

Bases de données
documentaires et distribuées,
<http://b3d.bdpedia.fr>

Scalabilité

Une définition (stricte) de la scalabilité

Définition

Un **système** est **scalable** si ses **performances** sont proportionnelles aux **ressources** qui lui sont allouées.

où

- le **système** est un système distribué de gestion de données ;
- les **ressources** sont les composants matériels (serveurs, connectique) ;
- la **performance** est mesurée
 - soit par la **latence** pour les systèmes temps réel (temps d'accès à un document) ;
 - soit par le **débit** pour les systèmes analytiques (temps de parcours d'une collection).

On quantifie les grandeurs "ressource" et "performance" ; on montre que leur rapport est constant pour un volume de données fixé.

Premier exemple : comptage des vidéos

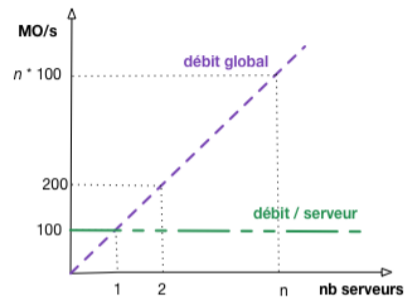
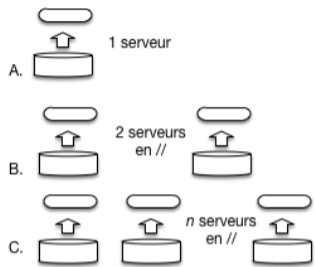
On veut compter les vidéos dans la collection.

- Avec un serveur : on lit la collection sur le disque, on compte : 100 MO/s
- Avec deux serveurs : on répartit la collection en deux parties égales, chaque serveur lit sur son disque : 200 MO/s
- avec n serveurs : on répartit en n , on lit en parallèle : $n \times 100$ MO/s

À la fin, le ou les serveurs envoie(nt) leur compte à l'application qui cumule (coût 0 ou presque).

Rapport performance/ressources = (débit global / nb serveurs) = 100 MO/s!

La méthode



Important

Le fait que le temps soit constant par serveur (courbe verte) est une condition nécessaire mais pas suffisante !

Si j'envoie **toutes** les vidéos au client, le coût réseau (500 TOs!) devient prépondérant.

Second exemple : on cherche des doublons

Comment savoir si on a des vidéos en double (des **contenus** identiques) ?

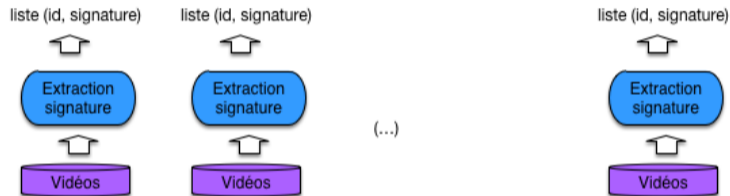
On procède en deux étapes

- parcours des vidéos, calcul d'une **signature** ; on produit la liste des paires (i, s) , i id de la vidéo, s signature ;
- regroupement des paires par signature : si deux signatures (ou plus) égales sont trouvées, c'est un doublon !

Et on envoie au client la liste des doublons.

Etape 1

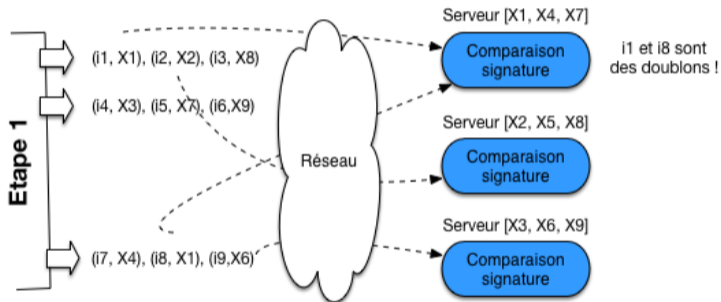
La première étape se déroule entièrement en parallèle sur les serveurs.



Scalable, pour les mêmes raisons que l'exemple 1.

Etape 2

La seconde étape implique des échanges réseaux : on alloue à chaque serveur un sous-ensemble des signatures, et on lui transmet les paires de la liste pour ces signatures.



Scalable? Peut-on augmenter linéairement le débit d'un réseau ?



Analyse

On échange sur le réseau un paire (i, s) pour chaque document.

La taille est **totalemment négligeable** par rapport à celle du document lui-même.

Conclusion : **le coût des accès disques est largement prépondérant ; la méthode est scalable (pas parfaitement...)**

A vérifier

Il faudrait **quantifier** les coûts respectifs des accès disques et réseau (cf. la suite)

Quelques conclusions générales

Quelques conditions favorables (voire indispensables)

- Données équitablement réparties.
- Traitement essentiellement **local**.
- Limitation des transferts réseaux, entre les serveurs et vers le client.
- Et un temps de calcul global qui reste raisonnable pour les ressources qu'on peut/veut lui allouer...

Scalable mais pas acceptable ?

Le fait d'être scalable (selon notre définition) ne veut pas dire que le temps de calcul est **acceptable**. Si un traitement prend 10 ans, il prendra encore 5 ans en doublant les ressources...