

Yelp Data Analysis: With Apache Spark, Google BigQuery, and Google Kubernetes Engine for Deployment

Darshan Bhadreshkumar
Kansara (40195725)
Concordia University
Montreal, Canada
da_kansa@live.concordia.ca

Nisarg Adalja (40276285)
Concordia University
Montreal, Canada
n_adalja@live.concordia.ca

Anuja Ajay Somthankar
(40265587)
Concordia University
Montreal, Canada
a_somtha@live.concordia.ca

Nimisha Mavjibhai Jadav
(40267767)
Concordia University
Montreal, Canada
n_jadav@live.concordia.ca

Prachi Jatinkumar Patel
(40261038)
Concordia University
Montreal, Canada
pa_prach@live.concordia.ca

ABSTRACT

In the current era of online evolution and assessments, data plays an indispensable role in aiding individuals to make well-informed choices regarding where they should dine-out, shop or watch. Yelp data provides a wide range of information that includes intricate details about different businesses and its users and reviews. This report endeavours to extract meaningful insights from this dataset.

By carrying out analyses on Yelp's data repository we identify not only recurring patterns but also uncover trends and available information that can be advantageous both for end-users and business establishments. We have created 4 questions to gain insight on the data and used Spark and BigQuery to find output. A Django application is created from this and deployed on Google Kubernetes Engine. Through this process, we understand the various distributed features of the Google Cloud Platform. Furthermore, we have also compared the processing time of Apache Spark and Google Big Query.

1 INTRODUCTION

The main objective of this project is to analyse the Yelp data using Spark, Google BigQuery and deploy using Google Kubernetes Engine, which forms a solid foundation for efficient data processing within a distributed system along with comparing Spark and BigQuery. Distributed features such as Kubernetes clustering, scalability, fault tolerance, availability, load balancing and naming are applied. The analysis was done in 2 ways: first was using Apache Spark was used for data loading, data cleaning and analyzing the data, second was using Google BigQuery, where we created complex SQL queries to find answers to the questions we designed. Both of their proficiency in parallelized operations enables faster computation time making it suited for large-scale analytics tasks.

The project centers around analysing the top 10 restaurants that have garnered 5-star rating from users over time. Moreover, it seeks to delve the top 5 users who have received the most compliments. Additionally, it aims to examine whose reviews were marked as the most useful ones. Furthermore, in this report we will analyse the unique businesses that were given more than 3-stars by at least 50 reviews in 2018.

We have used Yelp dataset with 3 files of total 8.81 GB named:
[2]

- (1) yelp_academic_dataset_business.json
- (2) yelp_academic_dataset_review.json
- (3) yelp_academic_dataset_user.json

2 ARCHITECTURE

2.1 Google Cloud Bucket

Google Cloud Platform provides Google Cloud Storage, commonly referred as "bucket", which is a reliable and scalable object storage service that is intended to store and retrieve data efficiently. Simple as well as sophisticated files can be organised and managed logically in the buckets. It has features like global distribution, access control and versioning where the users can easily upload, download and share the data.

2.2 Google Kubernetes Engine

Kubernetes is an open-source container platform that manages containerized workloads and services. Google Kubernetes Engine (GKE) expands the advantages of Kubernetes which runs on Google Cloud Platform. GKE is used for creating and deploying the clusters, replication controllers and updating and debugging clusters.

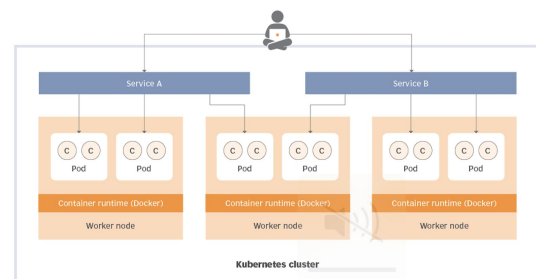


Figure 1: Working of Kubernetes [3]

2.3 Django

Django uses Python programming language for building web applications. The main components of Django are:

- URLs - They define the mapping between URLs and views. It specifies the endpoints that users can access and associates each endpoint with a specific view function.
- Views - It handles the logic of processing requests and sending back the response.
- Models - It is responsible for handling the logical part of the web application and how the data is stored in the database. [5]

2.4 Apache Spark

Apache Spark is an open-source distributed computing platform which offers a flexible cluster computing framework for handling huge amount of data. Because it can store intermediate data in memory rather than writing it to disk, Apache Spark's in memory processing capability is one of the main advantages. It can handle a range of workloads, such as streaming, interactive queries, iterative algorithms, and batch processing.

2.5 Google Big Query

Google Cloud Platform also offers Google BigQuery, a fully-managed serverless analytics and data warehouse platform. SQL-like queries are used real-time processing and analysis of large datasets. Google BigQuery can handle batch and streaming data processing and it can be used for variety of purposes including machine learning, data exploration and business intelligence. One of the key features is the division of compute and storage resources in integration with GCP.

3 METHODOLOGY

(1) Creating a Django project on GitHub:

Create a GitHub repository for saving all the Python files for Django web application. This is done to display the output in a HTTP web page. There are a total of 8 URLs, 4 each for Spark and BigQuery.

(2) Docker for the project:

Write a Dockerfile for the project. Also create a requirement.txt file which contains the version number for installing pyspark, django and pandas. Build the Docker image and push it to container registry in Google Cloud Platform.

- Creating Docker Image:
docker build -t gcr.io/focus-vertex-407519/gkeyelp:latest
- Pushing the docker Image:
docker push gcr.io/focus-vertex-407519/gkeyelp:latest

(3) Creating deployment.yaml, service.yaml and autoscaler.yaml:

Create 3 YAML files, each containing the deployment details, the service details and the autoscaler setup respectively. The files contain the name of the container, docker image name, port number and details horizontal auto scaling. Insert the these YAML files in the bucket in GCP.

(4) Creating a cluster in Google Kubernetes Engine:

GKE cluster consists of two things: nodes which contains the user pods and control panel. There are two type of

clusters, autopilot and standard cluster. For this project, we are using the standard cluster which gives us the flexibility over the infrastructure of the cluster.

- Creating the cluster:

```
gcloud container clusters create gkeyelp-django --num-nodes=3 --zone=us-east1-b
```

(5) Deploying to the cluster:

Set up the Kubernetes cluster credentials on local machine, fetch the deployment file from the Google Cloud storage bucket and deploy it to the GKE cluster using the 3 yaml files. Same commands would be applied for all 3 files as specified below.

- gcloud container clusters get-credentials gkeyelp-django --zone=us-east1-b
- gsutil cp gs://comp6231-yelp-bucket/deployment.yaml
- kubectl apply -f deployment.yaml

After deployment, go to workload in GCP and check for the deployment details of 'gkeyelp'. Here, we can see the CPU, memory and disk usage, number of active pods and the exposed service details.

4 DISTRIBUTED FEATURES OF GCP

4.1 Kubernetes Clustering

GKE environment consists of virtual machines called as nodes which are hosted on Google Cloud Engine. The nodes are grouped into clusters to facilitate the deployment of the application. Containers are deployed as parts of pods which run on nodes and can contain one or more container that share the same network. The Kubernetes API serves as the interface for interacting with workloads on the cluster.

Some of the advantages of using GKE is platform management, improved security, cost optimisation, reliability and availability. GKE can be used for AI and ML operations, data processing and gaming platforms.

Cluster: my-cluster			
Namespace	default		
Labels	No labels set		
Logs	Container logs, Audit logs		
Replicas	2 updated, 2 ready, 2 available, 0 unavailable		
Pod specification	Revision 1, containers: gkeyelp		
Horizontal Pod Autoscaler	Unable to read all metrics		
Vertical Pod Autoscaler	Not configured		

Active revisions						
Revision	Name	Status	Summary	Created on	Pods running	Pods total
1	gkeyelp-7f66ddcf56	OK	gkeyelp: gcr.io/focus-vertex-407519/gkeyelp:latest	Dec 16, 2023, 11:59:40 AM	2/2	

Managed pods					
Revision	Name	Status	Restarts	Created on	
1	gkeyelp-7f66ddcf56-4glnj	Running	0	Dec 16, 2023, 7:35:25 PM	
1	gkeyelp-7f66ddcf56-4nddb	Running	0	Dec 17, 2023, 12:30:03 AM	

Exposing services		
Name	Type	Endpoints
gkeyelp-service	Load balancer	34.71.172.58:8080/0
gkeyelp-service	Load balancer	35.222.248.128:80/0

Figure 2: Clustering

4.2 Scalability

When we deploy the cluster for the first time we may be unsure about the resource utilisation and how it can change based upon the usage in the future. We have created autoscaler.yaml to use

Horizontal Pod Autoscaling in this project which ensures that the workload is consistent and the cost usage is efficient. The horizontal pod autoscaler can scale the number of pods in the workload automatically. This can be done on the basis of one or more metrics types like when the resource utilisation exceeds a certain threshold, custom metrics given by Kubernetes or metric from an application or external cluster.

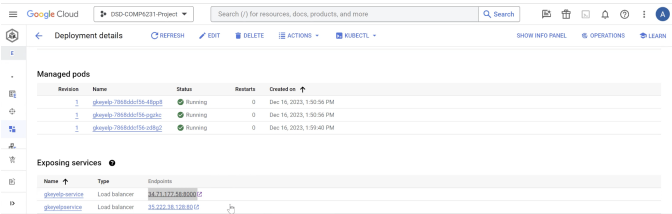


Figure 3: Before Scaling

Command used for scaling:
kubectl scale deployment gkeyelp --replicas=4

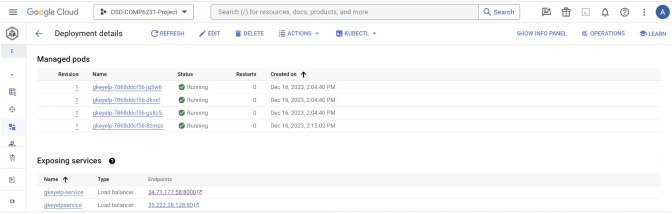


Figure 4: After Scaling

4.3 Fault Tolerance

Fault tolerance enables the system to run properly even after any failure in some of the components. In a study conducted by University of Massachusetts, Kubernetes clusters are fault tolerant because of resource reservations to ensure that enough resources are available during the time of failure to overcome the impact. [4]

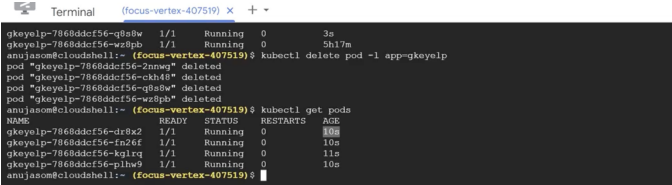


Figure 5: Fault Tolerance

As seen in Figure 6, even after deleting the pods we can see that pods are again back to running status showing us the fault tolerance feature of distributed system.

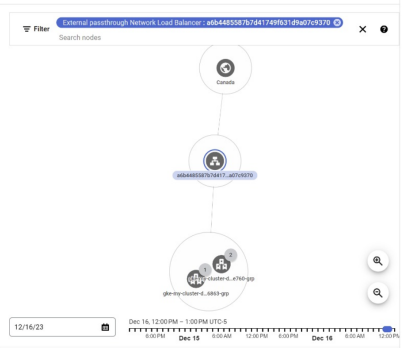


Figure 6: Load Balancer

4.4 Load Balancing

Load balancing distributes the traffic among different instances in the network. This reduces the risk of performance issue. In our project we have selected the Network Load Balancer which handles TCP, UDP or other IP protocols.

As seen in Figure 7, the type of Network Load Balancer used is External passthrough network load balancer. This means that regardless of the IP address of the pods, clients can connect with them anywhere from the internet through this Load Balancer. [1]

We created a service.yaml of type Load Balancer, creating a kubernetes service, which provides a single endpoint for accessing all the pods, providing distribution transparency to the user. In load balancer service, the IP address of an external network load balancer is used to access the pods. This service distributes workload across all available pods.

4.5 Availability

Availability is the ability of the system to remain available and accessible to the users at any given moment. To achieve availability in GKE, we can create node pools which groups the nodes within the cluster sharing the same configuration. GKE has a feature called as node auto-repair, which monitors the health of the node and fixes them in case of any issue. GKE also maintains the security updates and patches without causing any downtime of the application.

4.6 Naming

Naming services is a very important part of service discovery in GKE. Service discovery locates processes on a cluster network. Kubernetes has its own internal DNS. All service names follow a standard naming convention. There are 2 types of DNS in GKE, kube-DNS and Cloud DNS, default being kube-DNS. Being DNS, both are structured naming services.

Kube-DNS acts as a deployment and maps all pods to the kube-DNS namespace. A service is then created which the pods are mapped to, providing a single clusterIP. All pods in the cluster use it to resolve DNS queries, creating a central point for DNS resolution. The kube-dns-autoscaler ensures that the number of kube-dns replicas adjusts dynamically, based on the cluster's resource requirements.

5 COMPARISON OF GOOGLE BIG QUERY AND APACHE SPARK

We have written a Python script which uses Google Big Query client that runs a complex SQL query with the goal of analysing the data mentioned in the introduction. The script begins by connecting Google Big Query with a client variable. Django's HttpResponse object is used process the results and convert them into HTML table representation, which is then encapsulated in an HTTP response. This script is intended to be integrated into a Django web framework and provides a concise method for extracting meaningful insights from a BigQuery dataset, demonstrating the seamless integration of data analytics with web development.

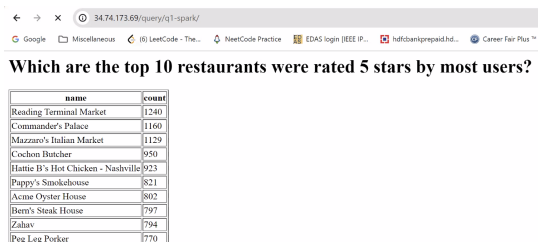
As we can see in Table 1 that **Google Big Query's processing time is much faster than the Apache Spark**. This is because BigQuery internally uses a columnar storage format, which makes it extremely efficient for reading large amounts of data as it can quickly skip over irrelevant data. When a query is executed, BigQuery distributes resources across its distributed architecture in order to process large datasets quickly. It employs the Dremel model, which allows queries to be executed on thousands of machines in parallel, providing high-speed analysis and returning results in seconds or minutes. This level of performance is achieved by combining massive parallel processing (MPP), a high level of query optimization by BigQuery's query planner, and a storage layer optimized for read access and analytics workloads.

Sr. no.	Google Big Query	Apache Spark
1	2.02	7.308
2	0.933	5.942
3	0.915	12.497
4	0.88	9.202

Table 1: Processing time for Big Query and Apache Spark in seconds

6 RESULT

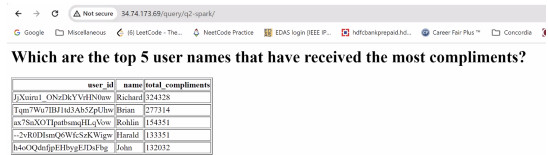
- Using Apache Spark
 - Question 1 - Which are the top 10 restaurants were rated 5 stars by most users? (Figure 7)



NAME	COUNT
Reading Terminal Market	1240
Commander's Palace	1160
Mazzaro's Italian Market	1129
Cochon Butcher	950
Hattie B's Hot Chicken - Nashville	923
Pappy's Smokehouse	821
Acme Oyster House	802
Bern's Steak House	797
Zahav	794
Peg Leg Porker	770

Figure 7: Top 10 Restaurants

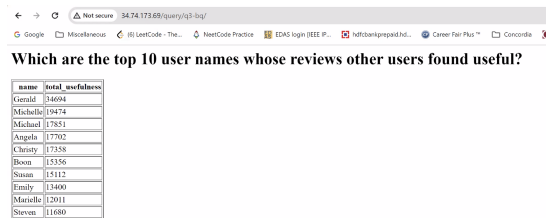
- Question 2 - Which are the top 5 user names that have received the most compliments? (Figure 8)



user_id	name	total compliments
JXoan1_ONoZkVYdNlhw	Richard	154128
Tamr7Wb7Bd1ASzZpLhw	Brian	127114
ac7baXOTtIpuhmgH-qNow	Robin	154351
-2nR0DlmsQW6sKwlgw	Harold	133351
hbcOQmGpEHrygZDdPg	John	13202

Figure 8: Top 5 Usernames with most compliments

- Using Google Big Query
 - Question 3 - Which are the top 10 user names whose reviews have been marked as most useful? (Figure 9)



name	total usefulness
Gerold	34604
Michelle	19474
Michael	17851
Angela	17762
Christy	17358
Brian	15356
Susan	15112
Family	15040
Marcia	12011
Sarven	11680

Figure 9: Top 10 usernames whose reviews were useful

- Question 4 - What are the total number of unique businesses that were given more than 3 stars by at least 50 reviews in 2018 in Philadelphia ? (Figure 10)



total count
100

Figure 10: Total businesses with more than 3 stars

REFERENCES

- Google Cloud. 2023. *Cloud Load Balancing overview*. <https://cloud.google.com/load-balancing/docs/load-balancing-overview>
- kaggle. 2023. *Yelp Dataset*. <https://www.kaggle.com/datasets/yelp-dataset/yelp-dataset/data>
- TechTarget Networks. 2023. *Google Kubernetes Engine (GKE)*. <https://www.techtarget.com/searchitoperations/definition/Google-Container-Engine-GKE>
- Yining Ou. 2023. Maximizing Resource Utilization in GKE: A Case Study on IP Address Management and Budget Allocation in Google Kubernetes Engine Config Page. *DeanandFrancis* 1, 1 (June 2023), 4. <https://doi.org/10.61173/zy9y5z35>
- Rajesh Kumar Singh, Himanshu Gore, Ashutosh Singh, and Arnav Pratap Singh. 2021. Django Web Development Simple Fast. *INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS* 9, 2, Article 9 (May 2021), b808-b815 pages. http://ijcrt.org/viewfull.php?p_id=IJCRT2105197