

ÉCOLE POLYTECHNIQUE DE MONTREAL**Département de génie informatique et génie logiciel****Cours INF4410: Systèmes répartis et infonuagique (Automne 2013)**3 crédits (3-1.5-4.5)

Contrôle périodique**DATE: Mardi le 29 octobre 2013****HEURE: 9h30 à 11h20****DUREE: 1H50****NOTE: Toute documentation permise, calculatrice non programmable permise****Ce questionnaire comprend 4 questions pour 20 points**

Question 1 (5 points)

Un service de forum de discussion réparti permet à des usagers de se connecter à un serveur géographiquement proche et de participer à une discussion en y lisant ou ajoutant des messages. Les serveurs collaborant pour offrir ce service forment un groupe de serveurs qui se diffusent les messages. Il est possible d'ajouter un message sur un nouveau sujet de discussion, de répondre à un message dans une discussion existante, ou de lire les messages disponibles en listant les messages sur de nouveaux sujets ou en listant les réponses à un autre message.

- a) Chaque message doit avoir un identifiant unique utilisé afin de demander un message, de lister les réponses à un message, ou pour répondre à un message. De plus, il ne faut pas afficher une réponse à un message avant d'afficher le message auquel il répond. Donnez i) le nom utilisé pour un tel ordonnancement de message de groupe, et proposez une méthode efficace ii) pour créer un identifiant unique pour chaque message, et iii) pour assurer le bon ordonnancement du traitement des messages dans chaque serveur. **(2 points)**

- b) Les messages du forum peuvent être diffusés entre les serveurs par TCP, ou par UDP en multi-diffusion. Chaque envoi de paquet par UDP ou TCP occupe le réseau, qui est ici une ressource partagée, pendant 100 microsecondes, plus le temps de transmission à 100Mbps/s. Un message moyen demande un paquet de 1500 octets tout compris. Par TCP, en plus du paquet du message d'envoi, le récipiendaire envoie un paquet d'accusé de réception de 100 octets. Combien de messages du forum par seconde le réseau peut-il supporter avec UDP et avec TCP s'il y a 4 serveurs? Proposez une organisation efficace pour détecter les messages perdus et les retransmettre en UDP avec multi-diffusion, puisque cette fonctionnalité est offerte par TCP mais pas pour UDP. **(2 points)**
- c) Dans un sujet de discussion du forum, portant sur une question politique délicate, un message incendiaire a été ajouté par un inconnu se faisant passer pour une autre personne, un candidat à une prochaine élection, afin de le mettre dans l'embarras. Expliquez brièvement ce qui est requis pour qu'un tel système de discussion soit sécuritaire à cet égard. **(1 point)**

Question 2 (5 points)

- a) Lors du premier travail pratique, vous avez étudié le temps requis pour effectuer un appel à distance avec Java RMI en fonction de la taille des arguments de l'appel. Décrivez comment varie la durée de l'appel en fonction de la taille des arguments. Comment peut-on expliquer cette variation en fonction de la taille des arguments? **(2 points)**
- b) Pour les appels à distance, les sémantiques *au moins une fois* de même que *au plus une fois* peuvent être utilisées. Par ailleurs, le protocole UDP et le protocole TCP peuvent être utilisés. Java RMI utilise généralement *au plus une fois* sur TCP alors que les Sun RPC utilisent habituellement *au moins une fois* sur UDP. Montrez dans chaque cas combien de paquets sont échangés pour un appel où aucun paquet n'est perdu, et dans le cas où le paquet de réponse à l'appel à distance est perdu. On suppose que les arguments pour l'appel de même que la réponse sont sans problème plus petits que la taille maximale pour un paquet. Pour chaque scénario, montrez bien la fonction de chaque paquet envoyé et indiquez à chaque fois lorsque la fonction à distance est exécutée. **(2 points)**
- c) Le langage C# offre les appels à distance avec l'interface *Remoting*. Une des difficultés avec les appels d'objets à distance est la gestion automatique de la mémoire. En effet, un objet peut être détruit et son espace recyclé seulement lorsqu'il n'est plus utilisé. Il faut donc savoir s'il existe localement des objets rejoignables qui utilisent cet objet, ou s'il en existe dans des applications distantes. Comment peut-on maintenir à jour l'information sur les utilisateurs distants? Comment est-ce que le C# se prémunit contre les utilisateurs distants qui disparaissent (e.g. ordinateur qui saute) sans prévenir qu'ils n'utilisent plus un objet? **(1 point)**

Question 3 (5 points)

- a) Dans une université, le service de fichiers est fourni par trois serveurs CODA. Les trois serveurs redondants contiennent les mêmes fichiers en réplication et la charge est bien répartie entre les trois serveurs. Chaque client en moyenne ouvre 2 fichiers par seconde et ferme un fichier modifié (à envoyer aux serveurs) par 4 secondes. Lors d'une ouverture, le client vérifie si une copie est déjà présente et si elle est à jour (il n'a pas reçu de notification d'invalidation). Sinon, une copie du fichier est prise auprès d'un des serveurs. Lorsqu'un fichier est écrit sur un serveur, le serveur doit envoyer des messages de notification si d'autres clients en ont présentement une copie. Cependant, pour simplifier le problème, on suppose que peu d'utilisateurs accèdent des fichiers différents et les messages de notification ou de validation seront rares et pourront être négligés. On suppose aussi que tous les fichiers ont une longueur inférieure à 64KiO et leur lecture ou écriture constitue un seul accès. Lors d'une ouverture d'un fichier sur un client, une copie du fichier est déjà présente dans 60% des cas et à jour dans 90% de ces cas. Si 200 clients sont actifs en parallèle, combien d'accès en lecture et en écriture par seconde chaque serveur verra-t-il? Si un disque permet d'effectuer un accès en 15ms, combien de disques devrait-on avoir sur chaque serveur, en supposant que la charge est bien répartie entre les disques. **(2 points)**
- b) Une compagnie offre des vidéos sur demande par Internet. Elle peut soit avoir des serveurs extrêmement bien connectés, soit utiliser les équipements et ressources des clients pour répartir la charge à l'aide de protocoles comme bittorrent. Un gros film doit sortir et la compagnie s'attend à voir 1000 clients le télécharger la première journée. Le film a une taille de 4GiO. La compagnie obtiendra 2\$ par copie mais doit payer \$3 par Megabit/s de bande passante pour la journée. Une solution alternative est de permettre à 100 clients de prendre une copie gratuitement, à condition qu'ils agissent comme serveurs (pairs) bittorrent pour les autres clients. On suppose que les téléchargements peuvent être uniformément répartis sur les 24 heures de la journée. Laquelle des deux solutions sera la plus payante pour la compagnie? Justifiez. **(2 points)**
- c) Avec le Global File System, plusieurs noeuds peuvent accéder directement des unités de stockage habituellement connectées par FibreChannel. Ces unités de stockage reçoivent de ces noeuds des commandes simples pour lire et écrire des blocs. Comment est-ce que cela se compare avec un système de fichiers comme Lustre? Quels sont les avantages de cette approche? **(1 point)**

Question 4 (5 points)

- a) Un processus implémente un service de nom de manière récursive. Dans 90% des cas, le processus peut répondre immédiatement après 0.5ms de temps CPU. Autrement, il doit faire une requête à un serveur de nom plus haut dans la hiérarchie, ce qui prend 0.3ms de temps CPU et aussi un temps d'attente pour la réponse. Ce temps d'attente est de 7ms dans 50% des cas, 25ms dans 30% des cas et 75ms dans 20% des cas. Combien de requêtes par seconde ce service peut-il soutenir avec un seul fil d'exécution qui traite les demandes

- séquentiellement? Combien peut-il en soutenir avec de nombreux fils d'exécution (mais un seul CPU)? Combien de fils d'exécution doit-on avoir au minimum? **(2 points)**
- b) Pour une application particulière, vous désirez qu'une mise à jour du service de nom soit faite de manière atomique. Deux clients ne devraient jamais en même temps voir auprès de serveurs de noms différents l'ancienne et la nouvelle adresse associées à un nom. Pour ce faire, vous devez effectuer des requêtes auprès du serveur responsable de ce nom (ajouter, effacer ou modifier un attribut comme l'adresse IP associée à un nom), tout en sachant que cette valeur peut être maintenue en cache pendant un certain temps (spécifié par l'attribut TTL) par les autres serveurs de noms. Indiquez comment vous pourriez obtenir une telle mise à jour atomique dans ce contexte? **(2 points)**
- c) Une entreprise veut se doter d'un annuaire comme service de nom pour ses listes d'ordinateurs, ses listes d'utilisateurs, etc. Un premier ingénieur informaticien propose d'utiliser le protocole LDAP et le serveur OpenLDAP. Un autre collègue est d'avis qu'un service de nom n'est rien d'autre qu'une base de données et propose de simplement mettre une base de données de type SQL avec un service d'accès à distance comme ODBC. Qu'en pensez-vous? Qu'est-ce qu'un service de nom offre qui lui est spécifique? **(1 point)**

Le professeur: Michel Dagenais