

## I. Définition

Un système réparti est une collection de processeurs qui ne partagent pas ni la mémoire centrale ni l'horloge. Chaque processeur possède sa propre mémoire centrale locale. Ils communiquent entre eux à travers des moyens de communication (**Exemple :** Les microprocesseurs, les stations de travaux, les mini ordinateurs, les grands systèmes). Ils sont caractérisés par : (les avantages)

- Accès aux ressources variées.
- Augmentation de la vitesse d'exécution.
- Disponibilité des données (multiple unités de stockage indépendantes).
- Fiabilité (le système peut continuer à fonctionner même si une des machines tombe en panne)

Les problèmes à prendre en compte pendant la conception d'un système réparti:

- La synchronisation des processus.
- La communication.
- Le problème d'interblocage.

## II. Les Réseaux

La répartition du calcul (la charge) entre plusieurs processeurs est une technique très à la mode, il existe deux schémas pour ce type de système :

**a) Système Multiprocesseur : (les systèmes Fortement Couplés) :** Une seule mémoire centrale et une seule horloge avec une communication par mémoire partagée. Ce sont les systèmes parallèles destinés pour traiter un seul problème à la fois. Ce sont des systèmes rapides dans la transmission des données avec un débit très élevé.

**b) Système faiblement Couplés (les réseaux) :** Chaque processeur possède sa propre mémoire locale et sa propre horloge. Ce sont des systèmes lents avec un débit faible. Ce sont les systèmes distribués destinés à traiter différents problèmes à la fois :

**1- Définition d'un Système Réparti :** C'est un ensemble de processeurs faiblement couplés. Chaque processeur peut être appelé soit : site, nœud, ordinateur, machine, hôte .... Le nom *Site* est le plus utilisé pour désigner un emplacement géographique de la machine. La *Machine* peut être un serveur : Il possède des ressources désirées par d'autres machines dans d'autres sites. Ces derniers sont appelés clients.

Le système réparti donne la possibilité aux utilisateurs d'accéder aux différentes ressources de deux manières différentes :

- L'utilisateur accède à la ressource éloignée par connexion à la machine possédant les ressources ou en transférant les données de cette machine éloignée à sa propre machine. (système d'exploitation des réseaux).
- Le système d'exploitation réparti : l'utilisateur n'a pas besoin d'être au courant de la multiplicité des machines. Il accède aux ressources éloignées de la même manière qu'il accède à ses ressources locales.

## **2. Avantages des Systèmes Répartis :**

- Partage de ressources.
- Accélération du calcul (répartition de la charge).
- Fiabilité.
- Communication.

## **3. Topologie Des Réseaux**

- Réseaux complètement connectés : chaque site est directement lié avec tous les autres sites.
- Réseaux partiellement connectés : la liaison directe n'est pas complète.
- Réseaux hiérarchiques : les sites forment un arbre.
- Réseaux en étoile.
- Réseaux en anneau.
- Réseaux de bus à accès multiples : il existe une liaison partagée unique par bus (Ethernet).
- Les réseaux hybrides : utilisation de plusieurs protocoles avec un routage de données très compliqué.

## **4. Types Des Réseaux :**

✦ **LAN ( Local - Area – Network ) : (les réseaux locaux)** ce sont des réseaux composés d'un certain nombre de processus répartis sur des petites zones géographiques.

✦ **WAN (Wide – Area – Network) :** Ce sont des réseaux à grandes distances. Ils sont constitués d'un certain nombre de processeurs autonomes répartis géographiquement. Ils ont précédé les réseaux locaux. Ils sont composés de machines différentes avec des systèmes d'exploitation différents. On ajoute à chaque système d'exploitation un sous-système de communication (s.s.c)

### III. Structure des Systèmes Répartis

Dans un système d'exploitation réparti, l'utilisateur exploite les ressources éloignées de la même façon que ses ressources locales. L'échange entre les différents sites est géré par le système d'exploitation réparti. Cet échange est peut être parmi un des trois types suivants :

**1. Echange de données :** Si l'utilisateur désire accéder à une donnée éloignée (Exemple un fichier) d'un autre site, il existe deux façons pour réaliser cet accès :

✦ Faire et apporter une copie de ces données : Le résultat est une donnée locale. Dans le cas où les données ont été modifiées, l'usager renvoie la nouvelle copie modifiée vers le site d'origine à la fin de l'utilisation de ces données.

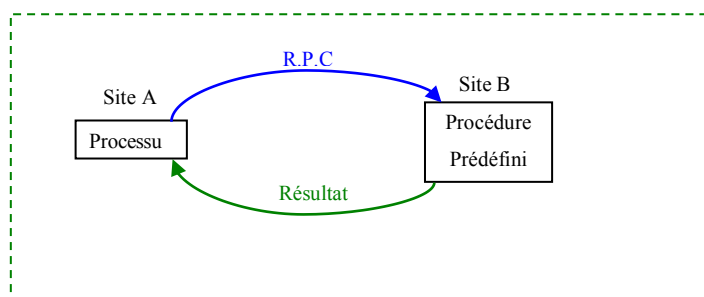
Exemple : En utilisant FTP {File Transfert Protocole}

Inconvénient : c'est une technique non efficace à cause du volume des fichiers de données et le temps de transfert.

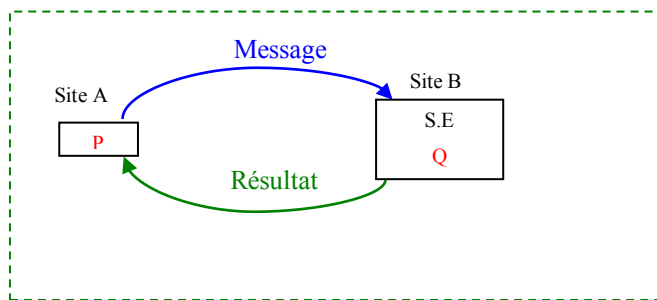
✦ Transfert par parties réellement utilisées : c.à.d pour le même fichier, par exemple, faire plusieurs transferts. Seulement les parties modifiées seront renvoyés au site d'origine.

Exemple : NFS {Network File Système}

**2. Transfert de calculs :** Dans le cas où un site a besoin de plusieurs données stockées dans des sites différents, il est plus efficace de transférer le calcul aux sites contenant ces données et le résultat est retourné au site demandeur du calcul. Cela peut être réalisé par un *appel de procédure à distance* notée RPC (Remote Procedure Call).



ou par envoie de message.



Les processus P et Q peuvent être exécutés en concurrence.

P envoie un message au système d'exploitation du site B pour créer un processus Q.

P et Q sont des processus autonome sur des sites autonomes.

**3. Transfert de processus :** Le transfert de processus est l'extension du transfert de calcul. Il est utilisé pour les raisons suivantes :

- Régler la charge.
- Accélérer le calcul.
- Le choix du matériel ou de logiciels plus adaptés
- Accès a des données multiples.

#### IV. Appelle de Procédure à Distance (RPC)

Le RPC est la technique la plus utilisée des services à distance. Il est utilisé pour connecter les différents systèmes reliés par un réseau. C'est une technique très semblable à un IPC (Inter Process Communication). Sauf que le message d'un IPC représente seulement un paquet de données.

- Les processus s'exécutent sur des systèmes répartis.
- La communication est par envoie de message.
- Le message est adressé à un **démon RPC** (un processus spécialisé) écoutant un port dans les systèmes éloignés.
- Le message contient un identificateur de la fonction à exécuter ainsi qu'une liste des paramètres.
- Après l'exécution de la fonction demandée, les résultats sont regroupés dans un message de réponse retourné au site demandeur.
- Chaque système à une seule adresse de réseau. Cette adresse peut avoir un aux plusieurs ports.
- Le port est un numéro identifiant un service.

**Problème 1 :** Un RPC peut échouer à cause des erreurs du réseau.

**Problème 2:** Le client ne peut pas connaître le numéro du port de l'autre système puisque il n'y a pas de mémoire partagée.

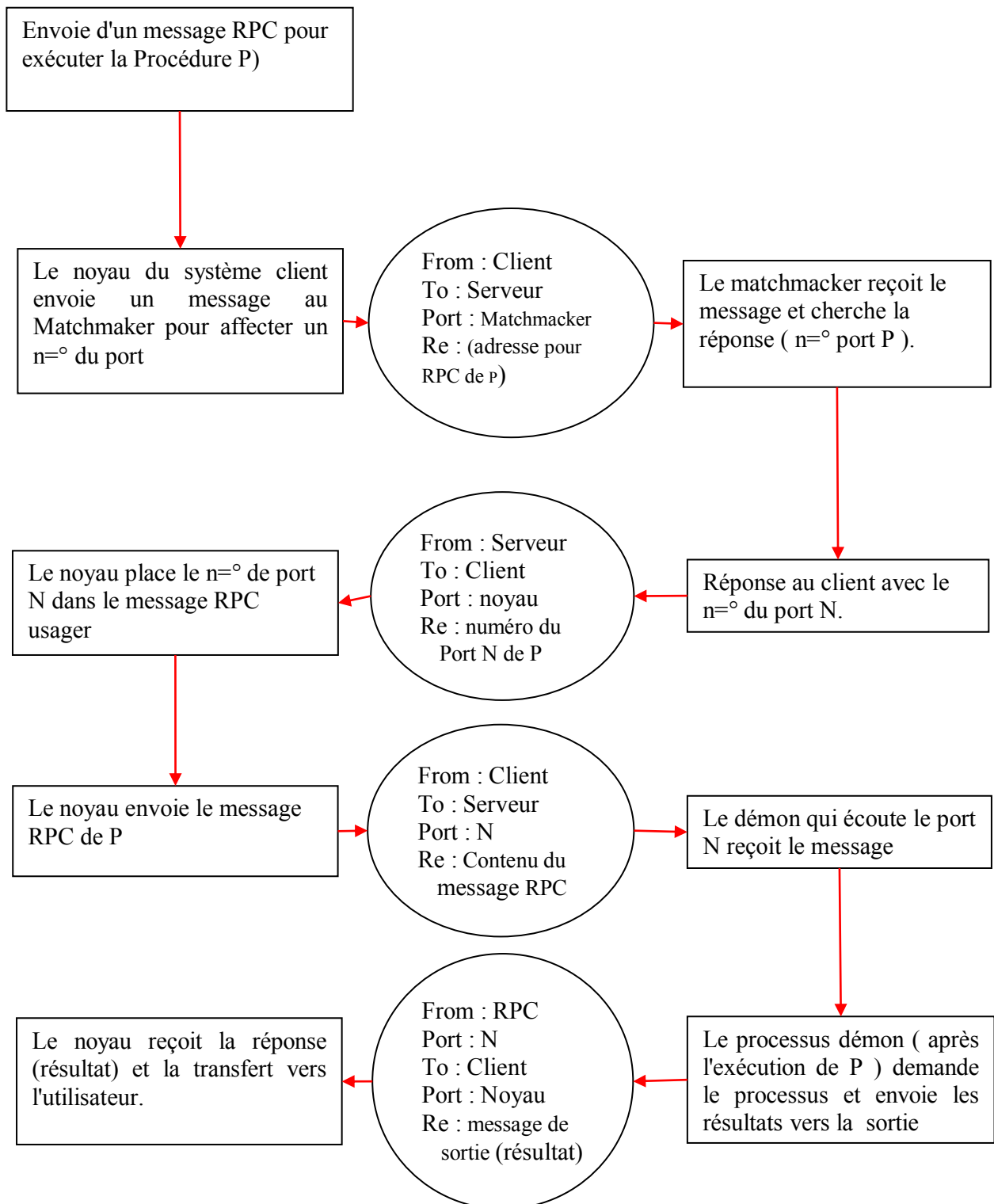
**Solutions :** Il existe deux types de solutions :

1. Connaître les adresses des ports à l'avance (les numéros des ports des différents services sont préfixés).
2. faire une liaison d'adresse d'une manière dynamique (par rendez – vous). Cette technique est réalisée par un programme appelé *démon de rendez – vous* (matchmaker) sur un port RPC fixe.

*Client*

*Messages*

*Serveur*



**Les Threads :** Les threads sont utilisés par les systèmes répartis pour l'envoi et la réception des messages en parallèle avec d'autres tâches (c'est une méthode asynchrone).

Les threads sont utilisés par les RPC pour pouvoir recevoir plusieurs messages en même temps pour rendre les RPC plus légers. Pour chaque requête entrante, le système crée un thread. Ce dernier sera éliminé après la fin d'exécution du travail demandé par la requête, car il est plus facile de créer un nouveau thread que de restaurer un ancien. (le temps et l'espace de stockage).

**Remarques :** Les systèmes répartis peuvent avoir plusieurs types de défaillances tel que :

- Défaillance d'une liaison.
- Défaillance d'un site.
- Les pertes de messages.

Le système repartit peut détecter l'existence ou non d'une défaillance, mais ne peut pas l'identifier (savoir le type de cette défaillance).