

Rappel

1. Application Répartie

Une application répartie (ou application distribuée) est un logiciel qui est conçu pour fonctionner sur un ensemble de machines interconnectées, souvent via un réseau, de manière à ce que ces machines travaillent ensemble pour fournir une fonctionnalité ou un service global. Contrairement à une application monolithique qui s'exécute sur une seule machine, une application répartie répartit ses tâches et ses données sur plusieurs machines.

Exemples d'applications réparties courantes incluent les systèmes de gestion de bases de données distribuées, les applications web à grande échelle, les services cloud, les réseaux sociaux, les systèmes de messagerie instantanée, et bien d'autres. Les applications réparties sont conçues pour offrir une évolutivité, une performance et une disponibilité améliorées, mais elles présentent également des défis tels que la gestion de la cohérence des données et la gestion des pannes.

2. "socket" et "thread"

Les termes "socket" et "thread" sont deux concepts fondamentaux en programmation, en particulier dans le contexte des communications réseau et de la programmation multithread.

Socket :

- Un socket est une interface de programmation qui permet à deux ordinateurs de communiquer entre eux sur un réseau, que ce soit en

utilisant le protocole TCP (Transmission Control Protocol) pour des connexions fiables et orientées connexion, ou en utilisant le protocole UDP (User Datagram Protocol) pour des communications non fiables et sans connexion.

- Un socket est un point de terminaison pour une communication réseau. Il est identifié par une adresse IP et un numéro de port.
- En programmation, les sockets sont utilisés pour établir des connexions réseau, envoyer et recevoir des données entre des machines distantes, ce qui permet de créer des applications réseau telles que des serveurs web, des applications de chat, des jeux en ligne, etc.

Thread :

- Un thread (ou fil d'exécution) est la plus petite unité d'exécution d'un programme informatique. Un programme peut fonctionner avec un seul thread (monotâche) ou avec plusieurs threads (multitâche).
- Les threads permettent d'exécuter des tâches simultanément, ce qui est particulièrement utile pour gérer des opérations concurrentes ou parallèles.
- En programmation multithread, chaque thread a son propre chemin d'exécution indépendant, mais ils partagent généralement les mêmes ressources mémoire (variables, données, etc.).
- Les threads sont couramment utilisés pour réaliser des opérations simultanées, telles que gérer plusieurs connexions clientes dans un serveur, exécuter des tâches en arrière-plan tout en maintenant une interface utilisateur réactive, etc.

En résumé, les sockets sont utilisés pour établir des connexions réseau et échanger des données entre des machines, tandis que les threads sont utilisés pour gérer des tâches simultanées ou concurrentes dans un programme, ce qui est essentiel dans de nombreuses applications, notamment celles impliquant des communications réseau.