Chapter 28
Admission Control in the Cloud:
Algorithms for SLA-Based Service Model

Jose Luis Vazquez-Poletti
Universidad Complutense de Madrid, Spain

Rafael Moreno-Vozmediano
Universidad Complutense de Madrid, Spain

Ignacio M. Llorente
Universidad Complutense de Madrid, Spain

ABSTRACT

Cloud computing is a paradigm that allows the flexible and on-demand provisioning of computing resources. For this reason, many institutions and enterprises have moved their data centers to the Cloud and, in particular, to public infrastructures. Unfortunately, an increase in the demand for Cloud results in resource shortages affecting both providers and consumers. With this factor in mind, Cloud service providers need Admission Control algorithms in order to make a good business decision on the types of requests to be fulfilled. Cloud providers have a desire to maximize the net income derived from provisioning the accepted service requests and minimize the impact of unprovisioned resources. This chapter introduces and compares Admission Control algorithms and proposes a service model that allows the definition of Service Level Agreements for the Cloud.

DOI: 10.4018/978-1-4666-6178-3.ch028

Copyright © 2014, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

INTRODUCTION

Cloud Computing is being actively used by academic institutions (Wang et al., 2008; Iosup et al., 2011) and business enterprises (Leukel, Kirn, & Schlegel, 2011; Lin, Dasmalchi, & Zhu, 2008). In the Cloud computing service provisioning paradigm, three main levels can be identified: *Infrastructure and Network as a Service* (IaaS and NaaS), the lowest level that refers to the computing resources and the networks connecting them; *Platform as a Service* (PaaS), which is a complete solution stack built regardless of the layer underneath for developing and deploying applications; and *Software as a Service* (SaaS), the level where the provided applications and their associated data reside.

Cloud services may be offered by either private or public infrastructures, depending on who owns the physical machines. Virtualization technologies such as OpenNebula (Moreno, Montero, & Llorente, 2009), Nimbus (Foster et al., 2006) or Eucalyptus (Nurmi et al., 2009) can be used for building private Clouds, also named internal or corporate Cloud. On the other hand, third party institutions offer public Cloud infrastructures, charging its users per service by unit of time. Amazon EC2 for IaaS and Google App Engine for PaaS are some examples of these pay-as-you-go infrastructures.

Much research has been conducted from the point of view of the public Cloud infrastructure user, and several optimal resource provisioning strategies have been proposed (Juve et al., 2009; Vazquez-Poletti, Barderas, Llorente, & Romero, 2010). However, this chapter is focused on the service provider side, considering that *Service Level Agreements* (SLAs) are to be satisfied when resources are provided. Not provisioning the required resources means breaking a SLA. These breaks, whose severity depends on the SLA, have been analyzed and mechanisms to minimize their impact from the client side have been proposed (Ramakrishnan et al., 2009; Cachin, Keidar, & Shraer, 2009).

Admission Control policies help avoiding or smoothing out these situations from the provider side, by deciding whether to accept a new service that is requesting resources or not. As it will be shown in this chapter, some of these policies consider accepting services even with a possibility of incurring SLA breaks to produce a bigger net benefit, while other policies follow a more conservative strategy.

The main objectives pursued in this chapter are twofold. The first one is to provide a novel definition of SLAs for Cloud service providers that will make the tailoring of Admission Control algorithms easier. The second one is to introduce a double set of Admission Control algorithms that are studied both from an economic and failure rate point of view.

Finally, this contribution represents an extension of the work[1] that was initiated by a previous research paper (Vazquez-Poletti, Moreno-Vozmediano, & Llorente, 2012). In this chapter, the experiments have been considerably expanded in order to consider three typical service provision scenarios.

BACKGROUND

Policies governing the deployment in a Cloud infrastructure include aspects such as the performance levels, the degree of trust expected from the service provider, or the level of risk according to cost thresholds (Ferrer et al., 2012). For example, SLA mechanisms offered by the OPTIMIST project (Comuzzi et al., 2009) aim to evaluate levels of trust and risk, even negotiating the use of license protected software, while actual mechanisms for Cloud computing are normally limited to cost/performance tradeoffs. In Cluster computing,

15 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:
www.igi-global.com/chapter/admission-control-in-the-cloud/115450

Related Content

Requirements for the Testable Specification and Test Case Derivation in Conformance Testing
Tanja Toroiand Anne Eerola (2007). *Verification, Validation and Testing in Software Engineering (pp. 136-156).*
www.irma-international.org/chapter/requirements-testable-specification-test-case/30750

A Service Component Model and Implementation for Institutional Repositories
Yong Zhang, Quansong Deng, Chunxiao Xing, Yigang Sunand Michael Whitney (2012). *Advanced Design Approaches to Emerging Software Systems: Principles, Methodologies and Tools (pp. 61-81).*
www.irma-international.org/chapter/service-component-model-implementation-institutional/55436

A Comparative Analysis of Software Engineering with Mature Engineering Disciplines using a Problem-Solving Perspective
Bedir Tekinerdoganand Mehmet Aksit (2011). *Modern Software Engineering Concepts and Practices: Advanced Approaches (pp. 1-18).*
www.irma-international.org/chapter/comparative-analysis-software-engineering-mature/51966

Retrofitting Existing Web Applications with Effective Dynamic Protection Against SQL Injection Attacks
San-Tsai Sunand Konstantin Beznosov (2010). *International Journal of Secure Software Engineering (pp. 20-40).*
www.irma-international.org/article/retrofitting-existing-web-applications-effective/39007

Validation and Verification of Software Systems Using Virtual Reality and Coloured Petri Nets
Hyggo Oliveira de Almeida, Leandro Silva, Glauber Ferreira, Emerson Loureiroand Angelo Perkusich (2009). *Software Applications: Concepts, Methodologies, Tools, and Applications (pp. 3361-3380).*
www.irma-international.org/chapter/validation-verification-software-systems-using/29566