Quantitative Evaluation of Service Dependability in Shared Execution Environments

Samuel Kounev

Chair of Software Engineering
Department of Computer Science
University of Würzburg
Am Hubland, 97074 Würzburg, Germany
skounev@acm.org

Recent reports indicate that ICT is currently responsible for 8-10% of EU's electricity consumption and up to 4% of its carbon emissions [2, 23]. By 2020, only in Western Europe, data centers will consume around 100 billion kilowatt hours each year [15] (the same as the total electricity consumption of the Netherlands), making energy a major factor in IT costs. However, according to [3], due to the growing number of underutilized servers, only 6 - 12% of the energy consumption in data centers nowadays is spent for performing computations.

Industry's answer to this challenge is cloud computing, promising both reductions in IT costs and improvements in energy efficiency. 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 [9]. Despite the hype around it, it is well established that if this new computing model ends up being widely adopted, it will transform a large part of the IT industry [8, 17].

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 [8, 9, 16, 1]. The term *dependability* is understood as a combination of service *availability* and *reliability*, commonly considered as the two major components of dependability [21], 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 [22]. According to [17, 16], concerns of organizations about service availability is a major obstacle to the adoption of cloud computing.

Today's cloud computing 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., high server utilization or 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 [6, 26]. 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 [7]. Thus, to be effective, triggers must also take into account the interactions between applications 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 described challenges, today's shared execution environments based on first generation cloud technologies rely on "best-effort" mechanisms and do not provide dependability guarantees. Nevertheless, 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, load spikes, and security attacks. We then present novel metrics and techniques for measuring and quantifying service dependability 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 three dependability aspects: i) the ability of the system to provision resources in an elastic manner, i.e., *system elasticity* [6, 5, 26, 25, 24, 4], 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* [12, 13, 11, 14, 10], and iii) the ability of the system to deal with attacks exploiting novel attack surfaces such as virtual machine monitors, i.e., *intrusion detection and prevention* [18–20]. We discuss the challenges in measuring and quantifying the mentioned three 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.

References

1. D. Durkee. Why Cloud Computing Will Never Be Free. *ACM Queue*, 8(4):20:20–20:29, April 2010.
2. European Commission - IP/13/231 - 18/03/2013. Digital Agenda: global tech sector measures its carbon footprint. http://europa.eu/rapid/press-release_IP-13-231_en.htm, March 2013.
3. J. Glanz. Power, Pollution and the Internet. New York Times, September 2012.
4. N. R. Herbst, N. Huber, S. Kounev, and E. Amrehn. Self-Adaptive Workload Classification and Forecasting for Proactive Resource Provisioning. In *4th ACM/SPEC International Conference on Performance Engineering (ICPE 2013)*, pages 187–198. ACM, April 2013.
5. 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, Special Issue with extended versions of the best papers from ICPE 2013*, 2014. John Wiley and Sons, Ltd.
6. 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.
7. N. Huber, M. von Quast, M. Hauck, and S. Kounev. Evaluating and Modeling Virtualization Performance Overhead for Cloud Environments. In *1st International Conference on Cloud Computing and Services Science (CLOSER 2011)*, pages 563 – 573. SciTePress, May 2011.
8. B. Jennings and R. Stadler. Resource Management in Clouds: Survey and Research Challenges. *Journal of Network and Systems Management*, March 2014. Springer.
9. S. Kounev, P. Reinecke, F. Brosig, J. T. Bradley, K. Joshi, V. Babka, A. Stefanek, and S. Gilmore. Providing dependability and resilience in the cloud: Challenges and opportunities. In K. Wolter, A. Avritzer, M. Vieira, and A. van Moorsel, editors, *Resilience Assessment and Evaluation of Computing Systems*, XVIII. Springer-Verlag, Berlin, Heidelberg, 2012. ISBN: 978-3-642-29031-2.
10. 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)*. SciTePress, April 2012.

11. 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)*. ACM Press, June 2012.
12. R. Krebs, C. Momm, and S. Kounev. Metrics and Techniques for Quantifying Performance Isolation in Cloud Environments. *Elsevier Science of Computer Programming Journal (SciCo)*, 2013. In print.
13. 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)*. IEEE/ACM, May 2014.
14. 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.
15. N. Kroes. Greening the digital world: major companies to measure the environmental footprint of ICT. http://ec.europa.eu/commission_2010-2014/kroes/en/blog/ict-footprint, 2012.
16. S. Lohr. Amazon's Trouble Raises Cloud Computing Doubts. The New York Times, April 22, 2011.
17. M. Armbrust et al. A View of Cloud Computing. *Communications of the ACM*, 53(4):50–58, 2010.
18. A. Milenkoski and S. Kounev. Towards Benchmarking Intrusion Detection Systems for Virtualized Cloud Environments. In *7th International Conference for Internet Technology and Secured Transactions (ICITST 2012)*. IEEE, December 2012.
19. A. Milenkoski, S. Kounev, A. Avritzer, N. Antunes, and M. Vieira. On Benchmarking Intrusion Detection Systems in Virtualized Environments. Technical Report SPEC-RG-2013-002 v.1.0, SPEC Research Group - IDS Benchmarking Working Group, Standard Performance Evaluation Corporation (SPEC), Gainesville, VA, USA, June 2013.
20. A. Milenkoski, B. D. Payne, N. Antunes, M. Vieira, and S. Kounev. HInjector: Injecting Hypercall Attacks for Evaluating VMI-based Intrusion Detection Systems (poster paper). In *The 2013 Annual Computer Security Applications Conference (ACSAC 2013)*. Applied Computer Security Associates (ACSA), December 2013.
21. J. Muppala, R. Fricks, and K. S. Trivedi. *Computational Probability*, volume 24, chapter Techniques for System Dependability Evaluation. Kluwer Academic Publishers, 2000.
22. 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.
23. K. Seneviratne. ICTs Increase Carbon Footprint Which They Can Reduce. IDN (InDepthNews), December 2013.
24. J. G. von Kistowski, N. R. Herbst, and S. Kounev. LIMBO: A Tool For Modeling Variable Load Intensities. In *5th ACM/SPEC International Conference on Performance Engineering (ICPE 2014)*, ICPE '14, pages 225–226. ACM, March 2014.
25. J. G. von Kistowski, N. R. Herbst, and S. Kounev. Modeling Variations in Load Intensity over Time. In *3rd International Workshop on Large-Scale Testing (LT 2014)*. ACM, March 2014.
26. A. Weber, N. R. Herbst, H. Groenda, and S. Kounev. Towards a Resource Elasticity Benchmark for Cloud Environments. In *2nd International Workshop on Hot Topics in Cloud Service Scalability (HotTopiCS 2014)*. ACM, March 2014.