ECOLE 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)

CORRIGÉ DE L'EXAMEN FINAL

DATE: Samedi le 7 décembre 2013

HEURE: 9h30 à 12h00

DUREE: 2H30

NOTE: Toute documentation permise, calculatrice non programmable permise

Ce questionnaire comprend 4 questions pour 20 points

Question 1 (5 points)

a) Trois processus utilisent des horloges logiques qui sont incrémentées à chaque événement, incluant les envois et réceptions de messages entre processus. Les événements vus par chacun sont listés. Associez un vecteur de compteurs d'événements à chacun (<p1,p2,p3>), fournissant ainsi ces mêmes trois listes, mais avec des vecteurs de compteurs plutôt que des compteurs. A l'aide de ces vecteurs, que pouvez-vous dire de l'ordre relatif (avant, concurrent, après) des événements 2 de p1 versus 2 de p3, et 4 de p1 versus 4 de p2. Justifiez. **(2 points)**

p1: 1 lecture; 2 message A vers p2; 3 écriture; 4 écriture; 5 message B de p3
p2: 1 message A de p1; 2 écriture; 3 message C vers p3; 4 lecture; 5 message D de p3
p3: 1 lecture; 2 message C de p2; 3 écriture; 4 message B vers p1; 5message D vers p2

Pour construire les vecteurs de compteurs d'événements, il faut initialiser les compteurs à 0 et joindre les contenus des vecteurs des envoyeurs lors de la réception d'un message, ceci donne ce qui suit.

p1: <1,0,0> lecture; <2,0,0> message A vers p2; <3,0,0> écriture; <4,0,0> écriture; <5,3,4> message B de p3
p2: <2,1,0> message A de p1; <2,2,0> écriture; <2,3,0> message C vers p3; <2,4,0> lecture; <2,5,5> message D de p3
p3: <0,0,1> lecture; <2,3,2> message C de p2; <2,3,3> écriture; <2,3,4> message B vers p1; <2,3,5> message D vers p2

En comparant p1:3, vecteur de <2,0,0>, avec p3:2, vecteur <2,3,2>, puisque toutes les entrées du second sont supérieures ou égales au premier, ce dernier est postérieur au premier. Dans l'autre cas, p1:4, vecteur <4,0,0>, versus p2:4, vecteur <2,4,0>, aucun vecteur n'est strictement supérieur ou égal à l'autre. On ne peut donc rien conclure et on doit considérer les événements comme concurrents.

b) Un ordinateur A envoie un message à B à 9h00m10.200s pour obtenir le temps et reçoit une réponse à 9h00m10.600s. L'ordinateur B reçoit la requête de A à 9h00m05.300s à son heure, envoie une requête à C à 9h00m05.400s, reçoit la réponse de C à 9h00m05.500s et envoie sa réponse à A à 9h00m05.550s. Le serveur C reçoit la demande de B à 9h00m00.300s et lui répond à 9h00m00.350s, heure de C. La réponse finale à A contient toutes ces valeurs de temps ainsi que leur origine. A peut donc appliquer l'algorithme utilisé par NTP pour synchroniser son horloge avec celle de C, celle-ci étant très précise. Quel est le décalage à appliquer sur A? Quel est l'intervalle d'incertitude associé? **(2 points)**

Nous pouvons appliquer de manière imbriquée l'algorithme de NTP. En regardant B par rapport à C, on a:

a = 9h00m00.300s - 9h00m05.400s = -5.100s
b = 9h00m00.350s - 9h00m05.500s = -5.150s
Ajustement = (a+b)/2 = (-5.100s - 5.150s)/2 = -5.125 s

Précision = (a-b)/2 = (-5.100s + 5.150s)/2 = 0.025s

Nous pouvons alors convertir les temps sur B en temps de C: 9h00m05.300 - 5.125 = 9h00m00.175 et 9h00m05.550 - 5.125 = 9h00m00.425. Le calcul de A par rapport à B (synchronisé à C) donne donc:

a = 9h00m00.175s - 9h00m10.200s = -10.025s
b = 9h00m00.425s - 9h00m10.600s = -10.175s
Ajustement = (a+b)/2 = (-10.025s - 10.175s)/2 = -10.100s
Précision = (a-b)/2 = (-10.025s + 10.175s)/2 = -0.075s

Le décalage à appliquer est ainsi de -10.1s et l'incertitude est la somme des deux et est donc de +/- (.075+.025) soit +/- .1s. Nous pouvons aussi calculer le décalage de A à B sans convertir le temps de B:

a = 9h00m05.300s - 9h00m10.200s = -4.9s
b = 9h00m05.550s - 9h00m10.600s = -5.05s
Ajustement = (a+b)/2 = (-4.9s - 5.05s)/2 = -4.975s
Précision = (a-b)/2 = (-4.9s + 5.05s)/2 = -0.075s

Il suffit alors d'additionner les deux décalages à appliquer (C à B et B à A), ce qui donne -5.125 - 4.975 = -10.1s ainsi que les imprécisions .075 + .025 = .1s, ce qui donne la même chose que le résultat précédent. Une troisième manière de résoudre le problème est de simplement soustraire le temps passé dans B à servir d'intermédiaire entre A et C et faire comme si A avait envoyé sa demande 9h00m05.400s - 9h00m05.300s = .1s plus tard et reçu la réponse 9h00m05.550s - 9h00m05.500s = .05s plus tôt. Ceci donnerait:

a = 9h00m00.300s - (9h00m10.200s + .100s) = -10s
b = 9h00m00.350s - (9h00m10.600s - .050s) = -10.200s
Ajustement = (a+b)/2 = (-10s - 10.200s)/2 = -10.1s
Précision = (a-b)/2 = (-10s + 9.8s)/2 = -.100s

c) Lorsque la méthode de l'élection hiérarchique est utilisée, afin de trouver le serveur de plus haute priorité disponible pour être élu coordonnateur, on dit que le nombre de messages envoyés peut être de l'ordre de n^2 dans le pire cas. Décrivez un cas où cela se produirait? **(1 point)**

Si le réseau est lent et que chaque ordinateur déclenche une élection en même temps, chacun aura le temps d'envoyer un message à chaque autre ordinateur de plus haute priorité avant d'obtenir une réponse. Ceci représente n-1 + n-2 + ... 1 message, soit $n^2/2 - n$ messages. Ensuite, chaque ordinateur, ne recevant pas encore de message de réponse, peut vouloir se déclarer élu, avec n - 1 messages (ou un envoi à tous) pour se proclamer coordonnateur, ce qui représente $n \times (n-1) = n^2 - n$ messages.

Question 2 (5 points)

Trois transactions, T, U et V, s'exécutent concurremment. Le séquencement des opérations est le suivant:

T: Début
U: Début
T: Read(a)
U: Read(a)
T: Read(b)
T: Write(b,1)
T: Compléter
V: Début
V: Read(b)
U: Read(b)
U: Read(c)
U: Write(b,2)
U: Write(c,3)
U: Compléter
V: Read(c)
V: Write(c)
V: Compléter

a) Lesquelles des transactions T, U et V pourraient s'effectuer ainsi, si un contrôle de la concurrence par prise de verrou (partagé pour la lecture et exclusif pour l'écriture) est utilisé? Justifiez. **(2 points)**

T acquière un verrou en lecture sur a, ensuite partagé avec U. T acquière un verrou en lecture sur b et le promeut en verrou d'écriture et peut compléter, relâchant ses verrous. V prend un verrou de lecture sur b, ensuite le partage avec U. U acquière un verrou sur c et veut ensuite promouvoir son verrou sur b pour l'écriture mais ne peut le faire à cause de V. S'il abandonne, cela laisse libre cours à V. S'il attend, U et V se bloqueront mutuellement car V voudra le verrou sur c.

b) Lesquelles des transactions T, U et V pourraient être validées si une validation en reculant était utilisée pour vérifier la cohérence des transactions? Une validation en avançant? Justifiez. **(2 points)**

En reculant, au moment de compléter T, il n'y a pas de problème. Pour compléter U, il faut vérifier ce qu'il a lu (a, b, c) versus ce qui a été écrit par les transactions précédentes (T qui a écrit b). Il y a intersection et donc potentiellement conflit. U est abandonné. Ceci laisse le libre champ à V, puisque V a commencé après la fin de T et n'est donc pas concurrent avec T.

En avançant, il faut vérifier ce qui est écrit par T (b) versus ce que U a lu à ce moment (a). Il n'y a pas d'intersection et T peut compléter. Au moment de compléter U, ses écritures (b, c) intersectent les lectures de V (b) et U doit être abandonné, laissant le champ libre à V.

c) Un client effectue une série d'opérations dans le cadre d'une transaction avec plusieurs sous-transactions imbriquées. Indiquez lesquelles des variables a à e seront modifiées par la transaction ainsi que leur nouvelle valeur. **(1 point)**

```
Début T1
  Write(a, 2)
  Début T1.1
    Write(b, 4)
    Début T1.1.1
      Write(c, 6)
      Commettre T1.1.1
    Write(d,8)
    Abandonner T1.1
  Write(b,10)
  Début T1.2
    Write(e,3)
    Début T1.2.1
      Write(e,5)
      Abandonner T1.2.1
    Commettre T1.2
  Commettre T1
```

Après avoir enlevé les transactions abandonnées (et celles imbriquées). il reste:

```
Début T1
  Write(a, 2)
  Write(b,10)
  Début T1.2
    Write(e,3)
    Commettre T1.2
  Commettre T1
```

Les variables affectées sont donc a=2, b=10, et e=3.

Question 3 (5 points)

a) Un jeune finissant veut offrir un nouveau service internet et doit se monter une infrastructure au plus bas coût possible. Son infrastructure est constituée de deux serveurs de base de données redondants et de 20 serveurs Web. Chaque serveur de base de données est constitué d'un ordinateur, et de deux disques redondants en miroir. Chaque serveur Web est constitué d'un ordinateur et d'un disque. Son fournisseur Internet s'occupe de répartir les requêtes entre ses serveurs Web. Le matériel utilisé a été recyclé et est en mauvais état. A chaque instant, chaque disque a une probabilité d'être en panne de 0.2 et chaque ordinateur (hormis

son ou ses disques) a une probabilité de 0.1. Quelle est la probabilité que tout (incluant chaque élément redondant) soit fonctionnel en même temps, (aucun disque ou ordinateur en panne)? Quelle est la probabilité que le service soit complètement non disponible?

(2 points)

Il y a 22 ordinateurs, chacun opérationnel avec une probabilité de .9, et 24 disques, chacun opérationnel avec une probabilité de .8. La probabilité que tout soit fonctionnel est de $.9^{22} \times .8^{24} = 0.000465$. *Chaque serveur de base de données sera opérationnel si l'ordinateur est opérationnel ainsi que les disques. Les disques le sont sauf si les deux sont en panne, une probabilité de* $1 - .2 \times .2 = .96$. *Chaque serveur a donc une probabilité de* $0.96 \times 0.9 = 0.864$ *d'être opérationnel. Le service sera offert sauf si les deux serveurs de base de données sont en panne* $1 - (1 - 0.864)^2 = 0.9815$. *Ensuite, chaque serveur Web est opérationnel si son disque et ordinateur le sont, soit une probabilité de* $0.8 \times 0.9 = 0.72$. *Le service Web sera opérationnel sauf si tous les serveurs Web sont en panne, une probabilité de* $1-(1-0.72)^{20} = 8.7 \times 10^{-12}$. *Le service sera non disponible sauf si les serveurs Web et les serveurs de base de données sont disponibles* $1 - ((1 - 8.7 \times 10^{-12}) \times 0.9815) = 0.0185$.

b) Une transaction répartie effectue les opérations suivantes sur un des serveurs impliqués. Expliquez qu'est-ce qui devra être écrit dans le jounal qui permet la récupération en cas de panne, et à quel moment ceci devra être fait au plus tôt et au plus tard (i.e. après quelle opération)? **(2 points)**

1: lire a
2: écrire 40, b
3: lire c
4: écrire 60, d
5: préparer à commettre
6: commettre

Les lectures n'ont pas à être indiquées dans le journal. Les valeurs écrites pour b et d peuvent être écrites dès que reçues ou au plus tard juste avant d'écrire l'instruction de préparation dans le journal. L'opération de se préparer à commettre doit être écrite lorsque reçue, avant de répondre pour confirmer que le serveur est prêt à commettre. L'opération commettre doit être ajoutée dans le journal après avoir été reçue. Il est avantageux (mais pas strictement requis) de le faire rapidement, pour éviter d'avoir à revérifier son statut en cas de panne avant de l'avoir écrit.

c) Dans le cadre du travail pratique 2, un répartiteur demandait à des serveurs de compter les incidences des mots dans des morceaux de texte. Lorsqu'un serveur faisait défaut et ne fournissait pas la réponse, le répartiteur pouvait demander à un autre serveur de refaire le travail, offrant ainsi une tolérance aux pannes. Cependant, le répartiteur demeure un maillon faible. Proposez une architecture qui permette d'améliorer la résilience du répartiteur. **(1 point)**

Le répartiteur pourrait écrire les résultats obtenus sur disque et les recouvrer après un redémarrage. Il pourrait y avoir deux répartiteurs redondants, tous deux actifs, qui se répartissent les serveurs et qui s'informent mutuellement des progrès réalisés.

Question 4 (5 points)

a) Une entreprise opère 9 ordinateurs identiques, 3 pour chacun de trois départements très semblables. Chacun des 3 ordinateurs pour un département est affecté à un service particulier (serveur LDAP, serveur de fichiers, serveur de compilation) et ces trois serveurs ont un taux d'occupation moyen de 10%, 20% et 30% respectivement. Un administrateur de système, inspiré par le cours INF4410, propose de consolider ces serveurs en les virtualisant, ce qui permettrait de réduire le nombre d'ordinateurs requis. Le surcoût de la virtualisation est de 20%, ce qui prenait 1 seconde en prendrait 1.2. Deux solutions sont envisagées, conserver un ordinateur par département qui roule les trois machines virtuelles de ce département, ou avoir un nuage de 3 ordinateurs physiques sur lesquels seraient répartis les 9 machines virtuelles. Que deviendrait le taux d'utilisation moyen dans chaque cas, en supposant que la charge sur chaque serveur était assez uniforme dans le temps? Quelle solution vous semble la plus intéressante? Pourquoi? **(2 points)**

Sur 100 secondes, les trois serveurs en prenaient respectivement 10, 20 et 30, soit un total de 60. Avec le surcoût de la virtualisation, ceci deviendrait $60 \times 1.2 = 72$*, soit un taux d'occupation de 72%. Il n'y a pas de différence de taux d'utilisation entre les deux cas, trois fois plus de charge sur trois ordinateurs au lieu d'un seul. La solution du nuage est plus intéressante car elle permet une certaine tolérance aux pannes. Par contre, elle peut être un peu plus difficile à mettre en oeuvre, et ne sépare pas tout à fait aussi bien les activités des trois départements au niveau de la sécurité informatique.*

b) Sur le nuage de la compagnie Amazon, un service de répartiteur existe qui envoie les requêtes reçues à tour de rôle à un des serveurs disponibles. Le répartiteur reçoit de l'information sur les différents serveurs (taux d'utilisation des serveurs, et temps de réponse aux requêtes). Le répartiteur peut aussi décider d'instancier des serveurs supplémentaires ou de retirer des serveurs instanciés. Quel critère est-ce que le répartiteur utilise pour choisir le prochain serveur auquel envoyer une requête reçue? Quel critère utilise-t-il pour décider d'activer une instance supplémentaire de serveur? Pour retirer une instance de serveur? **(2 points)**

Le répartiteur envoie la requête reçue au serveur qui répond le plus rapidement, ou les distribue à tour de rôle lorsque la différence de temps n'est pas importante. Lorsque le taux d'utilisation des serveurs est trop élevé, de nouvelles instances sont ajoutées. Lorsque le taux d'utilisation est trop faible, des instances sont retirées.

c) Les solutions de virtualisation comme KVM offrent la virtualisation complète ou la paravirtualisation. Quels sont les avantages et limitations de ces deux alternatives? Dans quelle situation choisirait-on de préférence chacune? **(1 point)**

La paravirtualisation est plus rapide puisque l'opération demandée est directement déléguée à la machine physique, plutôt que d'émuler le comportement d'un dispositif physique particulier, comme une carte réseau, avant de déléguer le travail à la machine physique. Toutefois, la paravirtualisation n'est possible que lorsqu'on peut facilement configurer de cette manière la machine virtuelle. Lorsqu'on a plein contrôle sur la configuration de la machine virtuelle, la paravirtualisation est préférable puisqu'elle est plus rapide et ne requiert pas de support matériel. Par contre, s'il n'est pas possible de modifier la machine virtualisée, par exemple

parce qu'elle est fournie par un client qui ne sait pas que son ordinateur est virtualisé ou qui n'est pas intéressé ou capable de modifier sa configuration, alors la virtualisation complète sera nécessaire.

Le professeur: Michel Dagenais