An Open-Simplex Approach: Serverless Multi-Tenancy Big Data Analytics On AWS and Google Cloud Platform

Project Proposal

[CPSC-597]

Vinay Khedekar

CWID – 893284166

California State University, Fullerton

(Computer Science Department)

Date: 11/15/2018

Dr. Yun Tian

Faculty Advisor

Dr. Bin Cong

Faculty Reviewer

Table of Content

1. Introduction

In the world of emerging internet technologies, communication over the internet has increased rapidly and that offered a facility to share the data over the internet. Today, tons of petabytes of data is generated every day by social media applications, financial applications, web applications and, research and scientific data. The increased data has created major issues in data management, data analytics, and data processing. The generic solution to the problem is to analyze the petabytes of data, extract meaningful data from it and use the refined data for further operations.

Analyzing data itself comes with the number of challenges including infrastructure needed to perform the analytical operations, cost associated with the infrastructure and operations, storage, security, and maintenance are few of the major challenges in the software industry. The major challenge in the data analytics process is the infrastructure needed to process the huge amount of data which needs high processing units to gain the performance in terms of execution time, large storage system, storing and operating the data stored in the distributed geographical locations, efficient software system embedded with the efficient algorithms, and highly skilled data scientist. [5]

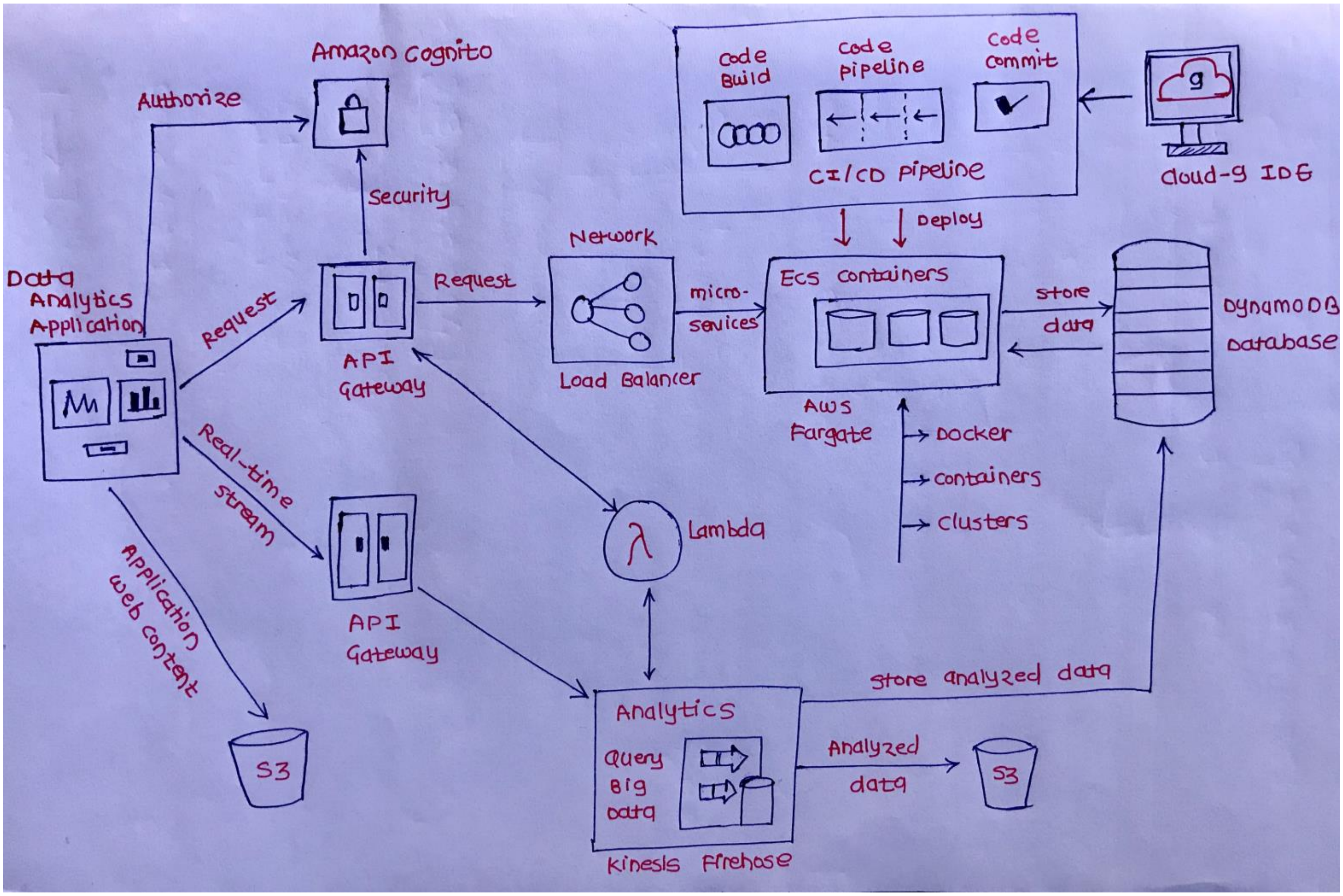
The solution to these problems is a Serverless cloud computing where eliminating the management of complex in-house infrastructure and focusing on the data analytical solutions only. The goal of my project is to conduct research and development on the Serverless cloud computing platforms like Amazon Cloud Platform (AWS) and Google Cloud Platform (GCP) to analyze huge amount of industry data and analyzing results in terms of multi-tenancy environment, time, performance, cost, efficiency by setting up an industry level architecture on the AWS and Google cloud platform.

The few business scenarios' I would be considered for the data analytics are-

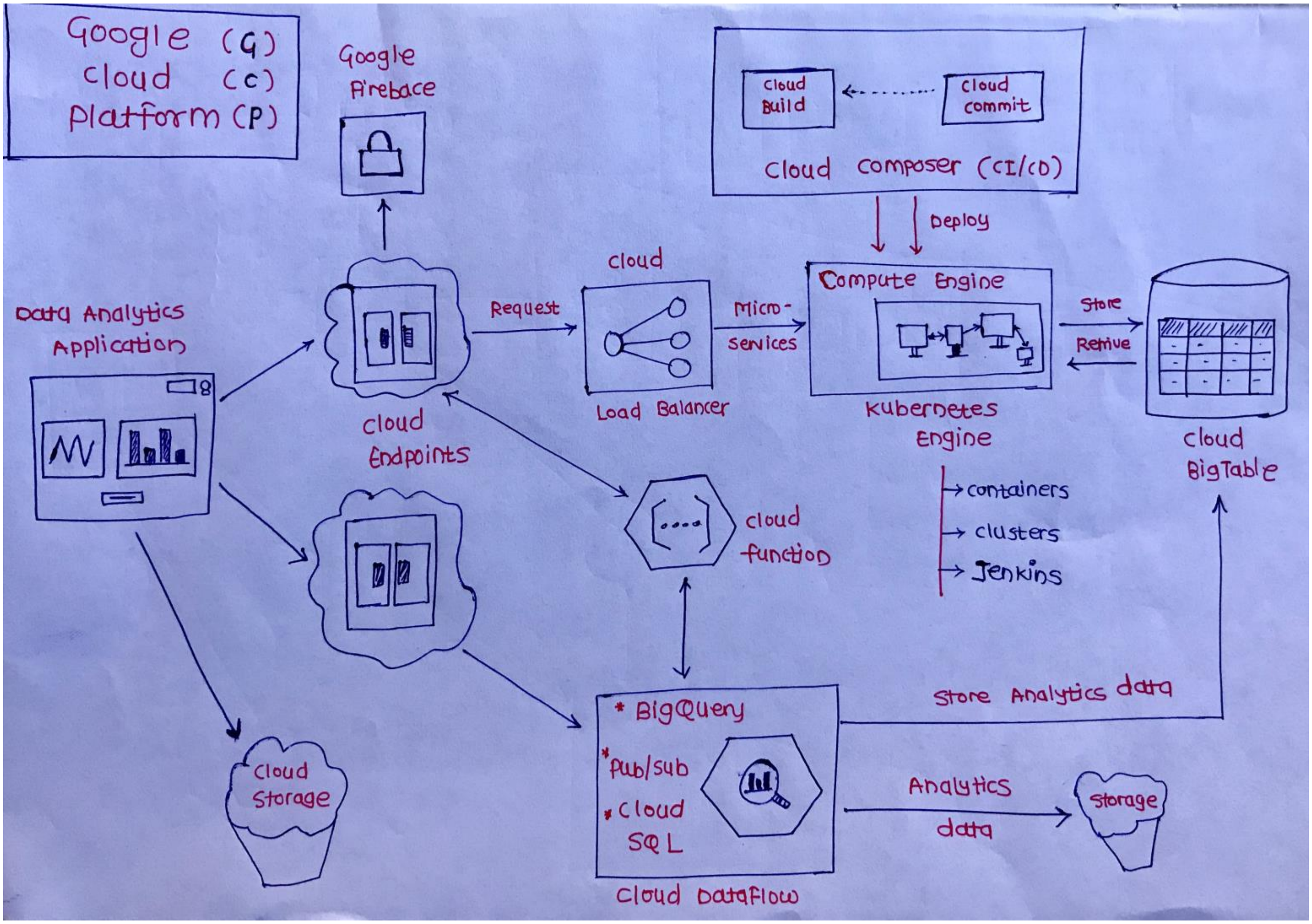
1. **Real-Time Stream data**
   Fictional wild rides companies real-time streaming data will be analyzed and data analytics will be performed on the real-time data stream which will give the Fictional Wild Rydes business teams insights to monitor the health and status of their unicorn fleet.
2. **The New York City Taxi & Limousine Commission dataset**
   A typical exploratory data analysis task described in a popular blog post by Todd Schneider [6]. The New York City Taxi & Limousine Commission has released a detailed historical dataset covering approximately 1.3 billion taxi trips in the city from January 2009 through June 2016. The entire dataset is stored on S3 and is around 215 GB. Each record contains information about pick-up and drop-off date/time, trip distance, payment type, tip amount, etc.

2. System Design and Architecture

2.1. Amazon Web Services (AWS) Cloud Platform [1]



2.2. Google Cloud Platform (GCP) [3]



3. Objectives

Project Phase – I

- Set up a fully functional project architecture on **Amazon Cloud Platform** described in the architecture and design above.
- Develop and implement a Kinesis data analytics application on the AWS platform.
- Develop and implement python, AWS Lambda, and Flask-based web application (Interface) to visualize the analytical data which will be used by the business users.
- Develop a platform-independent algorithm to analyze the performance of data processing on the multi-tenancy platforms.

Project Phase – II

- Set up a fully functional project architecture on **Google Cloud Platform** described in the architecture and design above.
- Develop and implement Google Query data analytics application on the GCP platform.
- Develop and implement python, Google function, and Flask-based web application (Interface) to visualize the analytical data which will be used by the business users.
- Reuse a platform-independent algorithm to analyze the performance of data processing on the multi-tenancy platforms.

Project Phase – III

- Comparative study of the analytical results from Amazon Cloud Platform and Google Cloud Platform.
- Analysis of algorithm performance on both cloud platforms

4. Activities

No.	Activities	Phase
1.	**Infrastructure setup on AWS Cloud Platform** 1. Setup Core Infrastructure a) Setup Amazon Virtual Private Cloud (VPC) b) Setup NAT API Gateways with 4 subnets c) Write CloudFormation scripts (yaml/Json) d) Setup AWS Cloud watch log group e) Setup security group f) Setup user roles and policies 2. Setup Clusters and Containers a) Setup an Elastic Container Service (ECS) b) Deploy a service with AWS Fargate c) Create AWS Fargate cluster d) Create Network Load balancers e) Create ECS and Docker service f) Setup Elastic Container Registry (ECR) g) Write a script to deploy Docker containers h) Configure services to communicate with the load balancer 3. Setup a database system a) Create NoSQL DynamoDB Database b) Create a Database schema and Indexes c) Create user roles and policies	**Phase - I**
2	**Data Analytics on AWS** 1. Create Data Analytics application on AWS 2. Create Amazon Kinesis stream 3. Develop an algorithm to analyze the performance on AWS 4. Write a script to create real-time streams, eg. Producer and Consumer 5. Write a lambda function to read and process the streams 6. Experiment with the stream data and record the results 7. Record and monitor stream activities over the time period 8. Setup a data storage using Amazon S3 buckets	**Phase - I**

	9. Create Amazon Data Firehose delivery stream 10. Setup Amazon Athena application to analyze the raw stream data and perform an analytics on it.	
3	**Application Programming Interfaces (API)** 1. Develop Micro-services a) Develop a Flask framework based micro-services and API's using python, AWS Lambda functions, and DynamoDB b) Create Docker Images and deploy to the Docker container. 2. Automate CI/CD Processes a) Write CloudFormation script to setup CI/CD services. b) Setup AWS CodeCommit service c) Setup AWS CodeBuild Service d) Setup AWS CodePipeline Service e) Deploy application code Docker image to clusters on the Fargate container.	**Phase - I**
1	**Infrastructure setup on Google Cloud Platform** 1. Setup Core Infrastructure a) Setup Google Virtual Private Cloud (VPC) b) Setup Network API Gateways with 4 subnets c) Write Cloud Deployment Management scripts d) Setup Google stack driver monitoring e) Setup security group f) Setup user roles and policies 2. Setup Clusters and Containers a) Setup Google Kubernetes Engine (Clusters) b) Create Google Compute Engine c) Create Network Load balancers d) Create Docker service e) Write a script to deploy Docker containers f) Configure services to communicate with the load balancer	**Phase - II**

	3. Setup a database system a) Create NoSQL Cloud BigTable Database b) Create Database schema and Indexes	
2	**Data Analytics on Google Cloud** 1. Create Data Analytics application on GCP (BigQuery) 2. Create a Cloud Data Flow stream 3. Develop an algorithm to analyze the performance on AWS 4. Write a script to create the real-time streams, eg. Producer and Consumer 5. Write a cloud function to read and process the streams 6. Experiment with the stream data and record the results 7. Record and monitor stream activities over the time period 8. Setup a data storage using cloud storage service	**Phase - II**
3	**Application Programming Interfaces (API)** 3. Develop Micro-services a) Develop a Flask framework based micro-services and API’s using python, cloud functions, and BigTable b) Develop Interface to visualize the analytics result 4. Automate CI/CD Processes a) Setup a Cloud deployment manager and CI/CD services. b) Setup workflow orchestration using Cloud Composer c) Deploy application code Docker image to clusters on the Kubernetes container.	**Phase – II**
1	**Comparative study of results** 1. Comparative study of the analytical results from Amazon Cloud Platform and Google Cloud Platform. 2. Analysis of algorithm performance on both cloud platform	**Phase– III**

5. Tools and Development Environment

Development Platforms

1. AWS Cloud Platform
2. Google Cloud Platform (GCP)

Operating System

1. Linux – AWS Virtual Machine (EC2)
2. Linux – Google Virtual Machine (Compute Engine)

Cloud Services I will be using for the project are listed below [1][3]

AWS Cloud Services	Google Cloud Services
• AWS CloudFormation • AWS Identity and Access Management • Amazon Virtual Private Cloud (VPC) • Amazon Elastic Load Balancing • Amazon Elastic Container Service (ECS) • Amazon Kinesis Data Streams • Amazon Kinesis Data Analytics • Amazon Kinesis Data Firehose • Amazon Athena • AWS Fargate • AWS Elastic Container Registry (ECR) • Amazon DynamoDB • Amazon Cognito • Amazon API Gateway • Amazon Simple Storage Service (S3) • AWS Kinesis Data Firehose • AWS Lambda • AWS CodeCommit • AWS CodePipeline • AWS CodeDeploy • AWS CodeBuild	• Compute Engine • App Engine • Google Kubernetes Engine • Cloud Functions • Virtual Private Cloud • Cloud Load Balancing • Google Domains, Cloud DNS • Cloud Storage • Cloud SQL • Cloud Spanner • Cloud Datastore • Cloud Bigtable • Cloud Dataflow • Cloud Pub/Sub • BigQuery • Cloud Composer • Cloud Deployment Manager

6. Project Results

1. Fully functional industry level Serverless cloud infrastructure to analyze the performance of Big Data processing on AWS Cloud Platform
2. Fully functional industry level Serverless cloud infrastructure to analyze the performance of Big Data processing on Google Cloud Platform
3. Kinesis data analytics application on the AWS platform.
4. Google Query data analytics application on the GCP platform
5. Comparative results of performance of large datasets on multi-tenancy environment (AWS and GCP)
6. Final project report

7. Future Research and Development

1. Research on the Serverless framework 'Flint' which is developed based on the AWS platform. [7]
2. Flint, a prototype Spark execution engine that takes advantage of AWS Lambda to provide a pure pay-as-you-go cost model. With Flint, a developer uses PySpark exactly as before, but without needing an actual Spark.

8. Project Schedule

Spring 2019	**January**		**February**				**March**				**April**				**May**			**Summary**	
Tasks:	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	Hrs	%
AWS Infrastructure Setup	20	20																40	25%
Data Analytics AWS			12	12	14													38	10%
API Development and Testing						12	10	5										27	8%
Google Infrastructure Setup								10	10	15	5							40	25%
Data Analytics Google											10	12	5					27	10%
API Development and Testing													15	10				25	8%
Write Final Report													5	10				15	12%
Demonstrate															5			5	2%
Hours	20	20	12	12	14	12	10	15	10	15	15	12	25	20	5	0	0	**217**	100.0%

9. References

[1] AWS Documentation- https://docs.aws.amazon.com/index.html#lang/en_us

[2] AWS Tutorials - https://docs.aws.amazon.com/index.html#lang/en_us#tutorials

[3] Google Documentation - https://cloud.google.com/docs/

[4] Google Tutorials - https://cloud.google.com/docs/tutorials

[5] Baldini, P. Castro, K. Chang, P. Cheng, S. Fink, V. Ishakian, N. Mitchell, V. Muthusamy, R. Rabbah Slominski, and P. Suter, "Serverless computing: Current trends and open problems," in arXiv:1706.03178v1, 2017.

[6] T. W. Schneider, "Analyzing 1.1 billion NYC taxi and Uber trips, with a vengeance," http://toddwschneider.com/posts/analyzing-1-1-billion-nyctaxi-and-uber-trips-with-a-vengeance/, 2015. http://toddwschneider.com/posts/analyzing-1-1-billion-nyc-taxi-and-uber-trips-with-a-vengeance/

[7] Y. Kim and J. Lin, "Serverless Data Analytics with Flint," *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, San Francisco, CA, 2018, pp. 451-455. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8457831&isnumber=8457768