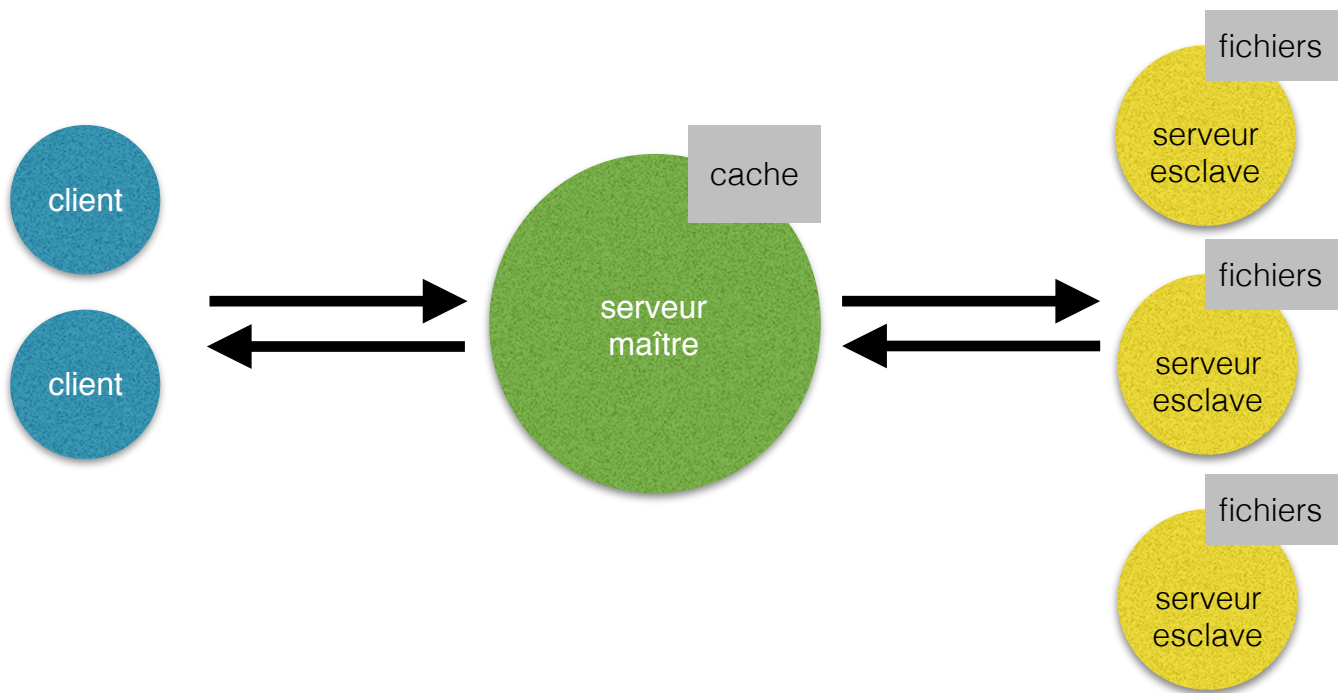


Serveur de fichiers distribué

On se propose de réaliser un système de serveur de fichiers qui repose sur trois composants :

- une **partie cliente** (autant de client souhaité par l'utilisateur) qui est utilisée pour interroger le système afin de retrouver un fichier
- une **partie serveur maître** (toujours en un seul exemplaire) qui est le partenaire privilégié de la partie cliente, qui connaît la localisation des fichiers
- une **partie serveur esclave** (autant d'esclaves que l'on veut) qui possède les fichiers

Le schéma synthétique suivant résume la situation :



Dans la suite, toutes les communications décrites s'effectuent en mode flot connecté (TCP).

La communication du point de vue du client

Cette communication s'effectue depuis n'importe quel port local vers le port 50000 du serveur maître.

Pour obtenir un fichier le client envoie au serveur maître une requête indiquant quel genre de fichier il désire (critères). En réponse, le serveur maître lui envoie une liste d'identifiants de fichiers correspondants aux critères ou une réponse signifiant l'échec de la requête (auquel cas, la communication s'arrête là). Après avoir reçu la liste, le client peut alors réclamer un fichier en spécifiant l'identifiant de celui qui l'intéresse, la réponse du serveur est alors l'envoi du contenu du fichier (ou une réponse signifiant l'échec en cas de problème), puis la communication s'arrête.

La communication du point de vue de l'esclave

Cette communication s'effectue depuis n'importe quel port local vers le port 50001 du serveur maître.

Un serveur esclave doit préalablement s'enregistrer auprès du serveur maître. Pour cela, il s'y connecte et envoie des données d'identification ainsi qu'une liste de fichiers (locaux) sous la forme d'une table constituée d'entrées contenant trois critères et un identifiant unique (local), de sorte que le serveur puisse peupler sa propre table globale d'indexation.

Ensuite le serveur esclave coupe la communication et passe dans un mode d'attente de connexion (sur un port de numéro >50001); c'est à l'initiative du serveur maître d'établir cette connexion lorsqu'il a besoin de récupérer un fichier réclamé par un client, auquel cas, le maître envoie l'identification locale du fichier et reçoit en réponse le contenu du fichier et la communication s'arrête.

Description des messages échangés

Attention les caractères [et] ne font pas partie des messages, ils permettent de délimiter clairement et visuellement le contenu réel des messages.

Requête de fichier de la part du client :

[**WANT** *C1, C2, C3*\n]

Réponses possibles du serveur maître :

[**NOSUCHFILE**]

[**FILES** *ID1, ID2, ID3, ..., IDN*\n]

Requête d'un fichier par son identifiant :

[**NEED** *ID*\n]

Réponses possibles du maître :

[**NOSUCHFILE**\n]

[**CONTENT** *S*\n...]

Notes :

- *C1, C2* et *C3* sont des critères sous la forme de suite de caractères ne contenant pas de caractère d'espacement, *i.e.* **film** ou **texte** ou **code** sont acceptables mais pas **le film**.
- *ID* sont des numéros codés sous la forme de chaînes de caractères et ne contenant que des chiffres décimaux, *i.e.* **32198763298562** est un identifiant valable.
- *S* est un nombre toujours codé sur 10 caractères (avec des zéros en tête si nécessaire) et qui représente la taille du fichier qui est ensuite envoyé tel quel. Ce qui suit le retour à la ligne dans le message **CONTENT** est donc la suite des *S* octets formant le contenu du fichier.

Demande d'enregistrement de la part d'un esclave :

[**SLAVE** *Pn*]

Réponses possibles du maître :

[**OK**\n]

[**BAD**\n]

Envoi des indexes depuis l'esclave :

[**INDEX** *C1, C2, C3, ID, C1, C2, C3, ID, ...*\n]

Réponses possibles du serveur :

[**OK**\n]

[**BAD**\n]

Demande de fichier de la part du serveur vers l'esclave :

[**NEED** *ID*\n]

Réponses possible de l'esclave :

[**NOSUCHFILE**\n]

[**CONTENT** *S*\n...]

Note :

- *P* désigne le numéro de port sur lequel l'esclave attendra les demandes fichiers en provenance du maître, c'est un numéro codé sous la forme d'une chaîne de caractères ne contenant que des chiffres.

Réalisation

Il est demandé :

- de réaliser les trois composants en **langage C sous Unix**
- de **documenter le code**
- de documenter le projet par un **manuel d'utilisateur** pour chaque commande
- de fournir un **rapport** expliquant les choix opérés. Certaines parties sont libres comme par exemple où sont stockés les fichiers du côté esclave ? Comment les identifiants sont-ils rendus uniques dans le maître ? Existence et fonctionnement d'un cache dans le maître ? etc. Attention, tout choix doit s'avérer raisonnable et doit pouvoir être défendu.

Notes :

- s'il n'est pas interdit de communiquer entre équipes-projet, cela pourrait même être très bénéfique si des équipes arrivaient à interchanger leurs composants une fois ceux-ci réalisés (ou même se proposent de coopérer de sorte que certains s'occupent d'une partie pendant que d'autres d'une autre, testent l'interopérabilité puis continuent ainsi), il est **formellement interdit de s'échanger du code**.
- on remarquera qu'il est tout à fait possible, **dans un premier temps**, de ne s'occuper que du client et du maître (celui-ci peut se contenter d'envoyer des faux fichiers). La réalisation peut donc être incrémentale.
- **attention**, malgré son apparente simplicité des détails techniques se cachent un peut partout dans le sujet. Il est donc nécessaire d'y réfléchir posément et sans coder en essayant d'imaginer les différentes situations. Ne vous laissez pas piéger par le temps...