Task 1 - External storage support

Yuan Fan <yfan@kth.se>

Xuwei Gong<xuweig@kth.se>

I. GCP Storage support

Google offers a variety of storage services on the GCP based on different types and needs. In this task, we are required to investigate different storage support that Google Cloud Platform provided including Cloud Storage(GCS)/ Cloud Datastore / Cloud Bigtable (GCB)/ Cloud SQL/Google Drive based on product features.

A. Comparison

Google Cloud Storage: Google Cloud Storage (GCS) is an Object storage that is easy to manage, durable, and highly practical performance. It is designed for a wide range of data types, including active data and archived data. It can be divided into storage categories: regional, multi-regional, near-line and far-line. All categories provide unlimited data, the same tools and APIs for data access, OAuth and fine-grained access control, access to other Google Cloud Storage services, and pay-per-use models.

Google Cloud Datastore: The Datastore is a highly scalable and fully managed NoSQL database that can be used at the back end of an application. It is a database that automatically processes data stored on multiple servers and is automatically copied and adjusted according to the application's loading. This feature is also used in Google App Engine's database.The Datastore is designed based on Bigtable's architecture and also complements what is missing in Bigtable.

Google Cloud Bigtable: Cloud Bigtable is a highly scalable NoSQL database for low latency and high throughput workloads. It is a decentralized repository that can scale to billions of rows and thousands of columns, allowing you to store data in terabytes and petabytes. It also has high practicality. When you write data, it will be synchronized to 3 (or more) servers. In Bigtable, data is stored in a highly extensible table. The table is composed of rows and columns. Each row has a value registered in the index. This value will become the key of the row. You can scan keywords but cannot scan the names of columns.

Google Cloud SQL: Google Cloud SQL provides fully-hosted My SQL and PostgreSQL databases on GCP. In terms of MySQL functionality, Cloud SQL is almost identical to those offering local hosting services. Cloud SQL is highly efficient and highly scalable, providing up to 10 TB of storage per VM, 25,000 IOPS, and 208 GB of memory. Google Cloud SQL can automate functions such as backup, copy, modify programs, update, and more.

II. Access Google Cloud Storage

After we done the throughly research of different types of storage options we choose GCS to further our project,Google cloud storage is highly structured, same with other GCP components, it is project-based, within the project the major container is called **bucket**, we wrote a simple code in Python with the help of GCS tutorial to realize the creating bucket, writing to the bucket through Blob API and downloading the file from the bucket that we created. The simple application structure is shown in figure 1:

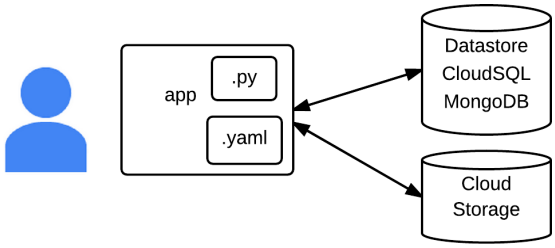

Fig. 1. Application Architecture [1]

There are several ways connecting with GCS, either you can write/read to the cluster using Google Cloud Storage manager which is a web-based interface console, you can upload with HTTP standard methods like *GET, PUT,POST*, or you can programmatically using REST APIs and finally the *gsutil* tool using command line.

The Blobstore API applies on many programming language, you can use the Blobstore API to upload objects to Cloud Storage or provide objects from Cloud Storage. The Blobstore API allows your application to serve data objects called Blobs. Applications can use Blobstore to accept large files uploaded from users and to serve them. Once uploaded, the file is called blobs. Blob in python is easy to understand and simple to implement, assigning the bucket with a created blob then we can do the upload and download actions with codes.

III. Storage performance Benchmark

To test out the performance of Google cloud storage, we got some help from our classmates and tried to upload several files in different file extensions and sizes. The whole test with single transfers we use 8 files in total 423.32mb, the experiment keeps record of each single file about the R/W information, then we calculate the average speed of these two operations. Table 1 summarize the findings:

[1]https://cloud.google.com/python/getting-started/using-cloud-storage

TABLE I
STORAGE BENCHMARK FOR DIFFERENT TYPE OF FILES

	movie.mp4	reference.pdf	picture.jpg	Total
Size	364.94mb	26.91mb	8.52mb	423.32mb
Write Speed	1.15mb/s	1.18mb/s	1.12mb/s	1.0mb/s(AVG)
Read speed	10.4mb/s	7.89mb/s	5.03mb/s	9.0mb/s(AVG)

From the result we could assume that the reading rate are much faster than the writing performance particularly for the large files over 100mb under a certain benchmarking environment.

For the parallel transmission, we took 2 different hosts (Mac OS X and Win 10) under the same network conditions to redoing the test with same amount of file and same sizes, the result shows that the R/W average speed of host1 are 1.0mb/s and 6.32mb/s where the other one gives the speed of 0.5mb/s and 4.17mb/s. These data shows that the synchronized transmission will share the data flow bandwidth.