through a set of rules to match as closely as possible the available resources with the current workload. Since the resource provisioning proposed approach is a combination of the autonomic computing and the reinforcement learning (as a machine learning technique), we will focus on dynamic resource provisioning techniques into the following two major categories: (i) resource provisioning based on the autonomic computing techniques [19–27], (ii) resource provisioning based on machine learning techniques [28–37].

### *2.1. Resource provisioning based on autonomic computing techniques*

Huebscher et al. [19] presented a survey on autonomic computing and the IBM's MAPE-Knowledge (MAPE-K) reference model. They introduced the motivation and concepts of autonomic computing, degrees, models, and applications. In [20] proposed a framework dynamic cloud provisioning of system topologies for common two-tier application scenarios based on a MAPE loop concept, while our framework applies to cloud services offered by SaaS providers. Pop et al. [21] reviewed advanced topics in resource management for ubiquitous cloud computing, and proposed an adaptive approach that maximizes the profit for service providers, while it minimizes the total cost to customers. Maurer et al. [22] proposed adaptive resource provisioning techniques based on the autonomic control MAPE loop for cloud infrastructural management. These techniques are the case-based reasoning and a rule-based approach, while our approach employs time series analysis and machine-learning techniques. In [23] designed an adaptive framework based on the control MAPE loop for optimizing the configuration of scientific applications in three layers, i.e., the application layer, execution environment layer, and the resource layer. In [24], the LoM2HiS framework is presented, which is used for managing the mappings of the low-level resource metrics into high-level SLA parameters. The LoM2HiS framework is embedded into the FoSII [25] infrastructure, which facilitates autonomic SLA management and enforcement. In [26], the authors developed a dynamic resources provisioning and monitoring (DRPM) system. Moreover, they proposed a multi-agent system to manage the cloud provider's resources, where the customers'