

# Module M1 - TP 1

## Création d'un serveur multiprocesseurs en mode datagramme

Bruno Jacob

8 Octobre 2001

### 1 But

- Maîtriser les principes de la communication par datagramme entre processus distants.

### 2 Introduction

Le but est de proposer à un processus client d'exécuter une liste de commandes shell, toutes indépendantes les unes des autres, sur plusieurs machines. La répartition des commandes sur ces machines doit être transparente au client. Le client ne doit correspondre qu'avec un seul serveur maître qui répartit le travail sur plusieurs serveurs esclaves. Le serveur maître donne ainsi l'impression d'être "multiprocesseurs".

Ce système se compose donc de

- 1 **serveur maître**
- $N$  **serveurs esclaves** (pour de TP choisissez  $N$  tel que  $3 \leq N \leq 5$ )

### 3 Un serveur esclave

Ce type de serveur est un *demon* (boucle sans fin) et offre un service dans le domaine Internet en mode datagramme. Son rôle est :

1. de recevoir un datagramme du serveur maître contenant :
  - la commande shell à exécuter (on suppose que la longueur de la commande ne peut excéder la taille d'un paquet IP)

- l’adresse du client qui a envoyé cette commande
- 2. d’exécuter la commande sur sa machine. On pourra se servir de la commande `system` qui exécute une commande et renvoie :
  - -1 si la commande n’a pas pu s’exécuter
  - >0 si la commande s’est mal exécutée
  - 0 si tout s’est bien passé
 (faire `man system` pour plus d’informations sur le compte rendu ou sur le fonctionnement de cette commande)
- 3. de renvoyer le compte rendu de l’exécution de cette commande au client. Ce compte rendu est un datagramme contenant :
  - la commande shell
  - la valeur renvoyée par la commande `system`
- 4. de signaler au serveur maître qu’il a fini d’exécuter une commande et qu’il peut en traiter une autre

## 4 Le serveur maître

C’est le serveur (*demon* lui aussi) qui offre le service ”multiprocesseurs” sur Internet en mode datagramme aux clients.

A son lancement, on doit lui fournir la liste des serveurs esclaves qu’il peut utiliser. La liste des serveurs esclaves est un fichier dont les lignes ont le format suivant :

```
<nom machine serveur esclave> <numéro de port esclave>
```

Pour chaque client qui lui demande ce service, son rôle est :

1. de recevoir le nom d’un fichier de commandes. Ceci peut être vu comme l’initialisation d’une demande de service par un client.
2. de transférer le contenu de ce fichier sur sa machine hôte
3. de déléguer l’exécution des commandes contenues dans ce fichier sur les serveurs esclaves dont la liste est passée en paramètre. L’exécution de ces commandes se fera donc en parallèle sur les machines des serveurs esclaves.

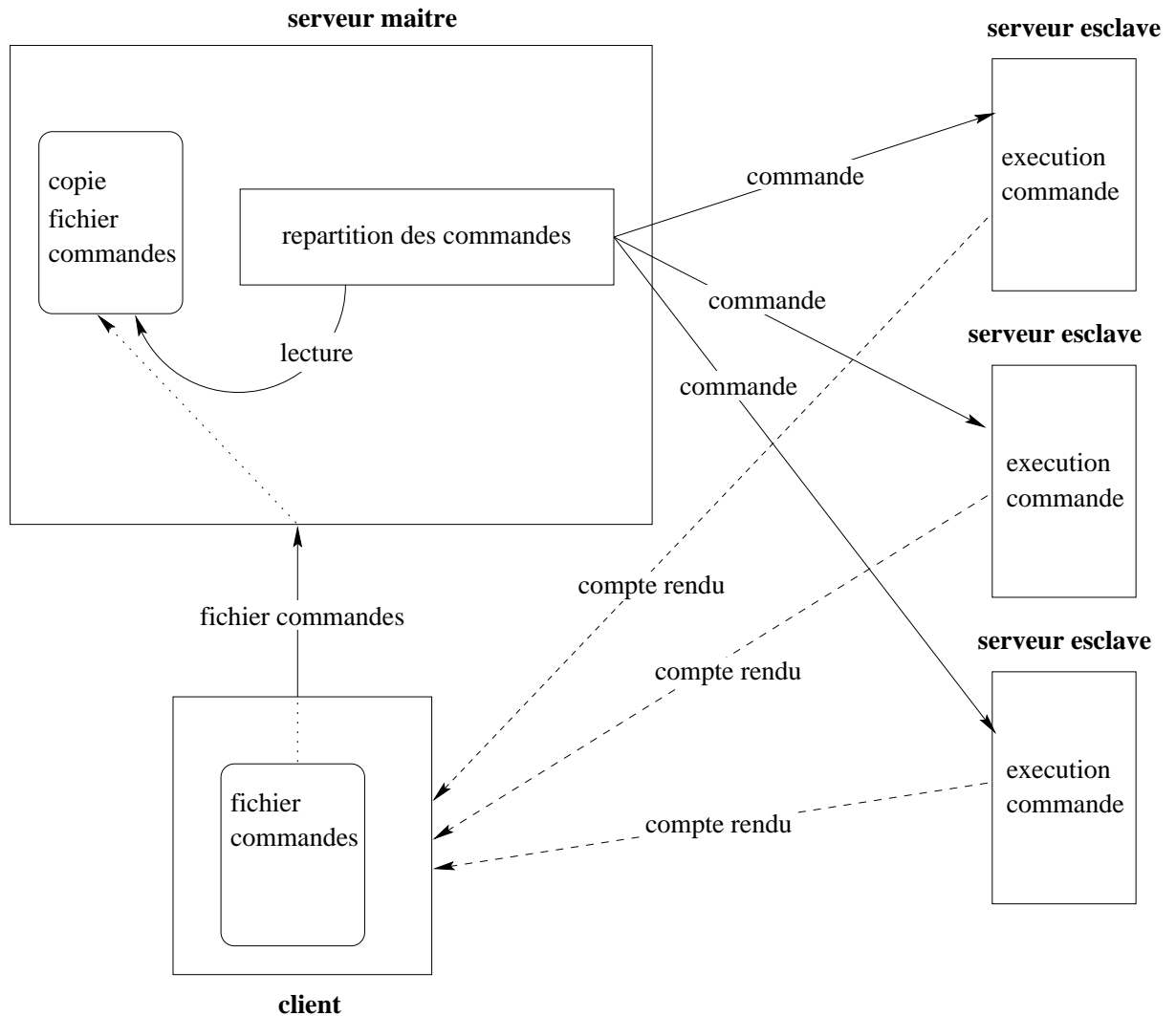
## 5 Un client

C’est le processus qui est client du service Internet en mode datagramme du serveur maître. Il doit donc respecter le protocole suivant :

1. Envoyer le nom du fichier de commandes au serveur maître
2. Lui transférer ensuite le contenu du fichier de commandes
3. Attendre le compte rendu de l'exécution de chaque commande pour vérifier si elle s'est bien déroulée.

Le fichier de commandes est un fichier texte. Les commandes sont séparées par un caractère que vous choisirez. Si, par exemple, le *séparateur* est un "retour chariot" ('`\n`') alors une commande correspond à une ligne du fichier texte.

## 6 synopsis



## 7 Rapport

Votre rapport sur ce TP devra comporter :

1. Une justification des choix de conception de vos programmes, c'est à dire une analyse du problème et une explication de votre solution.
2. Les structures de datagrammes utilisées
3. Les fichiers source des processus "client", "serveur\_maitre" et "serveur\_esclave".