Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 18 (2013) 159 – 168

Procedia Computer Science

International Conference on Computational Science, ICCS 2013

Elastic Memory Management of Virtualized Infrastructures for Applications with Dynamic Memory Requirements

Germán Moltó*, Miguel Caballer, Eloy Romero, Carlos de Alfonso

[a]*Instituto de Instrumentación para Imagen Molecular (I3M). Centro mixto CSIC Universitat Politècnica de València CIEMAT, camino de Vera s/n, 46022 Valencia, España*

Abstract

This paper addresses the impact of vertical elasticity for applications with dynamic memory requirements when running on a virtualized environment. Vertical elasticity is the ability to scale up and scale down the capabilities of a Virtual Machine (VM). In particular, we focus on dynamic memory management to automatically fit at runtime the underlying computing infrastructure to the application, thus adapting the memory size of the VM to the memory consumption pattern of the application. An architecture is described, together with a proof-of-concept implementation, that dynamically adapts the memory size of the VM to prevent thrashing while reducing the excess of unused VM memory. For the test case, a synthetic benchmark is employed that reproduces different memory consumption patterns that arise on real scientific applications. The results show that vertical elasticity, in the shape of dynamic memory management, enables to mitigate memory overprovisioning with controlled application performance penalty.

Keywords: Cloud computing, Cluster computing, Virtualization, Elasticity

1. Introduction

With the advent of virtualization, the wide use of commodity hardware and the advances in networks, the idea of utility computing, concerning the access to computing (and storage) resources on a pay-per-use basis, is taking shape. Cloud computing provides, at the moment, the closest implementation of utility computing, by providing a model for enabling ubiquitous, on-demand network access to a pool of configurable computing resources that can be rapidly provisioned and released with minimal provider interaction, according to the NIST definition [1].

Elasticity, i.e., the ability to rapidly provision and release resources, is often highlighted as one of the key features of Cloud computing [2], for it allows to dynamically adapt the underlying virtual computing infrastructure to the dynamic execution requirements of applications. This is specially true in the case of Infrastructure as a Service (IaaS) Cloud providers, where users request and release specific resources (mainly computational and storage capabilities) and pay for its usage.

On the one hand, horizontal elasticity has the ability of rapidly provisioning and releasing nodes in order to deal with an important change in the workload and to avoid additional costs (for example, from paying for unused resources). A typical example that uses horizontal elasticity is a web-based application with a fleet of n Virtual Machines (VMs) where incoming requests are handled by a load balancer that distributes them to those n VMs.

*Corresponding author. Tel.: +34963877007 Ext. 88254 ; fax: +34963877274.
E-mail address: gmolto@dsic.upv.es.

Selection and peer review under responsibility of the organizers of the 2013 International Conference on Computational Science
doi:10.1016/j.procs.2013.05.179

Whenever the (CPU) load of VMs exceeds a certain threshold, the fleet is increased to m VMs (where $m > n$). If the load decreases, some VMs are shut down in order to reduce costs. On the other hand, vertical elasticity has the ability to rapidly modify the capabilities of single VMs, typically in terms of CPU and RAM [3].

In the last years, many research efforts in the area of elasticity in the Cloud have focused on horizontal elasticity, while few works currently address vertical elasticity. In fact a report from the European Commission on the Future of Cloud Computing states that vertical elasticity is one of the areas not fully addressed by current commercial efforts, although it is acknowledged its importance for the efficient adaptation of infrastructures to applications [4].

As an example, the ability to dynamically modify the memory of a VM at runtime without any service disruption represents an important capability for applications with dynamic memory requirements [5]. This means to automatically adapt the underlying virtual computing platform (i.e., the VM or the set of VMs) to the runtime profile of memory consumption of the application. This introduces a benefit for resource providers, because a reduction of the memory size of the VM increases the available memory at the host on which the VM is running. This free memory could be dedicated to other concurrent VMs being run on the same physical machine. This could also lead to reduced costs for the user if public Cloud providers offered support to these techniques (which is not the case as of 2013, considering the main providers, such as Amazon EC2 or Windows Azure). Major public Cloud providers currently charge on a per-hour basis for a given computing capacity, regardless of their actual usage. If lower average memory consumption resulted in a lower cost, users could integrate these techniques in order to cut down costs, which would result in a win-win situation for both users and resource providers.

Concerning vertical elasticity, while open-source hypervisors such as KVM and Xen include support for techniques like memory ballooning, open source Virtual Infrastructure Managers (VIMs) such as OpenNebula and OpenStack do not currently include such support out of the box. In particular, this paper introduces support to vertical elasticity, through dynamic memory management, with a proof-of-concept implementation using an ad hoc modified version of OpenNebula [6] and the KVM hypervisor in order to dynamically shrink and grow the VMs' memory.

There are previous works in the area of elasticity on Cloud infrastructures. For example, the work by Ali-Eldin et al. [7] includes an adaptive horizontal elasticity controller for Cloud infrastructures, where a Cloud service is modelled by queue theory and service load is estimated to build proactive controllers. There are also works related to augmenting the computing capacities of mobile devices with the elastic capabilities of Cloud computing [8]. The aforementioned works focus on horizontal elasticity and do not address the topic of vertical elasticity, which is the main focus of this paper. Concerning vertical elasticity, the work by Kalyvianaki et al. [9], integrates the Kalman filter into feedback controllers to dynamically allocate CPU resources to VMs. The CPU utilization is tracked and the allocations are updated accordingly. The work by Zhao et al. [10] describes a system that monitors the memory usage of each VM to predict its memory needs and reallocate the host memory. They use the Xen hypervisor. In [11], Dawoud et al. focus on vertical elasticity and compare its benefits and drawbacks with respect to horizontal elasticity. They propose an Elastic VM architecture which scales number of cores, CPU capacity and memory, by using the Xen hypervisor. They show that by adapting the VM capacities to the requirements of the application (web-tier based), fine-grained resource provisioning is possible. Their case study exclusively focuses on dynamically altering the virtual CPUs and does not address memory scaling.

As opposed to previous works, this paper contributes a study of dynamic memory management on virtualized infrastructures for the execution of (scientific) applications with dynamic memory requirements. Techniques for vertical elasticity management are addressed and issues concerning the elasticity rules are covered, pointing out the main implications to be considered when deploying these techniques. A system has been implemented to support these techniques on our private Cloud infrastructure and a case study with different memory consumption patterns is executed in order to assess the effectiveness of an elastic management of the memory size of VMs.

After the introduction, the remainder of the paper is structured as follows. First of all, section 2 describes the methods employed to manage the vertical elasticity in terms of dynamic memory management. Then, section 3 describes the architecture employed to dynamically modify the memory size of the VMs, describing the proposed implementation. Next, section 4 describes a case study that executes a synthetic application that reproduces several dynamic memory consumption patterns on a virtual infrastructure. Later, section 5 discusses the main implications of the results both from the point of view of the user and the resource provider. Finally, section 6 summarises the paper and points to future work.