

Évaluation du stockage de notre application PolyShare

CANAVA Thomas, GARDAIRE Loïc, JUNAC Jérémy, MELKONIAN François



Solution de stockage de notre application

Nous avons identifié deux types de données que nous allons devoir stocker. Tout d'abord, les utilisateurs, avec leur email, leur score, et les identifiants des vidéos mises en ligne par cet utilisateur. Ensuite, nous devons stocker les vidéos publiées par les différents utilisateurs. Nous allons décrire et justifier les différentes solutions que nous avons choisi pour chaque type de données.

Détail de la hiérarchisation

Notre collection contiendrait une liste d'utilisateurs.

Chaque utilisateur possède ces données:

- Utilisateur :
 - Email (unique, clef privée)
 - Score
 - Liste des vidéos uploadées

Une vidéo uploadée correspond à une entrée de la liste et possède ces informations:

- Vidéo uploadée :
 - date d'upload
 - identifiant de la vidéo

Solutions de stockage

Pour les utilisateurs, deux choix s'offrent à nous. D'une part, nous pourrions utiliser une base de données relationnelle (Google Cloud SQL). Cela nous permettrait d'avoir une base de données cohérente en permanence. D'autre part, nous pourrions faire le choix d'une base de données orientée document (Google Cloud Datastore). Cette solution nous convient tout aussi bien. En effet, la mise à jour des profils sera plus rapide (notamment lors de l'ajout de vidéo) et il sera plus simple de faire évoluer notre modèle. De plus, nous ne possédons pas de relations entre les différentes table SQL. À cause de la taille des données, les vidéos ne seront pas stockées dans une table SQL. Nous ne tirerons donc pas pleinement avantage de la structuration de nos données. Dans notre cas, la hiérarchisation de nos données nous suffit, vu les requêtes nécessaires pour notre système.

Nous avons donc choisi la deuxième solution. L'inconvénient est que le *Leaderboard* ne sera pas à jour à tout moment. Mais au final, ce n'est pas très grave s'il met quelques minutes à se mettre à jour, d'autant plus que ce n'est pas la partie la plus importante de notre métier.

Pour la partie vidéo, nous avons encore une fois deux choix: le Google Cloud Storage ou la Blobstore API. Ils sont tous les deux faits pour stocker de gros fichiers, notamment des vidéos. Le Google Cloud Storage est cependant plus récent que le Blobstore. Nous avons tout de même choisi d'utiliser le Blobstore pour sa facilité d'interface.

Evaluation du coût de la persistance en fonction du type de donnée

Rappel du contexte

Ratio des niveaux sur 1 000 utilisateurs

Nous avons catégorisé nos potentiels utilisateurs en groupe nous semblant cohérent.

Noob : 600 utilisateurs, 768 Go de vidéos téléchargées (soit 19 200 vidéos par semaine)

- 50% : Uploader exceptionnel (1-2 vidéos) qui regarde peu de vidéos (1h / semaine)
- 50% : Uploader exceptionnel (1-2 vidéos) qui regarde beaucoup de vidéo (7h / semaine)

Casual : 300 utilisateurs

- 50% : Uploader occasionnel (4-5 vidéos) et regarde peu de vidéos (1h / semaine)
- 50% : Uploader occasionnel (4-5 vidéos) et regarde beaucoup de vidéos (7h / semaine)

Leet : 100 utilisateurs

- 50% : Uploader expérimenté (~6 vidéos) et regarde peu de vidéos (1h / semaine)
- 50% : Uploader expérimenté (~6 vidéos) et regarde beaucoup de vidéos (7h / semaine)

Leet et Casual : 512 Go de vidéos téléchargées (soit 12 800 vidéos par semaine)

Nous avons pris une moyenne de 6 vidéos mises en ligne par semaine pour les Leet. En effet, les personnes mettant en ligne plus de 1 vidéo par jour restent très rares, donc 6 nous semblait être une moyenne convenable.

Stockage des utilisateurs

Nous avons donc 1000 utilisateurs à stocker.

Voici la table de prix pour la solution que nous avons choisi, à savoir Google Cloud Datastore:

	FREE LIMIT PER DAY	PRICE ABOVE FREE LIMIT (PER UNIT)	PRICE UNIT
Stored data	1 GB storage	\$0.18	GB/Month
Entity Reads	50,000	\$0.06	per 100,000 entities
Entity Writes	20,000	\$0.18	per 100,000 entities
Entity Deletes	20,000	\$0.02	per 100,000 entities
Small Operations	50,000	Free	-

Figure n°1 : Prix du Google Cloud Datastore

NB: Sont compris dans les *Small Operations* toutes les lectures par clé.

Pour calculer le coût du stockage des utilisateurs, nous avons effectué le raisonnement suivant:

- Sachant que nous avons 1 000 utilisateurs, et que nous pouvons raisonnablement penser que la taille de chaque document ne dépassera pas 1 Ko. Donc nous ne dépasserons pas le quota gratuit de 1 Go.
- Nous avons aux alentours de 320 ajouts de vidéos (et donc suppression de vidéos) par jour. Nous avons donc environ 640 écritures d'entités (pour ajouter/supprimer les vidéos de la liste des vidéos de l'utilisateur). Encore une fois, nous sommes dans le quota gratuit.

Au final, le stockage des utilisateurs est gratuit (pour 1 000 utilisateurs).

Calcul de la taille maximale nécessaire pour stocker les vidéos

Dans nos estimations initiales, nous avons estimé que le trafic engendré par l'upload des vidéos est le suivant :

- En moyenne, l'ensemble des Noob publient **18 Go** de vidéos par semaine.
- Les Casual et Leet publient **72 Go** par semaine. (c.f. calculs ci-dessous)

Au total, 90 Go de vidéos passent par notre base de donnée au cours de la semaine.

Or, les vidéos étant supprimées après au maximum 30 min, nous estimons que le *pic* d'utilisateurs représente 50 % du trafic de la journée en 5 heures.

Les vidéos uploadées représente $90/7 = 12.86$ Go par jour. Au maximum, $12.86 * 50\% =$ **6.43 Go** d'espace disque nous permettent d'absorber la charge nécessaire pendant le pic d'utilisation. Or, avec cette solution de stockage, nous disposons de 5 Go gratuit.

L'espace de stockage de nos vidéos nous coûterait alors $(6.43-5) * \$0.026 =$ **0.04 \$** par mois.

A ce prix, nous devons ajouter le prix de la bande passante.

Le prix de l'insertion d'une vidéo est gratuite.

Dans notre estimation, les Noobs téléchargent 768 Go par semaine et les autres utilisateurs 512 Go par semaine, soit 1280 Go de téléchargement par semaine (soit **5 668.97 Go/mois**).
Le prix d'un Go téléchargé est de 0.11\$.

Nous avons donc : $5\,668.97 * 0.11 = \mathbf{680.16\$}$ par mois.

Nous avons donc, pour un mois : $0.04 + 680.16 = \mathbf{680.20 \$}$

Le stockage nous reviendrait donc à **0.68 \$ par utilisateur**.

En comptant l'estimation du coût de notre application qui est de **287.31 \$** par mois, nous avons un coût total par utilisateur par mois de **0.967 \$**

