



Communication dans les systèmes répartis

Exercices pour le Module 4

INF8480 Systèmes répartis et infonuagique

Michel Dagenais

École Polytechnique de Montréal
Département de génie informatique et génie logiciel

Attaque concertée

Les bleus sont divisés en deux camps de part et d'autre des verts et communiquent par messenger. S'ils attaquent ensemble, la victoire est à eux. Chacun n'attaquera que s'il a l'assurance d'un accord mutuel confirmé. Quel protocole peuvent-ils utiliser pour s'assurer de manière absolue que l'autre accepte d'attaquer avec eux?



Attaque concertée

Les bleus sont divisés en deux camps de part et d'autre des verts et communiquent par messenger. S'ils attaquent ensemble, la victoire est à eux. Chacun n'attaquera que s'il a l'assurance d'un accord mutuel confirmé. Quel protocole peuvent-ils utiliser pour s'assurer de manière absolue que l'autre accepte d'attaquer avec eux?

Ce n'est pas possible!



Ordre des mises à jour

Donnez un exemple pour lequel l'ordre d'un message de groupe (multicast) de deux clients n'est pas important? Un ou c'est important?



Ordre des mises à jour

Donnez un exemple pour lequel l'ordre d'un message de groupe (multicast) de deux clients n'est pas important? Un ou c'est important?

L'ordre n'est pas important pour une requête où un seul ordinateur peut envoyer la réponse (ou lorsque la valeur demandée ne change pas et peut venir de n'importe quel ordinateur). Par contre, pour des mises à jour de serveurs répliqués, il est important que tous les serveurs voient les mises à jour dans le même ordre lorsqu'elles s'intersectent.



Tempête de messages

Un réseau local comporte 250 ordinateurs et un routeur qui offre le service DHCP. Si au démarrage (après une panne d'électricité) chaque ordinateur demande à chaque autre s'il offre le service DHCP, combien de messages seront-ils envoyés à cette fin? Comment cela se compare-t-il à l'utilisation de diffusion générale?



Tempête de messages

Un réseau local comporte 250 ordinateurs et un routeur qui offre le service DHCP. Si au démarrage (après une panne d'électricité) chaque ordinateur demande à chaque autre s'il offre le service DHCP, combien de messages seront-ils envoyés à cette fin? Comment cela se compare-t-il à l'utilisation de diffusion générale?

Si chaque ordinateur envoie un message à chaque adresse, il peut envoyer 251 messages s'il sait l'intervalle ou sinon pourrait essayer les 254 adresses possibles. Le nombre total de messages serait de l'ordre de n au carré, soit $250 * 254 = 63500$. Autrement, chaque ordinateur pourrait n'envoyer qu'un seul message en diffusion générale pour 250 messages. Dans tous les cas, il y aura aussi 250 réponses.



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.



Service de vote

Un service de vote permet de voter (candidat et numéro d'électeur), et d'obtenir le résultat (nom du candidat et nombre de votes). Donnez le prototype de ces fonctions. Identifiez les paramètres d'entrée et de sortie.



Service de vote

Un service de vote permet de voter (candidat et numéro d'électeur), et d'obtenir le résultat (nom du candidat et nombre de votes). Donnez le prototype de ces fonctions. Identifiez les paramètres d'entrée et de sortie.

```
void Vote(char *nom, int identificateur);  
void Resultat(char **nom, int *nombre);
```

La première fonction a deux paramètres d'entrée alors que la seconde a deux paramètres de sortie.



Sémantique des appels à distance

Les électeurs veulent être certains que leur vote soit pris en considération. Quelle sémantique peut les satisfaire?



Sémantique des appels à distance

Les électeurs veulent être certains que leur vote soit pris en considération. Quelle sémantique peut les satisfaire?

La sémantique peut-être ne suffit pas. Par contre au moins une fois est acceptable dans la mesure où le numéro d'électeur sert déjà à filtrer les répétitions.



Réseau partiellement asynchrone

Un réseau avec paquets qui peuvent être perdus est utilisé pour obtenir une sémantique au moins une fois. L'implantation peut ou non utiliser une hypothèse de délai maximal. Quel est l'effet sur l'implantation?



Réseau partiellement asynchrone

Un réseau avec paquets qui peuvent être perdus est utilisé pour obtenir une sémantique au moins une fois. L'implantation peut ou non utiliser une hypothèse de délai maximal. Quel est l'effet sur l'implantation?

En retransmettant passé un délai maximal, ceci élimine en théorie la possibilité de recevoir deux réponses, une tardive et une pour la retransmission.



Création du code pour les proxy

Comment peut-on générer un proxy pour une méthode comme Vote dans un langage comme C++ qui ne supporte pas la réflexion?



Création du code pour les proxy

Comment peut-on générer un proxy pour une méthode comme `Vote` dans un langage comme C++ qui ne supporte pas la réflexion?

Le code pour le proxy est généré à partir du IDL. Le proxy contient une référence réseau vers l'objet réel et une méthode `Vote` qui envoie au processus de la référence réseau: le numéro d'interface, le numéro de méthode, l'identificateur de l'objet, le nom, et le numéro d'électeur. La procédure d'envoi identifie le type de l'ordinateur (gros/petit boutien) et retransmet le tout jusqu'à obtention d'un accusé de réception.



Performance d'un appel à distance

Un client prend 5ms pour calculer les arguments d'une requête RPC, le serveur prend 10ms pour traiter la requête. Chaque envoi ou réception de paquet demande 0.5ms au système d'exploitation et le temps d'envoi d'un message sur le réseau est de 3ms. L'encodage et le décodage des arguments prend 0.5ms par message. Le client et le serveur roulent chacun sur un seul coeur. Quel est le temps pour effectuer deux requêtes si le client est séquentiel? Si le client est multi-fils mais le serveur demeure séquentiel?



Performance d'un appel à distance

calculer args + encoder + envoi + transmission + réception + décodage
+ traitement + encoder + envoi + transmission + réception + décodage
= $5 + .5 + .5 + 3 + .5 + .5 + 10 + .5 + .5 + 3 + .5 + .5 = \mathbf{25ms}$, soit
50ms pour deux requêtes séquentielles.

En multi-fils, le client commence sa seconde requête après l'envoi à 6ms.
Sa seconde requête est envoyée après un autre 6ms, soit à 12ms. Le
serveur reçoit la première requête à 9ms, la traite et la renvoie à 21ms.
La seconde requête est déjà là et peut être traitée à ce moment. La
seconde réponse est envoyée à 33ms, et parvient au client à 36ms, pour
un total de **37ms**.



CORBA sans constructeur

Pourquoi CORBA et les systèmes similaires n'offrent-ils pas de constructeurs



CORBA sans constructeur

Pourquoi CORBA et les systèmes similaires n'offrent-ils pas de constructeurs

CORBA offre une interface et la notion d'instance n'est donc pas automatique. Le cas par défaut est une instance unique qui se confond avec l'interface. Au besoin il peut y avoir une méthode d'une interface qui exporte des objets réseau associés à une même interface.



Taille des messages

Vous devez programmer un système d'appel de procédure à distance pour vérifier la disponibilité et le prix de certains items:

```
void inventaire(in string id, out unsigned long  
quantity, out unsigned long price);
```

où `id` est un identifiant typiquement de **10 caractères**, la quantité environ **12** et le prix est un entier (en centièmes de dollars) aux environs de **\$100 (10000 cents)**. Vous avez le choix entre CORBA et gRPC. Quels sont les avantages de chacun. Si on ne tient pas compte des entêtes (e.g. HTTP2) mais seulement de l'encodage des arguments, CDR versus protobuf, lequel sera plus compact?



Taille des messages

Vous devez programmer un système d'appel de procédure à distance pour vérifier la disponibilité et le prix de certains items: `void inventaire(in string id, out unsigned long quantity, out unsigned long price);` où `id` est un identifiant typiquement de **10 caractères**, la quantité environ **12** et le prix est un entier (en centièmes de dollars) aux environs de **\$100 (10000 cents)**. Vous avez le choix entre CORBA et gRPC. Quels sont les avantages de chacun. Si on ne tient pas compte des entêtes (e.g. HTTP2) mais seulement de l'encodage des arguments, CDR versus protobuf, lequel sera plus compact?

CORBA et gRPC sont relativement semblables. Les deux permettent d'interfacer facilement des programmes écrits en différents langages et ont été développés avec une attention particulière pour l'optimisation. CORBA est mieux établi et bien normalisé. gRPC est plus moderne et permet une plus grande flexibilité au niveau de la compatibilité (vers l'avant et vers l'arrière) car les champs sont auto-identifiés.

CORBA prendra pour son encodage, outre le code identifiant la fonction, la longueur de la chaîne `id` arrondie au multiple de 4 octets supérieur plus 4 octets pour stocker la longueur, soit 16 octets pour un identifiant de 10 caractères, 4 octets pour la quantité et 4 octets pour le prix, pour un total de **24 octets**.

De son côté, protobuf demandera 1 octet pour identifier le champ et le type combinés pour `id`, 1 octet pour la longueur, 10 octets pour l'identifiant, 1 octet pour le champ et le type combinés et 1 octet varint pour la quantité, 1 octet pour le type et champ combinés et 2 octets pour le prix (10000 cents), pour un total de **17 octets**.

gRPC offre donc un compromis très intéressant de flexibilité (champs et types auto-identifiés), de simplicité d'encodage et de compacité.