Future Generation Computer Systems 56 (2016) 1–10

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Automatic memory-based vertical elasticity and oversubscription on cloud platforms

Germán Moltó *, Miguel Caballer, Carlos de Alfonso

Instituto de Instrumentación para Imagen Molecular (I3M), Centro mixto CSIC – Universitat Politècnica de València – CIEMAT, camino de Vera s/n, 46022 Valencia, Spain

HIGHLIGHTS

- We describe a memory oversubscription framework for Cloud Management Platforms.
- Transient overcommitment of physical hosts increases consolidation.
- Automatic vertical elasticity is managed via memory ballooning and live migration.
- Horizontal and vertical elastic virtual clusters are used in production.

ARTICLE INFO

Article history:
Received 15 April 2015
Received in revised form
10 September 2015
Accepted 6 October 2015
Available online 22 October 2015

Keywords:
Cloud computing
Cloud Management platform
Virtualisation
Vertical elasticity
Memory overcommitment
Oversubscription

ABSTRACT

Hypervisors and Operating Systems support vertical elasticity techniques such as memory ballooning to dynamically assign the memory of Virtual Machines (VMs). However, current Cloud Management Platforms (CMPs), such as OpenNebula or OpenStack, do not currently support dynamic vertical elasticity. This paper describes a system that integrates with the CMP to provide automatic vertical elasticity to adapt the memory size of the VMs to their current memory consumption, featuring live migration to prevent overload scenarios, without downtime for the VMs. This enables an enhanced VM-per-host consolidation ratio while maintaining the Quality of Service for VMs, since their memory is dynamically increased as necessary. The feasibility of the development is assessed via two case studies based on OpenNebula featuring (i) horizontal and vertical elastic virtual clusters on a production Grid infrastructure and (ii) elastic multi-tenant VMs that run Docker containers coupled with live migration techniques. The results show that memory oversubscription can be integrated on CMPs to deliver automatic memory management without severely impacting the performance of the VMs. This results in a memory management framework for on-premises Clouds that features live migration to safely enable transient oversubscription of physical resources in a CMP.

1. Introduction

Elasticity [1], or the ability to rapidly provision and release resources, is one of the integral characteristics of Cloud Computing. Horizontal elasticity is commonly employed to provision additional computational nodes in order to sustain the quality of service delivered by an architecture deployed on a Cloud platform, specially after an increase in the number of users or workload. Horizontal elasticity has been extensively studied in the past, with services already available for public Clouds, such as Auto Scaling[1] for Amazon Web Services (AWS), and Heat[2] for OpenStack.

Instead, vertical elasticity enables to increase and decrease the number of resources allocated to a single Virtual Machine (VM). The increased support to techniques such as *memory ballooning* [2] and *CPU hot plugging* by popular hypervisors such as KVM, Xen or VMware paves the way for vertical elasticity to be adopted by Cloud platforms. However, popular open source CMPs such as OpenNebula and OpenStack do not currently support vertical elasticity without stopping the VMs. As an example, the

* Corresponding author.
E-mail address: gmolto@dsic.upv.es (G. Moltó).

http://dx.doi.org/10.1016/j.future.2015.10.002
0167-739X/

[1] Auto Scaling: http://aws.amazon.com/autoscaling.
[2] Heat: https://wiki.openstack.org/wiki/Heat.

KVM hypervisor fully supports memory ballooning in order to dynamically modify the allocated memory to a given VM without any downtime, and the main Operating Systems (OSs) support this feature. However, CMPs require to stop the VM in order to change its allocated memory.

In our previous work [3] we demonstrated the benefits of introducing vertical elasticity to dynamically adjust the allocated memory of VMs to their current memory consumption, specially for applications with dynamic memory requirements during their execution. In fact, the number of VMs that one physical machine can support is typically limited by its memory size. Besides, users tend to overestimate the amount of memory required by their applications resulting in unused memory that could be dedicated to additional VMs running on the same physical machine [4]. In addition, CMPs typically provide templates, such as the *flavours* in OpenStack, which enforce a certain amount of memory size regardless of the actual memory requirements of the application. Just as airlines sell more tickets than available seats (i.e. oversubscribe the plane) in the hope that some passengers do not show up, Cloud providers can oversubscribe their resources by deploying additional VMs in a host, in the hope that VMs will actually use less memory than initially requested.

However, this situation might incur in memory overload for a host, where the sum of used memory of its VMs exceeds the physical memory of the host. Therefore, oversubscription [5] is a technique that can lead to an increase in the number of VMs per physical host though it can have an impact on the Quality of Service and probably violate the Service Level Agreement established by the Cloud provider. However, oversubscription can enable Cloud providers to better use the available memory in their physical systems if the appropriate countermeasures are introduced. As Williams et al. [6] state, in well-provisioned datacenters, overload is unpredictable, relatively rare, uncorrelated, and transient, indicating that an opportunity exists for memory oversubscription in those facilities.

In this paper we introduce CloudVAMP (*Cloud Virtual machine Automatic Memory Procurement*) a memory oversubscription framework that can be integrated in an on-premises CMP to automatically monitor the VMs and to dynamically adjust their allocated memory to adapt to the current memory requirements of their running applications. Without any user intervention, the system automatically manages the memory of the VMs (or a subset of VMs) irrespective of the memory initially allocated by the user. This introduces enhanced VM consolidation per physical node while live migration is employed to prevent overload of the physical machines.

The remainder of the paper is structured as follows. First, Section 2, describes the related works in the area of vertical elasticity and memory oversubscription. Next, Section 3 briefly describes the problem addressed and the underlying technologies employed. Later, Section 4 describes the architecture of CloudVAMP, in order to manage vertical elasticity in an on-premises Cloud. Then, Section 5 describes two case studies carried out to assess the behaviour and benefits of the developed platform. Finally, Section 6 summarises the paper and points to future work.

2. Related work

There can be found other works in the literature that have focused on vertical elasticity and memory oversubscription (also called in the literature memory overcommitment), though most of them are just focused on virtualisation platforms and, thus, not covering the intricacies of CMPs. In [7], the authors propose an Elastic VM architecture that scales the number of cores, CPU capacity and memory using the Xen hypervisor. They study the adaptation of the VM capacities to the requirements of a web application. However, their case study does not address memory scaling but only increasing the virtual CPU allocation.

In [8], a system to provide proactive dynamic memory allocation based on the Bayesian predictions is introduced to increase server consolidation. In [9], the Ginkgo memory overcommitting framework is introduced, which dynamically estimates VM memory requirements for applications and automates the distribution of memory across VMs through ballooning techniques. It uses performance profiles of the applications to characterise incoming load. The case study focuses on VMs running on a single physical host. These two works focus on a set of virtual machines running in a single hypervisor, while our work focuses at the whole infrastructure provided by the CMP, involving memory management across multiple physical hosts. In [10] an extension of ballooning techniques is applied to applications, using as example a database engine and the Java runtime, to reallocate memory between memory managers of different applications. However, these require modifications of the Xen Balloon Driver and does not address the overcommitment problems that arise in CMPs.

Overdriver [6] is a system to mitigate the problems that arise in oversubscribed virtualised hosts, by automatically deciding when to use network memory, using a cooperative swap approach, or live migration depending on whether the workload is considered to be transient or sustained, respectively. However, they do not consider memory ballooning as a mitigation strategy for oversubscription. This is the case of the work by Hwang et al. [11] where a system to opportunistically use memory during periods of light loads is introduced. For that, they allow the hypervisor to dynamically allocate memory at fine granularity, focusing on disk and application level caches. The work by Baset et al. [12] describes the different techniques employed to alleviate oversubscription and mitigate overload. They designed an event-driven simulator to develop an understanding of oversubscription. However, they focus exclusively on offline and live migration but ballooning techniques are discarded.

Regarding memory ballooning, the KVM hypervisor has a project called Automatic Ballooning [13] where the management of the balloon is automatic. When the host is under pressure, it asks guests to relinquish memory. When a guest detects memory pressure, it gets some memory back from the host. This requires Linux kernel 3.10+ and a specific version of QEMU. However, this approach focuses exclusively on the VMs running on a single physical machine and, thus, it does not solve the problems that arise when the host is overloaded, specially within an on-premises Cloud, where VMs could be live migrated across other physical hosts to restore the level of service.

The most similar work to our proposal is the one carried out by Litke [14], where the Memory Overcommitment Manager (MOM) is introduced. This system requires a daemon to be installed in the VMs to gather information regarding the memory usage from the VMs and a policy actuator that runs on the host's OS to decide when to increase or decrease memory through memory ballooning techniques. While this approach is of interest for a virtualisation platform where VMs have dynamic memory requirements, it does not introduce countermeasures for overloaded hosts.

As the authors of [5] state, much of the research conducted thus far has focused on managing oversubscription of a single physical machine, though this narrow focus is rather limiting. While other projects successfully manage memory overcommitment at a host level, we have not found any previous work that automatically manages oversubscription in an on-premises Cloud. Therefore, building on previous works in the area we introduce CloudVAMP, a memory management framework for on-premises Clouds that features live migration to safely enable transient oversubscription of physical resources in a CMP.

As opposed to previous work, our approach considers memory management not at a single physical host but at the whole