Tolérance aux pannes

Exercices pour le Module 10
INF8480 Systèmes répartis et infonuagique

Michel Dagenais

École Polytechnique de Montréal
Département de génie informatique et génie logiciel

Disponibilité

Trois ordinateurs offrent un service. Chaque ordinateur fait en moyenne une panne après 5 jours d'opération et cette panne prend 4 heures à réparer. Quelle est la disponibilité de ce système?

Disponibilité

Trois ordinateurs offrent un service. Chaque ordinateur fait en moyenne une panne après 5 jours d'opération et cette panne prend 4 heures à réparer. Quelle est la disponibilité de ce système?

La probabilité de ne pas être fonctionnel est de:

$$\frac{4\text{h}}{5 \times 24\text{h} + 4\text{h}} = 0.03$$

La probabilité d'avoir les trois ordinateurs simultanément en panne est de:

$$\binom{3}{3} \times (0.03)^3 \times (1 - 0.03)^0 = 1 \times (0.03)^3 \times 1 = 0.000027$$

La disponibilité est donc de $1 - 0.000027 = 0.999973$.

Serveurs DNS

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 i) par omission ce mécanisme peut-il tolérer avant de cesser de fonctionner? Si les serveurs pouvaient fournir de mauvaises réponses selon deux scénarios, ii) aléatoires et iii) byzantines, comment pourriez-vous modifier ce mécanisme pour aussi tolérer des pannes de type ii)? De type iii)? Dans chaque cas, ii) et iii), combien de serveurs en panne, (arrêtés ou compromis), pourriez-vous tolérer tout en maintenant le service opérationnel?

Serveurs DNS

- Avec des pannes i) par omission, il suffit d'un seul serveur opérationnel pour fonctionner, soit jusqu'à 4 serveurs en panne qui peuvent être tolérés.
- Pour les pannes ii) aléatoires, il faudrait comparer les réponses et retenir celle qui revient le plus souvent. En supposant que les probabilités de deux mauvaises réponses aléatoires identiques sont minimes, on peut tolérer 3 pannes.
- Pour les pannes iii) byzantines, il faudrait comparer les réponses et retenir aussi celle qui apparaît le plus souvent. Dans un tel cas, jusqu'à 2 serveurs en panne pourraient être tolérés (3 bonnes réponses contre 2 mauvaises). En effet, avec 3 serveurs compromis, la mauvaise réponse pourrait l'emporter.

Disques en miroir

On vous demande de choisir entre deux configurations pour le prochain serveur de fichiers de votre entreprise. La première configuration consiste en un ordinateur avec 4 disques en miroir. Chaque disque contient l'ensemble des données et un seul disque suffit donc. Le second système est constitué de deux ordinateurs, qui sont deux serveurs redondants, chacun étant connecté à deux disques en miroir et un seul disque suffit donc pour un serveur. La probabilité qu'un ordinateur soit opérationnel (hormis les disques) est de 0.95. La probabilité qu'un disque soit opérationnel est de 0.85. Quelle est la probabilité que le service soit disponible, pour chacune des deux configurations?

Disques en miroir

- Première configuration:
  - Le système de disque est disponible sauf si les 4 sont indisponibles: $1 - (1 - .85)^4 = 0.99949375$.
  - Le service sera disponible si les disques et l'ordinateur le sont: $0.99949375 \times 0.95 = 0.949519063$.
- Seconde configuration:
  - Sur un serveur, les disques seront disponibles sauf si les 2 sont indisponibles: $1 - (1 - .85)^2 = 0.9775$.
  - Un serveur sera disponible si l'ordinateur et les disques le sont: $0.9775 \times 0.95 = 0.928625$.
  - Le service sera disponible sauf si les deux serveurs redondants sont indisponibles: $1 - (1 - 0.928625)^2 = 0.994905609$.
- La deuxième configuration est donc nettement mieux pour réduire le temps d'indisponibilité. Elle est toutefois légèrement plus coûteuse puisqu'elle utilise deux ordinateurs plutôt qu'un, tout en conservant le même nombre de disques.

Services de fichiers

Avec CODA, pourquoi les utilisateurs doivent-ils parfois intervenir manuellement?

Services de fichiers

Avec CODA, pourquoi les utilisateurs doivent-ils parfois intervenir manuellement?

- Le contrôle de concurrence est optimiste sur CODA et les opérations ne peuvent être annulées.
- Ainsi, si deux utilisateurs ont mis à jour en même temps leur copie d'un fichier, le second à transmettre la mise à jour au serveur doit décider comment réconcilier les deux versions, plutôt que d'écraser le travail de l'autre.

Paxos

On veut appliquer l'algorithme du consensus de Paxos à l'élection d'un serveur primaire. Ce serveur primaire doit être unique en tout temps afin d'assurer la cohérence du système (e.g. attribuer les sièges d'une salle de spectacle à des clients qui se connectent via un réseau externe). Si le serveur primaire issu du consensus devient éventuellement non disponible, peut-on changer le consensus pour un nouveau serveur primaire? Comment peut-on distinguer entre un serveur primaire inopérant et un serveur primaire coupé du réseau? Comment empêcher un serveur primaire coupé du réseau de continuer à se considérer primaire (et vendre des sièges possiblement en double)? Donnez un exemple de deux propositions concurrentes où la seconde proposition commence alors que la première était presque terminée. Donnez un exemple de réseau de 5 serveurs participant au vote et partitionnés 2:1:2 et 2:3.

Paxos

Un consensus ne devrait normalement pas changer, le serveur primaire peut cependant devoir changer au fil du temps. Pour concilier les deux, on peut avoir un consensus sur le serveur primaire pour un intervalle donné. L'intervalle peut être dénoté par un rang (e.g. 14ème intervalle) ou par un temps de début. En cas de panne de ce serveur primaire, un nouveau consensus peut s'établir pour un nouveau serveur primaire pour le nouvel intervalle (e.g. 15ème intervalle) et tout le monde communique toujours avec le serveur primaire du plus récent intervalle. Le serveur primaire pourrait ne pas être en panne mais simplement coupé du réseau. On ne peut distinguer l'un de l'autre. Il peut alors être justifié d'élire un nouveau serveur primaire si l'ancien n'est pas rejoignable. Si l'ancien serveur primaire est en panne, il ne fait plus rien et ne cause pas de conflit.

Paxos

Si l'ancien serveur primaire est simplement coupé du réseau et continue à se considérer comme primaire, ce peut être problématique puisqu'il continue à se considérer en autorité pour effectuer des opérations critiques (attribution des sièges). Une solution que l'on retrouve souvent dans les configurations redondantes à deux serveurs (actif-passif) est que le nouveau serveur commande physiquement l'arrêt de l'alimentation électrique de l'ancien serveur pour prévenir ce genre de problème.

Paxos

Supposons 5 serveurs A, B, C, D, E. Au moment n=1, A se propose (v=A) comme serveur primaire et obtient une promesse de A, B et C, ce qui constitue un quorum. Presqu'en même temps, au temps n=2, E se propose (v=E) comme serveur primaire et obtient une promesse de D et E. Si A poursuit en demandant à D et E, il peut se faire ignorer ou se faire répondre qu'une proposition plus récente circule. En apprenant qu'une proposition plus récente circule, il peut abandonner. Autrement, il pensera que D et E sont en panne et, ayant déjà le quorum, A envoie un message demandant d'accepter n=1, v=A. Si ceci est effectué très rapidement, et que A reçoit la confirmation de A, B et C, la majorité est obtenue et le consensus atteint. Dans ce cas, E ne réussira pas à avoir un quorum de promesses et encore moins pour son acceptation.

Paxos

L'autre possibilité est que C reçoive la proposition de E avant la demande d'acceptation de A. Dans ce cas, il répond avec une promesse à la proposition de E, puisqu'elle est plus récente, et il ignorera la demande d'acceptation de A. A ne pourra obtenir un quorum d'acceptation. De son côté, E recevra une promesse de C avec l'information que sa promesse la plus récente acceptée était pour n=1, v=A. Il reprendra la proposition à son compte et enverra une demande d'acceptation pour n=2, v=A à tous et recevra la confirmation de C, D et E. Là encore, la majorité est obtenue et le consensus atteint.

Supposons les 5 mêmes serveurs A, B, C, D, E. Avec une partition du réseau 2:1:2, aucune partition ne regroupe une majorité de serveurs et il ne sera pas possible d'atteindre un consensus. Avec une partition 2:3, le consensus ne pourra être atteint que dans la partition de 3 serveurs, ce qui assure qu'un seul serveur primaire sera élu, et non pas un serveur par partition.

Raft

On veut appliquer l'algorithme du consensus de Raft à l'élection d'un serveur primaire pour une grappe de 5 serveurs. Le serveur 1 était le chef mais il vient de tomber en panne lorsqu'un *singe chaotique* le déconnecte. Décrivez comment se déroule l'élection si un seul membre déclenche l'élection? Si deux membres déclenchent une élection presqu'en même temps?

Raft

Ne recevant pas de battement de coeur (heartbeat) pendant un certain temps, les membres vont éventuellement conclure que le chef est inopérant. Ayant des délais aléatoires différents, un des membres déclenchera une élection avant les autres. Les autres accepteront sa candidature rapidement et il enverra alors à son tour un battement de coeur confirmant son statut. Si le réseau était simplement partitionné, et que le serveur 1 est toujours opérationnel, un nouveau chef pourrait néanmoins être élu si sa partition contient une majorité de membres. Il faudrait alors s'assurer de mettre hors service l'ancien chef.

Raft

Si deux membres déclenchent une élection en même temps, ils utiliseront le même numéro de terme et chaque autre membre ne votera qu'une seule fois. Si un candidat obtient une majorité, cela fonctionnera. Autrement, le vote ne fonctionnera pas et un nouveau vote, pour un nouveau terme sera déclenché.

Politique byzantine

Montrez qu'il est possible d'obtenir un consensus entre trois généraux byzantins dont un est fautif si les messages sont signés.

Politique byzantine

Montrez qu'il est possible d'obtenir un consensus entre trois généraux byzantins dont un est fautif si les messages sont signés.

Dans la mesure où les généraux qui agissent comme lieutenant peuvent vérifier la validité des ordres transmis, ils peuvent facilement s'entendre. Si le général en chef transmet des ordres contradictoires, les lieutenants peuvent s'en rendre compte. Si un lieutenant ment à propos des ordres reçus, il est tout de suite démasqué.

Mise à l'échelle des messages

Comparez les différentes garanties pour les messages de groupe (non fiable, fiable, atomique, totalement ordonnancé, causalement ordonnancé) en termes de mise à l'échelle et de nombre de messages, en supposant n membres et que peu de messages sont perdus?

Mise à l'échelle des messages

Comparez les différentes garanties pour les messages de groupe (non fiable, fiable, atomique, totalement ordonnancé, causalement ordonnancé) en termes de mise à l'échelle et de nombre de messages, en supposant n membres et que peu de messages sont perdus?

- **Non fiable**: un message en multi-diffusion par envoi de groupe.
- **Fiable** (localiser un répondant): un message en multi-diffusion et une réponse.
- **Atomique**: deux messages en multi-diffusion et n - 1 accusés de réception.
- **Ordonnancement total**: 2 messages pour le numéro de séquence plus un message en multi-diffusion; risque de surcharge du serveur de séquence pour la mise à l'échelle.
- **Ordonnancement partiel**: un message en multi-diffusion.