

Plan

- **Classes de Systèmes Distribués**
- Parallélisme et Concept de thread
- Mécanismes de Synchronisation
- Fine-Grain / Coarse-Grain Locking
- Example : Chorus IPC

Classes de Systèmes Distribués

- Client / Serveur simple
 - Services de base d'échanges de données
 - Peu ou pas de contraintes/exigences
- Systèmes Répartis
 - Répartition des service/ressources sur un grand nombre d'unités de calcul
 - Réseaux à hauts débits
 - Systèmes complexes

Client / Serveur Simple

- Echanges explicitement initiés par un utilisateur
- Communications temporaires
 - pas d'état rémanent
 - Pas d'état réparti
- Multiples clients indépendants
 - Pas de partage de données entre clients
 - Isolation/protection des échanges des clients

Examples

- Copie de fichiers
 - “ftp” (**F**ile **T**ransfert **P**rotocol)
 - “scp” (**S**ecure **C**opy)
- Login à distance
 - “ssh” (**S**ecure **S**hell)
- Courrier électronique

Contexte serveur

- Un processus serveur par client
- Créé à chaque [demande de] connexion établie par un client
- Contexte système privé
 - Connexion réseau avec le client
 - fichiers ouverts
- Espace mémoire (virtuel) privé

Serveur Mono-Tâche

- Modèle d'exécution synchrone
- Une seule opération E/S à la fois
 - Pas de parallélisme entre lecture/écriture d'un fichier et communications réseau
- Performances limitées

Systemes Répartis

- Tout ou partie des composants/ressource rémanents ou semi-rémanents
- Données partagées et/ou distribuées entre composants distants
- Système évolutif
 - Nombre de composants
 - Localisation

Propriétés Systèmes Répartis

- Tolérants aux pannes
- Robustes
- Hautement disponibles
- Reprise [automatique] sur erreur
- Prédicible
 - Temps de réponse acceptables
- Sécurisés

Exemples

- Service d'accès à des fichiers distants
- Grilles de calculs (“Grid computing”)
- Bases de données réparties
 - Banques, transports, etc...
- Systèmes de contrôle
 - Avionique, automobile
 - Chaînes de production automatisées

Example: Single System Image

- Système décomposé en modules répartis
- Répartition transparente
 - Pour les applications
 - Pour les utilisateurs
- Services/ressources distants accédés par RPC (**R**emote **P**rocedure **C**all)
- Propriétés critiques
 - Performances
 - “Scalability”

Serveurs Multi-Tâches

- Modèle d'exécution parallèle
 - Traitement simultané de multiples requêtes
 - Minimiser temps de latence
 - Partage équitable des ressources de communications (débit réseau, etc.)
- Problèmes exacerbés par exécution permanente
 - Fuites de mémoire
 - Cohérence d'états