

## Module 1 : Introduction

- 1.1 Donnez 5 types de matériel et de données/logiciels qui peuvent être partagés en réseau?

Le partage d'imprimantes, de disques, et d'autres périphériques (numériseur, unité d'archivage...) est fréquent.

Les serveurs de calcul permettent de partager un CPU rapide, et un serveur comme Squid permet de conserver en mémoire les pages fréquemment accédées (partage de mémoire).

Le protocole X windows permet l'accès à distance de l'affichage graphique.

Les données partagées peuvent être des pages Web, des fichiers, du vidéo, des objets CORBA...

- 1.2 Comment peut-on synchroniser les horloges entre les ordinateurs sur un réseau?

Il faut envoyer un message demandant l'heure et recevoir la réponse. L'imprécision est causée par les délais d'envoi. Si les délais sont symétriques, ils peuvent être calculés et leur effet compensé. Il est cependant impossible par un simple envoi de messages de savoir quelle est leur valeur relative.

Pour synchroniser plusieurs ordinateurs, il est préférable d'avoir une structure hiérarchique pour éviter que les imprécisions ne s'accumulent trop.

- 1.3 Comment peut-on établir une connexion réseau si on ne connaît pas les paramètres du réseau local, par exemple sur un PDA en entrant dans une gare?

En supposant que réseau puisse s'établir physiquement, par infra-rouge ou micro-ondes (IRda ou IEEE 802.11), il faut un point de ralliement. Le PDA peut envoyer un message à tous pour demander les informations de base du réseau, par exemple par DHCP/BOOTP qui fournit une adresse IP et plusieurs informations (réseau, masque, passerelle...). Le réseau local pourrait aussi envoyer périodiquement un message à tous contenant les informations de base ainsi qu'un URL pour trouver de l'information supplémentaire.

- 1.4 Deux objets qui se trouvent sur des ordinateurs différents et qui sont implantés dans des langages différents doivent interagir, comment cela est-il possible?

Le protocole TCP/IP fournit un service de communication indépendant de la plate-forme. Des bibliothèques pour accéder ce service et d'autres fonctions de base servent à isoler chaque programme de sa plate-forme. Une bibliothèque pour l'interaction entre les objets doit exister (e.g. CORBA) et doit aussi masquer les

différences. Finalement, un format standard d'échange doit être établi pour toutes les données échangées (entiers, chaînes de caractères, structures...) de manière à tenir compte des différences imputables à la plate-forme et aux langages.

- 1.5 Lors de l'invocation d'une méthode à distance, quelles sont les sources possibles de problèmes?

Le processus serveur peut être inopérant, le processus client peut défaillir, le réseau peut être inopérant, le système d'exploitation de l'un ou l'autre pourrait défaillir, un bris matériel à l'un ou à l'autre peut arriver.

Lorsque le client est en panne, l'utilisateur peut souvent s'en rendre compte rapidement. Si le processus serveur n'existe plus, le client s'apercevra aussi rapidement d'un problème. Si le processus serveur, l'ordinateur serveur, ou le réseau ne répond pas alors qu'il le devrait, il est impossible de savoir où se trouve le problème et il faut attendre longtemps avant d'être certain qu'il y a un problème et que ce n'est pas simplement un délai causé par un réseau surchargé.

- 1.6 Lorsque plusieurs serveurs sont utilisés, pourquoi voudrait-on migrer des ressources entre ces ordinateurs? Comment suivre les ressources qui migrent?

Ceci peut être fait pour des raisons de proximité (minimiser le délai et la charge du réseau) ou pour équilibrer la charge de calcul, de stockage, ou de réseau.

Il serait possible de faire un appel à tous pour retrouver les ressources mais cela peut être inefficace. Maintenir un répertoire de localisation des ressources, ou conserver un pointeur vers la nouvelle localisation sur l'ancien site sont souvent beaucoup plus efficaces.

- 1.7 Quels sont les besoins de synchronisation associés à un système réparti d'indexage et de recherche pour l'Internet?

Plusieurs ordinateurs peuvent bâtir morceau par morceau le nouvel index pendant que la copie actuelle est toujours en utilisation. Une fois que tous les morceaux sont fusionnés, le nouvel index peut être propagé partout et, au moment choisi, le remplacement se fait. Les nouvelles requêtes utilisent le nouvel index.

- 1.8 Donnez des exemples de code mobile?

Il est utile d'avoir du code mobile lorsqu'un traitement est plus efficace si effectué au bon endroit. Par exemple, un applet permet d'exécuter une application directement dans le navigateur du client, sans occuper le CPU du serveur, et en évitant les délais associés au réseau. De la même manière, il est utile d'avoir certains programmes d'analyse des données qui sont envoyés et exécutés par le serveur de base de données plutôt que de transmettre tout le contenu de la base de données par le réseau.

- 1.9 Donnez des exemples de défaillances logicielles et matérielles qui peuvent ou ne peuvent pas être tolérées grâce à la redondance?

Les défaillances indépendantes peuvent être masquées ou tolérées dans un système redondant (disques RAID, processeurs TMR, mémoire ECC, réseau redondant...). D'autres défaillances sont reliées et ne peuvent donc pas nécessairement être masquées: erreur de programmation, catastrophe naturelle dans la région géographique où sont localisés les serveurs redondants, action de vol, sabotage, vandalisme concertée.

- 1.10 Un service de messagerie A peut perdre, répéter ou retarder les messages et ne détecte les erreurs que sur les en-tête des messages. Un service B peut perdre, retarder, ou livrer à trop haut débit les messages, mais leur contenu et leur ordre est respecté. Que peut-on attendre dans chaque cas?

Dans les deux cas le système est asynchrone et les problèmes de temps sont donc absents.

Le système A n'est ni valide (messages perdus) ni intègre (messages corrompus). Il peut avoir des erreurs par omission et des erreurs arbitraires.

Le système B est intègre mais n'est pas valide/fiable puisqu'il peut présenter des erreurs par omission (messages perdus).