

Projet – Système de discussion tolérant aux pannes

1 Introduction

On désire réaliser un système de discussion (*chat*) à tolérance de panne. Les discussions sont organisées en *canaux*, c'est-à-dire que tous les utilisateurs connectés à un canal reçoivent les messages adressés au canal.

2 Description

Le système proposé repose sur 3 acteurs (c'est-à-dire 3 programmes distincts).

- Le **client**, lancé par l'utilisateur, demande d'abord aux serveurs d'annuaire l'identité des serveurs pour le canal souhaité. Il s'abonne ensuite auprès des serveurs du canal. Le client peut dès lors transmettre les messages saisis au clavier vers les serveurs du canal et afficher les messages reçus des serveurs.
- Le **serveur de canal** s'enregistre lors de son démarrage auprès des serveurs d'annuaire pour le canal fourni en paramètre. Il peut ensuite attendre les demandes d'abonnement des différents clients. Lorsqu'un message est reçu, le serveur de canal le rediffuse vers les autres clients.
- Le **serveur d'annuaire** maintient une liste à jour des serveurs associés à chaque canal, afin de servir les demandes des clients.

Pour assurer la tolérance aux pannes, les différents serveurs (canal et annuaire) doivent être répliqués. Ainsi, le client interroge plusieurs serveurs d'annuaire en parallèle et retient par exemple la première réponse reçue. De même, les serveurs de canal s'enregistrent auprès de tous les serveurs d'annuaire. Lorsqu'un utilisateur saisit un message, le client l'envoie à tous les serveurs du canal, qui le rediffusent aux clients des autres utilisateurs¹.

3 Tolérance aux pannes

L'ensemble du projet doit être tolérant aux pannes. Cette tolérance s'entend comme la capacité d'initier ou de continuer une discussion sans dégradation des fonctionnalités si un composant quelconque (serveur d'annuaire ou serveur de canal) devient indisponible. Cette tolérance s'obtient par deux points :

- redondance des serveurs (au moins deux instances de chaque serveur)
- échanges réalisés en parallèle avec les serveurs redondants, grâce aux communications non bloquantes offertes par le protocole UDP que vous utiliserez exclusivement.

Pour simplifier, on ne tiendra compte que du cas dégradé (défaillance d'un serveur) et pas de la reprise après panne (c'est-à-dire si un serveur est remis en service après une défaillance). De même, des messages peuvent se perdre, on ne demande pas de garantir que tout message sera reçu.

4 Configuration

Les seuls éléments de configuration pour les clients et les serveurs de canal sont les identités des serveurs d'annuaire. Ces identités sont des couples *<adresse, port>*, où *adresse* correspond à un nom, une adresse IPv4 ou IPv6, et *port* correspond à un port UDP afin de vous permettre de développer et tester tous les programmes même si vous n'avez qu'un seul ordinateur (turing) à votre disposition.

Les serveurs d'annuaire n'ont a priori pas besoin de configuration.

1. Chaque client ne devra bien sûr afficher le message qu'une seule fois.

5 Exemple

Cet exemple a pour but de donner une idée du fonctionnement pour l'utilisateur, ainsi que donner des idées pour une interface utilisateur minimale. On suppose ici que les serveurs d'annuaire et du canal demandé ont déjà été lancés.

| | |
|-----------------------|--|
| \$ whoami | <i>qui suis-je ?</i> |
| toto | <i>mon nom de login est « toto »</i> |
| \$ client m2csmi | <i>je démarre un client pour le canal « m2csmi »</i> |
| m salut les copains ! | <i>commande « m » : j'envoie un message</i> |
| titi: salut toto | <i>réponse de l'utilisateur « titi »</i> |
| tata: hello toto | <i>réponse de l'utilisateur « tata »</i> |
| m au revoir | <i>commande « m » : j'envoie un nouveau message</i> |
| q | <i>commande « q » : je quitte le client</i> |
| \$ | <i>je reviens au shell</i> |

6 Extensions possibles

Il est possible d'étendre le projet dans plusieurs directions. Voici par exemple quelques idées (non limitatives) pour meubler votre temps libre :

- les serveurs (de canal ou d'annuaire) maintiennent des informations d'abonnement/enregistrement qui devraient être tenues à jour en cas de défaillance. Par exemple, si un serveur de canal devient indisponible, il faudrait que les serveurs d'annuaire (ainsi que les clients) le détectent. De même, si un client devient défaillant, il faudrait le supprimer des abonnements au canal ;
- les informations de configuration (identité des serveurs d'annuaire) pourraient être stockées dans un fichier pour être facilement modifiables ;
- un serveur de canal pourrait gérer plusieurs canaux ;
- un client pourrait s'abonner à plusieurs canaux ;
- la reprise après défaillance pourrait être gérée ;
- le système pourrait garantir qu'aucun message ne soit perdu.

7 Travail demandé

Le projet est à réaliser sur la machine *turing* en langage C ou C++ à l'aide des *sockets* (vous n'utiliserez pas d'autre mécanisme de programmation réseau). Les différents programmes pourront être lancés sur *turing* grâce à des ports UDP distincts. Bien évidemment, vos programmes doivent être compatibles IPv4 et IPv6.

Pour simplifier, on ne tiendra pas compte des problèmes de sécurité et d'authentification.

Il vous est demandé :

- de décrire et de justifier l'architecture du système que vous concevez, les éventuels fichiers nécessaires et leur format, les principales structures de données, les algorithmes, etc. ;
- de décrire les protocoles que vous implémentez entre les différents acteurs, en justifiant vos choix. Votre description doit être suffisamment précise pour permettre à quelqu'un d'autre que vous de programmer un des acteurs ;
- de programmer les composants de l'application en langage C ou C++ sous Unix, avec les sockets.

8 Modalités de remise

Ce projet est à réaliser individuellement.

Vous déposerez sur Moodle, dans l'espace de devoirs prévu à cet effet, votre projet (documentation, sources, Makefile) sous forme d'une archive au format « *login.tar.gz* » où *login* est votre nom de login sur Turing. Vous prendrez soin de supprimer tous les fichiers binaires (exécutables, objets).

Date limite : à convenir.

Sauf contre-ordre, une soutenance est prévue.