POLYTECHNIQUE MONTRÉAL

Département de génie informatique et génie logiciel

Cours INF8480: Systèmes répartis et infonuagique (Automne 2021)
3 crédits (3-1.5-4.5)

CORRIGÉ DE L'EXAMEN FINAL

DATE: Samedi le 11 décembre 2021

HEURE: 13h30 à 16h00

DUREE: 2H30

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

Ce questionnaire comprend 5 questions pour 20 points

Question 1 (4 points)

a) Il y a 13 serveurs DNS racine en redondance, qui ont tous le même contenu, pour traduire les noms de domaine en adresses IP. Un client interroge les 13 serveurs pour une requête et retient la réponse qui revient le plus souvent. Combien de serveurs en panne peut-on tolérer avant que ce service DNS racine ne soit plus en mesure de fournir une bonne réponse, si le modèle de panne est i) détectée (un serveur en panne répond avec un message d'erreur), ii) par omission (un serveur défectueux ne répond pas) iii) panne de réponse (une mauvaise réponse aléatoire), ou iv) arbitraire (mauvaise réponse concertée entre les serveurs défectueux)? **(2 points)**

Les cas i) et ii) reviennent au même, à la différence que dans le premier cas on sait tout de suite que le serveur est défectueux, alors que dans le second cas on le déduit après un certain délai. On peut tolérer jusqu'à 12 serveurs en panne, car chaque serveur a le même contenu et, même s'il ne reste qu'un seul serveur, il peut offrir le service. Il se peut toutefois que ce serveur devienne très chargé. Pour le cas iii) on suppose qu'il y a très peu de chances que deux mauvaises réponses aléatoires coincident, et on peut donc obtenir la bonne réponse s'il reste deux serveurs valides. On peut alors tolérer 11 serveurs en panne. Finalement, dans le cas iv), il faut qu'une majorité de serveurs soient valides pour s'assurer d'une bonne réponse, car tous les serveurs compromis peuvent se liguer pour retourner la même mauvaise réponse. On peut ainsi tolérer 6 serveurs en panne, de sorte qu'il reste 7 serveurs valides dont la bonne réponse l'emportera.

b) Le serveur DNS d'un fournisseur de service Internet peut servir des requêtes DNS soit de manière itérative, soit de manière récursive. De manière itérative, le serveur traite la requête avec son CPU pendant 5ms. Il peut alors retourner la réponse à ce moment dans 70% des cas. Cependant, dans 30% des cas, il doit poursuivre le travail et lire en plus son disque pendant 15ms. Il trouve alors et retourne la réponse dans 60% des cas, mais dans 40% des cas il retourne une redirection vers un serveur plus haut dans la hiérarchie. En recevant une redirection, le client doit aller chercher la réponse sur un autre serveur, ce qui lui prend 20ms. Si le serveur est configuré pour fonctionner de manière récursive, il prendra aussi 5ms de CPU et retourne la réponse à ce moment dans 70% des cas. Il doit aussi en plus lire son disque en 15ms dans 30% des cas. Ensuite, il retournera la réponse demandée dans 90% des cas, mais dans 10% des cas il devra faire une requête et attendre la réponse avant de la retourner, ce qui lui demande en plus 1ms de CPU et 20ms d'attente. Quel sera le délai moyen vu par un client pour obtenir la réponse demandée dans chaque cas (récursif ou itératif)? Quel est le nombre de requêtes par seconde que peut soutenir le serveur, s'il n'utilise qu'un seul thread, de manière itérative? De manière récursive? **(2 points)**

De manière itérative, une requête prend en moyenne $5ms + .3 \times 15ms = 9.5ms$*. A cela, il faut ajouter 10ms dans 40% des cas où l'accès disque est requis, soit* $0.3 \times 0.4 \times 20ms = 2.4ms$*, pour un total de 11.9ms. De manière récursive, le délai est de* $5ms + .3 \times 15ms + 0.3 \times .1 \times (1ms + 20ms) = 10.13ms$*. La manière récursive est donc plus intéressante de ce point de vue. Dans le premier cas, le serveur peut traiter une requête en 9.5ms, puisqu'après le client est redirigé vers un autre serveur et le serveur du fournisseur Internet est libre pour la prochaine requête, soit* $1000ms/s/9.5ms/r = 105.26$ *requêtes par seconde. Dans le second cas, il peut fournir* $1000ms/s/10.13ms/r = 98.71$ *requêtes / seconde.*

Question 2 (5 points)

a) Un ordinateur A envoie une série de 3 messages à B pour obtenir l'heure exacte. Pour chaque envoi, l'heure d'envoi de la requête et l'heure de réception de la réponse sont notées avec l'horloge de A. La réponse de B donne la valeur de l'horloge de B au moment où il a traité la requête. Les différents messages ont été envoyés dans un intervalle très court, et on peut assumer que le décalage entre les deux horloges

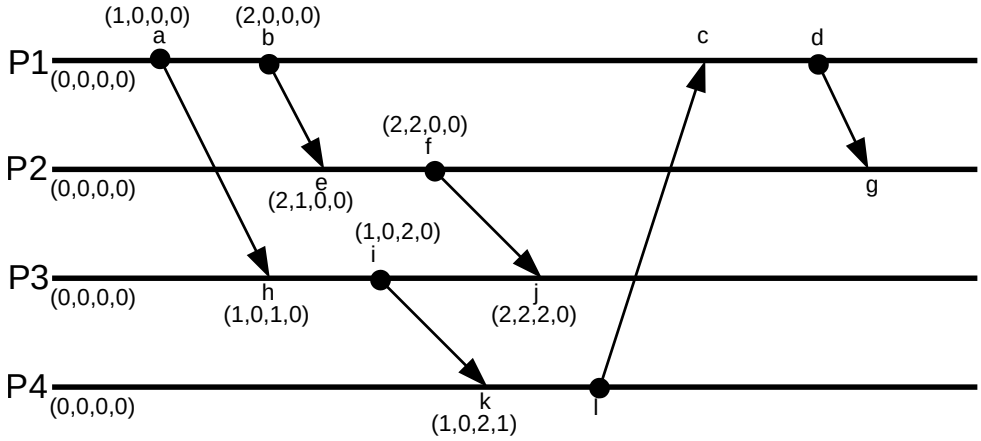
n'a pas changé pendant cet intervalle. La variabilité dans les réponses reçues dépend plutôt des délais variables pour l'envoi, la transmission et la réception des paquets via le réseau. Le détail des 3 envois est fourni dans le tableau suivant. Quels sont le décalage de temps et l'incertitude calculés pour chacun de ces 3 envois? Lequel doit-on retenir? Peut-on combiner les résultats des 3 requêtes pour obtenir une valeur de décalage de temps encore plus précise (avec moins d'incertitude)? Expliquez. **(2 points)**

Envoi	envoi	réception	réponse
Envoi 1	14h00m05.200	14h00m05.450	14h00m03.300
Envoi 2	14h00m06.250	14h00m06.450	14h00m04.400
Envoi 3	14h00m07.300	14h00m07.600	14h00m05.400

L'incertitude est donnée par le délai de réponse (réception - envoi), soit 14h00m05.450 - 14h00m05.200 = 0.250s pour l'envoi 1. En supposant que le délai est le même pour l'envoi et la réception, l'incertitude sera de +/- 0.250s / 2 = +/- 0.125s. L'heure de B à la réception sera ainsi l'heure de la réponse plus la moitié du délai, soit 14h00m03.300 + 0.250 / 2 = 14h00m03.425 +/- 0.125s. Le décalage est ainsi l'heure de réception de la réponse sur A moins l'heure de B à ce moment, soit 14h00m05.450 - 14h00m03.425 = 2.025s +/- 0.125s (ou l'intervalle [1.900s - 2.150s]), valeur qu'il faut soustraire à l'heure de A pour se synchroniser avec celle de B. Le même calcul est fait pour les envois 2 et 3. On obtient ainsi les intervalles résultants pour chacune des requêtes. Si on doit choisir une réponse parmi les 3, on retient celle qui donne la plus petite incertitude, soit l'envoi 2. Par contre, en combinant ces trois intervalles, on peut déduire que le décalage doit être supérieur à 1.900s et inférieur à 2.050s, soit de 1.975s +/- 0.075s, une réponse encore plus précise.

Envoi	*envoi*	*réception*	*réponse*	*incertitude*	*d min*	*d max*
Envoi 1	*14h00m05.200*	*14h00m05.450*	*14h00m03.300*	*+/- 0.125s*	*1.900s*	*2.150s*
Envoi 2	*14h00m06.250*	*14h00m06.450*	*14h00m04.400*	*+/- 0.100s*	*1.850s*	*2.050s*
Envoi 3	*14h00m07.300*	*14h00m07.600*	*14h00m05.400*	*+/- 0.150s*	*1.900s*	*2.200s*

b) Quatre processus, P1, P2, P3 et P4, démarrent à peu près en même temps et s'envoient des messages. Chaque processus maintient un vecteur de 4 compteurs d'événements, avec une entrée correspondant à chaque processus. Chaque processus, au moment d'envoyer un message, incrémente son compteur d'événements dans son vecteur et joint le vecteur au message. Chaque processus, lorsqu'il reçoit un message, incrémente son compteur d'événements dans son vecteur et fusionne son vecteur avec celui reçu dans le message. Les vecteurs sont donnés pour les points a, b, e, f, h, i, j et k. Que peut-on dire de la relation temporelle entre les points a et j, à l'aide de ces vecteurs de compteurs d'événements? De la relation entre les points b et k? Expliquez. Donnez finalement les vecteurs de compteurs d'événements pour les points c et g. **(2 points)**



Le point j est postérieur au point a car toutes les entrées de son vecteur de compteurs d'evénements sont supérieures, (2,2,2,0) vs (1,0,0,0). Par contre, les points b et k sont potentiellement concurrents car certaines valeurs dans les vecteurs sont supérieures à celles correspondantes dans l'autre vecteur, pour chacun des deux points, (2,0,0,0) vs (1,0,2,1). Pour le point c, le vecteur est (3,0,2,2). Pour le point g, le vecteur est (4,3,2,2).

c) Un serveur central d'exclusion mutuelle reçoit les requêtes de divers clients. Il accorde la ressource si elle est libre, ou met la requête en queue pour la ressource, si elle est déjà prise. Lorsqu'une ressource est libérée par un client, la ressource est accordée au prochain client en queue, ou la ressource devient libre si la queue est vide. Est-ce que ce service est sûr (un seul utilisateur de la ressource à la fois), vivace (chacun peut obtenir un tour) et respecte l'ordre (premier arrivé, premier servi)? **(1 point)**

Oui, le serveur central d'exclusion mutuelle est sûr, il n'accorde une ressource qu'à un seul client à la fois, il est vivace car chaque requête est maintenue en queue pour être servie à son tour, et respecte l'ordre, justement en raison des queues pour chaque ressource. On peut noter qu'il y a toutefois la possibilité que des interblocages se présentent, étant donné qu'il n'y a pas de contrainte imposée par le serveur sur comment sont pris les verrous.

Question 3 (4 points)

Les transactions T1, T2, T3 et T4 s'exécutent en même temps et leurs opérations de lecture et d'écriture sur des variables (x1, x2, ... x6) sont entrelacées. Les lectures d'une transaction sont effectuées sur les versions courantes des variables, et les écritures d'une transaction sont effectuées sur une version provisoire des variables pour la transaction. Lorsque la transaction se termine et est acceptée, la version provisoire des variables écrites par la transaction devient la version courante. Une validation de la cohérence par contrôle optimiste de la concurrence est effectuée pour accepter ou non chaque transaction. Une transaction est acceptée s'il n'y a aucun conflit sur les variables accédées (selon les critères de la méthode choisie) et est refusée s'il y a conflit sur certaines variables. Il faut tenir compte des transactions précédentes qui ont été validées (et ignorer celles qui ne l'ont pas été) dans le calcul des conflits pour savoir si chacune des transactions est acceptée ou non.

```
1  T1:  Begin                15 T4:  Write x2
2  T1:  Read x2              16 T1:  Write x1
3  T1:  Write x5             17 T1:  End
4  T2:  Begin                18 T2:  Read x6
5  T1:  Read x5              19 T3:  Read x3
6  T1:  Write x1             20 T2:  Write x3
7  T3:  Begin                21 T2:  Write x5
8  T1:  Read x4              22 T2:  End
9  T1:  Read x6              23 T4:  Read x5
10 T3:  Write x3             24 T3:  Write x4
11 T1:  Write x2             25 T3:  End
12 T4:  Begin                26 T4:  Read x6
13 T4:  Read x3              27 T4:  Write x3
14 T4:  Read x2              28 T4:  End
```

a) Lesquelles des transactions T1, T2, T3 et T4 pourraient être validées si une validation en reculant était utilisée pour vérifier la cohérence des transactions? Pour chaque transaction non validée, donnez la ou les variables en conflit. **(2 points)**

En reculant, au moment de compléter T1, il n'y a pas de problème. Pour compléter T2, il faut vérifier ce qu'il a lu (x6) versus ce qui a été écrit par les transactions précédentes concurrentes, T1 (x5, x1, x2). Il n'y a pas d'intersection et T2 peut compléter. Pour T3, ses lectures (x3) sont vérifiées versus les écritures de T1 (x5, x1, x2) et T2 (x3, x5). Il y a intersection (x3) et la transaction T3 doit être abandonnée. Pour T4, les lectures (x3, x2, x5, x6) doivent être vérifiées versus les écritures de T1 (x5, x1, x2) et T2 (x3, x5), mais pas de T3 qui a été abandonné. Il y a intersection (x2, x3, x5) et la transaction T4 doit être abandonnée.

b) Quel serait le contenu du journal écrit pour les transactions T1 et T2 (en supposant qu'il n'y aurait pas de conflit et qu'elles pourraient compléter, afin de bien découpler les sous-questions a et b). **(2 points)**

Les écritures des transactions peuvent être écrites dans le journal au fur et à mesure qu'elles apparaîssent dans les transactions, ou elles peuvent être regroupées et écrites au moment de commettre. Dans le premier cas, T1 sauverait x5, x1, x2 et une deuxième fois x1; au moment de la relecture, seulement la valeur de x1 la plus récente serait prise. Dans le second cas, en écrivant au moment de commettre, x5 , x2 et seule la dernière valeur de x1 seraient sauvés. Voici un contenu possible du journal.

```
P0: ...
P1: écrire x5
P2: écrire x2
P3: écrire x1
P4: Préparer T1(P1, P2, P3), P0
P5: Compléter T1, P4
P6: écrire x3
P7: écrire x5
P8: Préparer T2(P6, P7), P5
P9: Compléter T2, P8
```

Question 4 (4 points)

a) Un service de fichiers répartis est offert par 3 serveurs redondants. Au moins une majorité de serveurs, soit 2 sur 3, doit être disponible pour que le service soit disponible. Chaque serveur est constitué d'un boîtier et son électronique, avec une probabilité de disponibilité de 0.9, ainsi que d'un ensemble de disques en RAID, 5 disques dont au moins 4 doivent être fonctionnels. La probabilité d'être fonctionnel pour un disque est de 0.8. Quelle est la probabilité qu'un ensemble de disques en RAID soit fonctionnel? Un serveur de fichiers? Le service de fichiers répartis? **(2 points)**

L'ensemble de disques RAID a une probabilité de fonctionner de $0.8^5 = 0.32768$ *(5 disques fonctionnels) et* $5!/((5-4)!4!) \times 0.8^4 \times (1-0.8)^{5-4} = 0.4096$ *(exactement 4 disques fonctionnels) pour un total de* $0.32768 + 0.4096 = 0.73728$. *Un serveur est opérationnel si le boîtier est fonctionnel de même que son ensemble de disques RAID, ce qui donne* $0.9 \times 0.73728 = 0.663552$. *Le service sera disponible si 2 ou 3 serveurs le sont. La probabilité est de* $0.663552^3 = 0.292162779$ *(3 serveurs fonctionnels) et* $3!/((3-2)!2!) \times 0.663552^2 \times (1-0.663552)^{3-2} = 0.444415432$ *(exactement 2 serveurs fonctionnels), soit un total de* $0.292162779 + 0.444415432 = 0.736578211$.

b) Expliquez la différence entre un service répliqué avec des serveurs tous actifs, versus une réplication passive avec un serveur primaire actif et un serveur secondaire passif. Quels sont les avantages de chacun? **(1 point)**

Dans un service répliqué avec des serveurs tous actifs, chaque serveur peut accepter des requêtes. Ceci a l'avantage de permettre plus de parallélisme et une meilleure performance. De plus, en cas de panne, un autre serveur est tout de suite disponible pour prendre la relève. L'inconvénient est qu'il est plus difficile d'assurer la cohérence dans un tel environnement.

Avec un serveur primaire, celui-ci s'occupe de servir toutes les requêtes. Il communique ses changements d'état au serveur secondaire, pour que ce dernier puisse prendre la relève en cas de panne. Dans un tel système, la cohérence est très facile à assurer, tout passe par le serveur primaire, et les mises à jour sont envoyées dans l'ordre, sérialisées, au serveur secondaire. L'inconvénient est qu'il peut y avoir un court délai en cas de panne pour que le serveur secondaire bascule et devienne le serveur primaire. De plus, le serveur secondaire est largement inutilisé, puisque les pannes sont normalement rares, et cette ressource ne contribue pas à augmenter la performance du système.

c) Lors de vos travaux pratiques, vous avez utilisé Docker et Kubernetes. Expliquez brièvement la fonction de chacun. Lequel des deux vous permet d'obtenir une résilience aux pannes? Expliquez. **(1 point)**

Docker permet d'encapsuler dans une image une application avec tous les exécutables et librairies qui sont requis. Il est alors facile d'exécuter cette application sans avoir à se soucier si les bons exécutables ou les bonnes versions de librairies sont disponibles sur le système où on veut deployer cette application. Kubernetes fait l'orchestration d'applications, souvent contenues dans des images Docker. L'orchestration consiste en rouler les différentes applications requises pour un système, tout en connectant ces applications aux bons ports réseau. Kubernetes permet en outre de surveiller les applications, redémarrer celles qui s'arrêtent prématurément, et répartir la charge des requêtes entre plusieurs réplicats. C'est cette dernière fonctionalité de Kubernetes qui permet d'obtenir une résilience aux pannes; les requêtes ne sont pas envoyées à un réplicat en panne, et celui-ci est redémarré aussitôt.

Question 5 (3 points)

L'entreprise pour laquelle vous travaillez veut construire un nouveau centre de données. Vous faites partie d'une équipe d'ingénieurs qui doit concevoir et planifier la construction et l'opération de ce centre de données. Vous voulez mettre de l'avant un design qui optimise le rendement par rapport aux coûts. Devez-vous comme ingénieur vous limiter aux impacts financiers, ou devez-vous aussi vérifier l'impact environnemental du projet? i) Quelles sont les lois applicables qui imposent à l'ingénieur ou au promoteur de tenir compte de ces impacts et du développement durable? ii) Quelles sont les différentes phases à considérer pour évaluer dans son ensemble un tel projet dans un contexte de développement durable? iii) Quels sont les quatre différents types d'impact sur l'environnement normalement considérés dans le cadre d'une analyse de cycle de vie? **(3 points)**

L'ingénieur doit tenir compte des conséquences de l'exécution de ses travaux sur l'environnement dans une perspective de développement durable. i) Ceci est prescrit par la loi canadienne de 2008 sur le développement durable, la loi québécoise de 2006 sur le développement durable, et le code de déontologie de l'Ordre des Ingénieurs du Québec. La procédure d'évaluation environnementale du BAPE ne s'applique normalement pas à un centre de données. ii) Les phases du cycle de vie sont la fabrication (construction du site et fabrication des équipements), le transport (des matériaux pour la construction et des équipements), l'opération du site (entretien, alimentation électrique...) et finalement la fin de vie (la démolition ou conversion du bâtiment, la restauration du site, et le recyclage ou la mise aux rebuts des matériaux et équipements). iii) Pour toutes les activités qui se retrouvent dans ces différentes phases, il faut examiner l'impact sur la santé humaine, sur l'écologie, sur les changements climatiques et sur l'appauvrissement des ressources. Pour un centre de données typique, une grande partie de l'impact est reliée à la consommation d'énergie pendant l'opération de l'équipement informatique et de la climatisation. Un autre impact non négligeable est la fabrication des équipement informatiques et

électriques, en particulier le raffinage de l'or et du cuivre requis pour ces équipements. Etant donné l'importance de la consommation électrique, une source d'énergie propre, et un climat froid qui ne requiert pas de climatisation, peuvent diminuer considérablement l'impact négatif d'un centre de données.

Le professeur: Michel Dagenais