

Objectif :

L'objectif de ce travail est de mettre en œuvre des applications qui sont développées et exécutées sur un réseau en utilisant EJB3 technologie, sockets TCP, sockets UDP et RMI

Architecture :

Nous considérons l'architecture d'une application distribuée qui est composée de (voir Figure 1) :

Cinq serveurs (Servi). Chaque serveur doit gérer un groupe de services (S1, S2....Sn).

Six clients (Clienti). Chaque client a une chaîne de référence qui représente une demande de service.

Un serveur central (Central Server) avec une base de données qui héberge les services.

Un serveur intermédiaire (Inter) qui redirige les requêtes des clients vers les serveurs correspondants (Servi).

Processus :

Les services sont stockés dans une base de données. La base de données contient divers services pour tous les clients. Les services sont identifiés par un numéro, nom et description. Avant qu'un serveur (Servi) commence à communiquer, il doit s'enregistrer auprès du serveur central en fournir les informations nécessaires (adresse IP, nom du serveur, port de communication). Chaque client envoie ses différentes requêtes au serveur intermédiaire (Inter) qui redirige ces requêtes vers les serveur (Servi). Chaque serveur (Servi) appelle le serveur central qui effectue la recherche et la récupération de service à partir du base de données. Les informations concernant le service demandé doivent être renvoyées au client concerné (puisque chaque client dispose d'une chaîne de références, une requête est envoyée par référence).

Pour envoyer leurs demandes, tous les clients utilisent une méthode séquentielle à l'aide d'un jeton de rotation. Les clients sont initialement organisés en un anneau logique unidirectionnel. Seuls les clients qui reçoivent le jeton peuvent envoyer sa prochaine demande. Il est important de surveiller le jeton (par exemple, s'il est perdu en raison d'une interruption du réseau ou d'une panne de station). Pour ce faire, supposons que chaque station a une MINUTERIE qui est réinitialisée lorsqu'un jeton la traverse. Lorsque la station arrête d'envoyer le jeton, la MINUTERIE atteint le fin. La première station dont TIMER expire lance l'algorithme pour sélectionner un nouveau candidat. Il envoie la suppression information qui se déplace autour de l'anneau et émet un nouveau jeton.

Travailler

Pour montrer comment cela fonctionne, nous considérons les chaînes de référence client suivantes.

Client:	Reference chain
01	S7, S10, S1, S2, S4, S3, S0, S6, S12, S11, S8, S5, S2, S1, S0, S5, S1, S7, S9, S2.FIN
02	S15, S1, S4, S2, S3, S6, S7, S10, S1, S13, S6, S14, S2, S0, S1, S3, S2, S7, S0, S1.FIN
03	S10, S3, S6, S4, S2, S7, S0, S3, S12, S8, S7, S0, S6, S5, S3, S2, S1, S7, S0, S4. FIN
04	S13, S2, S1, S7, S0, S4, S6, S5, S11, S0, S14, S3, S9, S1, S0, S3, S6, S7, S12, S11, FIN
05	S2, S14, S1, S7, S6, S3, S0, S5, S2, S15, S7, S0, S4, S12, S3, S6, S7, S2, S1, S13. FIN
06	S5, S4, S1, S3, S10, S7, S6, S0, S13, S2, S1, S7, S6, S3, S15, S2, S1, S14, S5, S6. FIN

Server:	Service processing
01	S0, S1 S2 S3
02	S4 S5 S6
03	S7 S8 S9
04	S10 S11 S12
05	S13 S14 S15

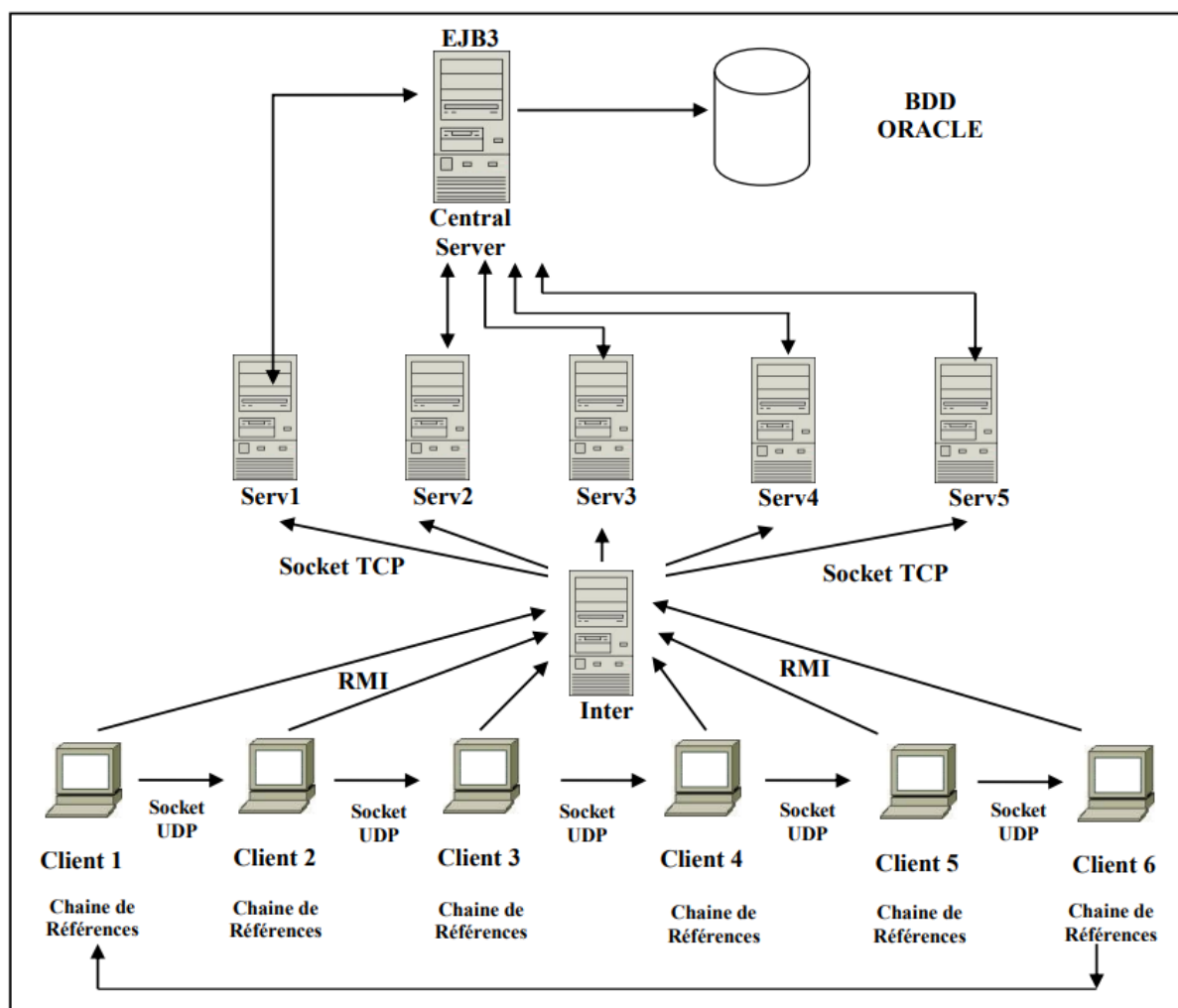


Figure 1 : Application architecture.