ED2 - Appel de procédure distante

Gérard Florin

Exercice 1 : Parallélisme et performance des communications en RPC

Un appel de procédure distante présente les caractéristiques temporelles suivantes (valeurs données à titre d'exemple):

Traitements client de préparation d'appel: 5 ms

- Traitements RPC émission de la requête coté client

Traitements de la souche client: 0,5 ms

Traitements logiciels réseaux coté client: 0,3 ms

- Réseau

Acheminement d'un message entre le client et le serveur 3 ms

- Traitements RPC réception de la requête coté serveur

Traitements logiciels réseaux coté serveur: 0,3 ms.

Traitements de la souche serveur: 0,5 ms

Traitements serveur de l'appel (exécution de la procédure): 10 ms

- Traitements RPC émission de la réponse coté serveur

Traitements de la souche serveur: 0,5 ms

Traitements logiciels réseaux coté serveur: 0,3 ms

- Réseau

Acheminement d'un message entre le serveur et le client 3 ms

- Traitements RPC réception de la réponse coté client

Traitements logiciels réseaux coté client: 0,3 ms.

Traitements de la souche client: 0,5 ms

Un client souhaite exécuter sur un même serveur deux appels de procédures distantes indépendantes (non reliées) ayant les caractéristiques temporelles précédentes.

1) On fait l'hypothèse que les RPC sont synchrones. Rappelez la définition d'un RPC synchrone.

2) On fait l'hypothèse que client et serveur travaillent séquentiellement. Le client exécute les deux appels au moyen d'un seul processus séquentiel. La procédure distante est exécutée pour les deux fois par le même processus. Quelle est la durée totale nécessaire à la réalisation des deux appels?

3) On cherche maintenant une exécution parallèle efficace en utilisant le parallélisme chez le client et chez le serveur. La procédure distante est toujours en mode synchrone. Le client et le serveur utilisent des processus sur la machine client et la machine serveur ("multi processing"). Les appels sont donc réalisés sur le client au moyen de deux processus parallèles associés aux deux appels indépendants. Les traitements sont réalisés sur le serveur au moyen de deux processus lancés en parallèle associés à chaque appel de procédure distante. On néglige les temps de commutation de contexte entre les processus. On se place dans le cas de machines mono processeur c'est à dire que les exécutions lancées en parallèles sont exécutées par un seul processeur (pseudo parallélisme).

Dessinez le diagramme d'ordonnancement des opérations dans le cas de l'ordonnancement parallèle le plus efficace des opérations client et serveur. Représentez par des segments verticaux sur un dessin les différentes opérations selon les quatre fils d'exécution associés aux quatre processus. Représentez également les messages de requête et de réponse? Quelle est la durée totale du traitement des deux appels?

4) On suppose maintenant que l'on dispose d'un RPC asynchrone. Rappelez la définition d'un RPC asynchrone ?

5) Le RPC asynchrone ne suppose pas l'existence de parallélisme sur le site client pour améliorer les performances. On considère dans cette question, que les clients et serveurs sont purement séquentiels. Le processus client réalise deux appels en RPC asynchrone et le processus serveur réalise les deux traitements des appels asynchrones. Dessinez l'ordonnancement le plus efficace des opérations pour les deux processus associés au client et au serveur. Représentez les messages échangés entre les deux processus. Quelle est la durée totale du traitement des deux appels ?

Exercice 2 : Traitement des pertes de messages

On souhaite réaliser très efficacement des appels de procédure distants en se plaçant non pas au dessus d'une couche transport fiable mais au dessus d'une couche réseau sans contrôle d'erreur.

Proposez des solutions efficaces au problème des pertes de messages d'appel et de réponse.

Exercice 3 : Exécutions très longues

On souhaite exécuter en mode appel de procédure distante des traitements de durée non prévisible (par exemple des interrogations distantes de bases de données dont les réponses peuvent prendre des durées très variables et très longues).

Proposez une solution pour le contrôle de l'exécution distante, en particulier pour pouvoir distinguer une panne de serveur d'une exécution longue.

Exercice 4 : Appels en parallèle (redondance massive)

L'appel distant active une seule procédure à distance à chaque fois. Un utilisateur peut souhaiter faire réaliser n exécutions à distance simultanément en lançant en diffusion une requête.

Explicititez le schéma du fonctionnement ainsi défini.

Utilisez le dans le cas de données répliquées. On suppose que l'on a n serveurs de données en copies multiples (pour la sécurité) qui peuvent réaliser les mêmes opérations lire et écrire. Comment sont réalisées les lectures, les écritures. Quelle propriété minimum doivent satisfaire les requêtes pour maintenir la cohérence des données si l'on considère qu'il n'y a pas de partage de données entre plusieurs activités?

Exercice 5 : Serveurs multiples (redondance sélective)

Dans certains types de service on souhaite disposer de plusieurs instances du même service sur plusieurs machines différentes (par exemple pour un service de compilation).

L'appel distant active une seule exécution à distance à chaque fois.

Dans quel but utiliserait-on de tels services à instances multiples?

Proposez une organisation pour rendre efficacement un tel service.