November 13, 2017
CS461 Fall 2017
Group 57

Technology Review and Implementation Plan (Rough Draft)

Prepared by

Parker Bruni

Abstract

This document contains an analysis of various technical components of the project. The technical components will be built from a specific framework and each framework will need to be chosen from a vast selection of potential frameworks. An analysis of three specific components of the project and their potential frameworks will be made. A framework will be chosen for each component based on the technological advantages that the framework will provide.

1 Introduction

Our scalable web application for monitoring energy usage on campus is an application that utilizes various technologies to accurately store, process, and display energy usage of buildings on the Oregon State campus. There are many different components and frameworks by which to build an application for this purpose and selection the right technologies for is proper functionality is crucial. Choosing technologies that may be outdated, have flaws specific to the needs of the application, contain compatibility issues, etc., could prove to be detrimental to the project as a whole. It is necessary to analyse the various technologies and select the technology that will provide the best performance and functionality prior to the creation of the application so that the development process will be fluid and efficient.

This document will discuss the pros and cons of various specific technologies that may be applicable to the components of our web application. Specifically, this document will compare technologies for the database framework of the application, the database host of the application, and the server framework that will be used. For the database framework, comparisons will be made between mongoDB, mySQL, and Apache Cassandra. For the database host, comparisons will be made between AWS EC2, Microsoft Azure VM, and Google CE. For the back end framework, comparisons will be made between Node.js, PHP, and Golang.

2 Technology 1: Database Framework

2.1 mongoDB

MongoDB is a noSQL scalable open source JavaScript based database system. It is a non-relational database and it's documents are stored in json format essentially making all JavaScript data types compatible with this database. It is often used in conjunction with Node.js projects because of this relationship with JavaScript. It is widely used because it is a "document" based database which essentially allows the data that it stores to be interpreted as objects of data, where you can have levels and sub-levels of data in one data entry. For example, you could have 1 entry in mongoDB that represents a store, and within that store you can have lower levels of data such as products within the store or employees that work in that store. It is widely used as the database for many web based applications because it is designed to have useful models for data structures such as logs, graphs, account and user profiles, form data, etc. It has been used by many major industries and companies such as MetLife, government websites, Expedia, and more. MongoDB is often referred to as new user friendly because it is widely documented and has many tutorials online. We choose mongoDB as our database framework because our application will be a JavaScript based stack application, it is scalable, it handles large amounts of unstructured data easily, and it is simple and relatively easy to learn.

2.2 mySQL

MySQL is perhaps one of the most popular open-source relational database management systems, written in c and c++. It is an extremely well documented and mature database system and is used widely by many applications, include web based applications. It is compatible with nearly all operating systems and supports many storage engines. Data using mySQL is stored in tables that may be accessed or manipulated using query requests. It is not a document oriented database system so data may not have various levels of hierarchy within each entry, but each entry may be logically bound to other parts of the database which would be managed by those who are designing the database. MySQL is proven reliable, high performance, and easy to use and is used by major company applications such as Facebook, Twitter, YouTube, and much more. While mySQL is widely used and provides efficient functionality, it is not a JavaScript based database system, which is a driving reason that it was not chosen for our JavaScript based stack application. Ultimately, our application is structured as a MEAN stack application because that method of building an application is proven to be efficient, compatible, functional, and quick to learn.

2.3 Apache Cassandra

Apache Cassandra is an open source linearly scalable storage system that is build to handle large amounts of data. It is build using two core technologies: Google Big Table and Amazon Dynamo. It is a noSQL class of database system developed to improve consistency, availability, and partition tolerance of the database. It is designed to handle arbitrary data types that may have interconnected relationships to other pieces of data within that database. Because of the way that Cassandra's architecture is designed, it is incapable of having a single point of failure among its nodes. Each node within the database is connected and running concurrently, there is no ”master-slave” relationship between nodes and nodes can be added to an existing cluster without taking down the server. This works well for applications that need to have access to a database at all times. Cassandra excels for real world uses such as messaging, fraud detection, recommendation engines, playlists or collections, and more. Although it would probably meet the needs of our scalable web application, it is less simple than either mongoDB or mySQL and is not directly associated with JavaScript frameworks. Another issue that presents itself is that it is not as widely used as either mongoDB or mySQL, and as a consequence is lacking the resources to learn about it as efficiently as either mongoDB or mySQL.

3 Technology 2: Database Host

3.1 AWS EC2

Amazon Web Services (AWS) provides a secure, scalable, and non-constricting web based service for applications which may use the non-relational database that we chose mongoDB. It allows complete control of the application (just as root access to computing capabilities) and scales elastically to the needs of the application. Along with control and scalability, the operating system environment in which the application is to be ran on may also be chosen. It is reliably available, secure, and has access to many other AWS computing applications that may be utilized by the project, offering the widest range of services to its users. It is designed to be simple to learn and cost effective. Because of AWS many features, it has been an excellent choice for a huge number of applications to be ran on, including Netflix, AirBnB, and Lamborghini. We have opted to use AWS because of its many great features, specifically cost of use, freedom of control, reliability, and ease to learn. While other web hosting database service may be good choices for our application as well, AWS is an extremely trustworthy and efficient service for our database.

3.2 Microsoft Azure Cosmos DB

Microsoft Azure Cosmos DB accomplishes a similar functionality as AWS EC2 but is not nearly as widely used as AWS. It has support for applications based in non-relational database services such as mongoDB. Essentially it offers all the functionality of a privately owned server but based in the ”cloud”, similar to AWS. It is open source and scalable, as well as provides security, good pay rates, and access to many of Microsoft Azures cloud computing applications. It has low latency and fast I/O performance, which may be scaled up for a larger price. Notable users that rely on this service include Pearson, Ford, and NBC News. While this service may be be more than qualified to run our database, we opted for the use of AWS EC2 because it is much more widely used and trusted than Azure VM as well as has a larger library of applications to use than Azure. Aside from that, Azure has been less reliable than AWS, having had a series of outages over the years.

3.3 Google CE

Google Compute Engine (CE) is Google cloud computing equivalent to Microsoft Azure and AWS. It is similar to other cloud computing services in that it supports all of the quality attributes of a good web hosting service, including scalability, security, control, area of distribution, reliability, support for non-relational databases such as mongoDB, etc., but it is not nearly as widely used or distributed as AWS. It may have an advantage over the other services with regards to machine-learning capability that an application may take advantage of, but in our case we do not require that unique resource. Google CE may be more appealing to niche markets, as it focuses on smaller scale companies. Essentially, Google CE is very similar to AWS but is much less widely adopted and does not offer as many resources to use as our choice of AWS.

4 Technology 3: Back-End Framework

4.1 Node.js

Node.js is an open source cross-platform runtime environment for developing applications using JavaScript. It can be run on various operating systems such as Linux, Windows, or OS X. It has a massive open source library of JavaScript libraries that may be easily installed and used by an application. It is extremely fast, scalable, asynchronous and event driven. Essentially, this framework can make multiple calls to a server without needing to be hung up on one specific call, it can reference a database dynamically. It is very popular and widely used by many large companies such as eBay, General Electric, PayPal, Yahoo!, and more. It performs exceptionally for applications that need to produce data functions quickly and dynamically so as not to refresh a browser to display data. This is our choice for our back-end framework because it is a JavaScript based library that will be compatible with our other technologies and frameworks as well as provides sufficient functionality.

4.2 PHP

PHP is a widely used and matured general purpose scripting language. It is perhaps more widely used than JavaScript as it is an older developed scripting language. PHP is perhaps simpler than node.js as it does not require a server, simply a php file that may be interpreted by a web server. PHP uses a multi-threaded blocking I/O to achieve parallel task completion whereas node.js uses a single threaded non-blocking I/O model. Many large companies also use PHP for their products, including Facebook, Wikipedia, Flickr, and many more big names. PHP is reliable and simple to use, but lacks modern functionality, organization, and resources that node.js has. As stated before, our application is JavaScript dependant so it is wise to keep as many technologies as possible within this language.

4.3 Golang

Golang, commonly referred to as ”Go”, is a more modern back-end framework developed by Google. It is not as widely used as node.js, but it is still and effective choice for many web based applications. Because it is in its early days of development use, it is still a relatively immature framework, but that is not necessarily a bad thing. It outperforms node.js with speed and performance of tasks because it is not based on an interpreted high language such as JavaScript.

It is used by companies such as Google, Docker, and Dropbox because of its excellence in scalability and concurrency. Large scale data can be effectively managed using Go but it may lack the broad scope of tools that node.js could provide. We have decided not to utilize Golang as our back-end framework because of its relative youth. It lacks the wide range of tools that we may utilize for our application, it lacks the simplicity of node.js, and it is not a JavaScript based environment such as node.js.

5 Conclusion

We have made a decision to base our application off of the JavaScript language, and as such we have chosen many of our technologies based on this so that they may be compatible and perform smoothly. Reducing the number of languages involved in a full stack application reduces the complexity of the project and allows development to happen more fluidly. We have opted for JavaScript based technologies because of it's wide use, large amount of applications and libraries to use, and overall high-level simplicity. Many of our technologies used, specifically our back-end framework and database framework, will utilize JavaScript based technologies, such as mongoDB and Node.js. Not only do these technologies have sufficient performance, scalability, and scope for our project, it allows the project to be easily compatible within itself. As a matter of our database host, all we need is a sufficient web service that can provide support for a non-relational database system such as mongoDB. The best choice to us is using Amazon Web Services DynamoDB, which supports non-relational database systems and allows us a vast array of tools as provided by Amazon Web Services.

6 Works Cited

[1] Y. Leven. June 15, 2017. *Top 5 alternatives to MongoDB*. panoply. [Online].
Available http://blog.panoply.io/top-5-alternatives-to-mongodb

[2] P. Wayner. Feb 9, 2017. *PHP vs. Node.js: An epic battle for developer mind share*. InfoWorld. [Online].
Available https://www.infoworld.com/article/3166109/application-development/php-vs-nodejs-an-epic-battle-for-developer-

[3] J. Nicholson. Nov 3, 2016 *Node.js vs Golang: Battle of the Next-Gen Languages*. HostingAdvice. [Online].
Available http://www.hostingadvice.com/blog/nodejs-vs-golang/

[4] S. Palmer. N/A. *Golang vs Node Comparison: Performance, Scalability and More*. devteam. [Online].
Available https://www.devteam.space/blog/golang-vs-node-comparison-performance-scalability-and-more/

[5] N/A. N/A. *Amazon EC2*. Amazon. [Online].
Available https://aws.amazon.com/ec2/

[6] Y. Shiotsu. N/A. *PHP vs. Node.js*. Upwork. [Online].
Available https://www.upwork.com/hiring/development/php-vs-node-js/

[7] E. Korotya. Mar 13, 2017. *MongoDB vs MySQL Comparison: Which Database is Better?*. hackernoon. [Online].
Available https://hackernoon.com/mongodb-vs-mysql-comparison-which-database-is-better-e714b699c38b

[8] N/A. N/A. *Node.js - Introduction*. tutorialspoint. [Online].
Available https://www.tutorialspoint.com/nodejs/nodejs_introduction.htm