Chapter 3

The Challenge of Service Level Scalability for the Cloud

Luis M. Vaquero
Telefónica Investigación y Desarrollo, Spain

Juan Cáceres
Telefónica, Spain

Daniel Morán
Universidad Nacional de Educación a Distancia, Spain

ABSTRACT

This paper presents a brief overview of the available literature on distributed systems scalability that serves as a justification for presenting some of the most prominent challenges that current Cloud systems need to face in order to deliver their pledged easy-to-use scalability. Through illustrative comparisons and examples, this paper aims to make the reader’s acquaintance with this long needed problem in distributed systems: user-oriented service-level scalability. Scalability issues are analyzed from the Infrastructure as a Service (IaaS) and the Platform as a Service (PaaS) point of view, as they deal with different functions and abstraction levels. Next generation Cloud provisioning models rely on advanced monitoring and automatic scaling decision capabilities to ensure quality of service (QoS), security and economic sustainability.

INTRODUCTION

Cloud computing is one of the most prominent terms employed in the latest years within the Information and Communication Technologies (ICT) field. This computing paradigm is commonly associated to new provisioning mechanisms of infrastructure resources. However, there is more to Cloud than mere infrastructure arrangement (Vaquero et al., 2009). The Cloud aims at offering every networked resource as a service to be consumed by a variety of actors far beyond any type of administrative domain or organization.

These general ideas may be familiar to those working in any type of distributed systems technology. The Cloud was born as the harvest of many different previous seeds; it is nourished
The Challenge of Service Level Scalability for the Cloud

by the maturity and joint of technologies such as virtualization or the Grid.

According to the functionality and abstraction level, Cloud technologies can be classified into three categories (Vaquero et al., 2009): 1) Infrastructure as a Service (IaaS), where Cloud vendors provide interfaces for the provision and management of virtualized resources (virtual machines, storage or network); 2) Platform as a Service (PaaS) which offers development tools, application containers and specialized platform technologies, and hides the details of IaaS infrastructure management; and 3) Software as a Service (SaaS), where applications are offered on demand. Currently, there are many services in the market that offer these features, for example: IaaS Clouds such as Amazon Elastic Compute Cloud (EC2) and Simple Storage Service (S3), GoGrid or Flexiscale; incipient PaaS Clouds Google App Engine, Windows Azure or SalesForce; and SaaS such as Web mail, remote desktops, Application Stores or Web 2.0 social networks.

The Cloud is already delivering important advantages that place it in an optimal position to overcome hype and evolve towards maturity. The illusion of a virtually infinite computing infrastructure, the employment of advanced billing mechanisms allowing for a pay-per-use mode on shared multitenant resources, the easiness to create and destroy new machines, the simplified programming mechanisms, etc. are already part of our common mindset and seem to be here to stay as a new ICT utility such as the electricity or water distribution systems. Figure 1 shows how Cloud types are layered by abstraction levels and management functions. In spite of the claimed advantages, the Cloud still needs some work with regards to a variety of different subjects.

Note that an upper layer may rely on intermediate layers or may skip some of them. For example, SaaS clouds can be directly constructed on physical hardware as it has been done traditionally, of course, but they could be also deployed using IaaS functions, or PaaS over IaaS. In the purest SaaS/PaaS/IaaS Cloud model: 1) software is offered as a service and it is composed by service components which are 2) hosted in platform containers (stack of OS, middleware, application containers and libraries); platform containers run in 3) virtual machines (VMs) which are hosted in

Figure 1. The cloud stack

![Figure 1. The cloud stack](image-url)