Future Generation Computer Systems 55 (2016) 41–50

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Impact of user patience on auto-scaling resource capacity for cloud services

Marcos Dias de Assunção [a,*], Carlos H. Cardonha [b], Marco A.S. Netto [b], Renato L.F. Cunha [b]

[a] *Inria, ENS de Lyon, France*

[b] *IBM Research, Brazil*

HIGHLIGHTS

- Mechanisms for resource auto-scaling in clouds considering users' patience.
- Methods for determining the step size of scaling operations under bound and unbounded maximum capacity.
- Users patience model inspired in prospect theory.

ARTICLE INFO

Article history:
Received 19 December 2014
Received in revised form
25 June 2015
Accepted 2 September 2015
Available online 10 September 2015

Keywords:
Cloud computing
Auto-scaling
Resource management
Scheduling

ABSTRACT

An important feature of most cloud computing solutions is auto-scaling, an operation that enables dynamic changes on resource capacity. Auto-scaling algorithms generally take into account aspects such as system load and response time to determine when and by how much a resource pool capacity should be extended or shrunk. In this article, we propose a scheduling algorithm and auto-scaling triggering strategies that explore user patience, a metric that estimates the perception end-users have from the Quality of Service (QoS) delivered by a service provider based on the ratio between expected and actual response times for each request. The proposed strategies help reduce costs with resource allocation while maintaining perceived QoS at adequate levels. Results show reductions on resource-hour consumption by up to approximately 9% compared to traditional approaches.

1. Introduction

Cloud computing has become a popular model for hosting enterprise and backend systems that provide services to end-users via Web browsers and mobile devices [1,2]. In this context, a typical scenario often comprises a provider of computing and storage infrastructure, generally termed as Infrastructure as a Service (IaaS) or simply a *cloud provider*; a *service provider* who offers a web-based service that can be deployed on Virtual Machines (VMs) hosted on the cloud; and the *clients*, or *end-users*, of such a service.

In this work, we investigate challenges faced by the service providers who compose their resource pools by allocating machines from cloud providers. Meeting end-users expectations is crucial for the service provider's business, and in the context of Web applications, these expectations typically refer to short requests response times. Simultaneously, there are costs associated with resource allocation, so resource pools with low utilisation are economically undesired. Moreover, clients demands are uncertain and fluctuate over time, so the problem of resource allocation faced by service providers is clearly non-trivial.

Elasticity, a selling point of cloud solutions, enables service providers to modify the size of resource pools in near real-time via auto-scaling strategies, allowing hence for reactions to fluctuations on clients' demand. Current strategies typically monitor and predict values of target system metrics, such as response time and utilisation level of relevant resources (*e.g.*, CPU, memory, and network bandwidth), and employ rule-based systems to trigger auto-scaling operations whenever predefined thresholds are violated. The parameters employed by these strategies do not allow them to explore *heterogeneity of expected response times* and *of tolerance to delays* that end-users have towards service providers. In combination with actual response times, these two elements define the perception end-users have from a service QoS, which we will refer to as *user patience*.

Previous work in other domains has shown that end-users can present heterogeneous patience levels depending on their

[*] Corresponding author.
E-mail address: assuncao@acm.org (M.D. de Assunção).

http://dx.doi.org/10.1016/j.future.2015.09.001
0167-739X/© 2015 Elsevier B.V. All rights reserved.

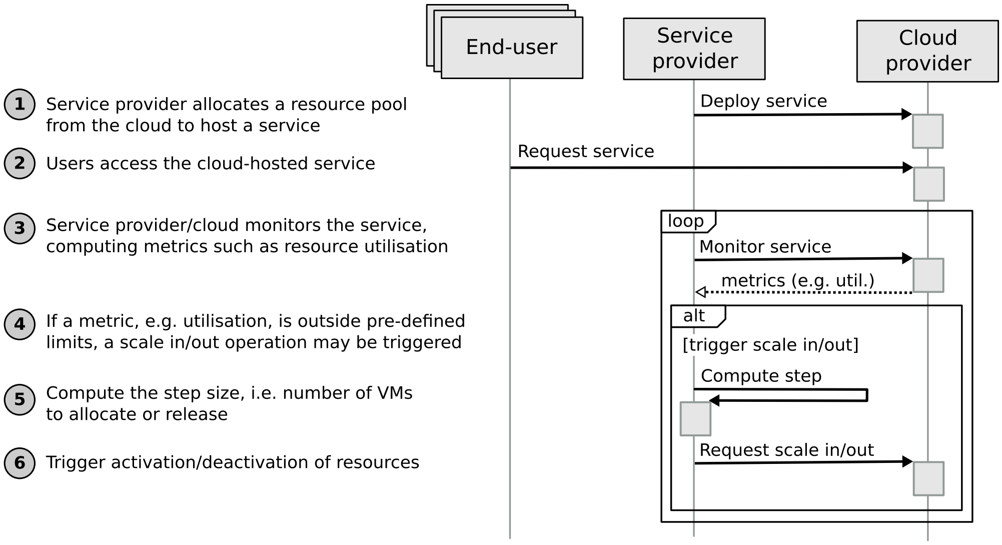
End-user
Service provider
Cloud provider
1 Service provider allocates a resource pool from the cloud to host a service
2 Users access the cloud-hosted service
3 Service provider/cloud monitors the service, computing metrics such as resource utilisation
4 If a metric, e.g. utilisation, is outside pre-defined limits, a scale in/out operation may be triggered
5 Compute the step size, i.e. number of VMs to allocate or release
6 Trigger activation/deactivation of resources
Deploy service
Request service
loop
Monitor service
metrics (e.g. util.)
alt
[trigger scale in/out]
Compute step
Request scale in/out

Fig. 1. Service provision model and auto-scaling scenario considered in this work.

context [3]. Moreover, in a society where human attention is increasingly becoming scarce, and where users perform multiple concurrent activities [4][1] and use multiple devices [5],[2] response time might not be the sole element defining the perceptions end-users have from a service's QoS. Our previous work investigated how application instrumentation [6–8], and end-user context and profiling [9] could be used in the collection of honest signals determining how clients interact with a service provider and what their levels of patience are when making requests.

The present work investigates how patience can be explored by auto-scaling strategies. Compared to our previous work [8], in addition to considering provisioning time, this article makes the following contributions:

- A scheduling algorithm and a patience-based auto-scaling triggering strategy for a service provider to minimise the number of resources allocated from a cloud provider, applicable to scenarios where the maximum number of available resources is unbounded (Section 3);
- Experiments with bounded and unbounded numbers of resources available to service providers that show reductions of up to nearly 9% on resource-hour consumption compared to traditional auto-scaling strategies (Section 4).

2. Problem description

In this section we describe the scenarios investigated in this work, present the assumptions regarding capacity limitations on the pool of resources offered by the cloud provider, and explain the elements of client behaviour which are taken into account by our algorithms.

Fig. 1 depicts the service hosting model and the main steps of auto-scaling operations. A service provider allocates a pool of VMs from a cloud provider to deploy its service, which is then accessed by end-users. The service provider and/or cloud provider periodically monitors the status of the pool, thus computing metrics such as resource utilisation. A metric lying outside certain lower and upper limits may trigger a scale in/out operation. A step size – the number of VMs to be allocated or released – is computed before a change to the capacity of the resource pool is requested.

Resources are pairwise indistinguishable, that is, they are VMs which have the same cost and performance characteristics. We also consider provisioning time; namely, after triggering the activation of resources, a service provider must wait for non-negligible time to use them. Finally, the scenarios either have a maximum number of resources that can be allocated (*bounded*) or do not have limitations of this nature (*unbounded*); these two scenarios are considered because they pose different levels of stress on resource utilisation.

We take into account that resources are paid only for the periods in which they were allocated in a per-minute billing model. This assumption is not unrealistic, as providers such as Microsoft Azure offer this type of service. Consequently, the strategies proposed in this article may allocate and deallocate a given resource several times within one hour in order to improve utilisation.

This work also considers short tasks whose execution times are all equal and in the order of a few seconds, reflecting hence scenarios typically faced by providers of web services. Nevertheless, the concepts and results can, without any loss of generality, be reproduced in scenarios with either shorter or longer tasks.

We assume that expectations on response time may vary across individuals, a realistic assumption in certain practical settings; for example, whereas end-users expect to get results from Web searches in a couple of seconds at maximum, clients performing large-scale graph mining operations, which can take from minutes to hours, tend to be more tolerant.

Differences between expected and actual response times for submitted requests have an impact on users' patience (either positive or negative) whose weight decreases over time. More precisely, changes on the patience level of an end-user take place after the execution of each request and are described by a function applied to the ratio between expected and actual response times. Based on Prospect Theory [10], we assume that the negative impact of delays on users' patience exceeds the benefits of fast responses (see Fig. 2).

The service provider periodically evaluates the system utilisation and/or user patience and decides on whether the capacity of its resource pool should be expanded (shrunk) by requesting (releasing) resources from (to) the cloud. The questions we therefore seek to address are: (i) *how to determine critical times when auto-scaling is necessary or avoidable by exploiting information on users'*

[1] Ofcom reports that up to 53% of UK adults media multi-task while watching TV.

[2] Accenture's study shows users expect other devices to augment content shown on TV.