

Cloud Service Benchmarking

AWS DynamoDB vs. Google Spanner

Denis Rangelov and Martin Dichev
Group H

I. INTRODUCTION

The purpose of this report is to summarize and evaluate the measurements collected during the performance evaluation process of two Cloud-based services - DynamoDB [3] and Google Spanner [4]. It reveals the benchmarking setup used throughout the project as well as the reasoning behind the selected benchmarking approach. Apart from the current passage, the report comprises of 5 more sections. The first one is *"Systems Under Test"* and illustrates the features that AWS DynamoDB and Google Spanner offer. In the *"Benchmark Design"* section, the main focus is placed on the design objectives, the selected metrics, the workload and the benchmarking setup. The sections that follow present the results collected when measuring the performance of the benchmarking targets (i.e. AWS DynamoDB and Google Spanner) as well as our interpretation of these results. The report concludes with a brief overview of the encountered challenges during the project and also includes an assessment of the degree to which we managed to accomplish the envisioned project goals.

II. SYSTEMS UNDER TEST

"If you know the enemy and know yourself, you need not fear the result of a hundred battles"
- Sun Tzu, *The Art of War*

Although the quote above is primarily intended as a playful and humorous way to open up the current section, there is a deeper meaning behind it. For the purposes of this writing, the idea of knowing your enemy can be translated as knowing well the systems one is trying to test. More specifically, when benchmarking a service is the main goal of a project, knowing the systems as close as possible is of crucial importance. Therefore, the current section is dedicated to summarizing some of the most well-known properties of DynamoDB and Google Spanner. Important to note here is that many of the properties of cloud services such as DynamoDB or Google Spanner remain "secret" for the general public. Therefore, despite our best efforts, it is possible that we might not be able to cover in much detail and with a complete accuracy how these systems operate behind the scenes. The information presented below was gathered by examining the reports from companies that actively use the services in question and also by reviewing the conference talks and papers the two service providers have published online.

III. BENCHMARK DESIGN

The main objective of this section is to briefly describe how we address the general benchmark design objectives, what performance metrics we decided to focus on and also what type of workloads we choose to utilize. Some of the answers to these questions are revealed implicitly by the benchmarking tool used in the scope of this project and namely - Yahoo! Cloud Serving Benchmark (YCSB) [1].

One challenging aspect of using a pre-defined benchmarking tool is that it allows us to only evaluate if the tool fulfills the benchmarking design objectives, but if it doesn't, we cannot change its internals to fix the issue we are facing. Therefore, in the subsections that follow we will address some of the limitations when using YCSB (or any out-of-the-box benchmarking tool for that matter) compared to a custom-made benchmarking solution (e.g. developing a dedicated, self-made tool that fits the desired benchmarking scenario).

Important to clarify here is that the discussion presented below follows the guidelines for benchmark- requirements, metrics and workload introduced in the Enterprise Computing lecture and the ones examined in the Cloud Service Benchmarking book [2] written by Professor Tai, David Bernbach and Erik Wittern.

A. Benchmark Design Objective Assessment

According to S. Tai, D. Bermach and E. Wittern, there are five general benchmark design objectives - relevance, reproducibility, fairness, portability and understandability. Each one is individually analyzed below:

Relevance:

B. Maintaining the Integrity of the Specifications

C. Abbreviations and Acronyms

D. Units

- Use
- Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to

E. Equations

$$\alpha + \beta = \chi \quad (1)$$

F. Some Common Mistakes

IV. USING THE TEMPLATE

A. Headings, etc

B. Figures and Tables

V. CONCLUSIONS

APPENDIX

ACKNOWLEDGMENT

REFERENCES

- [1] Brian F. Cooper et al. “Benchmarking Cloud Serving Systems with YCSB”. In: *Proceedings of the 1st ACM Symposium on Cloud Computing*. SoCC ’10. Indianapolis, Indiana, USA: ACM, 2010, pp. 143–154. ISBN: 978-1-4503-0036-0. DOI: 10 . 1145 / 1807128 . 1807152. URL: <http://doi.acm.org/10.1145/1807128.1807152>.
- [2] Erik Wittern Stefan Tai David Bermbach. *Cloud Service Benchmarking. Measuring Quality of Cloud Services from a Client Perspective*. Springer, Cham, 2017. DOI: 10.1007/978-3-319-55483-9.
- [3] *Amazon DynamoDB. Fast and flexible NoSQL database service for any scale*. Accessed: 11-Jan-2019. URL: <https://aws.amazon.com/de/dynamodb/>.
- [4] *CLOUD SPANNER. The first horizontally scalable, strongly consistent, relational database service*. Accessed: 10-Jan-2019. URL: <https://cloud.google.com/spanner/>.