

ECOLE POLYTECHNIQUE DE MONTREAL

Département de génie informatique et génie logiciel

Cours INF8480: Systèmes répartis et infonuagique (Hiver 2020)

3 crédits (3-1.5-4.5)

Contrôle périodique

DATE: Vendredi le 28 février 2020

HEURE: 9h30 à 11h20

DUREE: 1H50

NOTE: Aucune documentation permise sauf un aide-mémoire, préparé par l'étudiant, qui consiste en une feuille de format lettre manuscrite recto verso, calculatrice non programmable permise

Ce questionnaire comprend 4 questions pour 20 points

Question 1 (5 points)

- ✓ a) Un client envoie une requête de 4000 octets à un serveur et reçoit une réponse de 10 000 octets. Le client prend 200us pour préparer sa requête, le serveur 800us pour y répondre, et le client 100us pour traiter la réponse. La transmission de la requête et de la réponse se font sur un réseau qui a une latence de 100us et un débit de 10 000 000 octets / seconde. Combien de temps au total prend une requête? Combien de requêtes par seconde le client peut-il donc faire à ce serveur avec un seul thread d'exécution? Le client et le serveur ne font rien d'autre et ne sont donc pas interrompus par d'autres tâches. (2 points)
- ✓ b) Une machine virtuelle tourne sur un noeud physique A. L'image de cette machine virtuelle contient 8 000 000 pages. On veut migrer cette machine virtuelle vers un noeud B à travers un lien réseau qui permet d'envoyer 25 000 pages par seconde. Au premier tour, on copie l'ensemble des pages, aux tours subséquents, on copie les pages modifiées depuis le tour précédent. Pendant son exécution, la machine virtuelle modifie 1000 pages par seconde. La migration se fait d'abord en copiant les pages sans arrêter l'exécution, puis, lorsqu'il reste peu de pages modifiées, l'exécution est arrêtée le temps de copier les pages restantes. Cette phase d'arrêt ne doit pas durer plus de 0.25 seconde. Combien de temps durera la migration au total? Combien de temps est-ce que l'exécution sera arrêtée? (2 points)
- ✓ c) Un serveur prend 100us pour traiter chaque requête. Le serveur utilise un seul thread, sur un cœur dédié, pour traiter les requêtes, et les requêtes sont mises en queue d'attente lorsque le serveur est déjà occupé avec une requête. Si le serveur reçoit en moyenne 9500 requêtes par seconde, quel est le nombre moyen de requêtes en queue sur le serveur? (1 point)

Question 2 (5 points)

- ✓ a) Sur un réseau qui supporte la multi-diffusion, un message de groupe atomique est envoyé d'un membre aux autres. Le groupe compte N membres. Si sur ce réseau 1 message sur 4 envoyé par chaque noeud est perdu (i.e. pour cet échange, le premier message envoyé par un noeud donné est perdu, le second, troisième et quatrième passent, le cinquième est perdu...), combien de messages envoyés au total seront requis pour compléter ce message atomique. (2 points)
- ✗ b) Un message doit être envoyé avec gRPC utilisant l'encodage protobuf. Ce message suit la structure Recette et a pour contenu `{"eau": 80}, {"levure": 5}, {"sucre": 5}, {"farine": 250}, {"sel": 2}, {"yaourt": 15}`. Quel est le nombre d'octets requis pour encoder le message suivant sur un ordinateur 32 bits? 64 bits? (2 points)

```
message Ingredient {
    string nom = 1;
    int32 quantite = 2;
}

message Recette { repeated Ingredient = 1; }
```

- ✓ c) Quelle est la différence entre la sémantique "peut-être" et la sémantique "au plus une fois" pour les appels de procédure à distance? (1 point)

Question 3 (5 points)

- ✓ a) Trois serveurs CODA redondants sont pris pour servir des clients. Tous les fichiers lus ou écrits par les clients se retrouvent déjà dans leur cache CODA, et n'ont pas besoin d'être revalidés car le client se fie sur le service de notification des serveurs. Des étudiants sont en train de travailler sur un intéressant projet de programmation. Ils modifient et sauvent le code source, de 64KiO, et ensuite créent une nouvelle version de l'exécutable compilé, de 128KiO. Ceci se répète sur chaque client à chaque 5 minutes, le temps de trouver et corriger le prochain bogue. A chaque sauvegarde du fichier source, ou création du fichier exécutable par le compilateur, l'application ferme le fichier et son contenu est propagé en séquence, bloc par bloc, aux trois serveurs redondants. Le serveur peut lire ou écrire un bloc de 8KiO en 2ms de cœur de CPU et 8ms de disque. Combien de clients ces trois serveurs redondants peuvent-ils supporter, s'ils contiennent chacun 32 coeurs de CPU et 16 disques? (2 points)
- ✓ b) Le nouvel épisode de Pollux à Polytechnique vient de sortir et est offert par BitTorrent. Une première copie, qui contient les 1000 blocs de 1MiO, est disponible sur un serveur de départ. Les 8000 étudiants de Polytechnique en veulent une copie le plus tôt possible dès sa sortie. Tous sont connectés sur un réseau très rapide qui n'est pas un facteur limitant. Le serveur de départ et les ordinateurs des 8000 étudiants permettent tous de recevoir et de transmettre simultanément 1MiO par seconde dans chaque direction. Quel sera le temps requis avant que tous en obtiennent une copie? (2 points)
- ✓ c) Dans le travail pratique 3, plusieurs fichiers ont été créés avec la commande suivante. Donnez le contenu possible de /home/disk1, sur les conteneurs gluster1 et gluster2, pour les deux cas suivants de configuration: i) avec réPLICATION, ii) avec distribution. (1 point)

```
for (( i =1; i <= 20; i++ ) ) ; do touch $i; done
```

Question 4 (5 points)

- ✓ a) Dans une entreprise, chaque client effectue en moyenne 1 requête à un serveur donné à chaque 5 secondes. Le serveur peut répondre à 85% des requêtes en 1ms de CPU (information en mémoire), à 10% des requêtes en 2ms de CPU et 10ms de disque (information sur disque), et à 5% des requêtes en accédant d'autres serveurs en 5ms de CPU, 10ms de disque et 200ms d'attente après les autres serveurs. Ce service utilise plusieurs threads et le serveur possède 16 coeurs de CPU et 10 disques. Quel est le nombre maximal de clients que ce serveur peut desservir efficacement? Quel est le nombre minimal de threads qu'il devrait utiliser? (2 points)
- ✓ b) Un système client serveur utilise des objets Java qui communiquent avec Java RMI. Au moment d'une nouvelle requête, un objet Java RMI est créé sur le serveur et est accédé par le client.

Lorsque la requête est terminée, le proxy sur le client est libéré et un message est envoyé au serveur pour l'informer que l'objet Java RMI-correspondant n'est plus utilisé et peut être libéré. Le serveur reçoit environ 100 requêtes par seconde et chaque requête dure environ 5 minutes. Combien d'objets Java RMI pour ce service existent simultanément dans le serveur en moyenne? Si dans 1% des cas, le client termine de manière catastrophique pendant sa requête et le message du proxy libéré n'est pas envoyé au serveur, qu'arrivera-t-il? Est-ce que le serveur pourra continuer sans problème, ou au bout de combien de temps, et selon quels paramètres, deviendra-t-il en difficulté? (2 points)

- ✓ c) Une entreprise veut offrir un service en utilisant des serveurs virtuels obtenus d'un fournisseur comme AWS ou Azure. Sur ces serveurs virtuels, elle peut installer Linux et OpenStack afin de déployer et d'orchestrer des machines virtuelles. A la place, elle peut installer Linux et Kubernetes afin de déployer et d'orchestrer des conteneurs. Pour chacune de ces deux solutions, dites si c'est faisable et si c'est efficace. (1 point)

Le professeur: Michel Dagenais

Réponses

Question 2 :

$$a) 800 \mu s + 800 \frac{\mu s}{100} + 100 + 100 = \\ + \frac{1}{1000000} \text{ s} + \frac{10^6 \mu s}{1 s} \cdot (1000 + 10000) = 1000 \mu s$$

$$\begin{array}{r} 800 + 100 \\ \hline 900 \end{array} \quad \left. \begin{array}{l} 800 \\ 1000 \end{array} \right\} 800 \mu s$$

$$= 1000 \mu s + 800 + 1000 = 2600 \mu s. \text{ Temps total} = 2600 \mu s.$$

$$\frac{10^6 \mu s}{1 s} \times \frac{1 \text{ page}}{2600 \mu s} = 384,6 \frac{1}{s}. \text{ Le chat peut faire } 384,6 \frac{1}{s}$$

1.5
J

$$b) \frac{8000000}{25000} = 320 \text{ seconds}$$

$$1000 \times 320 = 320000 \text{ pages traitées}$$

$$\begin{aligned} & 320 + 12,8 + 0,512 + \\ & 30200 + 8 \times 10^{-4} \\ & = 35333 \text{ secondes} \\ & \text{de migration total.} \end{aligned}$$

$$\frac{320000}{25000} = 12,8 \text{ seconds}$$

$$1000 \times 12,8 = 12800 \text{ page med.}$$

$$\frac{12800}{25000} = 0,512 \text{ seconds}$$

$$1000 \times 0,512 = 512 \text{ page med.}$$

$$\frac{512}{25000} = 0,02048 \text{ sec}$$

$$1000 \times 0,02048 = 0,48 \text{ page med.}$$

$$\frac{0,48}{25000} = 8 \times 10^{-5}$$

l'exécution sera
achevée seulement
 8×10^{-5} secondes.

$$\text{c) } U = \frac{U}{1-U}$$

$$U = \frac{I}{P} = \frac{19500 \text{ A}}{\frac{12}{100 \times 10^{-6}} \text{ s}} = 0,95$$

$$N = \frac{0,95}{1-0,95} = 19$$

Le nombre moyen de négatifs en puissance est 19.

Question 2

- a) Pour un échange atomique alors si le réseau n'avait pas de portes, il ferait $1 \text{ messageDiffusia} + (N-1) \text{ ACK} + 1 \text{ mDiffusia}$.

Donc ce cas :

$$[(1 \times 0,75) + (2 \times 0,25)] + [(N-1) \times 0,75] + \left[\left(\frac{N-1}{4} \right) + (N-1) \right] \times 0,25] = [1 \times 0,75] + [2 \times 0,25]$$

- b) Ingrédient

$$eau = 1+1+3+1+1 = 7 \text{ octets}$$

$$lait = +6+ = 10 \text{ octets}$$

$$sucre = +5+ = 9 \text{ octets}$$

$$farine = +6+ = 10 \text{ octets}$$

$$sel = +3+ = 7 \text{ octets}$$

$$yogourt = +7+ = 11 \text{ octets}$$

$$\text{total} = \underline{\underline{55 \text{ octets}}}$$

Recette

$$1+5h = 55 \text{ octets requis}$$

Pour 36 bits; il faut ajouter ~~17~~ bits donc 2 octets 57 octets et 8 octets pour la partie

c) La terminologie "peut-être" signifie que ce n'est pas certain que le message se rende à la destination, il y a donc une possibilité que le RPC ne se fasse pas. En revanche, la terminologie "au plus une fois" signifie que l'on s'assure que le RPC se fait exactement une fois.

Question 3:

a) ~~64 ko + 128 ko = 192 ko~~ $\frac{192 \text{ ko}}{5 \text{ min}} = \frac{192 \times 10^3}{5 \times 60 \text{ s}} = 640 \text{ ko/s}$

serveur CPU: $32 \times \frac{8 \text{ ko}}{2 \text{ ms}} \times \frac{1000 \text{ ns}}{1 \text{ s}} = 128 \times 10^3 \text{ ko/s}$

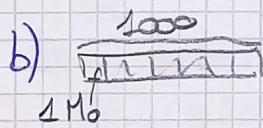
disk = $16 \times \frac{8 \text{ ko}}{8 \text{ ms}} \times \frac{1000 \text{ ns}}{1 \text{ s}} = 16 \times 10^3 \text{ ko/s}$

Le facteur limitant est le disque, qui limite à $16 \times 10^3 \text{ ko/s}$

$$\frac{16 \times 10^6 \text{ ko/s}}{640 \text{ ko/s}} = 25000$$

[Comme ils sont冗余的, on ne peut pas nécessairement écrire grâce au fait qu'ils sont 3. C'est donc l'écriture qui relatif face à la lecture.]

Ils peuvent supporter 25000 clients.



Le serveur envoie 1 bloc à chaque étudiant) 8000 s
La $\frac{1}{8}$ étudiants auront le même bloc.

Le réseau limite à 1 Mo/s par étudiant sur réception.

(les étudiants vont donner et recevoir le reste) 999 s
des blocs qu'ils marquent entre eux

Donc $8000 + 999 = 8999$ secondes.

Réponses

| | cluster 1 | cluster 2 | | cluster 1 | cluster 2 |
|----|--|---|-----|--|--|
| i) | 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 | 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 | ii) | 1 3 7 8 20 13 15 16 17 20 | 2 4 5 6 9 11 12 14 18 19 |

Question 6 :

a) client : $\frac{1 \text{ s}}{5 \text{ s}}$

serveur : 0,85 \rightarrow 1 ms CPU
 0,10 \rightarrow 2 ms CPU + 1ms disk
 0,05 \rightarrow 0,5 ms CPU + 5 ms disk + 20 ms attente.

CPU : $0,85 \times 1 + 0,1 \times 2 + 0,05 \times 5 = 1,3 \text{ ms/s}$

disk : 0 + 0,1 \times 10 + 0,05 \times 10 = 1,5 $\frac{\text{ms}}{\text{s}}$

16 CPU \Rightarrow $0,08125 \frac{\text{ms}}{\text{s}}$

16 disk \Rightarrow $0,15 \frac{\text{ms}}{\text{s}}$

Donc facteur limitant est le disque.

disk : $0,1 \times 10 + 0,05 \times (200 + 10) = 11,5 \frac{\text{ms}}{\text{s}}$ // 4000 ms $\Rightarrow 115 \frac{\text{ms}}{\text{s}}$

chaque répétition prend en moyenne : $5 \frac{\text{s}}{\text{n}} \times \frac{1000 \text{ ms}}{1 \text{ s}} = 5000 \frac{\text{ms}}{\text{s}}$

~~$\frac{1000}{115} = 869,57$ donc le nb maximal de clients est 869 clients.~~

~~$\frac{1 \text{ s}}{0,15 \text{ ms}} \times \frac{5000 \text{ ms}}{10} = 4347,8$~~

Le nombre minimum de threads qu'il devrait utiliser est donc 4348.

Réponses

b) serveur = $100 \frac{\text{objets}}{\text{s}}$ durée: $5 \frac{\text{min}}{\text{s}}$

$$\hookrightarrow \frac{5\text{min}}{1\text{min}} \times \frac{60\text{s}}{1\text{min}} = 300 \frac{\text{s}}{\text{s}}$$

$$100 \frac{\text{objets}}{\text{s}} \times 300 \frac{\text{s}}{\text{s}} = 30000 \quad \text{Il y a un moyen simultanément}$$

~~30 000 objets Java RMI.~~

Dans le cas où des clients terminent abruptement alors ils ne vont pas envoyer de message pour libérer l'objet.

Cela va poser un problème lorsque la mémoire sera pleine d'objets qui ne sont pas utilisés car ils n'ont pas été libérés. Il y a $100 \frac{\text{objets}}{\text{s}}$ donc si ~~1/s~~ ne termine pas, alors on ajoute 1 objet qui ne sera pas libéré. Si notre mémoire permet de contenir 100.000 objet RMI alors $100000 \times \frac{1\text{min}}{60\text{s}} \times \frac{1\text{h}}{60\text{min}} = 27,8$ heures plus tard

le système ne fonctionnera plus correctement.

c) On précise ici que l'entreprise obtient des serveurs virtuels, cela signifie que les machines qu'elle obtient ont déjà à niveau de virtualisation.

Si elle installe Linux et OpenStack dessus, cela est possible mais les performances seront médiocre, car ça va ajouter 1 niveau de virtualisation.

Si elle installe Linux et Kubernetes, ça va bien se passer et il y aura des performances très correctes : c'est courant de faire ça en entreprise.