

Preface

The advent of cloud computing has disruptively changed the way modern application systems are developed and delivered. The cloud offers affordable access to infrastructure resources of large-scale data centers, allowing consumers of cloud services to deploy production-quality application systems or to quickly prototype and experiment with systems under development. While this has lowered overall costs for development and operations and enabled new use cases, it has also shifted control over key parts of an application system to the cloud provider: Cloud consumers typically have to treat the cloud services that they are using as a black box. Furthermore, the quality of these cloud services often is unpredictable, changes over time, and can vary significantly between different cloud providers. As a consequence, the same holds for applications running on top of such services so that getting insights into the quality of underlying cloud services becomes a core business interest.

Benchmarking is the process of measuring and assessing the quality of some object, whether for purposes of experimentation or production. Traditionally, benchmarking has focused on objects that are under control, for example hardware or relational database systems, and on performance characteristics. Benchmarking the cloud, however, and using the cloud as a runtime environment for benchmarking are two new, relevant fields in benchmarking research and practice:

When benchmarking the cloud, we are interested in client-observable characteristics of cloud software services whose internals and implementation are hidden (“blackbox benchmarking”). The qualities of interest are not limited to performance but may also include fault-tolerance, data consistency, security, or business criteria such as operational costs. When benchmarking in the cloud, i.e., using the cloud as a runtime environment for benchmarking experiments, the cloud represents a cost-efficient experimentation platform that allows us to quickly provision and to deploy entire software application stacks. As such, benchmarking is no longer limited to low-level system aspects but can be applied to various layers of an application stack with opportunities to also run benchmarks across different geographic regions worldwide.

Overall, cloud service benchmarking can provide important, sometimes surprising insights into the quality of services and leads to a more quality-driven design

and engineering of complex software architectures that use such cloud services. Despite heightened interest, however, there is no comprehensive work yet that serves as an introduction to the field of cloud service benchmarking – we aim to close this gap with this book.

This book aims to cover all aspects of cloud service benchmarking, i.e., of both benchmarking the cloud and benchmarking in the cloud, at a very basic level. Specifically, this book aims to answer the following questions:

1. What is cloud service benchmarking and why should I care about it?
2. What are critical objectives and components I need to consider while designing a benchmark?
3. With which challenges do I have to deal when implementing and running an actual benchmark?
4. What and how can I learn from my measurement results?

Starting with a broad introduction to the field, this book aims to walk the reader step-by-step through the process of designing, implementing and executing a benchmark as well as understanding and dealing with results.

Organization of this Book

This book is divided into five parts where Part I introduces fundamentals, Parts II, III, and IV describe the different phases of a benchmark process (benchmark design, benchmark execution, benchmark analysis), and Part V concludes the book.

Part I focuses on fundamentals. It introduces what cloud benchmarking is (and what it is not, differentiating the field from traditional benchmarking practice and monitoring), gives an overview of cloud services and their key properties, and describes the notion of a cloud system and cloud service quality and how different qualities are interconnected. The part also addresses the aspect of a benchmarking lifecycle, i.e., different phases of a benchmarking process, as well as the motivations behind running benchmarks in particular phases of an application lifecycle.

Part II focuses on benchmark design (“pre-benchmark”). It discusses key objectives of benchmark design (e.g., repeatability, fairness, or understandability), distinguishes metrics (as the means to expressing measurement results) and measurement methods, and gives advice on developing own measurement methods and metrics. As benchmarks tend to be stress tests, there is a separate chapter on workloads, i.e., basic principles behind workloads and their generation strategies.

Part III focuses on the actual benchmark run (“benchmark execution”). It starts by discussing implementation challenges and objectives as well as the experiment setup before continuing to aspects like result collection, runtime monitoring, etc.

Part IV focuses on dealing with benchmark results (“post-benchmark”). It covers aspects such as an abstract process for turning data into insights, data preprocessing,

basic data analysis methods, as well as mechanisms for leveraging the acquired knowledge for different purposes depending on the motivation for the benchmark.

Part V concludes the book. It comes with a summary and outlook and points out directions for further reading in literature but also offers pointers to tools available on the web for hands-on cloud service benchmarking.

How to Read this Book

Overall, we recommend to read this book sequentially. However, chapter 14 has a special role: It does not introduce new information on the cloud service benchmarking process. Rather it aims to provide an incompressive overview of benchmarking toolkits that are publicly available. Readers that do not have a current benchmarking need, may simply skip that chapter and return to it at a later time.

Where it makes sense, we include examples and report on experiences that we or our closest colleagues made within the respective chapters. These examples are printed in gray boxes such as this one.

Where it makes sense, we put concrete advice on doing or specifically not doing something as part of a benchmarking process in boxes such as this one. Both example and advice boxes will start in Part II.

Tools, organizations, or web resources mentioned in the text are either cited directly in the text, e.g., as [1], and can be found in the references section on page 159. Alternately, they can be found in the directory of web links on page 164.

Intended Audience and Prerequisites

This book is intended for researchers and graduate students of computer science and related subjects, but also for industry practitioners, who are interested in evaluating the quality of cloud services or who want to assess key qualities of their own implementations through cloud-based experiments.

As such, the book requires knowledge of software systems at an undergraduate level. Specifically, readers should have a basic familiarity with distributed systems, data management and software engineering.

Acknowledgements

We would like to thank all current and former colleagues, co-authors of our papers, as well as fellow researchers and industry practitioners who provided inspiration for this book. Finally, we would also like to thank our families for all the patience with us and the support they provided us with while we were writing this book.

Berlin and New York City,
January 2017

David Bermbach
Erik Wittern
Stefan Tai

Cloud Service Benchmarking

Measuring Quality of Cloud Services from a Client
Perspective

Bermbach, D.; Wittern, E.; Tai, S.

2017, XIV, 167 p. 25 illus., 11 illus. in color., Hardcover

ISBN: 978-3-319-55482-2