**Introduction-**

This assignment introduces to Google App Engine and implementation of non-trivial distributed system. We built distributed file system over google app engine using Google cloud storage and blob storage.

Following are some objectives of this assignment:

* Generate 411 files of different sizes specified below.
* Upload 411 files to google cloud.
* Checking existence of file in database and memcache.
* Finding content of file.
* Remove the file from cloud storage.
* Listing of all files present in database and memcache.
* Search the file using ‘regex’.

**Design:**

* **Storage Statistics**- It is for storing stats of each file uploading to storage.
* **Blob Storage**- Blob Storage is used to store large size files.
* **Data Stores**- It is for storing metadata of file.
* **Memcache**- Memcache used for fast processing of data and fast retrieval of information since it deals with cached data not storage.

**Datasets:**

We generated files of different sizes using FileGenerator.java program. On compiling and running this program it will ask for size of file to be generated (should selected from menu) and number of files to be generated. Following are different sized 411 files generated in for this assignment.

* 100 files of 1KB (total size 100KB)
* 100 files of 10KB (total size 1MB)
* 100 files of 100KB (total size 10MB)
* 100 files of 1MB (total size 100MB)
* 10 files of 10MB (total size 100MB)
* 1 files of 100MB (total size 100MB)

We used java to implement various functions in this assignment. Each function listed below is implemented using its own java file. Description is given below.

We have implemented following functions in this assignment.

1. **Insert:**

This function is used to upload files on Google storage and memcache. Uploadfiles.java is java file to upload files using memcache and Uploadfileswom.java to upload files without memcache.

Initially webpage asks for operation to be completed using memcache or without using memcache. On selecting appropriate option it will redirect to next page and there we can browse the local storage to select files. We can upload multiple files at the same time. File having size <= 100KB uploaded to

1. **Find File:**

This function used for searching file in database and memcache. We have to specify the name of file which is to be searched.

DisplayContains.java is java file used to find the file specified by user.

1. **List Files:**

This function lists all files uploaded on cloud storage and memcache.

A java file ShowAll.java lists all files present on cloud storage and memcache.

1. **Remove File:**

This function removes specified file from cloud.

Delete.java- java file for removing files with memcache.

Deletewom.java- java file for removing files without memcache.

1. **Check File:**

Checking whether file is present in cloud or not.

Searchfile.java- java file to search file with memcache.

Searchfilewom.java- java file to search file without memcache.

**Extra Credits**

1. **Boolean = Check(key):** It checks whether file is present or not in distributed storage system. We have covered his part in main experiment.
2. **Boolean = checkCache(key):** It checks whether file is present or not in cache. We have covered his part in main experiment.
3. **Boolean = RemoveAllCache():** It removes all files from Cache. RemoveAllCache.java is java file which removes all files from cache.
4. **Boolean = RemoveAll():** It removes all files from cache as well as distributed file storage. RemoveAll.java is java file used to do so.
5. **Double = CacheSizeMB():** It gives total size allocated for file in cache. Stats.java is java file used to implement this function.
6. **Double = CacheSizeElem():** It gives total number of files in cache of our distributed file system. Stats.java is java file which gives the total number of files.
7. **Double = StorageSizeMB():** It gives total size in MB allocated for files uploaded in distributed file system. SizeStorage.java is java file used to get allocated space.
8. **Double = StorageSizeElem():** It gives total number of files in distributed file storage. SizeStorage.java is java file used to get total number of files is DSS.
9. **Boolean = FindInFile(key, string):** It searches for regular expression in file. Searchregexp.java is java file used to do this.
10. **Key[] = Listing(string):** It retrieve a list of all file names, whose names match the regular expression string. Searchregexp.java is java file used to do this.

**Extensions to our system:**

* + Uploading, retrieving and other functions requires internet access so ultimately it depends upon internet speed. So if we have better network speed then we could improve the performance.
  + Presently system is capable to deal with files with .txt extensions. We can change the system to deal with other extensions like .doc, .pdf, .gz, .jar, .tar, .tar.xz, .zoo, .zip etc.