Task 1 - External storage support

Hongyi Zhang <hongyiz@kth.se> Lida Liu <lidal@kth.se>

I. Different Types of Storage Support

Different applications and workloads require different storage and database solutions. Google Cloud Platform provides a variety of storage service to meet the needs for structured, unstructured, transactional, and relational data. The rest of the section we want to give you some examples and brief introductions of some storage types.

- **Cloud Storage buckets** GCS is a scalable, fully managed, highly reliable, and cost-effective object/Blob repository. Usually we use GCS to store streaming media, backup data, or disaster recovery. GCP can be accessed globally and support all the machine type. Besides that it have the highest IO throughput compared to the other storage types.
- **Google Cloud Bigtable** GCB is a scalable, fully-managed, NoSQL wide-column database for real-time access and analytics workloads. Regardless of size and application type, Bigtable can achieve low latency and high throughput.
- **Google Cloud Datastore** Cloud Datastore is a massively scalable NoSQL database that can be used in your application. There are many features of Cloud Datastore, such as ACID transactions, SQL class queries, indexes, and so on. Any deployment goal can easily access data through Cloud Datastore's RESTful interface.
- **Google BigQuery** It is a scalable and fully-managed Enterprise Data Warehouse (EDW) with SQL and fast response times. Usually Google BigQuery is used to construct large-scale data report or using SQL to process Big Data.
- **Google Drive** Google Drive is really popular and is a collaboration space for storing, sharing, and editing files. Users can access files from any location via web pages, apps, and sync clients.

The following figure shows how to choose the storage to fulfill your own requirements.

Fig. 1. How to choose a storage type

II. Accessing Google Cloud Storage

Google Cloud Storage is a flexible, scalable, and durable storage option for your virtual machine instances. You can write files to Cloud Storage buckets from almost anywhere, so you can use buckets as common storage between your instances, Google App Engine, your on-premises systems, and other cloud services. Therefore we decide to choose GCS as the external storage to access by using our own laptop.

For connecting to the GCS, you can using either Blobstore Python API or REST API such as Json, XML. For this time, we mainly use the Blob API to upload and download the objects.

The Blobstore API allows the application to serve data objects, called blobs, that are much larger than the size allowed for objects in the Datastore service. Blobs are useful for serving large files, such as video or image files, and for allowing users to upload large data files. Blobs are created by uploading a file through an HTTP request. Typically, applications will do this by presenting a form with a file upload field to the user. When the form is submitted, the Blobstore creates a blob from the file's contents and returns an opaque reference to the blob, called a blob key, which can later be used to serve the blob. The application can serve the complete blob value in response to a user request, or it can read the value directly using a streaming file-like interface.

III. Storage Operations Performance Benchmark

The code of this part is enclosed in the zip file and also you can find the code on the Github https://github.com/Mr-Hongyi/ID2210-Cloud_Project.git. Besides that, the testing is using the wireless network provided by KTH Kista.

During the single test we are going to upload and download 10 files with the size 557.47 mb in total. We have observed the average uploading and downloading time and speed to simulate the read/write operation. The following figure shows the Reading and Writing throughput.

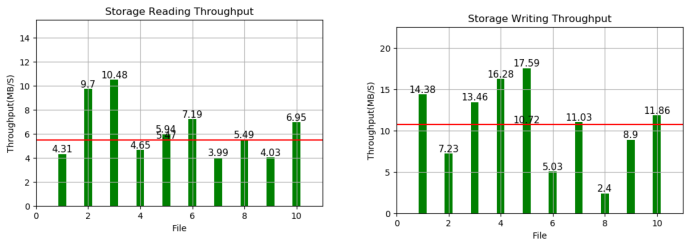

Fig. 2. R/W in Single Transmission

We can see that the average Reading throughput for single transfer is 5.47mb/s while Writing throughput is 10.72mb/s. And the maximum for reading is 10.48mb/s and for writing is 17.59mb/s. Overall the writing speed is twice as fast.

While in the parallel test, we are uploading and downloading two sets of identical files also 557.47mb for each.

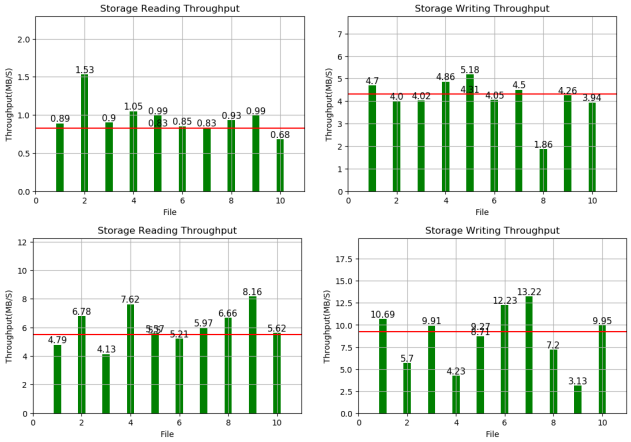

Fig. 3. R/W in Parallel Transmission(Statistics on two independable client)

As we can see the average reading speed for each are 0.83mb/s and 5.5mb/s, while the average writing speed are 4.31mb/s and 8.71mb/s. Addition of two sets of data is almost equal compared with the previous single transfer. In other words, the synchronous transmission still shares the data flow bandwidth to some extent.

REFERENCES

[1] Storage Options, Google Cloud Platform Document [Online]. Available: *https://cloud.google.com/storage-options/*

[2] Connecting to Cloud Storage buckets, Google Cloud Platform Document [Online]. Available: *https://cloud.google.com/compute/docs/disks/gcs-buckets*

[3] Blobstore API Overview, Google Cloud Platform Document [Online]. Available: *https://cloud.google.com/appengine/docs/standard/python/blobstore/*