Future Generation Computer Systems 78 (2018) 343–352

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

A coral-reefs and Game Theory-based approach for optimizing elastic cloud resource allocation

Massimo Ficco [a,*], Christian Esposito [b], Francesco Palmieri [b], Aniello Castiglione [b]

[a] Department of Industrial and Information Engineering, Second University of Naples, Via Roma 29, I-81031 Aversa (CE), Italy
[b] Department of Computer Science, University of Salerno, Via Giovanni Paolo II, 132 I-84084 Fisciano (SA), Italy

HIGHLIGHTS

- Bio-inspired coral-reefs optimization paradigm to model cloud elasticity.
- Game theory-based approach to identify the best cloud resource reallocation schema.
- Fuzzy linguistic SLA formalization.

ARTICLE INFO

Article history:
Received 15 January 2016
Received in revised form
16 May 2016
Accepted 22 May 2016
Available online 30 May 2016

Keywords:
Cloud computing
Elasticity
Live-migration
Coral-reefs optimization
Game Theory
Fuzzy linguistic SLA

ABSTRACT

Elasticity is a key feature in cloud computing, which distinguishes this paradigm from other ones, such as cluster and grid computing. On the other hand, dynamic resource reallocation is one of the most important and complex issues in cloud scenarios, which can be expressed as a multi-objective optimization problem with the opposing objectives of maximizing demand satisfaction and minimizing costs and resource consumptions. In this paper, we propose a meta-heuristic approach for cloud resource allocation based on the bio-inspired coral-reefs optimization paradigm to model cloud elasticity in a cloud-data center, and on the classic Game Theory to optimize the resource reallocation schema with respect to cloud provider's optimization objectives, as well as customer requirements, expressed through Service Level Agreements formalized by using a fuzzy linguistic method.

1. Introduction

In recent years, cloud computing has attracted attention from industry, government and academic worlds. An increasing amount of applications make extensive usage of cloud resources, according to an on-demand, self-service, and pay-by-use business model, with the progressive adoption of cloud-based services by many sectors of modern society. One of the main reasons of this success is the possibility of acquiring virtual resources in a dynamic and elastic way. In particular, the emerging virtualization technologies allow multiple virtual machines (VMs) to run concurrently on a single physical host, called host machine (HM). Each VM, in turn, hosts its operating system, middleware and applications, by using a partition of the underlying hardware resources capacity (CPU power, memory, store capability and network bandwidth) [1]. Cloud elasticity is the key feature for implementing server consolidation strategies. It allows for on-demand migration and dynamic reallocation of VMs [2]. NIST defines elasticity as the ability for customers to quickly purchase on-demand and automatically release as many resources as needed, ideally giving to the user the feeling that the cloud resource capabilities are unlimited [3]. Specific elasticity control mechanisms are implemented to decide when and how to scale-up or scale-down virtual resources, in accordance with the provider's own optimization objectives, as well as with user-defined settings and requirements formalized within the context of a specific Service Level Agreement (SLA) between the cloud provider and the customer. In order to do this, resource usage information, such as CPU load, free memory and network traffic volumes, for all the available HMs, has to be continuously collected and analyzed on-line. Three different techniques are used in the implementation of cloud elasticity solutions: replication, migration and resizing [4], resulting in different strategies and approaches for handling resource allocation within a cloud infrastructure.

* Corresponding author.
E-mail addresses: massimo.ficco@unina2.it (M. Ficco), christian.esposito@dia.unisa.it (C. Esposito), fpalmieri@unisa.it (F. Palmieri), castiglione@ieee.org (A. Castiglione).

http://dx.doi.org/10.1016/j.future.2016.05.025
0167-739X/

The dynamic resource allocation problem describes the decision of how many HMs are required overall to cope with the current demand, and how VMs are (re-)allocated to each HM in the individual time intervals. The optimization of resource allocation consists in minimizing the number of HMs necessary to host all the VMs associated to users' demands. This is a typical multi-objective optimization task, where the most important goals are the minimization of the needed hardware resources and the satisfaction of SLAs contracted with the customers. This problem, widely simplified, is closely related to a commonly known *NP*-hard combinatorial optimization problem, the "bin packing" problem [5], where the items to be packed can be viewed as the VMs and the bins as the HM nodes with their multi-dimensional capacity represented by the available computing power, free memory, storage capability and network bandwidth. Consequently, our elastic cloud resource allocation problem, that is inherently more complex, will be *NP*-hard, and thus, a reasonable heuristic solution, achieving near-optimal results is strongly desirable.

In such a complex multi-objective scenario, traditional optimization methods are not able to efficiently provide good results, due to their inherent difficulties in exploring huge solution spaces. Hence, novel approaches based on bio-inspired meta-heuristic schemes, seem to be the most promising options to achieve a better trade-off between the complexity of the search process and the optimization of the solutions found. Such schemes rely on models and strategies borrowed from the observation of behaviors, and evolution mechanisms available in nature, where only the fittest individuals are able to survive in organizations characterized by a high degree of competition. In this direction, to model the elastic resource reallocation dynamics of a typical cloud environment, we adopted a Coral-Reefs Optimization (CRO) approach, which artificially simulates the reefs evolution processes, including corals' reproduction and competition for the space in the reefs [6], which, by analogy, model the continuous demands for resource re-sizing, migration and replication, characterizing the operational life-cycle of VMs hosted within a cloud data center. This approach presents extremely interesting and promising features in achieving convergence towards global optima.

On the other hand, cloud customers can be seen as non-cooperative entities, whose fundamental interest is not the optimization of the overall system performance, but rather the maximization of their individual benefit. Therefore, a natural hint for carefully driving the VM allocation process comes from classic game theory. Accordingly, we used a game theory-based approach to identify the best VM reallocation schema with respect to the critical requirements specified by the cloud customers in their SLAs, formalized by using fuzzy linguistic label sets, and typically characterized by different and even conflicting performance objectives and optimization criteria. This approach is able to drive the complex interactions between the customers and the cloud provider towards a social optimum (i.e., an equilibrium point between their own demands and goals), by optimizing a global objective function that takes into account the individual interests of all the involved entities.

The rest of the paper is organized as follows. In Section 2, the strategies and methods currently adopted to implement cloud elasticity capabilities are presented. Section 3 defines the problem of interest. The proposed approach for elastic resource reallocation is presented in Section 4. The related experimental performance evaluation results are shown in Section 5. Finally, Section 6 presents some concluding considerations and remarks.

2. Background and related work

Elasticity represents a dynamic property of the cloud paradigm, which allows the system to scale on-demand within an operational system context [7]. It enables applications to evolve, by following the users' demand, without needing traditional "fork-lift" upgrades, and hence, introduces the degree of adaptiveness that is fundamental for modern big data-centric environments.

Generally, Infrastructure as a Service (IaaS) architectures provide an elasticity controller, which is responsible for monitoring information like CPU load, memory and network traffic, and for making decisions on whether or not the virtual resources must be scaled or migrated, in accordance with user-defined rules and settings [8]. From a user perspective, elasticity allows perfect matching between instantaneous customer's needs and resources available to him, by avoiding the unnecessary reservation of resources that are not needed for most of time, with obvious impacts on the overall system scalability and performance. From the provider perspective, elasticity ensures better use of computing resources, by providing economies of scale, and allowing a much larger number of users to be served simultaneously [4]. Moreover, it can be used to increase the local resources capacity, by simultaneously reducing operational costs and energy consumption [9–11].

Several strategies have been employed in order to support the implementation of elasticity capabilities, including resizing, replication, and migration:

- *Resizing (vertical scaling)*: according to such strategy, CPU, memory and storage resources can be re-sized on a running virtual instance, that can be a virtual machine or a container. Some of the most significant implementation of resizing mechanisms are PRESS [12], ElasticVM [13], and Kingfisher [10].
- *Replication (horizontal scaling)*: consists of adding/removing instances from users' virtual environments. It is currently the most widely used technique for providing elasticity in cloud environments [14–16]. Several cloud platforms, such as Amazon and AzureWatch, offer auto-scaling and load-balancing capabilities in order to split the load between the various instances. Specific thresholds can be set by customers to add or remove instances depending upon the actual usage.
- *Live migration*: consists in transferring the entire VM runtime status (CPU and memory pages) from the actual host machine to a new destination host, without preempting its execution [17,18]. It is performed in two steps, in which at first, a pre-copying of the VM status on the target host is performed (while the VM is still running); then, the execution of the source machine is stopped, a re-copy of the modified pages on the destination host is performed, and finally, the migrated VM is resumed.

  A cloud provider's resource management facility must orchestrate the VM migration in order to simultaneously minimize resource usage and maximize SLA adherence [19]. Specifically, it must provide:
  - *Cold-spots migration support*: that is related to over-provisioned resources with low utilization (server sprawl). In order to reduce severe financial losses for cloud providers, live migration can be used to consolidate VMs (optimize VMs placement) with the aim of minimizing the number of powered-on hosts and saving energy [20]. When the amount of resource usage of a host drops below a given threshold, VMs are migrated to another host providing enough available resources capacity (the freed-up HMs are switched-off).
  - *Load balancing migration support*: VM migration can be used to dynamically move VMs among the available hosts with the aim of rebalancing the utilization of resources over them. A specific decision-making process is enabled when the load imbalance within the cloud (among heavily loaded hosts and lightly loaded ones) exceeds a given threshold [21,22]. Its objective could achieve equal residual resource capabilities across all the available HMs in order to optimize local resource allocations in presence of increasing demands.