

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 conditions; while the features metric will concentrate on features available from each provider, such as provisioning, replication, 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.

CCS CONCEPTS

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

KEYWORDS

ACM proceedings, L^AT_EX, 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)*, 3 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 this approach was too complex and had too broad a scope.

2 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. Unfortunately, BigTable is difficult to deploy for even a simple test case so we were again forced to alter our plan, this time by swapping BigTable for CosmosDB from Azure.

^{*}Produces the permission block, and copyright information

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

[‡]

[§]

[¶]

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

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

¹SaaS : Software as a service.