

Performance Benchmarking of Infrastructure-as-a-Service (IaaS) Clouds with Cloud WorkBench (Tutorial)

Joel Scheuner
Software Engineering Division
Chalmers | University of Gothenburg
Gothenburg, Sweden
scheuner@chalmers.se

Philipp Leitner
Software Engineering Division
Chalmers | University of Gothenburg
Gothenburg, Sweden
philipp.leitner@chalmers.se

Abstract—The continuing growth of the cloud computing market has led to an unprecedented diversity of cloud services with different performance characteristics. To support service selection, researchers and practitioners conduct cloud performance benchmarking by measuring and objectively comparing the performance of different providers and configurations (e.g., instance types in different data center regions). In this tutorial, we demonstrate how to write performance tests for IaaS clouds using the Web-based benchmarking tool Cloud WorkBench (CWB). We will motivate and introduce benchmarking of IaaS cloud in general, demonstrate the execution of a simple benchmark in a public cloud environment, summarize the CWB tool architecture, and interactively develop and deploy a more advanced benchmark together with the participants.

Index Terms—Cloud Computing, Performance, Benchmarking

TUTORIAL OUTLINE

- 1) *IaaS cloud benchmarking (~20min)*: We present a general introduction into cloud performance benchmarking, with background, motivation, and selected results.
- 2) *My first benchmark with CWB¹ (~30min)*: We demonstrate how to modify an existing simple benchmark and execute it in a public IaaS cloud using the CWB web interface. Tutorial participants are invited and supported to follow these steps in their web browser.
- 3) *CWB Architecture Overview (~20min)*: We present a short overview of the basic architecture of CWB from a tool user perspective.
- 4) *Building an advanced benchmark with CWB (~90min)*: We demonstrate how to design and implement a more advanced benchmark from scratch, using Chef cookbooks to install dependencies and configure hooks to control the lifecycle of a benchmark. Interested tutorial participants will be given credentials and support to design, build, and execute such a benchmark from scratch using a template.

¹<https://github.com/sealuzh/cloud-workbench>

I. INTRODUCTION

Cloud computing has become the standard way of deploying computing infrastructure in many domains and has largely replaced privately owning computational resources such as server racks. In the Infrastructure-as-a-Service (IaaS) [1] service model, computing resources, such as CPU processing time, disk space, or networking capabilities, can be acquired and released as self-service via an application programming interface (API), prevalently in the form of virtual machines (VMs). VMs are typically available in different configurations or sizes also known as instance types, machine types, or flavors. This diversity ranges from tiny-sized VMs with a fractional CPU core and 0.6 GB RAM (e.g., *f1-micro*) to super-sized VMs with 128 CPU cores and 3904 GB RAM (e.g., *x1e.32xlarge*).

Selecting an appropriate VM configuration for an application is a non-trivial challenge given the large service diversity, which is exemplified in the 158 different VM types released by Amazon Elastic Compute (EC2) between 2006 and 2019. While functional properties can be compared by studying provider information or using tools such as Clouddorado², non-functional properties, such as performance, need to be quantified tediously. Cloud benchmarking is the field of research dedicated to objectively measuring and comparing the differences in performance between the various cloud services. A large body of literature [2]–[5] reports performance measurements for different workloads at the very resource-specific (e.g., CPU integer operations) and artificial micro-level or at the domain-specific (e.g., Web serving) and real-world application-level.

II. BASIC IAAS BENCHMARKING APPROACH

Figure 1 shows a very simplified view of an IaaS performance experiment. A benchmark manager acquires the instances that are to be benchmarked via the provider API, and provisions (i.e., configures) them. Once the setup is finished, it starts the execution of a benchmark within the instance, which returns the metrics (i.e., results) of its execution, and finally

²https://www.clouddorado.com/cloud_providers_comparison.jsp

destroys the instances once the benchmark is completed. Cloud environments typically deliver fairly unpredictable performance and therefore this execution lifecycle is repeated until a sample size with the desired statistical confidence is achieved. We refer to literature for the more detailed view [6] or a more generic architecture for IaaS cloud benchmarking [7].

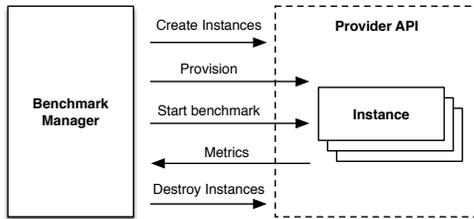


Figure 1. Basic IaaS Benchmarking Approach

III. CWB BENCHMARKING APPROACH

The following five steps explain how an experimenter can use the open source tool CWB to measure IaaS cloud performance in real cloud environments:

- 1) The experimenter writes a benchmark configuration as an Infrastructure-as-Code (IaC) [8] script using tools such as Chef³. The main idea is to express the entire configuration process in code and automate it in a repeatable and portable way for scaling out to many VMs and multiple providers. This configuration process includes installing dependencies (e.g., benchmark tools such as sysbench⁴), setting up configuration files or simulation data, and defining execution hooks for starting the measurements, submitting metrics, and notifying the termination of the execution. We published a collection of parametrizable benchmarks⁵, including a generic *cli-benchmark*, such that this step can be skipped for simple benchmarks.
- 2) The benchmark definition declares IaaS resources, and parametrizes the benchmark configuration from step 1. The resource declaration section specifies the type of IaaS resources (e.g., instance type, data center region, disk type, base image) and the authentication settings (e.g., API credentials, SSH login keys) for a specific provider. The benchmark parametrization section selects one or multiple benchmark configurations and optionally adjusts the default configuration via key-value pairs (e.g., duration of testing, number of repetitions). CWB uses Vagrant⁶ and its domain specific language (DSL) for Vagrantfiles to implement resource declaration and benchmark parametrization. This makes it easy to extend CWB with further cloud providers by leveraging the ecosystem of 30+ existing Vagrant provider plugins. In addition, CWB provides resource declaration defaults and abstracts the authentication setup using a centralized

configuration such that it is easy to setup an array of benchmark variations across multiple providers with minimal configuration effort.

- 3) The execution of a benchmark definition can be manually triggered via the CWB web interface or automated by attaching a periodic schedule written as crontab⁷ expression. CWB automates the entire benchmarking lifecycle such that no human intervention is required.
- 4) The metrics of a completed experiment can be downloaded as a comma-separated values (CSV) file.
- 5) The analysis and archiving of results is up to the experimenter and not supported by CWB.

IV. CONCLUSION

This tutorial addresses tools for performance benchmarking IaaS clouds. We motivate the importance of cloud benchmarking, demonstrate the execution of a simple IaaS benchmark in a public cloud provider using the CWB web interface, and develop a more advanced benchmark with configuration management integration.

ACKNOWLEDGMENT

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. We would like to thank the Ericsson Research Data Center (ERDC) for providing hardware resources for this tutorial.

REFERENCES

- [1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, and M. Zaharia, "Above the clouds: A Berkeley view of cloud computing," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-28, 2009.
- [2] P. Leitner and J. Cito, "Patterns in the chaos — a study of performance variation and predictability in public IaaS clouds," *ACM Trans. Internet Technol.*, vol. 16, no. 3, pp. 15:1–15:23, Apr. 2016.
- [3] A. Iosup, S. Ostermann, N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema, "Performance analysis of cloud computing services for many-tasks scientific computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 6, pp. 931–945, June 2011.
- [4] S. Ostermann, A. Iosup, N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema, "A performance analysis of EC2 cloud computing services for scientific computing," in *Cloud Computing*. Springer, 2009, vol. 34, pp. 115–131.
- [5] J. Schad, J. Dittrich, and J.-A. Quiané-Ruiz, "Runtime measurements in the cloud: Observing, analyzing, and reducing variance," *Proceedings of the VLDB Endowment*, vol. 3, no. 1, pp. 460–471, Sep. 2010.
- [6] C. Davatz, C. Inzinger, J. Scheuner, and P. Leitner, "An approach and case study of cloud instance type selection for multi-tier web applications," in *17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, 2017, pp. 534–543.
- [7] A. Iosup, R. Prodan, and D. Epema, "IaaS cloud benchmarking: Approaches, challenges, and experience," in *Cloud Computing for Data-Intensive Applications*. Springer, 2014, pp. 83–104.
- [8] M. Hüttermann, *Infrastructure as Code*, ser. DevOps for Developers. Apress, 2012.

³<https://www.chef.io/chef>

⁴<https://github.com/akopytov/sysbench>

⁵<https://github.com/sealuzh/cwb-benchmarks>

⁶<https://www.vagrantup.com>

⁷<https://linux.die.net/man/5/crontab>