

L3 Informatique - Module Système et réseaux
Projet



Système de fichiers distribués tolérant aux pannes

Première partie
Rapport d'analyse

Maréchal Anthony et Benoit

Table des matières

1	Introduction	3
2	Réflexion sur les différentes stratégies possibles	3
2.1	Le fichier de correspondance	3
2.2	Stratégie 1 : Un Serveur central réalise la maintenance du fichier de correspondance et la répartition des fichiers sur le réseau	4
2.3	Stratégie 2 : Un Serveur central effectue la maintenance du fichier de correspondance, les clients s'occupent de la répartition des fichiers eux-mêmes	5
2.4	Stratégie 3 : Les clients gèrent le fichier de correspondance à tour de rôle	7
3	La solution retenue	9
3.1	Justification du choix de la stratégie	9
3.2	Détail de la solution	9
4	Aller plus loin	10
5	Conclusion	10

1 Introduction

Dans le cadre du Module de Système et Réseau qui aborde les différents concepts des systèmes d'exploitations, notamment par une initiation au système Unix, et à la programmation réseaux, nous avons la responsabilité de réaliser un projet de *Système de fichiers distribués tolérant aux pannes*.

Les possibilités qui nous sont offertes pour accomplir ce projet sont nombreuses c'est pourquoi nous devons réaliser un rapport d'analyse, dont l'objectif est de nous faire réfléchir sur les différentes stratégies, d'en choisir une et de justifier ce choix par rapport aux autres stratégies possibles.

Faisons tout d'abord un rapide rappel du sujet qui nous permettra de définir le vocabulaire employé dans la suite du rapport. Le but de ce projet est de concevoir un système de fichiers distribué tolérant aux pannes. Ce système permettra à l'utilisateur de sauvegarder ses fichiers importants de manière sûre car ils seront copiés sur plusieurs machines du réseau. Il sera conçu grâce à un espace disque partagé et mutualisé sur chacune des machines participantes au système. Pour désigner l'ensemble des fichiers qui auront été choisis comme "importants" par les utilisateurs, et qui auront donc été sauvegardés par notre application nous utiliserons l'expression "espace virtuel". Les utilisateurs pourront à tout moment lire, écrire, ou sauvegarder de nouveaux fichiers sur cet espace virtuel ; à charge pour l'application de procéder aux traitements nécessaire sur les espaces partagés du réseau. Lorsqu'une machine tombe en panne, le programme devra recopier les fichiers de l'espace virtuel qui étaient hébergés sur cette machine sur une autre machine du réseau, afin de maintenir un nombre de copies des fichiers sur les machines du réseau estimé suffisant (ce nombre sera désigné par la lettre 'k' dans la suite du rapport).

Ce projet nous permettra de maîtriser les concepts d'opérations sur les fichiers et répertoires, de communications inter-systèmes et interprocessus et d'accès concurrents.

Nous verrons dans un premier temps notre réflexion sur les différentes stratégies d'implémentation possibles, puis nous verrons quelle solution nous avons retenue en justifiant ce choix, enfin nous définirons quelles sont les grandes lignes qu'il nous reste à approfondir et sur quels axes nous devons encore travailler pour améliorer le système.

2 Réflexion sur les différentes stratégies possibles

Les possibilités d'implémentation du projet sont très nombreuses et cette partie ne se prétend pas être une présentation exhaustive de toutes les stratégies qui s'offre à nous. Nous présentons, ici, trois stratégies qui nous semblent être les plus réalisables par rapport à nos connaissances sommes toutes assez récente en système et réseaux.

2.1 Le fichier de correspondance

Afin de savoir que tel fichier à été sauvegardé sur tel machine, il faut dans un premier temps définir un fichier de correspondance. Fichier indispensable à l'application, il représente le lien entre l'espace virtuel et les espaces partagés des ordinateurs clients participant au système. Il contient une liste de fichiers associés à une liste d'adresse ip correspondant aux ordinateurs client hébergeant le fichier. Il permet l'abstraction des espaces partagés physiques où se trouvent les fichiers, en un espace virtuel rassemblant tous ces fichiers sauvegardés sur le réseau. On peut donc retrouver n'importe quel fichier de l'espace virtuel, physiquement, sur le réseau.

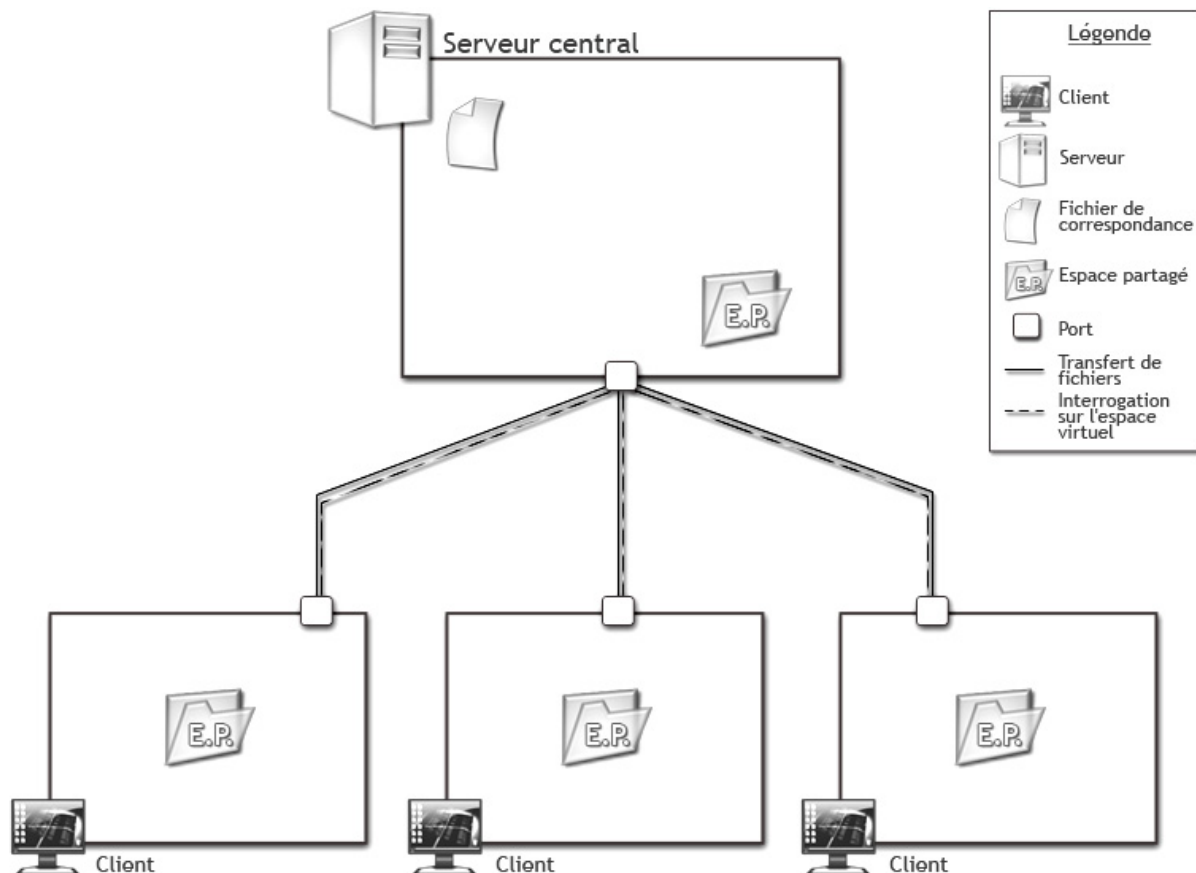
Représentation du fichier de correspondance :

Nom du fichier	Client hébergeant le fichier
fichier1.pdf	192.168.0.2 ; 192.168.0.5
fichier2.tex	192.168.0.11 ; 192.168.0.5

Remarque : La syntaxe exacte de ce fichier reste encore à définir. Une partie du système aura donc pour unique but la maintenance et la mise à jour de ce fichier. Voyons maintenant les stratégies possibles.

2.2 Stratégie 1 : Un Serveur central réalise la maintenance du fichier de correspondance et la répartition des fichiers sur le réseau

Dans cette stratégie on désigne une machine du réseau comme étant le *Serveur central* du système. Cette machine aura pour devoir la maintenance du fichier de correspondance ainsi que l'envoi ou la récupération d'un fichier appartenant à l'espace virtuel. Lorsqu'un ordinateur client voudra voir l'espace virtuel, récupérer ou sauvegarder un fichier, il devra demander au serveur de réaliser ces opérations pour lui. Le *Serveur central* agit donc comme un exécutant face aux ordres donnés par les clients.



Scénario type :

- Un Client se connecte au *Serveur central* (1 transaction)
- Le *Serveur central* lui affiche l'espace virtuel (1 transaction)
- Le Client demande à travailler sur un fichier de l'espace virtuel (1 transaction)
- Le Serveur récupère le fichier puis l'envoie au Client (3 transactions dont 2 envoient du fichier)
- Le Client travaille sur le fichier et demande à ce qu'il soit à nouveau sauvegardé (1 envoi du fichier)
- Le *Serveur central* récupère le fichier ainsi modifié, met à jour le fichier de correspondance et envoie le fichier sur les machines possédant son ancienne version, en l'écrasant ($2*k$ transactions dont k transactions sont des envois du fichier)

Avantages :

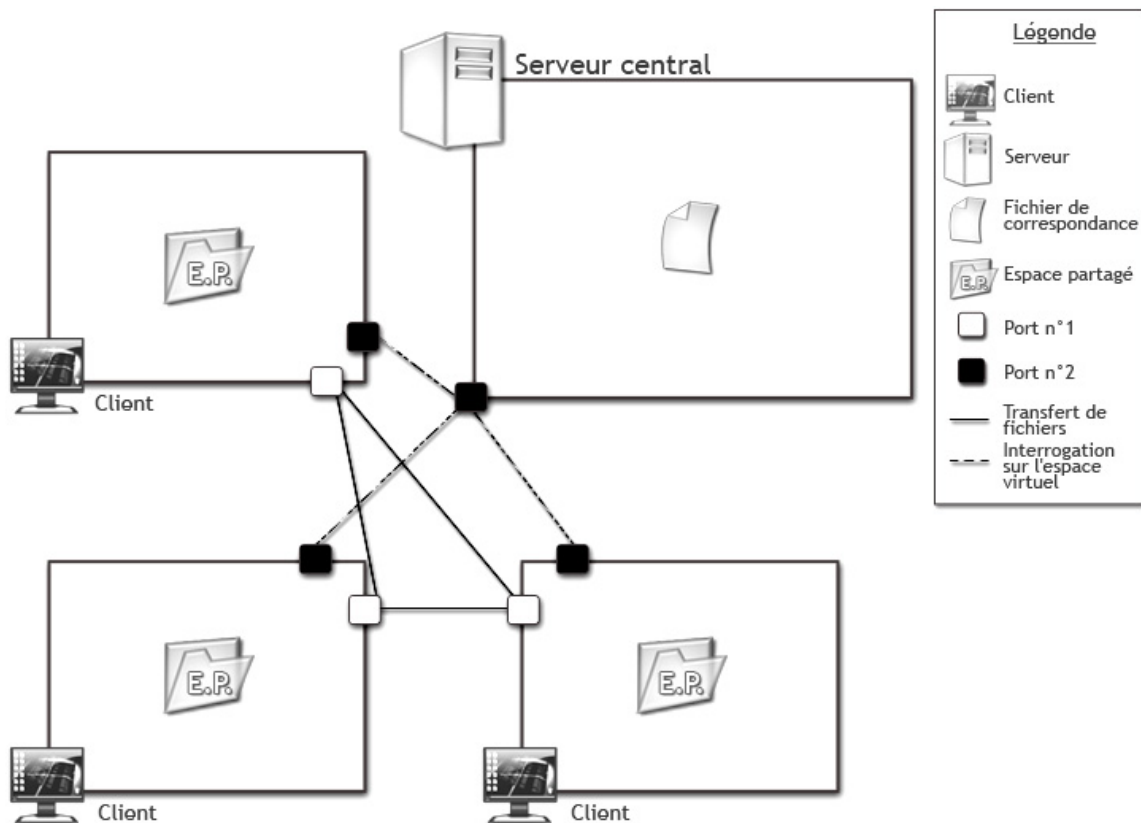
- Simplicité de mise en oeuvre du système
- Les ordinateurs client n'ont pas besoin d'intervenir dans la gestion du système

Inconvénients :

- Si le *Serveur central* tombe en panne le système ne fonctionne plus
- Grande activité sur le réseau, notamment pour le *Serveur central* (le scénario type comprend un minimum de $(6 + 2*k)$ transactions sur le réseau dont $(3+k)$ sont des envois du fichier)

2.3 Stratégie 2 : Un Serveur central effectue la maintenance du fichier de correspondance, les clients s'occupent de la répartition des fichiers eux-mêmes

Cette stratégie est une variante de la précédente, cette fois le *Serveur central* ne s'occupe que du fichier de correspondance et c'est au client de se connecter sur les autres machines du réseau pour récupérer ou sauvegarder ses fichiers.



Scénario type :

- Un Client se connecte au *Serveur central* (1 transaction)
- Le *Serveur central* lui affiche l'espace virtuel (1 transaction)
- Le Client demande à travailler sur un fichier de l'espace virtuel (1 transaction)
- Le *Serveur central* lui envoie les ip des ordinateurs hébergeant ce fichier (1 transaction)
- Le Client récupère le fichier en se connectant sur le bon ordinateur client (2 transactions dont 1 envoi de fichier)
- Le Client travaille sur le fichier puis indique au *Serveur central* quel fichier il a modifié (1 transaction)
- Le Client sauvegarde la nouvelle version sur les ordinateurs hébergeant l'ancienne version ($2*k+1$ transactions dont k transactions sont des envois du fichier)
- Le Client demande à travailler sur un fichier de l'espace virtuel (1 transaction)

Remarque : La mise à jour du fichier de correspondance n'est pas obligatoire ici puisqu'il s'agit d'une simple modification d'un fichier de l'espace virtuel.

Avantages :

- La machine désignée comme *Serveur central* est déchargée des transfères de fichiers.
- On a diminué de 2 transactions l'envoi de fichiers sur le réseau (le scénario type comprend un minimum de $(7+2*k)$ transactions sur le réseau dont $(1+k)$ sont des envois du fichier).

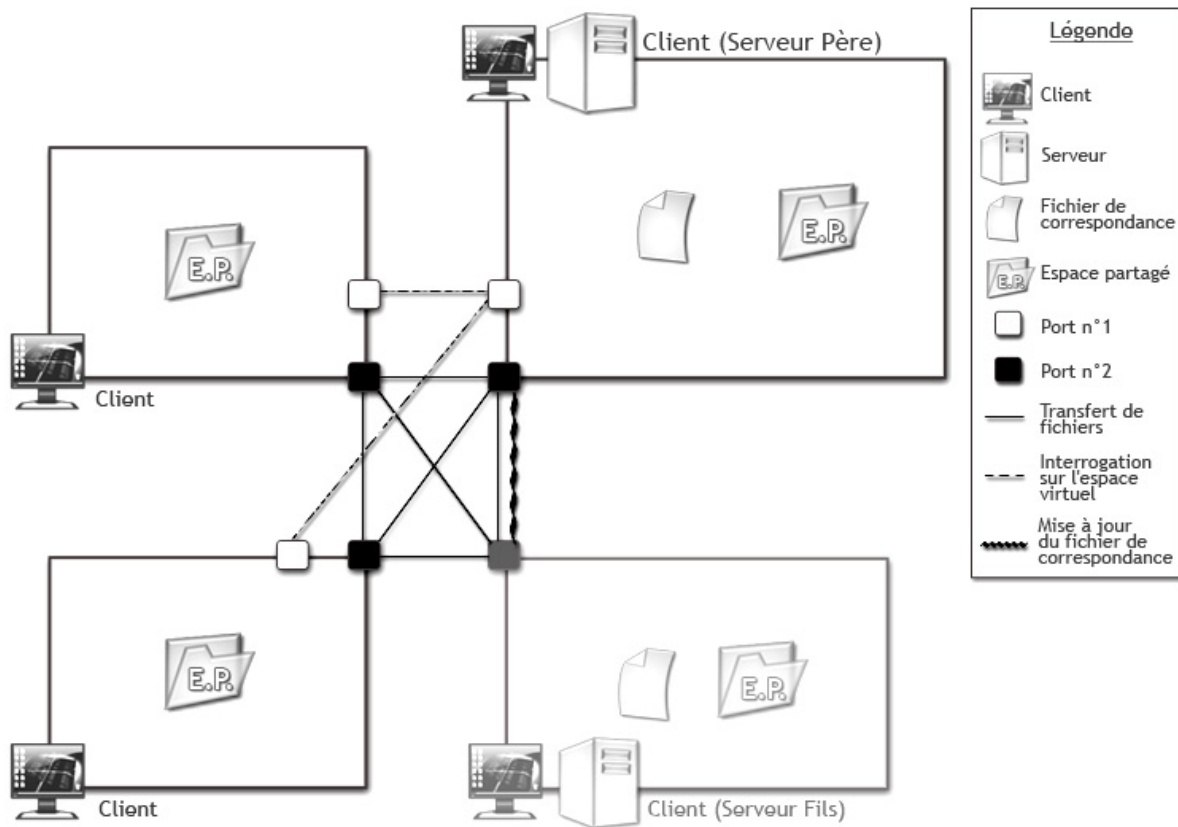
Inconvénients :

- Si le *Serveur central* tombe en panne le système ne fonctionne plus

2.4 Stratégie 3 : Les clients gèrent le fichier de correspondance à tour de rôle

De loin la plus complexe, cette stratégie, comme elle était indiquée dans le sujet, permet de se passer d'un *Serveur central*. Elle reprend le principe de la stratégie 2 en ce qui concerne la gestion des transferts de fichiers directement entre client, avec cette fois la particularité que les clients gèrent également la maintenance du fichier de correspondance. Bien sur l'application aura toujours besoin d'un serveur, mais celui-ci pourra changer d'une machine à l'autre selon les connexions et déconnexions des clients. Ainsi, à tout moment le système possède deux ordinateurs Serveur : Un *Serveur Père* et un *Serveur Fils*.

Ces deux serveurs maintiennent à jour le fichier de correspondance (tout comme le faisait le *Serveur central* de la stratégie 2). Le *Serveur Père* est un serveur actif, c'est sur lui que se connecteront les clients pour connaître l'espace virtuel. Par opposition le *Serveur Fils*, est un serveur inactif qui ne sert qu'à posséder une copie du fichier de correspondance ; il sera le serveur actif lorsque le *Serveur Père* se déconnectera du système. Il deviendra donc le *Serveur Père* et désignera un *Serveur Fils*. Dès lors, il n'y a plus besoin de *Serveur central*.



Scénario type :

- Un Client se connecte au *Serveur Père* (1 transaction)
- Le *Serveur Père* lui affiche l'espace virtuel (1 transaction)
- Le Client demande à travailler sur un fichier de l'espace virtuel (1 transaction)
- Le *Serveur Père* lui envoie les ip des ordinateurs hébergeant ce fichier (1 transaction)
- Le Client récupère le fichier en se connectant sur le bon ordinateur client (2 transactions dont 1 envoi de fichier)
- Le Client travaille sur le fichier puis dit au *Serveur Père* qu'il a modifié ce fichier (1 transaction)
- Le *Serveur Père* met également à jour le fichier de correspondance du *Serveur Fils* (1 transaction)
- Le Client sauvegarde la nouvelle version sur les ordinateurs hébergeant l'ancienne version (2*k transactions dont k transactions sont des envois du fichier)

Remarque : La mise à jour du fichier de correspondance n'est pas obligatoire ici puisqu'il s'agit d'une simple modification d'un fichier de l'espace virtuel.

Avantages :

- Pas de *Serveur central*, ce système est parfaitement tolérant aux pannes de n'importe quelle machine du réseau.

- Activité sur le réseau de $(8+2*k)$ transactions mais comme pour la stratégie 2 seulement $(1+k)$ transactions sont des transferts de fichiers.

Inconvénients :

- Complexité de mise en œuvre.
- Tous les clients, même les moins performants peuvent devenir *Serveur Père*.

Ces stratégies peuvent être vues comme incrémentales, chacune est l'amélioration de la précédente. Voyons maintenant quelle stratégie nous avons choisi d'implémenter pour ce projet.

3 La solution retenue

Comme il était suggéré dans l'énoncé, nous avons choisi d'implémenter une solution minimale, que nous essaierons de faire évoluer en une version plus élaborée se passant de *Serveur central*. C'est pourquoi nous avons logiquement choisi de développer la stratégie 2 que nous tenterons de faire évoluer vers la troisième stratégie.

3.1 Justification du choix de la stratégie

Nous avons volontairement écarté la première stratégie de notre choix, pour plusieurs raisons. Tout d'abord à cause des activités excessives de la machine désignée comme serveur sur le réseau. En effet la présentation du projet se faisant dans les salles de TP, aucune machine ne se distingue pour ses facultés à être plus un serveur que les autres. Deuxièmement, l'inconvénient "Si le serveur central tombe en panne le système ne fonctionne plus" trouve une solution assez satisfaisante dans la stratégie 2, puisqu'il suffirait de faire régulièrement une copie du fichier de correspondance pour conserver l'espace virtuel, et de relancer le serveur sur un autre client pour que le système fonctionne à nouveau.

3.2 Détail de la solution

L'explication des deux stratégies étant donnée dans la première partie, nous allons voir ici comment nous allons implémenter la stratégie 2 dans un souci de réutilisation pour que son passage en stratégie 3 se fasse le plus simplement possible, puis nous verrons plus en détail la stratégie 3.

En regardant bien les trois stratégies, on peut remarquer que le système est toujours basé sur deux sous programmes (modules) dont seule la répartition sur les machines du réseau change :

- Le premier module s'occupe de la maintenance du fichier de correspondance, et permet d'avoir l'espace virtuel ; on appellera par la suite ce module *gereEV* car il gère l'Espace Virtuel.
- Le second module s'occupe de tout ce qui concerne les transfères de fichiers ; on l'appellera *transfereFichier*.

Le programme principal est constitué d'un programme serveurCentral et d'un programme client pour la stratégie 2. Le développement des deux modules cités peut se faire de manière détachée du programme principal. Par conséquent ces deux modules se retrouveront dans nos deux applications, c'est donc eux qu'il faudra programmer de manière la plus simple et la plus claire afin d'assurer une parfaite réutilisation.

Pour ce qui est du choix du langage, les signaux et les tubes offerts par le Shell ne sont pas assez évolués pour créer correctement des programmes client/serveur, nous avons donc choisi d'utiliser le C.

4 Aller plus loin

Le système peut être grandement amélioré par l'ajout de diverses fonctionnalités telles que :

- Arborescence à plusieurs niveaux
- Interface web
- Diverses choix optimaux
- La gestion des accès concurrents
- La gestion des accès concurrents
- La création d'une liste rouge des machines clients qui ne seront pas autorisées à devenir Serveur Père
- Gestion des utilisateurs dans l'espace virtuel

Afin de rendre le système plus agréable, nous développerons : l'arborescence à plusieurs niveaux, la gestion des accès concurrents (pour réutiliser les notions de synchronisation vues en cours), et nous verrons quels choix pourront encore plus optimiser notre système.

5 Conclusion

Nous espérons avoir été le plus clair possible tout au long de cet exposé, et nous pensons que le temps qu'il nous a demandé à nécessaire pour arriver à avoir un certain recul par rapport au projet demandé. Nos connaissances en système réseaux et notamment en programmation C étant vraiment très récente, l'élaboration de ce système risque de nous demander un travail plus que conséquent, le challenge est de taille mais l'enjeu aussi.