

Cloud Computing

TOUTAIN Xavier – BENZA Amandine – HAJJI Amine - FORNALI Damien

Livraison 2 – 01/11/18

Architecture de la couche de stockage de données

1) Type de stockage utilisé

Nous utiliserons une base de données **NoSQL** offerte par le **Google Cloud Storage**. Contrairement aux bases de données relationnelles, les bases de données **NoSQL** sont plus flexibles et évolutives. Ce sont les caractéristiques qui nous intéressent pour ce projet.

Au niveau des méta-données que nous allons stocker, nous pensons important de stocker l'identifiant de l'utilisateur, son score, son rang, la date des derniers envois ainsi que les envois eux-mêmes (dans un autre espace de stockage).

Ces données ne nécessitent pas de cohérence particulière hormis la mise à jour de score ainsi que la date des derniers envois. La mise à jour du score serait faussée par une mauvaise cohérence (valeur présente dans la base de données invalide). De plus, la date des derniers envois est utilisée pour la conservation des documents. Nous pouvons cependant tenter de garder cette valeur cohérente à l'aide des **Ancestors** (introduits dans le dernier TP).

2) Coût du stockage par utilisateur

Afin d'établir un coût de stockage par utilisateur, nous allons nous référer à nos précédentes estimations.

Rappel des estimations précédentes:

- Échelle du nombre de requêtes
 - **8500** requêtes par heure (**75%** de téléchargements et **25%** d'envoi)
 - **2** requêtes par seconde
- Temps moyen de traitement d'une requête
 - **300 ms** (chargement + téléchargement non inclus)
- Nombre d'instances
 - deux instances pour avoir une marge de manœuvre (type: **B2**)
 - deux supplémentaires pour traiter les pics de demandes (type: **B2**)
- Nombre d'utilisateurs
 - **1000** au total avec 50% **Noob**, 35% **Casual** et 15% **Leet**
- Taille moyenne des fichiers traités
 - fichiers texte: 100KB
 - fichiers image: 3MB

Afin d'estimer nos coûts de stockage nous nous référons au calculateur de prix de **Google**. Voici ci-dessous notre estimation du poids total des fichiers stockés en simultané.

- Catégorie **Noob**:
Nous avons 500 utilisateurs ($1000 * 50\%$).
Calculons le poids des fichiers texte à stocker sur une heure:

- La répartition de nos fichiers est la suivante : 60 % pour les fichiers texte et 40% pour les images. Nous avons donc $500 * 60\% = 300$ fichiers d'une taille moyenne de **100KB**, ce qui nous une taille de **30 000KB**.
- Pour les images, nous avons $500 * 40\% = 200$ fichiers d'une taille moyenne de **3MB**, ce qui nous donne une taille de **600 000KB**.
- Ayant 25 % d'upload on ne garde que **150 000KB** par heure.
- Un fichier **Noob** n'étant conservé que 5 minutes on obtient le calcul suivant $(150\ 000 * 5) / 60$, où 60 est le nombre de minutes par heure. Cela nous donne **12,5MB** de stockage en simultané pour une requête par utilisateur. Au total cela nous donne $12,5 * 4 = \mathbf{50MB}$.
- Catégorie **Casual**:
Nous avons 350 utilisateurs ($1000 * 35\%$).
Calculons le poids des fichiers texte à stocker sur une heure:
 - La répartition de nos fichiers est la suivante : 60 % pour les fichiers texte et 40% pour les images. Nous avons donc $350 * 60\% = 210$ fichiers d'une taille moyenne de **100KB**, ce qui nous une taille de **21 000KB**.
 - Pour les images, nous avons $350 * 40\% = 140$ fichiers d'une taille moyenne de **3MB**, ce qui nous donne une taille de **420 000KB**.
 - Ayant 25 % d'upload on ne garde que **110 250KB** par heure.
 - Un fichier **Casual** n'étant conservé que 10 minutes on obtient le calcul suivant $(110\ 250 * 10) / 60$, où 60 est le nombre de minutes par heure. Cela nous donne **18,3** de stockage en simultané pour une requête par utilisateur. Au total cela nous donne $18,3 * 10 = \mathbf{183MB}$.
- Catégorie **Leet**:
Nous avons 150 utilisateurs ($1000 * 15\%$).
Calculons le poids des fichiers texte à stocker sur une heure:
 - La répartition de nos fichiers est la suivante : 60 % pour les fichiers texte et 40% pour les images. Nous avons donc $150 * 60\% = 90$ fichiers d'une taille moyenne de **100KB**, ce qui nous une taille de **9000 KB**.
 - Pour les images, nous avons $150 * 40\% = 60$ fichiers d'une taille moyenne de **3MB**, ce qui nous donne une taille de **180 000KB**.
 - Ayant 25 % d'upload on ne garde que **47 250KB** par heure.
 - Un fichier **Casual** n'étant conservé que 10 minutes on obtient le calcul suivant $(47\ 250 * 30) / 60$, où 60 est le nombre de minutes par heure. Cela nous donne **23,6MB** de stockage en simultané pour une requête par utilisateur. Au total cela nous donne $23,6 * 20 = \mathbf{472,5MB}$.

Le stockage toute catégorie confondue est donc égal à $50 + 183 + 473 = \mathbf{606MB}$.

De plus, pour pouvoir gérer les métadonnées ainsi que prévoir une hausse importante de trafic il nous semble être raisonnable de prévoir le double de ce stockage (**1,2GB**).

Nous avons par mois 6 120 000 requêtes, ce qui nous donne 4 590 000 téléchargements ainsi que 1 530 000 envois.

Avec l'ensemble de ces données, à l'aide de la plateforme de calcul de prix de **Google**, nous obtenons un coût de **9,30 dollars par mois**.