Laboratoire de programmation repartie PRR

Objectifs

- Concevoir et simplifier une architecture répartie.
- Analyser la problématique à un lieur de service.

Enoncé

Dans ce laboratoire il est question de réaliser un lieur redondant entre des clients et des serveurs d'application.
L'objectif du lieur étant d'associer un client à un service fourni par l'un des serveurs d'application disponible.

Comportement des clients

Le client doit obtenir l'adresse et le port du service fournit par un serveur auprès d'un lieur.
Par la suite il pourra alors utiliser l'adresse obtenue pour réclamer directement le service auprès du serveur.
Le client va choisir un lieur aléatoire et lui formuler sa demande. Si le lieur ne répond pas, le client va s'arrêter.
Si le service est inatteignable, le client le fera savoir au lieur et se terminera.

Les arguments pour le lancement d’un client doivent etre les suivants :

- 1er argument = port d’écoute du client
- 2eme argument = type de service que le client va utiliser
- 3eme arguement = ip du lieur
- 4eme arguement = port d’écoute du lieur
  les arguments 3 et 4 peuvent être répétés si nous avons plus d’un lieur

Exemple de paramètres minimaux pour le lancement d’un client

> *2226 1 127.0.0.1 2222*

Comportement des serveurs

Lors du démarrage, tout serveur s’inscrit auprès d’un lieur en lui transmettant son adresse IP, son port de service
ainsi que le type du service rendu. Il va ensuite répondre aux demandes des clients et aux demandes d’existances des serveurs.

Les arguments pour le lancement d’un serveur doivent etre les suivants :

- 1er argument = port d’écoute du serveur
- 2eme argument = type de service
- 3eme arguement = ip du lieur
- 4eme arguement = port d’écoute du lieur
  les arguments 3 et 4 peuvent être répétés si nous avons plus

d’un lieur

Exemple de paramètres minimaux pour le lancement d’un serveur

> *2227 1 127.0.0.1 2222*

Comportement des lieurs

Lors du démarrage, le lieur va demander la liste des services existant à un autre lieur opérationnel. Il va ensuite répondre aux requêtes des serveurs, des clients et des autres lieurs (voir protocole).

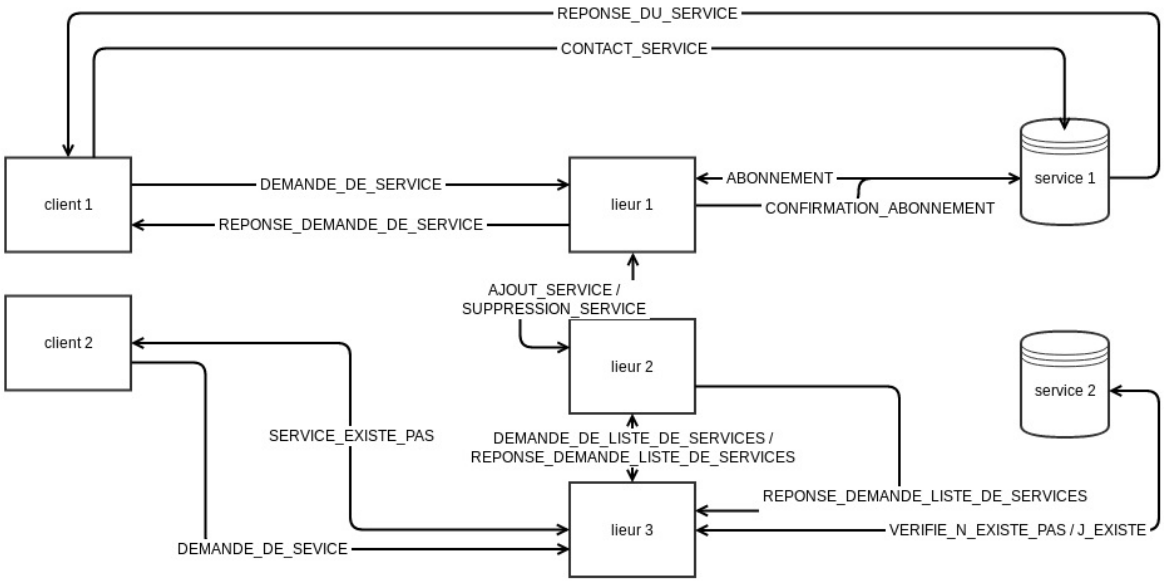
Les arguments pour le lancement d’un lieur doivent etre les suivants :

- 1er argument = port d’écoute du lieur
- 2eme argument = port d’écoute du lieur pour l’envoi/réception des messages de vérifications d’existence
- (optionel) 3eme arguement = ip d’un autre lieur
- (optionel) 4eme arguement = port d’écoute d’un autre lieur
  les arguments 3 et 4 peuvent être répétés si nous avons plus d’un lieur

Exemple de paramètres minimaux pour le lancement d’un lieur

> *2222 2223 127.0.0.1 1111*

Protocole



CONTACT_SERVICE

Ce paquet est envoyé quand nous voulons faire une requête à un service.

il est constitué de la manière suivante :

> *[type de paquet][longueur du message][message]*

REPONSE_DU_SERVICE

Réponse du service questionné.

Envoyé après réception de « CONTACT_SERVICE »

il est constitué de la manière suivante :

> *[type de paquet][longueur du message][message]*

DEMANDE_DE_SERVICE

Ce paquet est envoyé à un lieur quand un client veux accéder à un service.

Il est constitué de la manière suivante :

> *[type de paquet][type de service demandé]*

REPONSE_DEMANDE_DE_SERVICE

Réponse envoyée par le lieur après une demande de service d'un client.
Envoyé après réception de « DEMANDE_DE_SERVICE »
Il est constitué de la manière suivante :

> *[type de paquet][IP du service][port du service]*

SERVICE_EXISTE_PAS

Cas client → lieur : Paquet envoyé au lieur si un service n'a pas été atteint par le client.
Il est constitué de la manière suivante :

> *[type de paquet][type du service][IP du service][port du service]*

Cas lieur → client : Paquet envoyé si le lieur ne connait pas le type de service demandé.
Envoyé après réception de « DEMANDE_DE_SERVICE »
Il est constitué de la manière suivante :

> *[type de paquet]*

ABONNEMENT

Demande d'adhésion d'un service à un lieur.

Il est constitué de la manière suivante :

> *[type de paquet][type de service]*

CONFIRMATION_ABONNEMENT

Ce paquet est envoyé comme confirmation d'adhésion d'un service à un lieur, une fois ce paquet reçu par le service, ce dernier tournera dans une boucle infinie.
Envoyé après réception de « ABONNEMENT »
Il est constitué de la manière suivante :

> *[type de paquet]*

DEMANDE_DE_LISTE_DE_SERVICES

Ce paquet est envoyé à un lieur quand un autre lieur démarre et met à jour sa liste de service.
Il est constitué de la manière suivante :

> *[type de paquet]*

REPONSE_DEMANDE_LISTE_DE_SERVICES

Ce paquet est envoyé en réponse à la demande de mise à jour d'un lieur.
Envoyé après réception de «DEMANDE_DE_LISTE_DE_SERVICES»
Il est constitué de la manière suivante :

> *[type de paquet][nombre de service][type de service][IP du service][port du service]*

AJOUT_SERVICE

Ce paquet est envoyé par un lieur aux les autres lieurs après réception du paquet « ABONNEMENT »
Il est constitué de la manière suivante :

> *[type de paquet][type de service][IP du service][port du service]*

SUPPRESSION_SERVICE

Ce paquet est envoyé arpès l'émission du paquet « VERIFIE_N_EXISTE_PAS » qui n'a pas donné de réponse « J_EXISTE » . Ce paquet est envoyé aux autres lieurs pour leur dire de supprimer le service incriminé.
Il est constitué de la manière suivante :

> *[type de paquet][type de service][IP du service][port du service]*

VERIFIE_N_EXISTE_PAS

Ce paquet est envoyé par le lieur après réception du « SERVICE_EXISTE_PAS » . Il est envoyé du lieur vers le service incriminé pour vérifier si ce dernier est bien injoignable. Il est constitué de la manière suivante :

> *[type de paquet]*

J_EXISTE

Ce paquet est envoyé par le service après réception du paquet « VERIFIE_N_EXISTE_PAS » il permet de confirmer son existence. Il est

constitué de la manière suivante :

> *[type de paquet]*

Tests effectués

Le tableau suivant présente les tests qui seront effectués.

Tests	**Resutats**	**Commentaires **
Un serveur doit se souscrire à un lieur de la liste des lieurs quand il démarre	succès	Ok
Un client doit demander un service à un lieur quand il démarre puis utiliser le service	succès	Ok
Un client doit informer le lieur quand un service donné ne répond pas	succès	Ok
Un lieur doit envoyer une confirmation aux services quand il s’inscrit	succès	Ok
Un lieur doit notifier les autres lieurs quand un service s’inscrit à lui et ils doivent se mettre à jour	succès	Ok s’ils se connaissent

Un lieur doit vérifier l’existence d’un service si un client lui indique qu’il ne répond pas et doit le supprimer de sa liste de services	succès	Ok
Un lieur doit notifier les autres lieurs quand un service est indisponible et ils doivent se mettre à jour	succès	ok s’ils se connaissent
le serveur doit fournir son service d’echo aux clients après s’être souscrit au lieur	succès	Ok
les serveurs doivent répondre aux demande d’existence des lieurs après s’être souscrit à un lieur	succès	Ok
Un lieur doit être obténu aléatoirement à partir d’une liste dans le client et le serveur	succès	Aucun
Un lieur ou un serveur doit être redemarré aussitôt qu’il est tombé en panne	-	Non pris en compte, redemarrage manuel
Au démarrage et au redémarrage, un lieur doit se mettre à jour par rapport à un autre lieur disponible	succès	lancement du 1er lieur, lancement du service, lancement du second lieur. le second lieur possède le service dans sa liste

Un lieur doit envoyer envoyer sa liste de service quand un autre lieur la lui demande	succès	Ok
Le client doit s’arrêter ou attendre un délai après qu’il redemande un service inconnu auprès d’un lieur	succès	le client s’arrête si le lieur ne connais pas le service demandé
Après notification d’un client qu’un service ne répond pas, un lieur doit le vérifier et notifier les autres lieurs	succès	Ok
Le lieur doit distribuer les services de même type de façon cyclique entre les clients	succès	
Deux clients notifient presque simultanément qu’un service ne répond pas	succès	Le lieur utilise un autre port pour l’envoi et la récéption du message de vérification d’un service, il ne recevra donc pas le 2ème message du client dans le receive de la confirmation d’existence du service.

Simulation effectuée

Nous avons fait une simulation mettant en place :

- un lieur avec les parmatres suivants

        - port principal : 2222
        - port de verification : 2223
- un client
        - port : 2226
        - type de service voulu : 1
        - ip du lieur : 127.0.0.1
        - port du lieur : 2222
- un service
        - port : 2224
        - type de service : 1
        - ip du lieur : 127.0.0.1
        - port du lieur : 2222

Nous lançons nos jar dans cette ordre :

1. lieur
2. service
3. client

client

Démarrage du client

Le client va demander le service1 au lieur:

Service: ip 127.0.0.1, port 2222

Reponse du lieur recue

Le service a été trouvé il est joignable a l’adresse: 127.0.0.1:2224

Message envoyé au service

Reponse du serveur reçue

taille 4

0 : 1

1 : 1

2 : 1

3 : 1

serveur

Démarrage du serveur

Tentative de souscription au lieur:

Service: ip 127.0.0.1, port 2222

Confirmation de souscription reçue

Attente d’une nouvelle demande d’un client

Reception d’une nouvelle demande du client 127.0.0.1 2226

lieur

Démarrage du lieur

Reception de la liste des services

La liste des services est à jour

Attente d’une nouvelle demande…

Liste actuelle

Nouvelle demande recue

Type de message: ABONNEMENT

Nouvelle souscription du service:

Service: id 1, ip 127.0.0.1, port 2224

Notification aux autres lieurs de l’ajout du service

Envoi de la confirmation de souscription au service

Attente d’une nouvelle demande…

Liste actuelle

Service: id 1, ip 127.0.0.1, port 2224

Nouvelle demande recue

Type de message: DEMANDE_DE_SERVICE

Envoi du service au client

Le cas montré ici est le cas le plus simple. Nous constatons que nous suivons bien le scénario voulu.