NoSQL SaaS Comparison[*]

Extended Abstract[†]

J. McFadden[‡]
Institute of Technology
Univ. of Washington: Tacoma
Tacoma, WA 98402
mcfaddja@uw.edu

Y. Tamta[§]
Institute of Technology
Univ. of Washington: Tacoma
Tacoma, WA 98402
yashaswitamta@gmail.com

J. N. Gandhi[¶]
Institute of Technology
Univ. of Washington: Tacoma
Tacoma, WA 98402
jugalg.uw.edu

ABSTRACT

Two different No-SQL database cloud services, from two different providers are compared. These comparisons will be based, broadly, on the performance, features, usability, and cost of each service. The performance metric includes measurements of both data throughput and query latency under a variety of load conditions; while the usability metric will focus on the number of available means for interacting with the database, as well as the ease of utilizing those means. Along with comparing standard NoSQL database features, our features metric will also consider tools for interacting with the each database which available from the respective provider.

CCS CONCEPTS

•**Computer systems organization** → **Cloud based systems;** *Software as a Service;* •**Database systems** → NoSQL; •**Cloud Systems** → Perfomance; Cost;

KEYWORDS

ACM proceedings, LaTeX, text tagging

ACM Reference format:

J. McFadden, Y. Tamta, and J. N. Gandhi. 1997. NoSQL SaaS Comparison. In *Proceedings of TCSS 562: Cloud Computing Term Project, Tacoma, WA USA, April 2017 (TCSS 562: Spring 2017),* 1 pages.

DOI: 10.475/123_4

1 INTRODUCTION

Initially, we were going to do a software/systems comparison, with the software/systems chosen for comparison in this project being two different NoSQL database system. These systems would have been deployed/run/operated in several different ways, including *SaaS*[1] implementations, *containerized* implementations, and *native installations*. The goal of the project would have been to understand the performance characteristics of each deployment method ***and*** to quantify the costs of each deployment method, with the costs being calculated based on the hourly cost to operate, the initial time & costs required for setup, and the maintenance requirement of a deployment. Additionally, performance of the systems and deployments would have been measured using the time required to carry out various database operations, under a set of several different conditions, as well as the CPU, memory, and network loads imposed by the various deployments under the same set of conditions. As we proceeded with the project, however, we discovered that there was no way to properly cover all of these topics within a single project

1.1 Changed Questions

Ultimately, we decided that our initial approach was too complex and beyond the intended scope of the project; therefore, were limited ourselves to comparing DynamoDB from AWS and BigTable from Google's Cloud Services. We also changed our comparison of the services to be based on metrics measuring their performance, features, usability, and cost; while also setting a specific and narrow scope for each metric. The performance metric is limited to separately measuring both data throughput and query latency under a variety of load a replication conditions; while the features metric will concentrate on features available from each provider, such as provisioning and tools for interacting with their respective service. Our usability metric will focus on both the ease of implementing software to use each service, as well as the availability in terms of the number of supported languages, the number of available APIs & SDKs, and the availability of third-party software for interacting with each service. Finally, our cost metric will focus on the costs associated with the setup and operation of each service, however, it will also attempt to cover costs which might be encountered when implementing software making use of each service.

1.2 Service Change

Unfortunately, BigTable is difficult to deploy for even a simple test case, as it requires the configuration of multiple cloud services on Google's cloud and the installation of multiple Linux-only software packages, namely rBase & Hadoop. Thus, we were again forced to alter our plan, this time by swapping BigTable from Google for CosmosDB from Azure. This swap also lead to an additional feature to consider and condition to test under, the replication two or more of each cloud service's availability zones.

[*] Produces the permission block, and copyright information

[†] The full version of the author's guide is available as `acmart.pdf` document

[‡]

[§]

[¶]

[1] **SaaS :** Software as a service.

TCSS 562: Spring 2017, Tacoma, WA USA

© 2016 Copyright held by the owner/author(s). 123-4567-24-567/08/06…$15.00

DOI: 10.475/123_4