



Activité 11.2 : Appels de fonctions à distance en utilisant des sockets

Objectif scientifique

Le but de ce projet est d'étudier et de comprendre la méthode et le fonctionnement correct de RPC en envoyant et en recevant des informations, tout cela avec l'appel à des fonctions distantes en utilisant des sockets. En plus de donner une explication correcte du programme vu en classe.

Plan du rapport

Tout d'abord, nous avons besoin d'une mise en contexte du fonctionnement du programme, ainsi que de la manière dont les fonctions distantes sont appelées et appliquées à l'aide de sockets, et enfin nous devons expliquer nos objectifs.

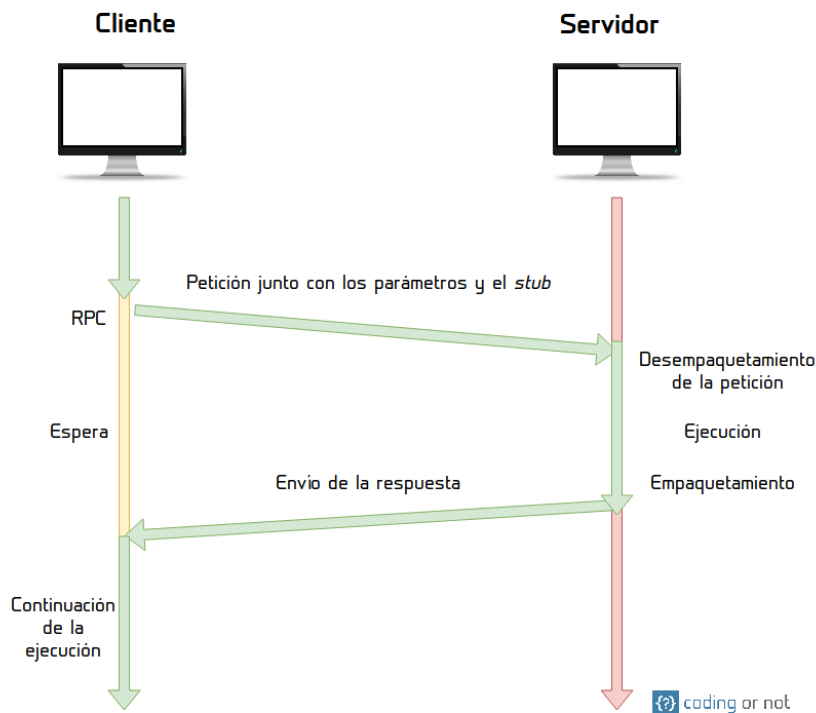
Introduction

Dans ce rapport nous étudierons le modèle d'architecture RPC, ce modèle est utilisé pour distribuer une application centralisée, pour essayer de mieux expliquer ce modèle nous utilisons un problème très simple dans lequel nous allons essayer de reconnaître ce même processus, c'est-à-dire le processus de distributions en général et l'appliquer à des situations plus complexes.

Tout d'abord, nous pouvons définir le RPC comme un outil chargé de réguler la communication entre les processus, c'est-à-dire un protocole de communication logiciel utilisé par un programme pour demander un logiciel, un service ou un autre élément situé sur un autre ordinateur. Il est important de se rappeler que pour que cet outil fonctionne, les appareils doivent être sur le même réseau.

D'autre part, il est important de mentionner le modèle que cet outil utilise. D'une certaine manière, le modèle le plus clair est le modèle client-serveur. Nous pouvons trouver le programme demandeur comme notre client et le programme dont nous avons besoin et que nous demandons comme celui qui est servi. Un point important de ce modèle pour qu'il fonctionne correctement est très important qu'il y ait une synchronisation, c'est-à-dire que le programme demandeur (client) soit maintenu en suspension pour que l'autre programme puisse travailler et nous donner les résultats que nous recherchons.

D'autre part, l'un des inconvénients de RPC est qu'il n'existe pas de norme unifiée pour cette technologie. Les différentes implémentations, dont la plupart sont spécifiques à une entreprise (par exemple, ONC-RPC de Sun), ne sont souvent pas compatibles entre elles. Voici un exemple :



Comment fonctionne le RPC ?

Voici les étapes à suivre pour faire fonctionner RPC, toutes tirées du diagramme ci-dessus :

- Le client appelle le stub et les paramètres dont il a besoin auprès du client, qui est responsable de la définition des paramètres et des adresses mémoire d'un environnement (le client) à un autre (le serveur). Il appelle la procédure locale avec des paramètres insérés sur la pile de la manière habituelle.
- Le stub client regroupe les paramètres nécessaires de la procédure dans un message et appelle le système pour envoyer le message. L'emballage des paramètres est un processus très important de la procédure car il est connu sous le nom de marshalling.
- Le système d'exploitation client envoie le message de la machine client à la machine serveur distante.
- Le système d'exploitation du serveur est chargé de transmettre les paquets entrants au serveur.
- Le stub du serveur dépile les paramètres (mieux connu sous le nom de démarchage) du message.
- Le serveur met fin à ce processus et retourne au code d'aide du serveur, qui met en forme les valeurs renvoyées dans un message.
- Le serveur renvoie le message résultant au stub client.

Arturo Arellano Coyotl
Victor Jules Paulin BARBE
PIERRE-LOUIS GAUCHER
Rubén Alfredo Jaramillo Gómez

- Pour terminer ce processus, le stub client dématérialise les paramètres de retour et l'exécution revient à l'appelant.

Un exemple clair de cela est ce que nous avons fait : en tant qu'étudiants, nous avons envoyé des messages à l'ordinateur du Docteur et ceux-ci ont été stockés dans un espace de l'ordinateur. Malheureusement nous n'avons pas de preuve graphique de ce processus, nous avons cherché quelques images en classe, mais l'angle n'était pas assez clair.

Conclusions

RPC est un excellent modèle pour réaliser des appels de fonctions à distance à l'aide de sockets. Il facilite grandement la connexion entre un programme demandeur, qui est un client, et le programme fournisseur de services, qui est le serveur. Après toutes les observations que nous avons faites, nous pouvons dire ce qui suit :

- Il est utile pour les clients de communiquer avec les serveurs par l'utilisation traditionnelle des appels de procédure dans les langages de haut niveau.
- Si nous en avons une connaissance approfondie, il est très facile de réécrire et de redévelopper le code.
- Nous pouvons l'utiliser aussi bien pour un environnement distribué que pour un environnement local.

D'un autre côté, il y a aussi des inconvénients :

- L'utilisation de RPC peut devenir un peu complexe et abstraite si son application n'est pas maîtrisée.

Après cette étude de cas, nous pouvons dire que nous avons atteint l'objectif de comprendre l'utilisation de la RPC et, bien que ce soit un peu difficile, nous ne pensons pas avoir la capacité de l'appliquer.

Arturo Arellano Coyotl
Victor Jules Paulin BARBE
PIERRE-LOUIS GAUCHER
Rubén Alfredo Jaramillo Gómez

Bibliographie.

Appel de procédure à distance (RPC) (2021, 26 octobre).

TechEdu. <https://techlib.net/techedu/llamada-a-procedimiento-remoto-rpc/>

Appel de procédure à distance : communication dans les systèmes. (2020, 24 client-serveur

mars). Guide numérique
IONOS.

<https://www.ionos.es/digitalguide/servidores/know->

[comment/qu'est
-ce que c'est-
rpc/](#)

Ruelas, U. (2018, 19 juin). *Qu'est-ce que RPC (appel de
procédure)*

à distance)
?
<https://codingornot.com/que-es-rpc-llamada-a-procedimiento-remoto>

Zechinelli Martini, José Luis (2022). Programmation concurrente UDLAP (2022).