

## Définition de l'architecture de l'application:

### A. Le serveur:

- Un processus père qui va mettre en place l'état initial, l'attente des différents clients et créer un processus fils pour chaque client qui se connecte grâce à la fonction fork().
- Les processus fils vont d'abord envoyer l'état initial aux clients et en boucle attendre la réception de demande de réservation ou de suppression de ressources de calcul.
- Chaque processus fils va créer un thread qui se chargera d'envoyer les mises à jour de l'état à son client.
- L'état est placé dans une mémoire partagée et est manipulé grâce à un pointeur vers une structure (\*etatshm)
- Il y a 3 sémaphores pour gérer l'accès à l'état et notifier les processus.
- Le premier sémaphore (id1) est utilisé pour garantir qu'il y a un seul processus qui accède à l'état à la fois.
- Le deuxième sémaphore (id2) est utilisé pour notifier les processus quand il y a une mise à jour.
- Le troisième sémaphore (id3) est utilisé pour potentiellement débloquer les demandes qui ne pouvaient être satisfaites

### B. L'état du système :

- L'état des ressources est regroupé dans une structure composée du site géographique où se trouve la ressource, le nombre de processeurs disponibles, le volume de stockage etc.... et un tableau contenant des structures "Sreservation".
- Sreservation étant une autre structure contenant le nombre de cpu, le volume de stockage, le mode de réservation ainsi que le nom du client.

### C. Le client :

- Le client se connecte au serveur et récupère l'état actuel.
- Ensuite il crée un thread qui va se charger de faire les demandes et les envoyer au serveur.
- Le thread précédent crée un autre thread qui aura pour but de récupérer les états venant du serveur après chaque mise à jour.
- Un verrou est utilisé pour s'assurer qu'un seul thread affiche à la fois, afin d'avoir un programme cohérent.

#### **D. La communication Client- Serveur :**

- La communication entre le serveur et les clients est établie via une communication TCP. Étant donné qu'un client se connecte d'abord au serveur pour pouvoir recevoir l'état du système et pouvoir par la suite effectuer des demandes de réservation de ressources.