

Khadija BEN ALI - Pascal MANDENDE - Mickaël GOGUELIN

PROJET

ALGORITHMES PARALLELES ET DISTRIBUES

Rapport n°1

Sujet: Implémenter en MPI un système de gestion de fichiers distribué partagé simplifié. Les primitives PUT fichier et GET fichier sont à implémenter. On peut imaginer une source unique de lectures/écritures mais plusieurs machines de stockage. Possible répartition de charge à la round-robin (tourniquet). Attention: commencer par définir des scénarios types.

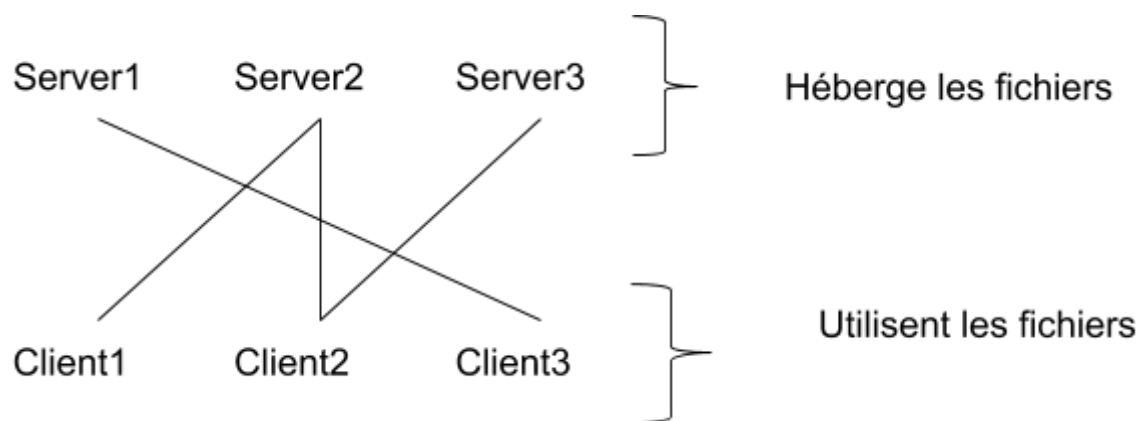
Pour commencer, quelques définitions et recherches concernant différents termes du sujet pouvant nous aider à sa compréhension :

Système de gestion de fichiers distribué et partagé:

Ce système permet une vision logique du stockage physique : plusieurs machines servent au stockage mais on ne voit de manière logicielle qu'une seule machine. La notion de "partagé" ajoute la possibilité d'accès au stockage à distance.

Exemples : NFS, AFS (dérivé de NFS), CXFS \approx , VFS...

AFS:



AFS se compose de cellule. Chaque cellule représente une zone d'autorité indépendante.

Au niveau des clients: Ces cellules sont regroupées au sein de la hiérarchie AFS (/afs/) => qui peut être intégrée dans le système de fichier d'un client grâce à VFS.

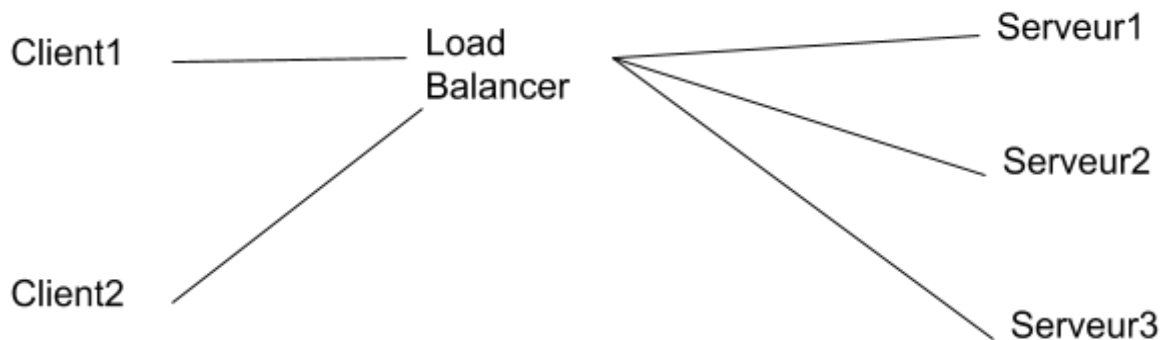
Fonctionnalités de AFS:

- Répartition de volume sur plusieurs serveurs
- Sécurité: technique d'authentification mutuelle par le biais de Kerberos. Ceci afin d'offrir un accès simplifié à un utilisateur distant après acquisition d'un token (jeton) après authentification.
- Gestion de caches de fichiers: les fichiers peuvent être mis en cache localement.

Round-robin (tourniquet) : algorithme de répartition de charge :

- on dispose d'un pool de serveurs
- une tâche attribuée à chaque serveur, l'un après l'autre, jusqu'au dernier puis on recommence.

On retrouve cette notion dans les architectures client serveur avec un load balancer.



Définition du périmètre du sujet:

- Mettre en place un système de gestion de fichiers distribué
- L'existence d'une et une seule source capable d'écrire ou de lire sur les fichiers. Si on se limite dans notre projet à une unique source ayant les droits d'écriture et de lecture sur notre SFD, alors il ne sera pas nécessaire d'implémenter la gestion d'accès concurrentes aux ressources (ici les fichiers).
- Le choix d'avoir plusieurs machines de stockage nous est suggéré, c'est pourquoi notre système sera distribué. Ceci nous obligent à nous poser quelques questions et à y répondre:
 - Un noeud peut être à la fois client (source) et serveur à la fois?
Nous disposons d'un pool de machine. Chacune d'elle est serveur de stockage et le client peut se connecter sur n'importe laquelle de ces machines et les répercussions des modifications se font au fur et à mesure. Par exemple, pour la recherche de fichier, on la décompose sous forme de tâches attribuées à chacune des machines les après les autres (se basant par conséquent sur le principe round robin).
 - Tous les serveurs de stockage contiennent-ils l'ensemble des données? Si oui alors il faut implémenter un protocole de réplication des données. Sinon comment répartir la totalité des données sur les serveurs de stockage? Est-ce possible d'accéder à la totalité des données lors d'une panne d'un des serveurs de stockage? Ceci pose également un problème au niveau de la répartition de charges (load balancing) qui ne peut pas vraiment être implémentée si on considère que les serveurs de stockage contiennent chacun des données

différentes. Comment faire de load balancing si la source accède toujours aux fichiers présent dans le même serveur de stockage?

- Authentification: Doit-on mettre en place un système d'authentification? login+mdp?
- Droits: Doit-on gérer les droits d'accès dans notre système de gestion de fichiers? La question se pose mais vu qu'on a qu'une seule source de lecture et d'écriture mais si nous en faisons plusieurs, ceci devient nécessaire. Dans un premier temps nous pouvons développer le système pour un utilisateur et ensuite ajouter cette notion de plusieurs utilisateurs.

Primitives file/dir:

- PUT
- GET
- DELETE
- MOVE (si on a une arborescence)
- REN : renommage de fichier.
- D'autres primitives peuvent apparaître dans le cas où nous laissons plusieurs utilisateurs.

Quelques scénarii

- Un client souhaite virtualiser son espace de stockage (fichiers avec arborescence).
Pour cela, il souhaite :
 - pousser son fichier ou rep en choisissant où le mettre dans son arborescence
 - récupérer son fichier ou rep dans un répertoire cible
 - supprimer - renommer - supprimer au sein de l'arborescence virtualisée
 - Protéger le contenu par un login et pw
- Comment réagir lors de la création ou la modification d'un fichier?
- Déclencher toute de suite la réplication du fichier concerné sur tous les serveurs de stockage? Combien de messages sont générés? combien de temps prend cette réplication?
- Doit-on mettre en place un système de priorité dans la réplication des données? (Du genre commencer par répliquer les données dans le serveur de stockage successeur du serveur de stockage courant dans la liste ou tourniquet)

-Que se passe t-il lorsqu'on accède à un fichier dans un serveur de stockage où il n'est pas encore répliqué? Peut- on détecter le fait que le fichier n'est pas encore répliqué mais il le sera?

-L'utilisateur aura-t-il la possibilité de chercher un fichier dans la système de fichiers?

Comment va se comporter le système lors de la suppression d'un fichier?

- Le fichier sera recherché puis supprimé (sur tous les serveurs si nous choisissons une copie de l'ensemble des serveurs ou non)

-Copie des fichiers

-Déplacement de fichiers

-Gestion des alias et de liens

-Gestion des noms de fichiers (Que faire lorsque 2 fichiers ont le mm nom?)

-Mise en place de verrou sur les fichiers en cours d'utilisation? Pas forcément nécessaire si on n'a qu'une unique source.

-Lister les fichiers d'un répertoire

Propriétés d'un fichier?

-Taille

(-Droits)

-Propriétaire

-Date de création

-Date de la dernière modification?

Webographie

https://perso.limsi.fr/sabouret/archives-info32b/cours/cours_12_fr.pdf

<https://tel.archives-ouvertes.fr/tel-00004373/document>

<https://technet.microsoft.com/fr-fr/library/bb967408.aspx>