



UNIVERSITÉ LIBRE DE BRUXELLES

Rapport Analyse

[INFO-F309]

JACOBS Alexandre ENGELMAN David ENGELMAN Benjamin BONAERT Gregory NOGUERIA CABAÇO Pedro Filipe
BA3 Informatique

Novembre 2017

Table des matières

1	Problématique	2
2	Solution	2
3	<i>File Storage</i> vs <i>Object Storage</i>	2
4	Comparaison de Swift et Ceph	2
4.1	Performance	2
4.2	Scalability	3
4.3	Facilité d'utilisation et d'installation	3
4.4	Choix final : Ceph	3
5	Travail à mettre en oeuvre	3
5.1	Planning	3
5.2	Matériel	3

1 Problématique

Facebook devient de plus en plus populaire et commence à ne plus savoir comment stocker ses données. L'architecture n'étant plus capable de gérer un volume de données gigantesque et grandissant chaque jour, notre objectif est de trouver une solution qui perdurera et qui sera capable de gérer et stocker les millions de photos et vidéos partagées par des dizaines de millions d'utilisateurs. Les anciennes données devront être transférées vers la nouvelle architecture et la migration ne pourra pas engendrer de downtime. Le nouveau système devra effectuer des copies dans différents data centers afin de s'assurer qu'aucune donnée ne soit perdue et que le système résiste aux défaillances de disques durs ou aux pannes. Il devra également être facilement extensible pour faire face au grand nombre d'ajouts de photos et vidéos par jour.

2 Solution

Pour résoudre ce problème, nous allons mettre en place un système de stockage. Ce système répartit une large quantité de données, dans notre cas les images et vidéos des utilisateurs de facebook, sur plusieurs serveurs (disques durs). Cela permet de pouvoir stocker une quantité importante de données et aussi d'en assurer la redondance et la disponibilité grâce à la réplication.

Initialement, nous nous étions intéressés à 3 solutions basées sur des fichiers :

1. Hadoop File System (HDFS)
2. Ceph
3. Gluster

Après analyse, nous avons rapidement éliminé HDFS car ce système est plus adapté au stockage d'un petit nombre de gros fichiers. Dans notre cas nous avons un nombre important de fichiers relativement petits, donc HDFS ne correspond pas à nos besoins.

Nous nous sommes alors intéressé à la manière avec laquelle Ceph et Gluster gèrent les données. Gluster utilise un système dit de *File Storage* et Ceph un système d'*Object Storage*.

3 *File Storage* vs *Object Storage*

Il existe plusieurs systèmes de stockage. Nous nous sommes concentrés sur deux d'entre eux :

1. *File Storage* : c'est le système classique que tout le monde connaît. On a des dossiers dans des dossiers jusqu'à arriver dans un fichier. Il y a donc une hiérarchie. Ce système commence à montrer ses faiblesses quand on commence à avoir beaucoup de fichiers (des millions).
2. *Object Storage* : on stocke les données dans des "objets" (similaires à des fichiers) mais il n'y a pas de notion de hiérarchie (tous les objets sont au même niveau). Chaque objet a un identifiant unique qui lui est attribué.

Les avantages du *Object Storage System* sont :

1. La performance : Très rapide de trouver/accéder à un objet.
2. La *Scalability*¹ : De plus, cette performance n'est pas dégradée avec la croissance continue de données.

Nous avons décidé de comparer 2 systèmes de stockage d'objets répandus (Ceph et Swift) afin de prendre une décision finale.

4 Comparaison de Swift et Ceph

4.1 Performance

Ceph et Swift possèdent 2 architectures différentes, ce qui influence leur performance.

Ceph est plus performant pour lire les données.

1. La capacité d'un système à s'agrandir

Ceph réplique immédiatement l'information, ce qui lui permet de garantir une meilleure fiabilité mais ajoute un peu de délai dans l'écriture, car certains data centers sont très éloignés. Néanmoins, nos data centers sont reliés par de la fibre optique, donc la latence n'est pas très élevée.

Swift n'écrit qu'une seule copie immédiatement, les autres étant écrites après. Ainsi, il peut être plus rapide pour réaliser des opérations d'écriture dans le cas où une donnée doit être écrite sur 2 serveurs éloignés.

Comme un utilisateur consulte en moyenne beaucoup plus de photos et vidéos qu'il n'en ajoute, la vaste majorité des opérations sont des lectures. On préfère donc optimiser la lecture et ce n'est pas un problème si l'écriture est un peu plus lente.

4.2 Scalability

Lorsque le nombre de clients augmente, la performance de Swift a tendance à se dégrader de manière plus significative que Ceph.

4.3 Facilité d'utilisation et d'installation

L'installation et la configuration de Ceph sont plus faciles à réaliser et moins complexe que celles de Swift. Ceci nous fera gagner du temps lors de la mise en place du nouveau système.

4.4 Choix final : Ceph

En nous basant sur la comparaison ci-dessus, nous avons opté pour Ceph parce qu'il permet d'avoir des performances excellentes tout en étant plus facile à installer et configurer.

5 Travail à mettre en oeuvre

5.1 Planning

Pour arriver à mettre en oeuvre la nouvelle architecture, nous devons donc passer par les différentes phases de développement suivantes :

1. Achat matériel - mi-Novembre 2006
2. Mise en place du matériel & Configuration du matériel - mi-Décembre 2006
3. Installation et configuration de Ceph sur les serveurs - mi-Décembre à fin Décembre 2006 (environ 1-2 semaines)
4. Testing & Correction architecture - Janvier à mi-Février 2007 (environ 5 semaines)
5. Transfert des données par tranche de 15 % des données totales existantes & analyse des données transférées & correction des bugs éventuels - mi-Février à début Mars 2007 (environ 2-3 semaines)

5.2 Matériel

En ce qui concerne le matériel, nous avons choisi de réutiliser celui utilisé pour l'ancien système de stockage mais pour faire face à l'augmentation rapide de la charge de photos et vidéos nous avons décidé de faire l'acquisition de 5% de matériel supplémentaire.

Au total, nous aurons 667 serveurs gérant chacun 12 téra octets de données utilisateurs, au total 8,004 péta octets de données utilisateurs (2001 disques durs de 4 téra octets).

Ceph nécessitant une distribution linux pour fonctionner, tous les serveurs tourneront sous Debian, celle-ci étant une distribution stable et fortement utilisée.