Future Generation Computer Systems 65 (2016) 57–72

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

A hybrid cloud controller for vertical memory elasticity: A control-theoretic approach

Soodeh Farokhi [a,*], Pooyan Jamshidi [b], Ewnetu Bayuh Lakew [c], Ivona Brandic [a], Erik Elmroth [c]

[a] Faculty of Informatics, Vienna University of Technology, Austria
[b] Department of Computing, Imperial College London, United Kingdom
[c] Department of Computing Science, Umeå University, Sweden

HIGHLIGHTS

- A feedback controller for vertically scale the memory of cloud applications is proposed.
- The controller is able to tune the memory in order to meet the desired performance.
- The application performance and memory utilization are used as decision making criteria.
- The feedback controller guarantees the stability of the cloud application.
- The results show the efficiency in memory usage when the feedback controller is used.

ARTICLE INFO

Article history:
Received 28 October 2015
Received in revised form
15 February 2016
Accepted 22 May 2016
Available online 23 June 2016

Keywords:
Cloud computing
Control theory
Vertical memory elasticity
Interactive application performance
Memory utilization

ABSTRACT

Web-facing applications are expected to provide certain performance guarantees despite dynamic and continuous workload changes. As a result, application owners are using cloud computing as it offers the ability to dynamically provision computing resources (e.g., memory, CPU) in response to changes in workload demands to meet performance targets and eliminates upfront costs. Horizontal, vertical, and the combination of the two are the possible dimensions that cloud application can be scaled in terms of the allocated resources. In vertical elasticity as the focus of this work, the size of virtual machines (VMs) can be adjusted in terms of allocated computing resources according to the runtime workload. A commonly used vertical resource elasticity approach is realized by deciding based on resource utilization, named capacity-based. While a new trend is to use the application performance as a decision making criterion, and such an approach is named performance-based. This paper discusses these two approaches and proposes a novel hybrid elasticity approach that takes into account both the application performance and the resource utilization to leverage the benefits of both approaches. The proposed approach is used in realizing vertical elasticity of memory (named as vertical memory elasticity), where the allocated memory of the VM is auto-scaled at runtime. To this aim, we use control theory to synthesize a feedback controller that meets the application performance constraints by auto-scaling the allocated memory, i.e., applying vertical memory elasticity. Different from the existing vertical resource elasticity approaches, the novelty of our work lies in utilizing both the memory utilization and application response time as decision making criteria. To verify the resource efficiency and the ability of the controller in handling unexpected workloads, we have implemented the controller on top of the Xen hypervisor and performed a series of experiments using the RUBBoS interactive benchmark application, under synthetic and real workloads including Wikipedia and FIFA. The results reveal that the hybrid controller meets the application performance target with better performance stability (i.e., lower standard deviation of response time), while achieving a high memory utilization (close to 83%), and allocating less memory compared to all other baseline controllers.

1. Introduction

Cloud applications face dynamic and bursty workloads generated by variable numbers of users. Therefore, dynamic resource provisioning is necessary not only to avoid the application

[*] Corresponding author.
E-mail addresses: soodeh.farokhi@tuwien.ac.at (S. Farokhi), p.jamshidi@imperial.ac.uk (P. Jamshidi), ewnetu@cs.umu.se (E. Bayuh Lakew), ivona.brandic@tuwien.ac.at (I. Brandic), elmroth@cs.umu.se (E. Elmroth).

http://dx.doi.org/10.1016/j.future.2016.05.028
0167-739X/© 2016 Elsevier B.V. All rights reserved.

performance degradation, but also to efficiently utilize resources. Consequently, infrastructure providers ought to have a resource provisioning technique that allocates resources according to application demands in order to attract customers and to use their resources efficiently.

Since the users of modern interactive applications are becoming increasingly interested to have high and predictable, if not guaranteed, performance, the need for having robust auto-scaling[1] solutions that would meet their SLA is rising for cloud environments. Otherwise, unexpected workloads can cause a poor service performance that kills user's satisfaction. Several studies have shown that increased response times reduce, revenue [1]. For instance, Amazon found a page load slowdown of just 1 s could cost $1.6 billion in sales each year [2]. Google stated only half a second delay in search page generation time dropped traffic by 20% [3].

Resource elasticity, as one of the main selling points of cloud computing [4–6], is defined as the degree to which a cloud service is able to accommodate the varying demands at runtime by dynamically provisioning and releasing resources, such that the available resources match the current demands closely [6]. Two types of resource elasticity are defined: horizontal and vertical. While horizontal resource elasticity allows virtual machines (VMs) to be acquired and released on-demand, vertical resource elasticity allows adjusting computing resources (e.g., CPU or memory) of individual VMs to cope with runtime changes. Accordingly, vertical memory elasticity, as the focus of our work, is the case where the size of the allocated memory of the VM is dynamically changed at runtime. Generally speaking, horizontal resource elasticity is coarse-grained, i.e., VMs are considered as resources, which have static and fixed size configurations. Vertical resource elasticity, on the other hand, is fine-grained: the size of the VMs in terms of a particular computing resource such as CPU or memory can be dynamically changed to an arbitrary size for as short as a few seconds [5].

Horizontal elasticity has been widely adopted by commercial clouds due to its simplicity as it does not require any extra support from the hypervisor. However, due to the static nature and fixed VM size of the horizontal elasticity, applications cannot be provisioned with arbitrary configurations of resources based on their demands. This leads to inefficient resource utilization as well as SLA violations since the demand cannot always exactly fit the size of the VM. To efficiently utilize resources and avoid SLA violations, horizontal elasticity should be complemented with fine-grained resource allocations where the VM sizes can be dynamically adjusted to an arbitrary allocated computing resources according to runtime demands. Moreover, based on a European Commission report on the future of cloud computing [7], vertical elasticity is one of the areas that is not fully addressed by current commercial efforts, although its importance is acknowledged. For example, vertical elasticity is considered as a key enabling technology to realize resource-as-a-service (RaaS) clouds and one of the main driving features of the second-generation Infrastructure as a Service (IaaS 2.0) [8], in which users pay only for the resources they actually use, and cloud providers can use their resources more efficiently and serve more users [5,9].

Nevertheless, from the research point of view, in the last decade, most elasticity research has focused on horizontal, while only few research efforts have addressed vertical elasticity [10] due to lack of support from hypervisors. However, vertical scaling of resources has recently started to be supported by the hypervisors such as Xen [11] and KVM [10]. Unlike horizontal elasticity that is widely supported by almost all commercial cloud providers, only a few cloud providers such as *DotCloud*[2] and *ProfitBricks*[3] have started commercial support for vertical elasticity. However, with the current rate of technological developments and user expectations, the support of vertical elasticity techniques will become necessary for any public cloud providers in the future [10].

In theory, one can turn any computing resource, like CPU or memory, vertically elastic if there is a way to measure its behavior continuously over time and if there is at least one knob to change its behavior. However, the practical exploitation of vertical resource elasticity is very challenging due to the following reasons: (i) intrinsically dynamic and unpredictable nature of the and applications' workloads; (ii) the difficulty in determining which resource (e.g., memory or CPU) is the bottleneck [12]; (iii) non-trivial relationship between the performance metrics (e.g., throughput or response time) and the amount of required resources; (iv) detecting when and how much of resources can be added to or removed from the VM while maintaining the desired application performance.

In this work, we used control theory to synthesize a controller for vertical memory elasticity of cloud applications, i.e., the elastic resource is memory and it is adjusted by the controller. The main motivation behind the choice of control theory in our work is to use this well-established theory for modeling and designing feedback loops to make the cloud applications self-adaptive and achieve a proper balance between fast reaction and better stability. Moreover, since the time to adjust memory at runtime is close to instantaneous, control theory is a good fit. The proposed approach, named *hybrid memory controller*, takes the advantages of both the performance-based (PC) and the capacity-based (CC) elasticity control approaches. As commonly used vertical resource elasticity approaches, CC approaches take the resource utilization as a decision making criterion to do the resource elasticity, while PC approaches, give the priority to the application performance and adjust the resources in accordance to the application performance metrics such as response time. However, a CC approach is inadequate to ensure the application performance, and a PC approach may not be able to provide the level of performance assurance that a CC approach can provide efficient resource utilization as the performance of an application can also be affected, for example, by bugs inside the application or by other resources which are not considered by the controller. Therefore, using both the application performance and the resource utilization at the same time would allocate the right amount of resources for the application while preventing both under and over-provisioning. In summary, at the designed feedback loop of the proposed *hybrid memory controller*, scaling up or down of the allocated memory is considered as the control knob parameter, while the application response time (RT) and VM memory utilization are used as feedback variables.

We evaluated the *hybrid memory controller* using RUBBoS [13] as an interactive benchmark application and compare the results of the hybrid controller with a performance-based controller [14,15] and a capacity-based controller [10]. We validate our approach using synthetic traces generated based on open and closed user loop models [16] along with the real user request traces of Wikipedia [17] and FIFA WorldCup [18] websites, which are the number of requests/users accessing these two websites per unit time. Results show that the *hybrid memory controller* ensures efficient resource utilization as well as meeting application performance.

Contribution. The contribution of this paper lies in developing and experimentally evaluating a vertical memory controller

[1] "Auto-scaling" and "Elasticity" are used alternatively in this paper.

[2] docCloud: https://www.dotcloud.com/.

[3] ProfitBricks: www.profitbricks.com.