



# Introduction aux systèmes répartis

Exercices pour le Module 1

INF8480 Systèmes répartis et infonuagique

Michel Dagenais

École Polytechnique de Montréal  
Département de génie informatique et génie logiciel

# Introduction aux systèmes répartis

---

① Systèmes répartis

② Réseautique



## Partage en réseau

---

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



## Partage en réseau

---

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 ou GPU rapide

Les données partagées peuvent être des pages Web, des fichiers, du vidéo, ou du code source. Par exemple, un serveur Web comme Apache rend des fichiers html disponibles à tous, ou un serveur comme Squid permet de conserver en mémoire, localement sur un serveur du réseau, les pages Web statiques fréquemment accédées agissant comme un cache partagé.



# Synchronisation

---

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



# Synchronisation

---

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.



## Initier une connexion

---

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



## Initier une connexion

---

Comment peut-on établir une connexion réseau si on ne connaît pas les paramètres du réseau local, par exemple sur une tablette 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 périphérique 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.



## Interaction entre objets

---

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



## Interaction entre objets

---

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.



## Appel de procédure à distance

---

Lors d'un appel ou de l'invocation d'une méthode à distance, comparé à une opération interne à un programme, quelles sont les sources possibles de problèmes?



## Appel de procédure à distance

---

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. Il se peut aussi que tout fonctionne mais que le réseau soit extrêmement lent.

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 car la connexion sera refusée par le système d'exploitation du serveur. 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. S'il n'y a pas de limite au temps que peut prendre un réseau surchargé, il faut attendre un temps infini avant d'être certain que le service est en panne et que ce n'est pas le réseau qui est lent!

## Migration de ressources

---

Lorsque plusieurs serveurs sont utilisés, pourquoi voudrait-on migrer des ressources (e.g., fichiers sur disque, programmes en exécution...) entre ces ordinateurs? Comment suivre les ressources qui migrent?



## Migration de ressources

---

Lorsque plusieurs serveurs sont utilisés, pourquoi voudrait-on migrer des ressources (e.g., fichiers sur disque, programmes en exécution...) 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. On pourrait aussi avertir les clients du changement d'adresse mais ce n'est habituellement pas la meilleure solution car souvent les clients sont très nombreux et moins fiables.

# Synchronisation de données réparties

---

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



## Synchronisation de données réparties

---

Quels sont les besoins de synchronisation associés à un système réparti d'indexation 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.



# Code mobile

---

Donnez des exemples de code mobile?



## Code mobile

---

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 fureteur 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 donnée par le réseau.



# Tolérance aux défaillances

---

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



## Tolérance aux défaillances

---

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.



# Introduction aux systèmes répartis

---

① Systèmes répartis

② Réseautique



## Temps de service d'une requête

Un client envoie une requête de 200 octets et reçoit une réponse de 5000 octets. Le temps de traitement de la requête est de 2ms. En local, avec les processus client et serveur sur le même ordinateur, une latence de 1ms s'ajoute (e.g., changement de contexte), en plus du temps de transfert mémoire-mémoire pour la requête et la réponse à 500Mo/s. Par réseau, chaque paquet prend une latence de 5ms en plus du temps de transfert à 10Mbits/s. Les paquets ont une taille maximale sur ce réseau de 1000 octets. Si une connexion TCP doit être établie, une latence de 5ms s'ajoute pour initier la connexion au début de la requête. Dans ce contexte, quel est le temps total requis avec pour recevoir la réponse à une requête en local, par UDP et par TCP?



## Temps de service d'une requête

Un client envoie une requête de 200 octets et reçoit une réponse de 5000 octets. En local nous avons la latence de 1ms, le transfert de 200 octets à 500Mo/s, le traitement de 2ms, la latence de 1ms, et le transfert de 5000 octets à 500MO/s.

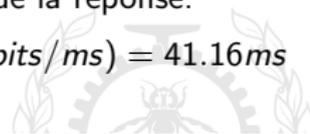
$$1ms + 200/500000 \text{ octets/ms} + 2ms + 1ms + 5000/500000 \text{ octets/ms} = 4.0104ms$$

Par réseau avec UDP (sans connexion) on a la latence de 5ms, le transfert de 200 octets à 10Mbits/s, le traitement de 2ms et la réponse en 5 paquets de 1000 octets, chacun demandant une latence de 5ms et le transfert de 1000 octets à 10Mbits/s.

$$5ms + (200 \cdot 8)/10000 \text{ bits/ms} + 2ms + 5(5 + (1000 \cdot 8)/10000 \text{ bits/ms}) = 36.16ms$$

Par réseau avec TCP, il faut simplement ajouter 5ms pour initier la connexion. Nous ne tenons pas compte des accusés de réception prévus dans TCP qui pourront arriver plus tard mais qui ne retardent pas la réception de la réponse.

$$5ms + 5ms + (200 \cdot 8)/10000 \text{ bits/ms} + 2ms + 5(5 + (1000 \cdot 8)/10000 \text{ bits/ms}) = 41.16ms$$



# Routeur

---

Quel est la tâche d'un routeur sur l'Internet? Quelles tables doit-il maintenir?



# Routeur

---

Quel est la tâche d'un routeur sur l'Internet? Quelles tables doit-il maintenir?

Un routeur connecte plusieurs réseaux IP. Il reçoit des paquets de ces réseaux et pour chacun doit décider à quel réseau l'envoyer. Il doit avoir une table qui dit, pour chaque groupe d'adresse IP possible, vers quel réseau l'envoyer pour que le paquet atteigne sa destination par le meilleur chemin possible. Etant donné le très grand nombre de réseaux sur l'Internet, on ne peut nécessairement avoir une table complète à chaque routeur. A la place, des routes par défaut et des regroupements de réseaux existent et servent à minimiser les tables (e.g. route par défaut vers un routeur avec une table plus complète, groupe de réseaux qui commencent par XXXX pour lesquels les paquets doivent tous passer par un routeur en Asie de chemin connu).



# Unicité des adresses

---

Comment sait-on que toutes les adresses IP sont uniques?



# Unicité des adresses

---

Comment sait-on que toutes les adresses IP sont uniques?

Un centre s'occupe d'allouer les adresses IP et ne donne jamais la même adresse 2 fois. Si quelqu'un choisit une adresse au hasard sans passer par ce centre, cela peut causer des ennuis sérieux.



## Avec ou sans connexion

---

Quel est le meilleur choix, TCP ou UDP, pour chacune des applications suivantes: Telnet, FTP, HTTP, RPC?



## Avec ou sans connexion

---

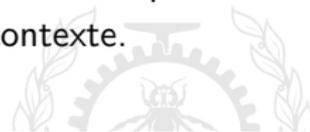
Quel est le meilleur choix, TCP ou UDP, pour chacune des applications suivantes: Telnet, FTP, HTTP, RPC?

Telnet: la fiabilité est importante et la performance n'est pas un problème, TCP est plus simple à utiliser.

FTP: pour le transfert de très gros fichiers, les possibilités de fenêtres d'accusés de réception, retransmission, contrôle du débit... de TCP sont très utiles.

HTTP: pour de courtes réponses, UDP pourrait être intéressant. Cependant, les transferts impliquent assez souvent de gros fichiers ou des requêtes multiples sur la même connexion et TCP est donc utilisé.

RPC: les messages sont généralement courts et un système de délai expiré est déjà implanté. UDP est donc un choix efficace dans ce contexte.



## Localisation d'un service

---

Un serveur ouvre un port et lui assigne un nom (numéro).  
Comment les clients peuvent-ils s'y connecter?



## Localisation d'un service

---

Un serveur ouvre un port et lui assigne un nom (numéro).  
Comment les clients peuvent-ils s'y connecter?

L'adresse d'un service est constituée de l'adresse IP et du numéro de port. On peut utiliser une adresse pré-établie (e.g. 132.207.99.1 pour la passerelle du réseau 132.207.99), un nom qui doit être communiqué (e.g. charles.polymtl.ca), ou un nom par convention (e.g. www.polymtl.ca, news.polymtl.ca).

Pour les numéros de ports, les numéros pré-établis sont souvent utilisés (e.g. 21 FTP, 22 SSH, 23 TELNET). Le service portmap existe aussi mais est principalement utilisé pour les Sun RPC.



# Problèmes de sécurité

---

Discutez des principaux types de problèmes de sécurité sur l'Internet.



## Problèmes de sécurité

---

Discutez des principaux types de problèmes de sécurité sur l'Internet.

Le premier type d'attaque est d'utiliser un processus sur le serveur pour y prendre pied. Ce processus peut être mal configuré (contrôle des accès), mal protégé (mot de passe par défaut), ou mal programmé (débordement de tampon). Un deuxième type d'attaque est de corrompre le réseau en espionnant ce qui s'y passe ou en falsifiant la provenance de messages. Un troisième type d'attaque est le déni de service en surchargeant un serveur ou un réseau de manière à le rendre inopérant.



## File d'attente

---

Un service peut traiter  $\mu=200$  requêtes par seconde. Il en reçoit présentement  $\lambda=100/s$  et on estime qu'il en recevra bientôt  $\lambda=180/s$ , toujours selon un processus de Poisson. Quel est  $N$  le nombre moyen de requêtes dans le système et  $W$  le temps de réponse moyen pour ces deux cas?



## File d'attente

Un service peut traiter  $u=200$  requêtes par seconde. Il en reçoit présentement  $l=100/s$  et on estime qu'il en recevra bientôt  $l=180/s$ , toujours selon un processus de Poisson. Quel est  $N$  le nombre moyen de requêtes dans le système et  $W$  le temps de réponse moyen pour ces deux cas?

Le taux d'utilisation  $U$  est de  $l/u = 100/200 = 0.5$  pour le premier cas et de  $180/200 = 0.9$  dans le second. Le nombre moyen de requêtes  $N = 0.5 / (1 - 0.5) = 1$  pour le premier cas et  $0.9 / (1 - 0.9) = 9$  dans le second cas. Le temps de réponse est de  $W = N/l = 1 / 100/s = .01s$  dans le premier cas et  $9 / 180/s = .05s$  dans le second cas.



## Nombre de threads

---

Un système reçoit des requêtes qui demandent en moyenne 5ms de coeur de CPU, 8ms de disque et 20ms d'attente pour une requête imbriquée. Ces requêtes sont réparties uniformément entre les ressources et sont traitées par un ordinateur avec 8 coeurs de CPU et 16 disques. Combien de requêtes par seconde est-ce que ce système peut traiter au maximum? Quel est le nombre minimal de threads requis pour ce nombre de requêtes?



## Nombre de threads

Un système reçoit des requêtes qui demandent en moyenne 5ms de coeur de CPU, 8ms de disque et 20ms d'attente pour une requête imbriquée. Ces requêtes sont réparties uniformément entre les ressources et sont traitées par un ordinateur avec 8 coeurs de CPU et 16 disques. Combien de requêtes par seconde est-ce que ce système peut traiter au maximum? Quel est le nombre minimal de threads requis pour ce nombre de requêtes?

Les 8 coeurs de CPU peuvent servir  $8 \times 1000\text{ms/s} / 5\text{ms/r} = 1600\text{r/s}$ .  
Les 16 disques peuvent servir  $16 \times 1000\text{ms/s} / 8\text{ms/r} = 2000\text{r/s}$ . Le facteur limitant est donc le CPU dans ce cas-ci. Puisque chaque requête prend au total  $5\text{ms} + 8\text{ms} + 20\text{ms} = 33\text{ms}$ . Nous avons donc  $1600\text{r/s} \times .033\text{s/r} = 52.8$  requêtes en même temps dans le système et donc 53 threads.

