

Subject

<https://docs.google.com/document/d/1VVrKrDH9Hj0iGAbUADNwkXJDBd2HnzYe52RD3ASlBMA/edit>

Team

- AMINE Hajji
- TOUTAIN Xavier
- BENZA Amandine
- FORNALI Damien

Master II - SI5

University / Polytech Sophia-Nice

Cloud Computing - 2018

Delivery 1 - 19/10

To simplify, our upload/download files formats will be standard like .txt or even images (.png) according to the time we have.

We make the following suppositions:

- 60% of our users upload/download text files
- An average text file size is around 100 KB.
- Remaining users upload/download only png files
- An average png file size is around 3 MB
- We got 1000 users / h, 50% are Noob, 35% are Casual and remaining Leet
- One typical Noob user makes 4 uploads/downloads a hour
- One typical Casual user makes 10 uploads/downloads a hour
- One typical Leet user makes 20 uploads/downloads a hour
- It gives us $(4 * 500) + (10 * 350) + (20 * 150) = 8500$ requests / hour * 24 = 204000 requests / day and 141 requests / minute ≈ 2 requests by second
- An average request handle time is about 300ms (upload / download time not included)
- One instance could handle this number of requests but we would lack of margin, thus we decide to use two instances (at least as a first scale measuring)
- To handle urgent cases we will be able to provision two additional instances
- We chose instances of B2 type which seem enough to handle this work load
- $(2 * 0.1) / 1000 = 0.0002\$/user$
- $(0.20 * 24 * 31) = 148\$/month$
- In case we need to use elasticity, we will need to double this amount, so it will be near

300\$/month.

- We will use a database offered by the Google Cloud Storage (NoSQL or RDBMS). When compared to relational databases, NoSQL databases are more flexible and scalable. These are attributes that interest us for this project.
- For the architecture overview, please refer to the 'ArchitectureOverview.pdf' file which contains our different schemas. It contains our architectures for the different user grades.

Subject resume and notes

Noob [0-100]:

We will use a push queue to implement this grade.

The Noob' tasks are handled consecutively.

He is able to send only one operation by minute. If he tries to send more he will get a mail which title/content will be "lol non noob".

This grade users' files are deleted after 5 minutes.

Casual [101-200]:

We will use a pull queue to implement this grade.

The system offers to the Casual 2 operations in parallel by minute.

Casual files are deleted after 10 minutes.

Leet [> 201]:

We will use a pull queue to implement this grade.

The system offers to the Leet 4 operations in parallel (1 min delay).

Its files are stored for only 30 minutes long.

Questions

- Does the 1 minute delay timer starts at the start or the end of the upload ?

Links

Intro: <https://fhermeni.github.io/sacc/sacc-introduction.pdf>

sacc-architecture: <https://fhermeni.github.io/sacc/sacc-architecture.pdf>

Console google cloud: <https://console.cloud.google.com/>

Course page: <https://fhermeni.github.io/sacc/>

Postman: <https://www.getpostman.com/>

Push queue: <https://cloud.google.com/appengine/docs/standard/java/taskqueue/push/>

Pull queue: <https://cloud.google.com/appengine/docs/standard/java/taskqueue/pull/>

Export billing data to a file:

https://cloud.google.com/billing/docs/how-to/export-data-file?rd=2&visit_id=636754899995029268-3782129590

Mail API overview: <https://cloud.google.com/appengine/docs/standard/java/mail/>

Sending email from an instance: <https://cloud.google.com/compute/docs/tutorials/sending-mail/>

Platform pricing calculator: <https://cloud.google.com/products/calculator/>

Compute engine pricing: <https://cloud.google.com/compute/pricing>

Google Machine Types: <https://cloud.google.com/compute/docs/machine-types>

Google cloud storages: <https://cloud.google.com/products/storage/>

Price calculator: <https://cloud.google.com/products/calculator/>