

Définition 1 [Tanenbaum] : Système Réparti :Un ensemble d’ordinateurs indépendants qui apparaît à un utilisateur comme un système unique et cohérent

* Les machines sont autonomes
* Les utilisateurs ont l’impression d’utiliser un seul système (la transparence).

Définition 2 [Lamport]: Un système réparti est un système qui vous empêche de travailler quand une machine dont vous n’avez jamais entendu parler tombe en panne.

Définition :

Un Système Réparti (distribué) est un ensemble d’entités autonomes de calcul (ordinateurs, processeurs, processus, processus léger etc.) interconnectées et qui peuvent communiquer par un réseau de communication et qui communiquent par envoi de messages.

Exemples :

* réseau physique de machines
* Un logiciel avec plusieurs processus sur une même machine.

Réparti = distribué ?

Distribué à fabricants d’ordinateurs Réparti à Génie Logiciel

Objectifs généraux

* Aspects économiques (rapport prix/performance)
* Adaptation de la structure d’un système à celle des applications (géographique ou fonctionnelle)
* Besoin d’intégration (applications existantes)
* Besoin de communication et de partage d’information

Objectifs spécifiques de l'Informatique Repartie

* Proposer des méthodes et outils pour simplifier le développement d'application réseau client/serveur, en essayant de s'abstraire de l'aspect " distant" : proposer une programmation " naturelle"
* Pour les applications "lourdes" :

🞑 Décomposer les applications en ensembles de services

🞑 Rationaliser la répartition des services pour limiter les échanges d'informations

une application informatique est un programme exécutable sur une machine ou plusieurs machines qui représente la logique de traitement des données manipulées par l’application.

Interfaces

* Partie “opérationnelle”

Interface Definition Language (IDL)

Pas de standard, mais s’appuie sur un langage existant

* + IDL CORBA sur C++
  + Java et C# définissent leur propre IDL
* Partie “contractuelle”

Plusieurs niveaux de contrats

🞑 Sur la forme : spécification de types à conformité syntaxique

🞑 Sur le comportement (1 méthode) : assertions à conformité

🞑 sémantique

🞑 Sur les interactions entre méthodes : synchronisation

🞑 Sur les aspects non fonctionnels (performances, etc.) : contrats de

🞑 QoS