

Systèmes distribués et répartis

SDR 2022 / Labo 4

Temps à disposition : 8 périodes.

Travail à réaliser par groupe de 2 étudiants (pas forcément les mêmes que pour les labos précédents, mais en en informant l'enseignant dans ce cas).

Un étudiant qui effectuerait le labo seul (avec justification et après autorisation explicite de l'enseignant) n'aura à rendre qu'une des deux parties demandées.

La présence aux labos est obligatoire.

En cas de copie manifeste entre les rendus de deux labos, tous les étudiants des deux groupes se verront affecter la note de 1.

*Labo distribué le vendredi 23 décembre 2022 à 14h55. Vous utiliserez un repo github **privé** dont vous donnerez l'accès à l'enseignant et à l'assistant (comptes **patricklac** et **raphaelracine** sous github) et dont vous leur communiquerez l'adresse par email avec le nom des étudiants du groupe. Vous devrez pousser votre travail en l'état sur ce repo au moins une fois par semaine, en début de chaque séance de labo.*

***A rendre le vendredi 27 janvier 2023 à 16h35.** Un readme indiquant comment utiliser votre programme et précisant ce qui a été réalisé, ce qui fonctionne et ce qui reste à faire, devra être déposé dans votre repository. Un email rappelant vos noms et repository confirmera votre rendu qui sera accessible dans la branche master (ou main).*

Objectifs

- Comprendre le fonctionnement de paradigmes de programmation dans un réseau réparti.
- Réaliser des communications UDP unicast en GO.
- Gérer la concurrence d'accès aux variables avec des goroutines et des channels en GO.

Introduction

Le labo est composé de 2 parties qui permettent chacune de compter de façon distribuée le nombre d'occurrences de lettres dans un texte. Chaque nœud du réseau est responsable du comptage des occurrences d'une lettre. Le réseau physique interconnecte tous les nœuds mais les algorithmes distribués utilisent uniquement un réseau logique dont la topologie est un graphe quelconque (pas forcément une arborescence). Pour la première partie, on utilisera un algorithme synchrone (ondulatoire) et pour la deuxième partie, un algorithme asynchrone (sondes et échos).

Description commune aux 2 parties

Pour les 2 parties, une fois le réseau réparti établi et sur une demande portant sur un texte, il s'agit d'obtenir par échange de messages entre nœuds voisins le nombre d'occurrence de chaque lettre de l'alphabet (ou seulement de certaines d'entre elles) dans ce texte.

Les communications se feront en mode non-connecté (UDP unicast). On supposera le réseau stable et sans erreurs : pas de perte de message ni de panne de serveur.

Un fichier de configuration contiendra la description du réseau :

- Le nombre de processus serveur, leurs numéros, adresses et numéros de port
- La description du graphe logique qui relie les serveurs (liste des numéros des voisins immédiats de chaque nœud).
- La lettre de l'alphabet traitée par chaque serveur.

Chaque serveur devra pouvoir être démarré avec son numéro en paramètre.

Une application client permettra de soumettre un texte saisi au clavier à analyser. On ne fera pas de nouvelle demande tant qu'une recherche précédente n'est pas terminée : il n'est pas demandé de gérer de recherches simultanées.

A l'issue de la soumission d'un texte, l'application client affichera les comptages pour chacune des lettres traitées.

Il n'est pas demandé de test automatisé mais votre readme contiendra la description de tests manuels et des résultats obtenus.

Comme pour les labos précédents, il sera tenu compte dans chaque partie de la qualité de votre code source (packages, ...), de sa conformité aux bonnes pratiques du langage GO, et aussi de la concurrence d'accès avec des goroutines et des canaux, et non avec des verrous.

Chaque partie sera valorisée pour 25 points, 15 points pour son bon fonctionnement et 10 points pour la qualité de votre rendu (documentation, tests, code source).

Première partie : algorithme ondulatoire

Une fois le réseau établi (pas de contrôle demandé), l'application client permettra de soumettre un texte simultanément sur chaque nœud du réseau (par message UDP direct, sans tenir compte du graphe logique). Elle en demandera ensuite le résultat à un nœud quelconque du réseau de numéro saisi par l'utilisateur.

Dès la réception du texte, chaque serveur participera à des vagues successives selon l'algorithme ondulatoire jusqu'à obtenir les comptages pour chaque lettre. On suppose qu'il ne connaît pas le « diamètre » du réseau : il s'arrêtera après avoir obtenu la liste complète des comptages de lettres, sans laisser de messages non lus sur le réseau, selon l'algorithme expliqué en cours.

Deuxième partie : algorithme sondes et échos

Une fois le réseau établi (pas de contrôle demandé), l'application client permettra de soumettre un texte sur un nœud de numéro saisi par l'utilisateur. Elle transmettra le message de soumission au seul serveur correspondant.

Ce serveur initial activera alors de proche en proche les comptages sur chaque nœud selon l'algorithme de sondes et échos jusqu'à obtenir la liste complète des comptages de lettres qu'elle retournera au client.

L'algorithme implémenté permettra de traiter sans boucler les cycles dans le graphe, et sans laisser de messages non lus sur le réseau, selon l'algorithme expliqué en cours.