Intergiciels

1h45, documents autorisés

11 avril 2017

Les exercices sont indépendants.

1 Questions de cours (2 pt)

1. Dans le tp socket avec authentification, qu'est-ce que cela change de remplacer l'utilisation de socket datagramme (UDP) par un socket connecté (stream) ?
2. Quel service de base est nécessaire dans tous les intergiciels pour permettre la désignation des entités à distance : procédures, méthodes, files, thèmes, etc ?

2 Problème (1 pt / question)

On considère un système client-serveur dans lequel le serveur répond aux requêtes des clients et entretient des données internes représentant son état. L'exécution d'une requête modifie l'état du serveur. Par exemple, le service est une centrale de réservation de places de concert.

La tolérance aux pannes côté serveur est assurée par une organisation dite « serveur primaire – serveur de secours ». Ce schéma met en jeu deux machines interconnectées, un serveur primaire (SP) et un unique serveur de secours (SS). Le principe est que chacune des machines gère son propre état interne, et que ces états doivent être maintenus en cohérence afin qu'à tout moment SS puisse remplacer SP, en cas de défaillance de SP (on ne considère que des pannes franches, où le serveur s'arrête sèchement).

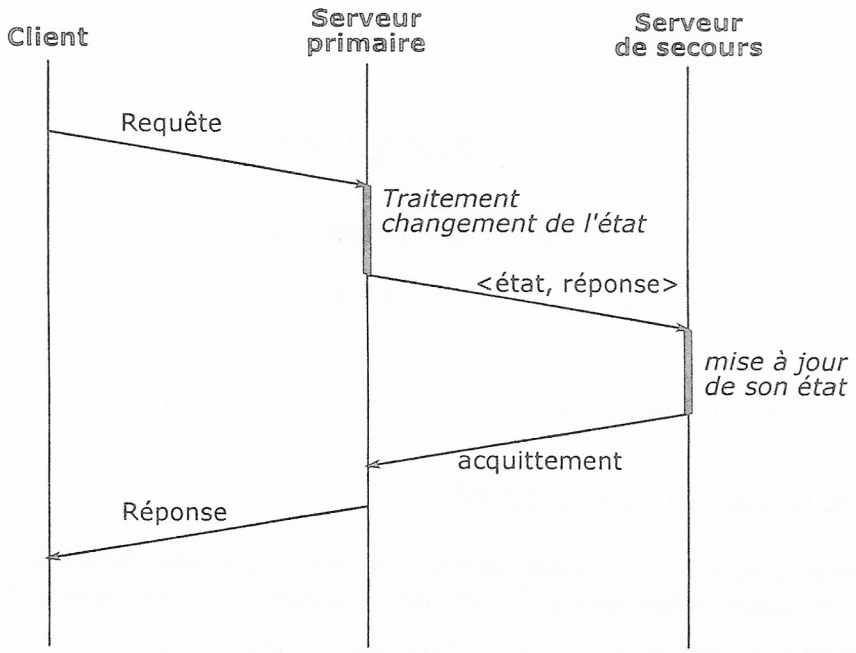
On propose le schéma de fonctionnement suivant : un client envoie une requête au serveur primaire SP. Celui-ci exécute la requête, modifie son état, envoie au serveur de secours SS une copie de la réponse et de l'état résultant du traitement de la requête, et se bloque. À la réception de ceci, SS met à jour son état et envoie un accusé de réception à SP. Lorsque SP reçoit cet accusé de réception, il envoie la réponse au client et devient prêt à traiter une nouvelle requête.

1. Pourquoi SP attend-il l'acquittement avant de répondre au client ?
2. Pourquoi SP fournit-il à SS, outre l'état résultant, la réponse qu'il va envoyer au client ?

2.1 Mise en œuvre avec des sockets

On souhaite réaliser l'architecture SS-SP avec des sockets.

3. Combien de liaisons distinctes faut-il entre SS et SP ?
4. Est-il préférable d'utiliser le mode connecté ou le mode non connecté ?



5. Comment l'un des serveurs peut-il détecter la panne de l'autre ?
6. On suppose que le client utilise lui-même des sockets pour envoyer ses requêtes au service. Comment peut-on rendre transparente la panne d'un des serveurs, afin que le client ignore lequel de SP/SS exécute effectivement la requête demandée ?

2.2 Mise en œuvre avec des RMI

On souhaite réaliser l'architecture SS-SP avec des RMI. Soit les entités `SS`, `SP`, `Requête`, `Réponse`, `État`, `AccuséDeRéception` correspondant aux entités du protocole proposé. On ne s'intéresse dans la suite qu'aux interactions entre SS et SP, et on ne se préoccupe pas de savoir comment le client accède au service. Noter aussi que le service effectivement réalisé par notre système est sans importance pour la suite (tout est caché dans les objets `Requête`, `Réponse`)

7. Pour chacune des six entités (`SS`, `SP`, `Requête`, `Réponse`, `État`, `AccuséDeRéception`), préciser si elles correspondent à des objets sérialisables ou accessibles à distance.
8. Proposer les deux interfaces `SS` et `SP` qui permettent la mise en œuvre de l'architecture décrite.
9. En s'appuyant sur les interfaces et entités proposées, donner le code algorithmique de `SP`, sans se préoccuper de la possibilité de panne.
10. En s'appuyant sur les interfaces et entités proposées, donner le code algorithmique de `SS`, sans se préoccuper de la possibilité de panne.
11. Comment `SP` peut-il détecter la panne de `SS` ?
12. Comment `SS` peut-il détecter la panne de `SP` ?

2.3 Mise en œuvre avec un intergiciel à messages (MOM/JMS)

On suppose maintenant que le client, SP et SS utilise un même intergiciel à messages pour communiquer.

13. Combien de `Destinations` faut-il, en précisant pour chacune si c'est une Queue ou un Topic.
14. Comment peut-on rendre transparente la panne d'un des serveurs, afin que le client ignore lequel de SP/SS exécute effectivement la requête demandée ?

3 Intergiciel à messages – JMS (4 pt)

On considère un système de supervision d'une chaîne de production. Le système est constitué de capteurs et de superviseurs (cf schéma). Il existe trois types de capteurs (température, pression, humidité), et un nombre arbitraire de capteurs physiques sont installés pour chaque type. Chaque capteur peut envoyer des informations (par exemple la température courante), ou des alertes (par exemple une panne ou une température anormale).

Les superviseurs (en nombre quelconque) peuvent aussi bien traiter des messages d'informations que des alertes. Une alerte ne doit être traitée que par un seul superviseur (le premier à la prendre en compte) pour une réaction unique. Les messages d'informations peuvent par contre être utiles à plusieurs superviseurs, selon le contrôle qu'ils appliquent à la chaîne de production.

1. Indiquer quelles sont les Destinations nécessaires, en précisant s'il s'agit de Topic ou de Queue.
2. Comment peut-on garantir qu'un message d'alerte sera pris en compte ? Ou inversement comment peut-on détecter qu'un message d'alerte a été ignoré ?
3. Quand un nouveau superviseur démarre, il peut avoir besoin des messages d'informations passés. Cela est-il possible avec votre architecture ?
4. Un message d'information peut intéresser plusieurs superviseurs, sans qu'on ne sache lesquels. Quelle difficulté cela pose-t-il ?