Science of Computer Programming 90 (2014) 116–134

Contents lists available at ScienceDirect

Science of Computer Programming

www.elsevier.com/locate/scico

Metrics and techniques for quantifying performance isolation in cloud environments

Rouven Krebs [a,*], Christof Momm [a], Samuel Kounev [b]

[a] SAP AG, 69190 Walldorf, Germany
[b] Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany

HIGHLIGHTS

- Three basic metrics for quantifying performance isolation.
- Decoupled from a concrete scenario and evaluated.
- Round robin or blacklist scheduling for isolation on SaaS.
- System provider or developer could use the isolation metrics.
- Case study quantifying Xen's performance isolation capabilities.

ARTICLE INFO

Article history:
Received 15 October 2012
Received in revised form 5 August 2013
Accepted 6 August 2013
Available online 20 August 2013

Keywords:
Performance
Isolation
Metric
SaaS
Cloud
Multi-tenancy

ABSTRACT

The cloud computing paradigm enables the provision of cost efficient IT-services by leveraging economies of scale and sharing data center resources efficiently among multiple independent applications and customers. However, the sharing of resources leads to possible interference between users and performance problems are one of the major obstacles for potential cloud customers. Consequently, it is one of the primary goals of cloud service providers to have different customers and their hosted applications isolated as much as possible in terms of the performance they observe. To make different offerings, comparable with regards to their performance isolation capabilities, a representative metric is needed to quantify the level of performance isolation in cloud environments. Such a metric should allow to measure externally by running benchmarks from the outside treating the cloud as a black box. In this article, we propose three different types of novel metrics for quantifying the performance isolation of cloud-based systems.

We consider four new approaches to achieve performance isolation in Software-as-a-Service (SaaS) offerings and evaluate them based on the proposed metrics as part of a simulation-based case study. To demonstrate the effectiveness and practical applicability of the proposed metrics for quantifying the performance isolation in various scenarios, we present a second case study evaluating performance isolation of the hypervisor Xen.

1. Introduction

Resource sharing promises significant cost savings in cloud environments, thanks to the reduced per customer overheads and economies of scale [25,3]. The most significant obstacle for potential cloud users, besides data isolation and security aspects, is unreliable performance [15,3,5]. Therefore, providing performance guarantees is a major research issue in the

* Corresponding author.
E-mail addresses: Rouven.Krebs@sap.com (R. Krebs), Christof.Momm@sap.com (C. Momm), Kounev@kit.edu (S. Kounev).

0167-6423/$ – see front matter
http://dx.doi.org/10.1016/j.scico.2013.08.003

area of cloud computing [12]. Providing one cloud customer constant Quality of Service (QoS) independent from the load induced by others is referred to as performance isolation.

The National Institute of Standards and Technology [23] defines three service models for cloud computing. The Infrastructure-as-a-Service (IaaS) model leverages virtualization to share hardware resources among customers. The Platform-as-a-Service (PaaS) model hosts applications of different customers within one middleware instance. Software-as-a-Service (SaaS) is the last model which provides a ready to run, hosted application. Isolating cloud customers in terms of the performance they experience is an important concern in each of these scenarios.

The allocation of hardware resources is handled by the lower levels (e.g., infrastructure level) in the stack. Therefore, we see performance isolation as a bigger challenge in the upper levels (e.g., platform and software) as they have no direct resource control. Within a group of users sharing the same view onto the application are referred to as *tenant*. This view includes the data they access, the application configuration, Service-Level-Agreements (SLAs) and Quality-of-Service (QoS) aspects. Multi-tenant Applications (MTA) share one application instance between multiple tenants and provide every tenant a dedicated share of the instance, isolated from each other. In our opinion the isolation of data and configuration aspects in MTAs is fairly easy as these issues are handled by the application domain. Usually all tenants use an application in a similar way with regards to the configuration and navigation paths. Nevertheless, the amount of users and peak times might defer and the tight coupling of tenants results in strong interference of non-functional system properties. Consequently, dealing with inference, especially considering non-functional system properties is still an open research issue in the area of SaaS (e.g., Bezemer [4], Fehling [8] and Wang [28]) and a challenging task for developers and architects of such systems.

In contrast to MTAs a hypervisor runs several virtual machine (VM) on the same hardware. A VM is a computer which is not directly accessing the hardware by leveraging virtualization. Thus, several virtual machines can run in parallel and share the resources. This technology is used to provide several customers access to SaaS offering whereby several instances of the application serve the load. Furthermore, it is the enabling technology of IaaS. In the traditional datacenters without cloud context the technology is also widely accepted and applied. We also observe considerable research interest in the field of performance isolation for IaaS clouds and virtualization technologies. Huber et al. [14] pointed out that different virtual machines (VM) have significant influence on each other. Gupta et al. [11] were already aware of this issue and developed an advanced scheduler resource scheduler for a specific hypervisor to solve the problem. Nevertheless, performance isolation in existing implementations has still potential to improve, especially in I/O intensive scenarios.

Performance isolation is an important aspect for various stakeholders. When a developer or architect has to develop a mechanism to ensure performance isolation between customers they need to validate the effectiveness of their approach to ensure the quality of the product. Furthermore, to improve an existing mechanism they need an isolation metric to compare different variants of the solution. When a system owner has to decide for one particular deployment in a virtual environment not only traditional questions like the separation of components on various hosts are of importance. Also the configuration of the hypervisor with regards to resource allocation mechanism have to be considered. For a concrete decision several concerns might be important. Performance, efficiency, administrative costs and security are mostly the basis for the decision. With a metric quantifying isolation, one more parameter could be used for the decision making process.

To the best of our knowledge, no metrics and techniques for quantifying performance isolation have been proposed before. In this article, we present two different methodologies and several alternative metrics along with appropriate measurement techniques for quantifying the isolation capabilities of IT systems. Although our focus is on cloud environments and cloud enabling technologies the metrics are not limited to these. The metrics presented are applicable for performance benchmarks, and preferable in situations where various customers use similar functionality with various load. In addition to this, we introduce general approaches for performance isolation in SaaS environments at the architectural level using four concrete isolation mechanisms. Finally, we apply the proposed metrics and measurement techniques for quantifying isolation in two independent case studies to on the one hand demonstrate the practical applicability of the proposed metrics. On the other hand, the metrics allow us to evaluate the effectiveness of the proposed performance isolation methods for SaaS environments and the impact of different deployment options in a virtual infrastructure that also allows us to reason for IaaS environments.

The remainder of this article is structured as follows. In Section 2, we first define performance isolation. Based on this definition, Section 3 presents the proposed isolation metrics. Section 4 discusses different approaches to ensure performance isolation within a MTA. Section 5 presents the first experiment setup for the evaluation of the isolation approaches and the metrics. Section 6 presents the second case study using virtualization. Based on the results a discussion and final assessment of the metrics and their usability in various scenarios is given in Section 7. In Section 8, we briefly introduce further ideas for enhancements of the metrics and measurement approaches. Section 9 surveys related work, while Section 10 concludes the article.

2. Performance isolation in shared environments

This section provides a comprehensive definition of performance isolation and differentiation to related concepts in shared environments. The metrics presented in Section 3 are based on these definitions.