

**ECOLE POLYTECHNIQUE DE MONTREAL**

**Département de génie informatique et génie logiciel**

**Cours INF4410: Systèmes répartis et infonuagique (Automne 2016)**

3 crédits (3-1.5-4.5)

---

**Contrôle périodique**

**DATE: Lundi le 31 octobre 2016**

**HEURE: 13h45 à 15h35**

**DUREE: 1H50**

**NOTE: Toute documentation permise, calculatrice non programmable permise**

**Ce questionnaire comprend 4 questions pour 20 points**

---

## Question 1 (5 points)

- a) Une banque achète et vend des devises (e.g. dollars US ou Euros) à des taux très concurrentiels. Lorsque les prix sont mis à jour, il est essentiel que ces mises à jour rejoignent toutes les succursales en même temps et que la mise à jour se fasse simultanément. Il ne faut jamais que des prix différents, en raison d'une mise à jour en cours, soient vus dans deux succursales différentes, puisqu'alors des spéculateurs pourraient prendre de l'argent à la banque simplement en profitant de cette différence de prix. Il est aussi important de s'assurer que les messages sont valides (non modifiés) et authentiques (viennent bien de la maison-mère de la banque). Décrivez comment un tel service peut être implanté (messages envoyés et reçus pour une mise à jour, contenu de ces messages en termes de champs ajoutés et de transformation). **(2 points)**
- b) Un gros fichier doit être transmis d'un serveur vers un grand nombre de clients. Chaque client est rejoignable soit par un réseau sans-fil commun, qui supporte la multi-diffusion et offre un débit de 100Mbit/s, soit par un réseau filaire commuté, dont chaque prise supporte un débit de 1Gbit/s, mais qui ne supporte que les communications point à point. Pour évaluer le temps requis pour les transmissions, on néglige la latence d'envoi sur le réseau, les paquets perdus et les bits requis pour les en-têtes de paquets, et on ne tient compte que du débit. Les choix disponibles sont soit de faire un envoi du fichier à tous les clients simultanément par multi-diffusion sur le réseau sans-fil, soit de faire une chaîne de distribution en arborescence en utilisant le réseau filaire (chaque client qui a une copie du fichier envoie une copie à un autre client qui ne le possède pas encore). Quel est le temps pour transmettre un fichier de taille  $k$  (en bits) à  $n$  clients dans chacun des deux cas? Pour quelles valeurs de  $n$  et de  $k$  est-ce que chaque solution (sans-fil versus filaire) est plus rapide? **(2 points)**
- c) Quels sont les avantages et inconvénients de CORBA? En quoi pourrait-il être plus intéressant que SOAP? **(1 point)**

## Question 2 (5 points)

- a) Un service similaire à celui que vous avez programmé dans votre premier TP offre la lecture de fichiers à distance par Java RMI. Voici l'interface pour ce service qui retourne le contenu du fichier dont le nom est reçu en argument. Fournissez le contenu complet du fichier `Server.java` qui permettrait d'implémenter ce service en java, incluant la classe `Server`, les déclarations et les initialisations, mais sans les inclusions (`import`). **(2 points)**

```
package ca.polymtl.inf4410.tpl.shared;
import java.rmi.Remote;
import java.rmi.RemoteException;

public interface ServerInterface extends Remote {
    byte[] ReadFile(String name) throws RemoteException;
}
```

- b) Dans le premier TP, un client, un serveur et un processus `registry` s'exécutaient et utilisaient Java RMI. Dans le serveur, un objet de type `Server` était initialisé puis enregistré auprès du `registry`. Java fonctionne avec un ramasse-miettes (garbage collector) et libère la mémoire associée aux objets qui ne sont plus utilisés. A quel moment est-ce que l'objet de type `Server` du serveur ne sera plus utilisé et le système de support à l'exécution saura qu'il peut récupérer la mémoire associée? Toujours dans le client et le serveur fournis dans le premier TP, on retrouve les deux appels distants suivants, `UnicastRemoteObject.exportObject(this, 0)`; pour exporter l'objet de type `Server` vers le `registry` et `distantServerStub.execute(a, b)`; lorsque le client demande au serveur d'effectuer un calcul. Expliquez quelle information est transmise pour chaque appel entre les deux processus en cause, et en particulier pour chaque argument (`this`, `a`, `b`) dites si une copie distincte est effectuée ou si un autre mécanisme est employé. **(2 points)**
- c) Dans les fichiers XDR pour les SUN RPC, il est possible de spécifier un numéro de version pour un service. Quel usage peut-on en faire? **(1 point)**

### Question 3 (5 points)

- a) Lors de la migration d'une machine virtuelle (VM) d'un serveur physique à l'autre, il faut copier toutes les pages qui constituent son image en mémoire virtuelle d'un serveur à l'autre. On suppose que les mêmes fichiers sont accessibles des deux serveurs et un peu d'état du logiciel de virtualisation (taille négligeable) doit aussi être copié. On vous propose deux techniques possibles pour effectuer la migration tout en minimisant le temps pendant lequel la machine virtuelle est complètement arrêtée. La première méthode consiste en tout copier sans arrêter la VM mais en notant les pages qui ont été modifiées après que cette itération de copie ait commencé. On continue ainsi pour quelques itérations jusqu'à ce que le nombre de pages encore modifiées soit petit. A ce moment, on arrête la VM, copie les pages restantes et redémarre la VM sur le nouveau serveur. La seconde méthode consiste en commencer dès le départ en même temps la copie des pages et l'exécution sur le nouveau serveur. Cependant, lorsque l'exécution de la VM requiert une page qui n'est pas encore copiée, la VM bloque, priorise la copie de la page après laquelle elle attend, et reprend dès que cette page arrive. Avec cette seconde méthode, il n'y a pas d'arrêt complet de la VM mais plusieurs pauses en attente de pages accédées qui n'ont pas encore été copiées. L'image en mémoire virtuelle contient 2 000 000 pages. Les pages sont copiées d'un serveur à l'autre au rythme de 20 000 pages par seconde. Pour la première méthode, la VM en exécution modifie 2 000 pages par seconde et initie l'arrêt et le transfert des pages restantes après trois itérations (la première copie et deux itérations pour les pages modifiées depuis lors). Pour la seconde méthode, la VM accède 4 000 pages par seconde non encore copiées. Pour la première méthode, quel est le temps pendant lequel la VM est arrêtée? Quel est le temps total de migration? Pour la seconde méthode, quel est le nombre de fois et la durée des arrêts en attente d'une page accédée non déjà copiée? Quel est le temps total de migration? **(2 points)**
- b) Lors de vos travaux pratiques, vous utilisez un système infonuagique basé sur OpenStack. Au moment de créer une nouvelle instance de machine virtuelle, OpenStack doit choisir sur quel

- noeud physique la placer. Quelle composante de OpenStack est responsable de choisir le noeud physique et démarrer l'instance virtuelle? Comment se fait le choix du noeud physique? **(1 point)**
- c) La fonctionnalité KSM (Kernel Same page Merging) a été ajoutée au noyau Linux pour aider la virtualisation. Que fait KSM? Pourquoi est-ce particulièrement intéressant pour la virtualisation? **(1 point)**
- d) Sur Amazon EC2, trois types de services de stockage (disque) sont disponibles, quels sont-ils? Quels sont les composants de OpenStack qui offrent les services équivalents? **(1 point)**

#### Question 4 (5 points)

- a) Un serveur de disque reçoit des requêtes à partir de clients. Chaque client requiert en moyenne des données pour 10 megabits/s. Le réseau est entièrement commuté. Le serveur est connecté au réseau par une prise qui fournit 10 gigabits/s. Son bus a une capacité de 16 gigaoctets/s et 6 disques y sont connectés. Les disques actuels fournissent chacun 100 megaoctets/s. Combien de clients est-ce que ce serveur peut supporter maintenant? Combien de clients peut-il supporter si on remplace les disques actuels par des disques SSD qui fournissent 1 gigaoctets/s chacun? **(2 points)**
- b) Sur un client NFS, le système d'exploitation reçoit 2 lectures et 1 écriture de page de 4Kio par seconde en moyenne. Pour une lecture, la page cherchée ne se trouve pas sur le client dans 25% des cas, se trouve déjà sur le client et a été validée depuis moins de 3 secondes dans 40% des cas, et se trouve sur le client mais a été validée depuis plus de 3s dans le reste des cas, soit 35%. Lorsqu'une page est présente mais sa validation est trop vieille, une revalidation suffit dans 60% des cas et une relecture doit être faite en plus (car la validation indique que la page a changé) dans 40% des cas. Sur le serveur, une demande de validation demande 2ms de CPU, une lecture demande 4ms de CPU et dans 30% des cas un accès disque de 12ms, et une écriture demande 5ms de CPU et 12ms de disque. Si 25 clients accèdent ce serveur qui comporte 1 CPU et 1 disque, quel sera le pourcentage d'utilisation du CPU? Du disque? **(2 points)**
- c) Avec le système de fichiers CEPH, la répartition des groupes de placement (PGID) sur les serveurs de fichiers est déterminée par un algorithme de hachage (CRUSH). Quel est l'intérêt d'un tel algorithme pour déterminer le placement sur un service de fichiers réparti? **(1 point)**

Le professeur: Michel Dagenais