Comparison of NoSQL on the Cloud Software, Systems and Platforms

J. McFadden
Univ. of Washington: Tacoma
Tacoma, WA
mcfaddja@uw.edu

Y. Tamta
Univ. of Washington: Tacoma
Tacoma, WA
yashaswitamta@gmail.com

J. N. Gandhi
Univ. of Washington: Tacoma
Tacoma, WA
jugalg@uw.edu

April 11, 2017

*Project coordinator indicated by **

Abstract

Two different NoSQL database software packages will be implemented in and/or installed on several different types of systems. In turn, these systems will be deployed on several different platforms. This will allow different combinations of software, systems, and platforms to be compared based on both performance, difficulty of setup & maintenance, and both setup & operating costs.

1 Introduction

The software/systems chosen for comparison in this project are two different NoSQL database system. These systems will be deployed/run/operated in several different ways. These include *SaaS*[1] implementations, *containerized* implementations, and *native installations*. The goal of the project is to understand the performance characteristics of each deployment method ***and*** to quantify the costs of each deployment method. These costs will be 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 will be 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.

2 Systems and Platforms

We will be using two NoSQL database software packages. The first software package is **DynamoDB** from Amazon Web Services (*AWS*), while the second software package will be **Cassandra**, an open-source NoSQL database software package. These software packages will be deployed using several different systems and platforms, as described below.

2.1 Systems

This project will run the software packages on four different systems (*or types of systems*). We have chosen systems which range from hosted *SaaS* through various degrees of virtualization and then all the way to non-virtualized machines. These systems are as follows

A): **AWS *SaaS* system(s)**

B): **Virtualization using Docker containers**

C): **Virtualization on AWS's EC2 VMs**

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

D): **Dedicated, non-virtualized machines**

These four systems will be deployed using several different platforms which we will describe in the next part of this section.

2.2 Platforms

We have chosen three different platforms on which to deploy our systems. The chosen platforms span the range of cloud service paradigms from *SaaS* to *PaaS*[2] to *IaaS*[3]. We list the three platforms, along with two variations on one of the platforms, below

1): **AWS's DynamoDB Service (***SaaS***)**

2): **Containerized implementations (**using *Docker***) running on**

  i): AWS's Container Service (*PaaS*)

  ii): AWS EC2 Machines running the docker run-time in Linux (*hybrid: Pass/IaaS*)

3): **AWS EC2 Machines running native installations of the software in Linux (***IaaS***)**

4): **A dedicated, non-virtualized server (***server***)**

In the next section, we list which systems will run each software package, along with a explanation why each software-system pairing was chosen. Additionally, we will describe which platforms will be used to deploy each system and why those deployment choices were made.

3 Deployment

The deployment strategy will be described in two parts which can be breifly described as "*what systems will be used to run and/or implement each software package*" and "*which platforms will be used to deploy each of these systems*". In particular, this

[2] **PaaS :** Platform as a service.

[3] **IaaS :** Infrastructure as a service.