PolyMessages — Sprint 1

Eri Agnese, Marvin Bontemps, Lucas Nouguier

17 Avril 2022

Contents

1 Protocole de communication

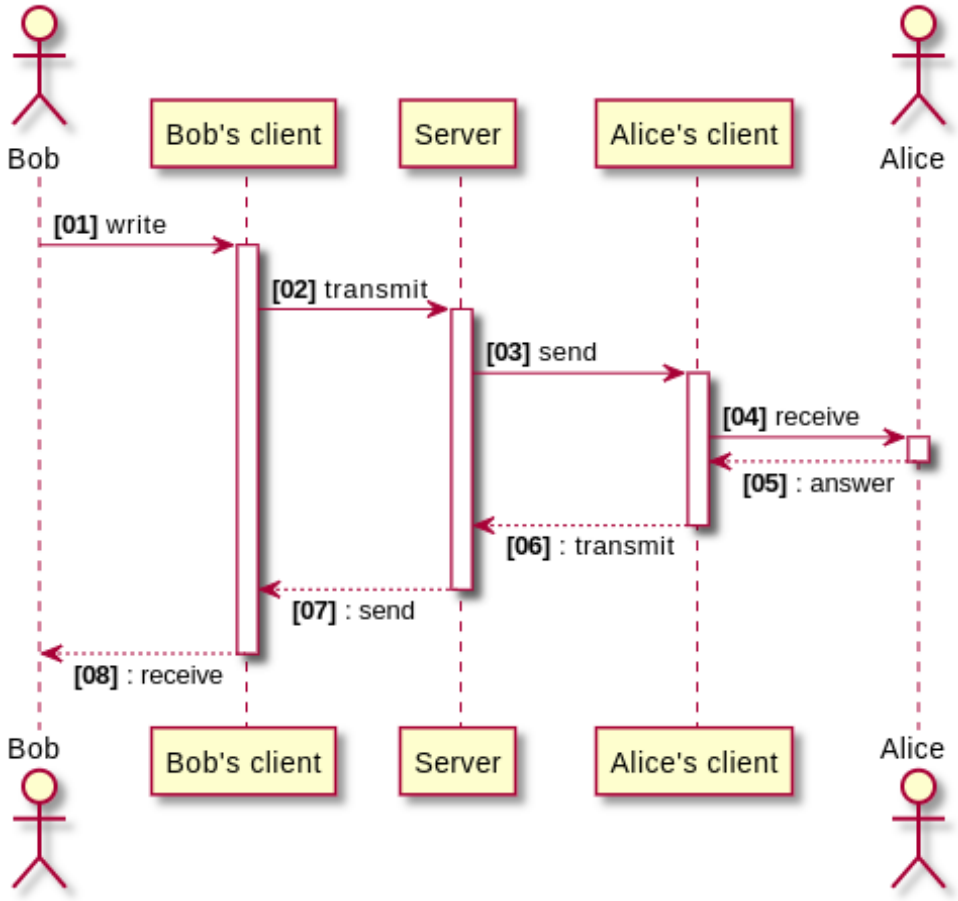

Figure 1: *Protocole de communication clients/serveur*

2 Architecture

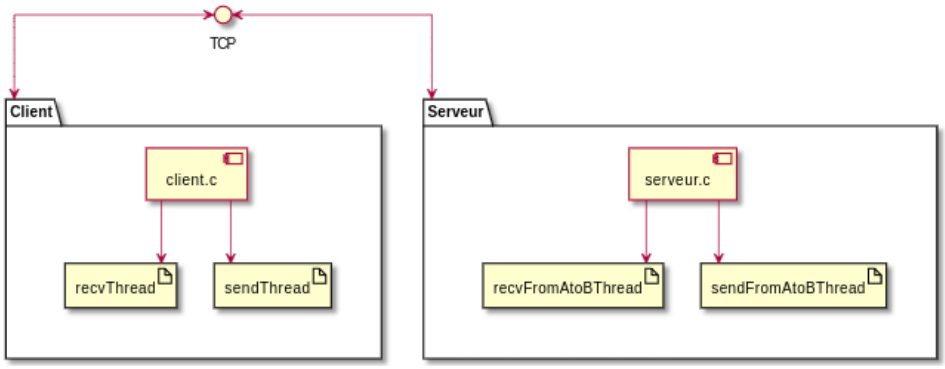

Figure 2: *Architecture de la messagerie*

3 Difficultés

Les difficultés rencontrées ont principalement été d'ordre technique. En parler entre nous ou avec les autres nous a permis de résoudre les problèmes.

4 Répartition du travail

Au lancement du projet, une messagerie basique au tour par tour a été montée par Lucas car il était impossible de travailler de manière efficiente à plusieurs sur une seule fonctionnalité. Une fois cette base complète, il était possible de travailler en parallèle donc le threading du serveur et des clients ont respectivement été fait par Marvin et Éri (voir Table. 1 & Fig. 3). La fusion de ces 2 travaux asynchrones a été effectuée par Lucas.

Tâches	Étudiants		
	Lucas	Éri	Marvin
Initialisation	X		
Threading client		X	
Threading serveur			X
Fusion	X		

Table 1: *Tâche effectuée par étudiant*

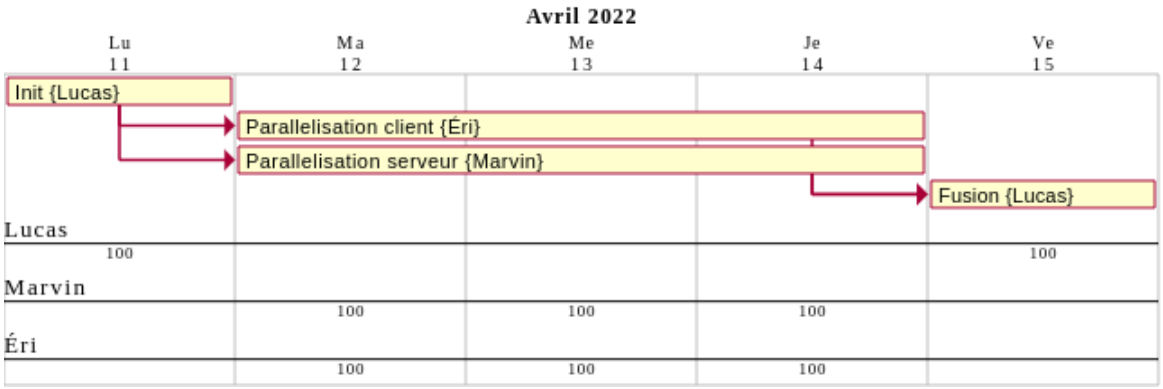

Figure 3: *Diagramme de Grantt sur la réalisation du projet*

5 Exécution du code

Pour lancer la messagerie, il faut commencer par compiler et lancer le serveur

```
[lucas@xps-lucas ~]$ gcc -o server serveur.c
[lucas@xps-lucas ~]$ ./server <PORT>
```

On peut alors lancer les clients (actuellement il en faut 2 avant qu'un échange puisse avoir lieu)

```
[lucas@xps-lucas ~]$ gcc -o client client.c
[lucas@xps-lucas ~]$ ./client <IP> <PORT>
```