Load Testing Elasticity and Performance Isolation in Shared Execution Environments

Samuel Kounev
Chair of Software Engineering
University of Würzburg
97074 Würzburg, Germany
skounev@acm.org

ABSTRACT

The inability to provide performance guarantees is a major challenge for the widespread adoption of shared execution environments, based on paradigms such as virtualization and cloud computing. Performance is a major distinguishing factor between different service offerings. To make such offerings comparable, novel metrics and techniques are needed allowing to measure and quantify the performance of shared execution environments under load, e.g., public cloud or general virtualized service infrastructures. In this talk, we first discuss the inherent challenges of providing performance guarantees in the presence of highly variable workloads and load spikes. We then present novel metrics and techniques for shared execution environments, specifically considering the dynamics of modern service infrastructures.

Categories and Subject Descriptors

D.2.8 [**Software Engineering**]: Metrics—*performance measures*

1. INTRODUCTION

The proliferation of shared execution environments, based on paradigms such as virtualization and cloud computing, is becoming increasingly ubiquitous in today's IT landscape. Cloud computing is a novel paradigm for providing data center resources as on demand services in a pay-as-you-go manner. It promises significant cost savings by making it possible to consolidate workloads and share infrastructure resources among multiple applications resulting in higher cost- and energy-efficiency. Despite the hype around it, it is well established that this new computing model is already transforming a large part of the IT industry [5, 10].

However, the inability of today's cloud technologies to provide dependability guarantees is a major showstopper for the widespread adoption of the cloud paradigm, especially for mission-critical applications [5]. The term *dependability* is understood as a combination of service *availability* and *reliability*, commonly considered as the two major components of dependability [11], in the presence of variable workloads (e.g., load spikes), security attacks, and operational failures. Given that an overloaded system appears as unavailable to its users, and that failures typically occur during overload conditions, a prerequisite for providing dependable services is to ensure that the system has sufficient *capacity* to handle its dynamic workload [12]. According to [10], concerns of organizations about service availability is a major obstacle to the adoption of cloud computing.

Today's cloud platforms generally follow a trigger-based approach when it comes to enforcing application-level service-level agreements (SLAs), e.g., concerning availability or responsiveness. Triggers can be defined that fire in a reactive manner when an observed metric reaches a certain threshold (e.g., long service response times) and execute certain predefined reconfiguration actions until given stopping criteria are fulfilled (e.g., response times drop). Triggers are typically used to implement *elastic* resource provisioning mechanisms. The term *elasticity* is understood as the degree to which a system is able to adapt to workload changes by provisioning and deprovisioning resources in an autonomic manner, such that at each point in time the available resources match the current demand as closely as possible [3]. Better elasticity leads to higher availability and responsiveness, as well as to higher resource- and cost-efficiency.

However, application-level metrics, such as availability and responsiveness, normally exhibit a highly non-linear behavior on system load, and they typically depend on the behavior of multiple virtual machines (VMs) across several application tiers. Thus, for example, if a workload change is observed, the platform cannot know in advance *how much*, and at what level of granularity, additional resources in the various application tiers will be required (e.g., vCores, VMs, physical machines, network bandwidth), and *where and how* the newly started VMs should be deployed and configured to ensure dependability without sacrificing efficiency. Moreover, the platform cannot know *how fast* new resources should be allocated and *for how long* they should be reserved. Hence, it is hard to determine general thresholds of when triggers should be fired, given that the appropriate triggering points typically depend on the architecture of the hosted services and their workload profiles, which can change frequently during operation.

Furthermore, in case of contention at the physical resource layer, the availability and responsiveness of an individual application may be significantly influenced by applications running in other co-located virtual machines (VMs) sharing the physical infrastructure [4]. Thus, to be effective, triggers must also take into account the interactions between applica-

LT'15, February 1, 2015, Austin, Texas, USA.
Copyright © 2015 ACM 978-1-4503-3337-5/15/02 ...$15.00.
http://dx.doi.org/10.1145/2693182.2693186.

tions and workloads at the physical resource layer. The complexity of such interactions and the inability to predict how changes in application workload profiles propagate through the layers of the system architecture down to the physical resource layer render conventional trigger-based approaches unable to reliably enforce SLAs in an efficient and proactive fashion (i.e., allocating only as much resources as are actually needed and reconfiguring proactively before SLA violations have occurred).

As a result of the above challenges, today's shared execution environments based on 1st generation cloud technologies rely on "best-effort" mechanisms and do not provide dependability guarantees. Although no guarantees are given, the provided level of dependability is a major distinguishing factor between different service offerings. To make such offerings comparable, novel metrics and techniques are needed allowing to measure and quantify the dependability of shared execution environments, e.g., cloud computing platforms or general virtualized service infrastructures.

In this keynote talk, we first discuss the inherent challenges of providing service dependability in shared execution environments in the presence of highly variable workloads and load spikes. We then present novel metrics and techniques for measuring and quantifying platform elasticity and performance isolation specifically taking into account the dynamics of modern service infrastructures. We consider both environments where virtualization is used as a basis for enabling resource sharing, e.g., as in Infrastructure-as-a-Service (IaaS) offerings, as well as multi-tenant Software-as-a-Service (SaaS) applications, where the whole hardware and software stack (including the application layer) is shared among different customers (i.e., tenants). We focus on evaluating two dependability aspects: i) the ability of the system to provision resources in an elastic manner, i.e., *system elasticity* [3, 2, 13, 1], ii) the ability of the system to isolate different applications and customers sharing the physical infrastructure in terms of the performance they observe, i.e., *performance isolation* [8, 7, 9, 6]. We discuss the challenges in measuring and quantifying the mentioned two dependability properties presenting existing approaches to tackle them. Finally, we discuss open issues and emerging directions for future work in the area of dependability benchmarking.

2. BIOGRAPHY

Samuel Kounev is a Professor and Chair of Computer Science at the Department of Computer Science, University of Würzburg, Germany. His research focusses on developing methods, techniques and tools for the engineering of dependable and efficient software systems. Relevant research areas include: software design, modeling and architecture-based analysis; systems benchmarking, monitoring and experimental analysis; and autonomic and self-aware systems management. He received a PhD degree in computer science from Technische Universitaet Darmstadt (2005). From February 2006 to May 2008, he was a research fellow at Cambridge University. In April 2009, he received the Emmy-Noether Career award (1 Mil. EUR) for excellent young scientists by the German Research Foundation (DFG). He currently serves as elected Chair of the Research Group of the Standard Performance Evaluation Corporation (SPEC), which he co-founded in 2010, providing a platform for collaborative research efforts between academia and industry in the area of quantitative system evaluation. He also serves as Co-Chair of the Steering Committee of the ACM/SPEC International Conference on Performance Engineering (ICPE), which he co-founded in 2010 as a first joint event between ACM and SPEC. He is a member of the ACM, IEEE, and the German Computer Science Society, and recipient of several honors including the SPEC 2014 Presidential Award for "Excellence in Research" recognizing lasting contributions to the field of performance evaluation and benchmarking of computing systems.

3. REFERENCES

[1] N. R. Herbst, N. Huber, S. Kounev, and E. Amrehn. Self-Adaptive Workload Classification and Forecasting for Proactive Resource Provisioning. 2013.

[2] N. R. Herbst, N. Huber, S. Kounev, and E. Amrehn. Self-Adaptive Workload Classification and Forecasting for Proactive Resource Provisioning. *Concurrency and Computation - Practice and Experience, John Wiley and Sons, Ltd.*, 26(12):2053–2078, 2014.

[3] N. R. Herbst, S. Kounev, and R. Reussner. Elasticity in Cloud Computing: What it is, and What it is Not. In *10th International Conference on Autonomic Computing (ICAC 2013)*. USENIX, June 2013.

[4] N. Huber, M. von Quast, M. Hauck, and S. Kounev. Evaluating and Modeling Virtualization Performance Overhead for Cloud Environments. In *International Conference on Cloud Computing and Services Science (CLOSER 2011)*, May 2011.

[5] B. Jennings and R. Stadler. Resource Management in Clouds: Survey and Research Challenges. *Journal of Network and Systems Management*, March 2014.

[6] R. Krebs, C. Momm, and S. Kounev. Architectural Concerns in Multi-Tenant SaaS Applications. In *2nd International Conference on Cloud Computing and Services Science (CLOSER 2012)*, April 2012.

[7] R. Krebs, C. Momm, and S. Kounev. Metrics and Techniques for Quantifying Performance Isolation in Cloud Environments. In *8th ACM SIGSOFT International Conference on the Quality of Software Architectures (QoSA 2012)*, June 2012.

[8] R. Krebs, S. Spinner, N. Ahmed, and S. Kounev. Resource Usage Control In Multi-Tenant Applications. In *14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, 2014.

[9] R. Krebs, A. Wert, and S. Kounev. Multi-Tenancy Performance Benchmark for Web Application Platforms. In *13th International Conference on Web Engineering (ICWE 2013)*. Springer-Verlag, July 2013.

[10] M. Armbrust et al. A View of Cloud Computing. *Communications of the ACM*, 53(4):50–58, 2010.

[11] J. Muppala, R. Fricks, and K. S. Trivedi. *Computational Probability*, volume 24, chapter Techniques for System Dependability Evaluation. Kluwer Academic Publishers, 2000.

[12] R. Nou, S. Kounev, F. Julia, and J. Torres. Autonomic QoS Control in Enterprise Grid Environments using Online Simulation. *Journal of Systems and Software*, 82(3):486–502, Mar. 2009.

[13] J. G. von Kistowski, N. R. Herbst, and S. Kounev. LIMBO: A Tool For Modeling Variable Load Intensities. In *5th Intl. Conf. on Performance Engineering (ICPE 2014)*. ACM, March 2014.