

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)

EXAMEN FINAL

DATE: Vendredi le 16 décembre 2016

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) Sur les ordinateurs Linux, le fichier `resolv.conf` permet de lister plusieurs serveurs de nom (DNS), qui seront interrogés dans l'ordre selon lequel ils apparaissent dans le fichier, jusqu'à ce qu'une réponse soit obtenue. Ceci offre donc une tolérance aux pannes de serveur DNS par redondance, lorsqu'une traduction de nom à adresse IP est requise. Si 5 serveurs sont listés dans le fichier de configuration, combien de serveurs en panne par omission ce mécanisme peut-il tolérer avant de cesser de fonctionner? Si vous étiez dans un environnement hostile, où certains serveurs compromis fourniraient de mauvaises réponses, comment pourriez-vous modifier ce mécanisme pour aussi tolérer des pannes byzantines? Dans ce cas, combien de serveurs en panne (arrêtés ou compromis), pourriez-vous tolérer tout en maintenant le service opérationnel? **(2 points)**
- b) Un serveur de nom contient un CPU à un coeur ainsi qu'un disque. Chaque requête prend 150us (us pour microseconde) de temps CPU pour traiter la requête au complet lorsque la réponse est présente en mémoire centrale. Lorsque la réponse n'est pas présente en mémoire centrale, ce qui arrive pour 10% des requêtes reçues, il faut ajouter un 150us supplémentaire de temps CPU et une lecture au disque qui prend 500us (disque rapide de type SSD). Finalement, lorsque la réponse n'est pas en mémoire ni sur disque, ce qui arrive pour 1% des requêtes reçues, il faut ajouter à ces deux temps le temps pour une demande récursive au serveur plus haut dans la hiérarchie, ce qui ajoute 200us de temps CPU et 20ms (ms pour milliseconde) d'attente. Si le processus pour ce service traite séquentiellement les requêtes avec un seul fil d'exécution (thread), combien de requêtes par seconde peut-il servir? Si plusieurs fils d'exécution (thread) sont utilisés de manière à maximiser la performance, combien de requêtes par seconde peut-il servir et combien de fils d'exécution doit-on utiliser? **(2 points)**
- c) Il y a eu, au cours des dernières années, quelques épisodes où des attaques en déni de service ont été lancées contre les serveurs de nom (DNS) principaux sur l'Internet. Dans un tel cas, il peut arriver que les serveurs racines, complètement submergés, soient pratiquement inopérants. Cependant, la plupart des utilisateurs n'ont rien remarqué et ils ont pu continuer à utiliser la plupart des sites populaires. Toutefois, certains sites moins populaires, mais aussi quelques sites populaires qui font un usage plus dynamique des serveurs de nom, n'étaient effectivement pas accessibles en raison de l'attaque. Comment expliquez-vous cela? **(1 point)**

Question 2 (5 points)

- a) A l'occasion du nouvel an, vous désirez inviter à la maison les parents de votre conjoint. Cependant, ceux-ci, des ingénieurs, accordent une très grande importance à la précision du temps. A 10h05m00s, heure de votre domicile, vous envoyez un drone avec caméra lire le temps sur l'horloge atomique des parents qui n'habitent pas très loin. Rendu là, le drone filme l'horloge qui affiche 10h10m05s, se pose 5 minutes pour reposer ses moteurs, et filme à nouveau l'heure sur l'horloge atomique qui est maintenant 10h15m05s. Le drone revient à votre domicile à 10h14m00s, heure de votre domicile. Vous pouvez alors visionner le film et connaître les deux temps qui ont été filmés, à l'arrivée et au départ du drone de chez les parents. Quel ajustement apporterez-vous à l'horloge de votre domicile pour la synchroniser avec l'horloge atomique des

- parents suite à ces lectures? Quelle est l'incertitude sur cet ajustement? Si vous connaissez la direction et la force du vent ainsi que la localisation du domicile des parents, est-ce que cela vous permettrait d'obtenir un meilleur ajustement du temps? **(2 points)**
- b) Le 31 décembre prochain, une seconde sera ajoutée en fin de journée de sorte qu'il sera éventuellement 23h59m60s, et que seulement une seconde plus tard il sera 0h00m00s. Pourquoi l'organisme en charge de maintenir le temps fait-il une telle modification qui pourrait perturber le décompte de nombreux fêtards lors de la veille du jour de l'an? La compagnie Google, soucieuse d'aplanir le problème, offre un service public de serveurs de temps (NTP) et compte cacher cette seconde supplémentaire, qui pourrait causer des ennuis à plusieurs logiciels, en ralentissant le temps d'une seconde sur une période allant de 10 heures avant et 10 heures après la fin de la journée du 31 décembre. Vous désirez profiter de ce ralentissement du temps pour améliorer votre record de 20s pour courir un 100 mètres. Que deviendra ce temps de 20s lorsque mesuré en utilisant le temps ralenti par Google? **(2 points)**
- c) Lors du travail pratique 2, un répartiteur envoyait des tâches à plusieurs serveurs. Il pouvait arriver que certains serveurs soient tués au beau milieu de l'exécution d'une tâche. Comment est-ce que le répartiteur pouvait s'apercevoir d'une telle panne et comment pouvait-il réagir afin de tolérer cette panne? **(1 point)**

Question 3 (5 points)

- a) 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)**

T: Début

T: Read(a)

T: Write(b,1)

U: Début

U: Read(a)

U: Write(c,2)

V: Début

T: Compléter

U: Read(b)

V: Read(a)

V: Read(b)

V: Read(c)

V: Write(c,3)

V: Compléter

U: Compléter

- b) Une transaction répartie implique un coordonnateur, un client et plusieurs serveurs. Dans un premier cas, i), le client a ouvert une transaction (e.g., acheter des billets de spectacle pour sa

famille), spécifié plusieurs opérations pour la transaction et demandé de compléter la transaction. Il est maintenant en attente de la confirmation que la transaction est acceptée. Il attend un certain temps et ne reçoit toujours pas de réponse. Peut-il en conclure que la transaction est effectuée ou non? Peut-il demander à ce point d'annuler la transaction? Que peut-il faire? Dans un second cas, ii), le coordonnateur d'une transaction reçoit la demande du client de compléter la transaction et envoie à chaque serveur impliqué une demande à savoir s'ils sont prêts à commettre la transaction. Il reçoit des réponses (affirmatives) de tous les serveurs sauf 1 (dont le réseau est possiblement en panne). Peut-il relancer le serveur qui ne répond pas? Peut-il attendre indéfiniment? Que peut-il faire? **(2 points)**

- c) Sachant que vous avez complété le travail pratique 3 et êtes un expert avec OpenStack et Heat, on vous consulte car le gabarit suivant comporte un problème. Dites quelle ligne devrait être corrigée et comment. **(1 point)**

```
resources:
  web_nodes:
    type: OS::Heat::ResourceGroup
    properties:
      count: 2
      resource_def:
        #Définition des serveur.

  pool:
    type: OS::Neutron::Pool
    #Définition du Pool

  lbalancer:
    type: OS::Neutron::LoadBalancer
    properties:
      protocol_port: 8000
      members: {get_resource: web_nodes}
      pool_id: {get_resource: pool}
```

Question 4 (5 points)

- a) Le système informatique d'une centrale solaire commande l'inclinaison des panneaux solaires et doit offrir une très grande fiabilité. Un arrêt peut causer une diminution importante de la production électrique voire même concentrer les rayons du soleil au mauvais endroit et faire griller les maisons voisines. Le système est composé de 3 ordinateurs en redondance (Triple Modular Redundancy) qui reçoivent les mêmes entrées et dont les sorties sont passées par un voteur. Le voteur est très fiable et produit une sortie tant que deux des trois ordinateurs sont en bon ordre (donnent le même résultat). Chaque ordinateur a une probabilité de défaillance (donner un mauvais résultat) de 0.1. Quelle est la probabilité de défaillance de ce système de trois ordinateurs redondants? On vous propose de changer le voteur pour que, s'il obtient 3

- réponses différentes, il propagera en sortie la sortie d'un des ordinateurs pré-sélectionné (e.g., l'ordinateur #1, car il est plus neuf). Est-ce que cela augmente la disponibilité du système? Est-ce qu'il y a des inconvénients à faire cela? **(2 points)**
- b) Un dicton dit que les configurations de serveurs actif-passif sont plus simples à gérer mais moins performantes. Sur le système de fichiers CODA, chaque fichier est stocké sur plusieurs serveurs (typiquement 3). Lorsqu'un client veut lire un fichier, il lui suffit de le lire sur un seul serveur. Pour écrire un fichier, il doit l'écrire sur tous les serveurs, ou du moins ceux qui sont rejoignables puisque ce système est tolérant aux pannes. Est-ce que CODA utilise une configuration actif-passif ou actif-actif? Est-ce que le dicton s'applique ici? Donnez un exemple concret de performance différente et de cas complexe à gérer par rapport à un serveur seul? **(2 points)**
- c) Dans le processus de développement classique, chaque développeur d'un gros projet extrait du système de gestion de code source (e.g., SVN ou Git) les fichiers du programme et en obtient une copie sur son poste. Ensuite, celui-ci peut modifier quelques fichiers et effectuer la compilation sur son poste. Une compagnie qui dispose d'une grappe de serveurs de compilation pourrait permettre aux développeurs d'envoyer la demande de compilation sur ces serveurs (envoyer les fichiers source et recevoir les fichiers compilés). Avec le Google Build System, la situation est différente. Expliquez ces différences et en quoi permettent-elles de sauver du temps de compilation. **(1 point)**

Le professeur: Michel Dagenais