Future Generation Computer Systems 32 (2014) 82–98

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Enabling cost-aware and adaptive elasticity of multi-tier cloud applications

Rui Han, Moustafa M. Ghanem, Li Guo, Yike Guo *, Michelle Osmond

Department of Computing, Imperial College London, London SW7 2AZ, UK

ARTICLE INFO

Article history:
Received 30 October 2011
Received in revised form
30 March 2012
Accepted 17 May 2012
Available online 9 June 2012

Keywords:
Cloud computing
Elasticity
Multi-tier applications
Cost-aware criteria
Adaptive scaling algorithm

ABSTRACT

Elasticity (on-demand scaling) of applications is one of the most important features of cloud computing. This elasticity is the ability to adaptively scale resources up and down in order to meet varying application demands. To date, most existing scaling techniques can maintain applications' Quality of Service (QoS) but do not adequately address issues relating to minimizing the costs of using the service. In this paper, we propose an elastic scaling approach that makes use of cost-aware criteria to detect and analyse the bottlenecks within multi-tier cloud-based applications. We present an adaptive scaling algorithm that reduces the costs incurred by users of cloud infrastructure services, allowing them to scale their applications only at bottleneck tiers, and present the design of an intelligent platform that automates the scaling process. Our approach is generic for a wide class of multi-tier applications, and we demonstrate its effectiveness against other approaches by studying the behaviour of an example e-commerce application using a standard workload benchmark.

1. Introduction

Cloud computing has received wide attention over the past few years. New services offered by cloud IaaS (Infrastructure-as-a-Service) providers, such as Amazon Web Services (WS) [1], GoGrid [2] and IBM [3], are generating a huge demand from application owners. The pay-as-you go model used by such providers is appealing to most application owners. It removes the costs of buying, installing and maintaining a dedicated infrastructure for running an application. Moreover, most IaaS providers allow the application owners to scale up and down the resources used based on the computational demands of their applications, thus letting them pay only for the amount of resources they use. This model is appealing for deploying applications that provide services for third parties, e.g. traditional e-commerce sites, financial services applications, online healthcare applications, gaming applications, media servers and bioinformatics applications. If the workload of a service increases (e.g. more end users start submitting requests at the same time), the application owner can ideally scale up the resources used to maintain the Quality of Service (QoS) of their service. When the workload eases down, they can then scale down the resources used. Within this context, elasticity (on-demand scaling), also known as redeploying or dynamic provisioning, of applications has become one of the most important features of a cloud computing platform. This elasticity enables real-time acquisition/release of computing resources to scale the applications themselves up and down in order to meet their run-time requirements, while letting application owners pay only for the resources used.

Our motivation in this paper is investigating the development of new methods that assist the owners of applications deployed on IaaS clouds in managing the costs of their own applications while still maintaining the Quality of Service (QoS) they provide to their end users. Addressing this issue effectively requires taking a closer look at the structure of most common services and applications deployed on IaaS clouds to provide services to other parties. Such applications are typically implemented as multi-tier applications running on distributed software platforms. Taking the example of an e-commerce website, there are at least three tiers: a frontend web server for handling HTTP requests; a middle-tier application server for implementing business logic; and a backend database with data store and processing. Each of the tiers can be implemented using one or more servers. Depending on different types of incoming requests, servers at each tier can be stressed by heavy workloads, or can become idle due to light workloads. When scaling up and down an application, it is thus crucial to discover the real bottlenecks that may be caused at any, or all, of the servers.

Although some of the existing scaling techniques [1,4–16] address the question of how to maintain an applications' Quality of Service (QoS), they rarely consider the equally important aspect of cloud computing—the cost of using the resources themselves. Applications deployed in a cloud environment require both good performance and cost-efficient resource usage.

In this paper, we propose a scaling approach that is both cost-aware and workload-adaptive, allowing application owners to

* Corresponding author. Tel.: +44 07869562039.
E-mail addresses: r.han10@imperial.ac.uk (R. Han), mmg@imperial.ac.uk (M.M. Ghanem), liguo@imperial.ac.uk (L. Guo), y.guo@imperial.ac.uk (Y. Guo), mo197@imperial.ac.uk (M. Osmond).

0167-739X/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.future.2012.05.018

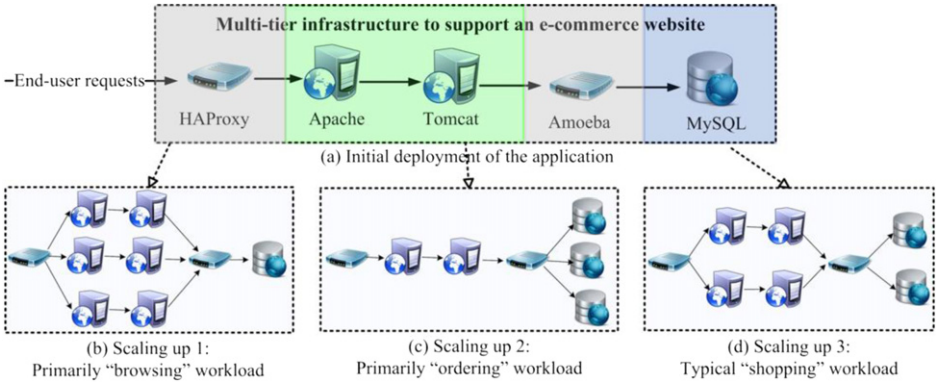

Fig. 1. Multi-tier infrastructure of an e-commerce website and three types of workloads.

perform more efficient cloud elasticity management. The paper features four key elements:

- Cost-aware criteria: a flexible analytical model is developed to capture the behaviour of multi-tier applications. Cost-aware criteria are introduced to measure the effect of cost of resources on every unit of response time.
- Workload-adaptive scaling: using the above criteria, a Cost-Aware Scaling (CAS) algorithm is designed to handle changing workloads of multi-tier applications by adaptively scaling up and down bottlenecked tiers within applications.
- Automation of application scaling: a standard and extensible specification is introduced to describe the properties of the servers, including their VM configuration, IaaS user settings, linking relationships and other constraints. Based on this specification, the best cost-aware scaling approach required for an application can be automatically computed and executed.
- Implementation and experimental evaluation: an intelligent platform based on the CAS algorithm is implemented to automate the scaling process of cloud applications on the IC-Cloud infrastructure [17]. The proposed cost-aware approach is tested using an industry standard benchmark [18] and the test results show: (1) the CAS algorithm responds to changing workloads effectively by scaling applications up and down appropriately to meet their QoS requirements; (2) deployment costs are reduced compared to other scaling techniques.

The remainder of this paper is organised as follows: Section 2 illustrates the need for this new approach to elasticity by describing some current examples and challenges; Section 3 discusses related work; Section 4 provides a more detailed overview of the properties of typical multi-tier applications and describes the architecture of the Imperial Smart Scaling engine (iSSe) implemented to support our approach; Section 5 explains the proposed CAS algorithm and its details; Section 6 reports the experimental evaluation of the algorithm's effectiveness; Sections 7 and 8 present discussion of the approach and summarise directions for future work.

2. Motivation

This section illustrates two challenges that need to be addressed in order to achieve elastic scaling in a large class of multi-tier applications deployed on IaaS clouds. Without loss of generality, we use a simple example based on an e-commerce website to capture the typical behaviour of such applications. Also for simplicity, we focus only on applications that are deployed on the resources of single IaaS cloud provider. As discussed in the introduction, the workload for such applications depends on the number of end users submitting requests at the same time. The workload may be composed of different types of requests that need to be handled by different parts of the application. For example some end users may be browsing the web site itself, while others may be querying the product catalogue or making a payment transaction. To highlight the key challenges, consider the typical infrastructure for a multi-tier e-commerce application as shown in Fig. 1(a). This application is composed of five tiers of components (servers): the HAProxy and Amoeba load-balancing servers, the Apache HTTP server, the Tomcat application server and the MySQL database server. These servers work together to handle end users' requests. Depending on the application workload, servers at each tier can be stressed at different times and the application owner would need to scale up or down the appropriate resources to maintain the QoS of their application.

Challenge 1: *Cost-aware scaling*. In a highly scalable cloud environment where computing resources are consumed as a utility such as water and electricity [19], application owners would expect to spend the least cost for the desired application performance. To this aim, the elastic scaling must take cost-aware criteria into consideration and use them to guide application scaling. Take Fig. 1(a)'s application for example, these criteria should be aware of both the cost of adding a server (e.g., an Apache or a MySQL) and the performance effect brought by this scaling up (e.g., reducing response time).

When the application is initially deployed (Fig. 1(a)), five servers of this application are hosted across different VMs to support a small number of customers. When the demand increases, the application should be scaled up. An interesting point here is that this scaling process is greatly influenced by the behaviour (i.e., the type of workload) of the application itself. We examine three typical types of workloads, where each workload places varying demands on different tiers of the application. In the primarily "browsing" workload (Fig. 1(b)), end users mainly browse webpages and preview products. This workload mainly stresses the service tier including the Apache and Tomcat servers, so their resources are saturated and the number of these servers needs to be increased. By contrast, the primarily "ordering" workload mainly stresses the storage tier including the MySQL database and so the number of these database servers needs increasing (Fig. 1(c)). Finally, the typical "shopping" workload simultaneously stresses the service and storage tiers and so the number of servers in both two tiers is increased (Fig. 1(d)).

Challenge 2: *Workload-adaptive scaling*. Due to the dynamic cloud environment, two types of uncertainties exist in the application workload: (1) the type of workload, such as Fig. 1's three types of workload; (2) the volume of workload, which is denoted in terms of the arrival rate of incoming requests, namely the number of incoming requests per time unit. In this context, the elastic scaling must be adaptive to the changing workload, and such adaptive scaling has triple meanings. First of all, bottleneck tiers of applications should be automatically identified both for scaling up and down. Secondly, scaling should be performed as