A Layered Coordination Framework for Optimizing Resource Allocation in Adapting Cloud-based Applications

Patrizia Scandurra  
Università degli Studi di Bergamo  
DIIMM, Dalmine (BG), Italy  
patrizia.scandurra@unibg.it

Claudia Raibulet  
Università degli Studi di Milano-Bicocca  
DISCo, Milano, Italy  
raibulet@disco.unimib.it

Pasqualina Potena  
Università degli Studi di Bergamo  
DIIMM, Dalmine (BG), Italy  
pasqualina.potena@unibg.it

Raffaela Mirandola  
Politecnico di Milano  
DEI, Milano, Italy  
mirandola@elet.polimi.it

Rafael Capilla  
Universidad Rey Juan Carlos  
Madrid, Spain  
rafael.capilla@urjc.es

ABSTRACT

In this paper we propose a framework that adapts a cloud-based software application by providing an enhanced assembly of resources using the Pareto-optimal solution to optimize the resource allocation with tight cooperation between the cloud layers.

Categories and Subject Descriptors

D.2.11 [Software Engineering]: Software Architectures—Domain-specific architectures; C.2.4 [Computer Communication Networks]: Distributed Systems—Distributed applications; K.6.3 [Management of Computing and Information Systems]: Software Management—Software maintenance

General Terms

Design, Management

Keywords

cloud computing, software adaptation, resource allocation

1. INTRODUCTION

Cloud computing is an emerging computational model where systems, both hardware and software, are seamlessly delivered and administered over the Internet as services, giving the illusion to have them on a local machine. This is allowed by the exploitation of web technologies, virtualization and large web data centers. Depending on the content of the service, a cloud can offer Infrastructure (raw computing services such as CPU and storage) as a Service (IaaS), Platform (COTS, tools, middleware for developing and deploying applications) as a Service (PaaS), and Software (end user applications) as a Service (SaaS). The SaaS layer generally exploits the Service-Oriented Architecture (SOA) technology because, as remarked in [4], SOA and cloud computing coexist, complement and support each other.

The approach presented in this paper has been inspired by the work in [2] and proposes a tight interlayer coordination schema for optimizing resource allocation able to deal with the adaptation of cloud-based applications.

2. A COORDINATION FRAMEWORK FOR CLOUD LAYERS

This section presents a three-layered coordination approach based on three frameworks (i.e., the SaaS, the PaaS and the IaaS framework) that dynamically perform resource allocation for each cloud layer. At each layer, each framework provides a set of managers that handle the requests coming from the other layers. Each framework also interchanges sensors data (related to measures of the optimization) and actuators data (to drive the optimization) with the managers of the other layers thus performing a continuous optimization activity.

The IaaS layer aims at maximizing the use of its Hardware Resources (HRs). Such resources are virtualized and offered as computing services (e.g., storage, CPU and memory). The IaaS layer handles service requests from the PaaS layer, such as Virtual Machines (VMs) with particular performance characteristics on memory, storage and processing capacity. The PaaS layer, which offers platform services to the SaaS layer, aims at optimizing the profit by maximizing the number of applications it hosts and minimizing the resources it uses, and penalties it pays. The PaaS layer uses resources of the IaaS layer by requesting VMs and storage and deploying application containers on the VMs. Finally, the SaaS layer, provides services to end users and maximizes the profit, for example setting the revenue proportional to the number of users or to the throughput. It deploys the applications in PaaS containers considering topologies specific to each application.

In the following we describe how our coordination framework allows the combination of the adaptation – performed at the SaaS layer and driven by quality attributes – of a cloud-based application and of resource allocation – performed at all layers.
An example of a coordination scenario. Figure 1 shows a concrete example of coordination scenario as it would be performed by our coordination framework for the adaptation of a cloud-based application. Below, we describe the main steps of this process, which could be merely adopted in case of system evolution. In case of self-adaptation, some simplifications are required in the optimization algorithms involved.

**Step 1: SaaS Framework defines adaption actions.** The SaaS framework defines software adaptation actions for addressing the required changes (claimed by a user or triggered by a Monitor entity [3]) by using a multi-objective optimization approach[1]. A solution of such an optimization, called Pareto solution, is the one that minimizes a set of objectives (e.g., adaptation cost, probability of failure and response time) under quality constraints, such as on the required minimum level of reliability. The solutions may differ in the adaptation actions for adapting the applications or the resource allocation (e.g., the deployment of services on the containers). If the SaaS layer needs more resources (or does not find an admissible solution with the containers assigned by the PaaS layer), then the SaaS framework proposes the Pareto solutions to the PaaS framework. For each solution the resources necessary for its application are indicated (e.g., a new deployment topology) or ranges for container parameters (e.g., utilizations, throughput) are reported. The SaaS framework tries to minimize, other than the cost for the software adaptation (e.g., the one for embedding a new service), the payment for resources to the PaaS layer.

**Step 2: PaaS Framework tries to satisfy the SaaS demands.** The PaaS framework analyzes the Pareto solutions received from the SaaS framework by trying to find feasible resource allocations. If it does not find feasible solutions, then it proposes Pareto solutions to the IaaS framework. Each solution proposes, for example, additional VMs, the allocations of VMs to containers. Similarly to the SaaS framework, the PaaS framework tries to minimize the payment for resources to the IaaS layer.

**Step 3: IaaS Framework tries to satisfy the PaaS demands.** The IaaS framework analyzes the Pareto set received from the PaaS framework. Similarly to the SaaS and PaaS frameworks, the IaaS framework finds a Pareto solution that minimize a set of objectives (e.g., the resources that it does not use and the penalties it pays) by trying to satisfy the PaaS requests (e.g., by allocating VMs to processors).

**Step 4: Contract re-negotiation between layers.** If the layers do not find a resource allocation for addressing the required systems changes, then layers’ contracts may be (re)-negotiated, e.g., the PaaS layer contracts additional resources with the IaaS layer or the SLA is re-negotiated. This step will mostly depend on runtime adaptation needs.

3. CONCLUSIONS AND FUTURE WORK

We have argued the need that cooperation between the cloud layers should be improved in order to satisfy the resource allocation in an optimal way thus avoiding collapsing the cloud.

Currently, we are implementing a prototype of our coordination framework to handle multiple distributed applications and large scale infrastructures by following a centralized or a decentralized paradigm. We intend to apply our approach on realistic examples to validate it, study its scalability, and comparing it with existing approaches.

4. REFERENCES