

Modèles et Systèmes de programmation distribuée

Nhan LE THANH
Cours LpSIL – Spécialité IDEE
Octobre 2004

Plan du cours

I. États de l'art

- 1- Objectifs, caractéristiques
- 2- Communication en réseaux
- 3- Structures logiques
- 4- Problématique
- 5- Développement, déploiement

II. Modèles d'exécution

- 1- Modèles client-serveur
- 2- Modèles de communication par messages
- 3- Modèles de communication par événements
- 4- Autres modèles

III. Systèmes pair à pair

- 1- Principes et composantes
- 2- Calcul distribué Pair à Pair
- 3- Échange de données Pair à Pair
- 4- Étude de cas
- 5- Problématique et évolution

IV. Systèmes transactionnels

- 1- Transactions et transactions réparties
- 2- Systèmes transactionnels
- 3- Programmation transactionnelle
- 4- SGBD répartis

Part 1: Etat de l'art

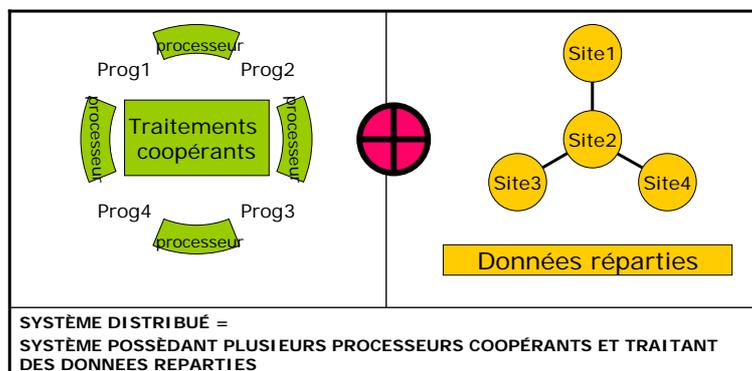
Plan

- 1- Objectifs, caractéristiques
- 2- Communication en réseaux
- 3- Structures logiques
- 4- Problématique
- 5- Développement, déploiement

I. États de l'art

1. Objectifs, Caractéristiques

- Définitions générales :



I. États de l'art

1. Objectifs, Caractéristiques

□ Définitions générales

coopération = communication + synchronisation

- Un modèle de communication et d'exécution
- Une interface de programmation (et/ou langage)
 - Un modèle de programmation
- Des outils de développement
- Un environnement d'exécution : services systèmes (pour différents types d'infrastructures)

I. États de l'art

1. Caractéristiques générales

□ Pourquoi des systèmes répartis

- **Aspects économiques**
- **Adaptation de la structure d'un système à celle des applications**
- **Besoin d'intégration**
- **Besoin de communication et de partage d'information**
- **Réalisation de systèmes à haute disponibilité**
- **Partage de ressources (programmes, données, services)**
- **Réalisation de systèmes à grande capacité d'évolution**

I. États de l'art

1. Objectifs, Caractéristiques

□ Objectifs, Avantages et inconvénients:

OBJECTIFS	
<ul style="list-style-type: none">- <i>Coût</i> : plusieurs processeurs à bas prix- <i>Puissance de calcul</i> : aucune machine centralisée peut réaliser- <i>Performance</i> : calcul parallèle- <i>Adaptation</i> : à des classes d'applications réelles naturellement distribuées- <i>Fiabilité</i> : résistance aux pannes logicielles ou matérielles- <i>Extensibilité</i> : croissance progressive selon le besoin	
Avantages	Inconvénients
<ul style="list-style-type: none">- partage de données- partage de périphériques- communication- souplesse (politiques de placements)	<ul style="list-style-type: none">- logiciels : peu de logiciels disponibles- réseaux : la saturation et délais- sécurité : piratage

I. États de l'art

1. Objectifs, Caractéristiques

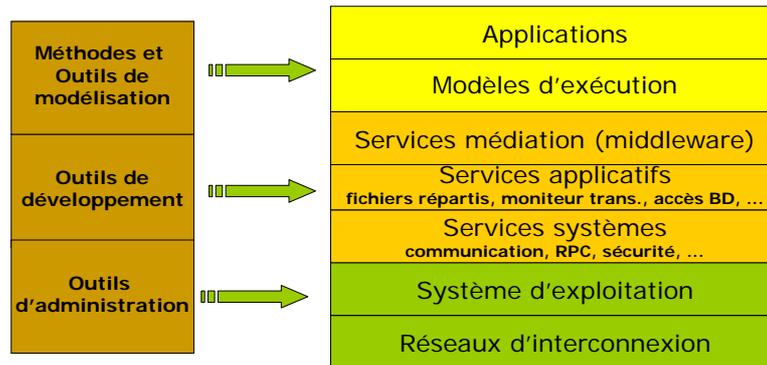
□ Caractéristiques générales

- Ensemble d'éléments (traitement, stockage)
 - Processeurs, mémoires, organes d'E/S
- Système d'interconnexion
- Intégration (vue globale uniforme)
 - "Transparence"
- Défaillances locales possibles sans compromettre nécessairement l'ensemble du système
- État commun (partition + duplication)
 - Coopération à une tâche commune
 - Redondance permettant la reprise après erreur locale

I. États de l'art

1. Objectifs, Caractéristiques

□ Problématique



I. États de l'art

1. Objectifs, Caractéristiques

□ Champs d'application

- **CFAO, Ingénierie simultanée**
 - Coopération d'équipes pour la conception d'un produit
 - Production coopérative de documents
 - Partage cohérent d'information
- **Gestion intégrée des informations d'une entreprise**
 - Intégration de l'existant
- **Contrôle et organisation d'activités en temps réel**
 - Work flows
 - Travail coopératif
- **Centres de documentation, bibliothèques**
 - Recherche, navigation, visualisation multimédia
- **Web sémantique**

I. États de l'art

1. Objectifs, Caractéristiques

□ Besoins des applications

- **Ouverture**
 - Interopérabilité, portabilité, fédération ; réutilisation de l'existant
- **Coopération, coordination, partage**
 - Vision commune cohérente d'informations partagées (globalement, par groupes). Interaction en temps réel, support multimédia
- **Transparence**
 - Accès (mobilité des usagers avec préservation de l'environnement)
 - Localisation (de l'information, des services, ...)
- **Qualité de service**
 - Disponibilité, délais, coûts, qualité de perception, .. avec niveau garanti
- **Sécurité**
 - Authentification, intégrité, confidentialité, ...
- **Évolutivité, administrabilité**
 - Reconfiguration, gestion dynamique de services

I. États de l'art

1. Objectifs, Caractéristiques

□ Évolution: historique

Années	Recherche	Produit
1970	Stations de travail + serveurs	
	Ethernet	
	Mémoire stable	
	Fichiers Unix répartis	
1980	Micro-noyaux	Stations de travail
	Serveurs de fichiers évolués	Ethernet
	Appel de procédure à distance	Mémoire virtuelle répartie
1985	Systèmes transactionnels répartis	Systèmes "non-stop"
	Grands systèmes intégrés	SGF répartis, RPC
	Systèmes répartis à objets	Gestion répartie de fenêtres
1990	Travail coopératif	Micro-noyaux
	Support système pour multimédia	...

I. États de l'art

2. Communication en réseaux

□ Rappel de l'architecture OSI

O.S.I. = OPEN SYSTEME INTERCONNECTION

- Modèle fondé sur un principe énoncé par Jules César « Diviser pour mieux régner »
- Le principe de base est la description des réseaux sous forme d'un ensemble de couches superposées les unes aux autres
- L'étude du tout est réduit à celle de ses parties
- L'ensemble devient plus facile à construire et à manipuler

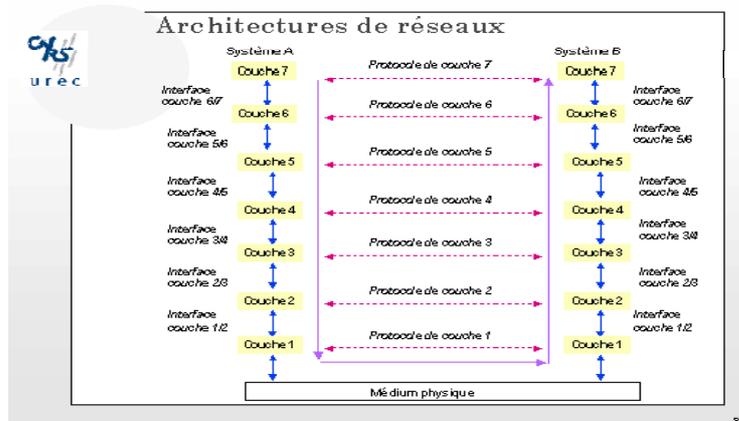
Deux organisation de normalisation pour les réseaux informatiques

- ISO (International Standardization Organization) dépendante de l'ONU avec les représentants nationaux : ANSI pour les USA, AFNOR pour la France, DIN pour l'Allemagne, BSI pour le Royaume Uni, HSC pour le Japon, ...
- UIT-T (Union Internationale des Télécommunication) comprend des opérateurs et des industriels des télécommunications

I. États de l'art

2. Communication en réseaux

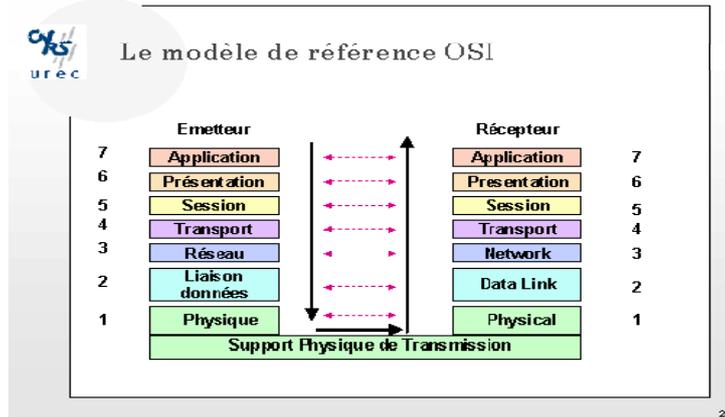
□ Rappel de l'architecture OSI



I. États de l'art

2. Communication en réseaux

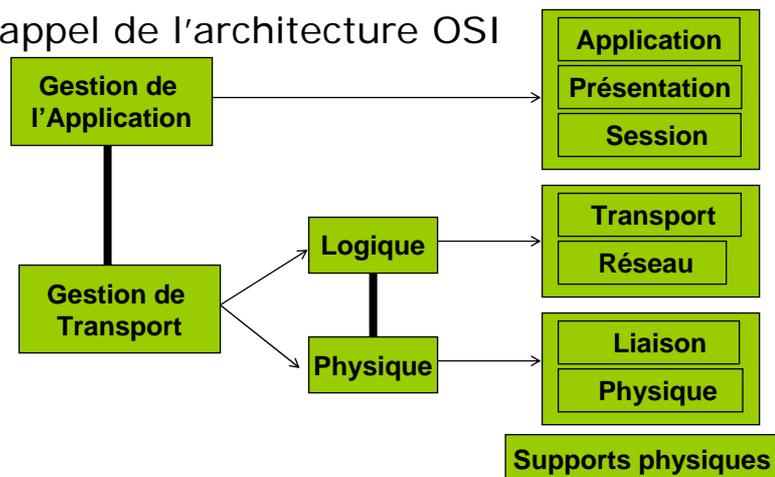
□ Rappel de l'architecture OSI



I. États de l'art

2. Communication en réseaux

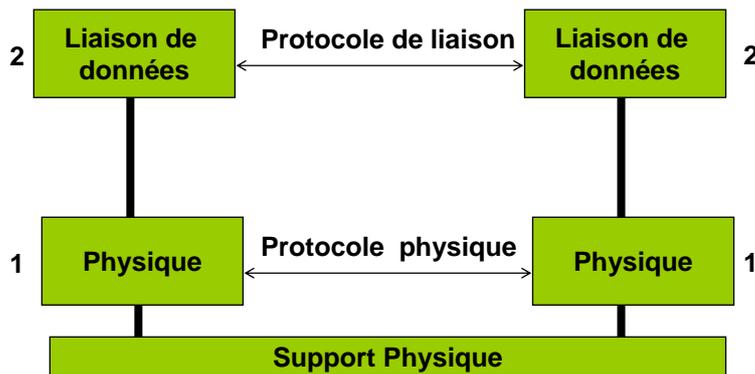
□ Rappel de l'architecture OSI



I. États de l'art

2. Communication en réseaux

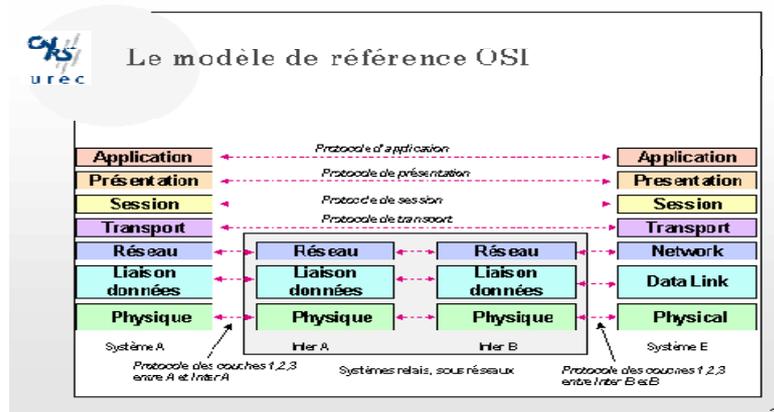
OSI : Communication directe



I. États de l'art

2. Communication en réseaux

OSI : Communication indirecte



I. États de l'art

2. Communication en réseaux

- OSI : Communication entre application
 - Gestion des sessions : synchronisation
 - Gestion des terminaux virtuels
 - Gestion des communication entre applications



I. États de l'art

2. Communication en réseaux

- TCP/IP : Internet
 - Le concept d'interconnexion ou d'*internet* repose sur la mise en oeuvre d'une couche réseau masquant les détails de la communication physique du réseau et détachant les applications des problèmes de routage.

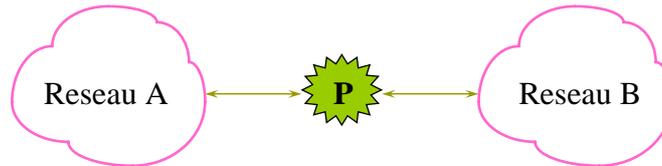
 - L'interconnexion : faire transiter des informations depuis un réseau vers un autre réseau par des noeuds spécialisés appelés passerelles (*gateway*) ou routeurs (*router*)

I. États de l'art

2. Communication en réseaux

□ TCP/IP : Internet

- Les routeurs possèdent une connexion sur chacun des réseaux:



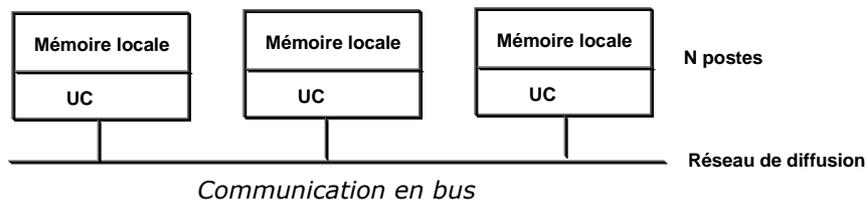
La passerelle P interconnecte les réseaux A et B.

- Le rôle de la passerelle P est de transférer sur le réseau B, les paquets circulant sur le réseau A et destinés au réseau B et inversement.

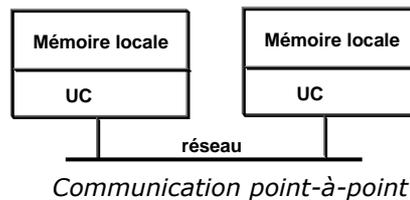
I. États de l'art

2. Communication en réseaux

□ Communication en bus et point-à-point



- Facile à construire
- Bas coût
- Le trafic est moins important (pas d'échange UC-mémoire)
- souvent utilisé dans les réseaux locaux avec une vitesse suffisamment rapide

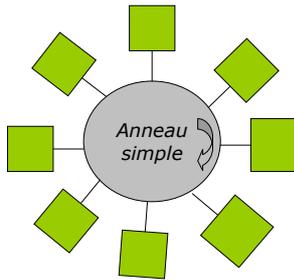


I. États de l'art

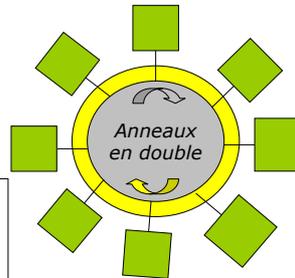
2. Communication en réseaux

□ Communication en anneau

Les informations circulent sans un sens unique



Souvent réalisé dans un réseau campus pour faire un grappe (cluster) de serveurs



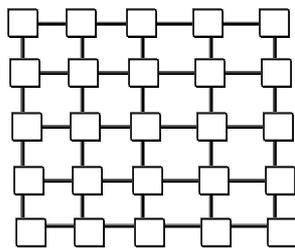
Les informations circulent dans les deux voies dont chacune dans un sens opposé

- Avantage : diminution du risque de perte d'information. L'existence de mécanismes de synchronisation sans conflit
- Inconvénient : coût élevé

I. États de l'art

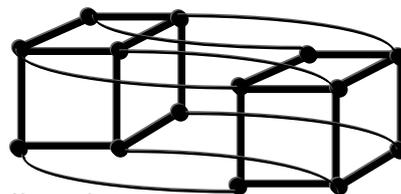
2. Communication en réseaux

□ Communication en réseau commuté



Treillis

- câblage simple
- le chemin plus long croît en racine carré du nombre d'UC



Hypercubes (4 dimensions)

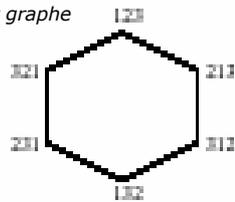
- Un hypercube est un cube à n dimensions
- 4 dimensions : 2 cubes de 3 dimensions avec les sommets homologues reliés
- 5 dimensions : 2 hypercubes de 4 dimensions avec les sommets homologues reliés
- la complexité croît en log2 du nombre de noeuds
- le chemin plus long croît en log2 du nombre de -noeuds

I. États de l'art

2. Communication en réseaux

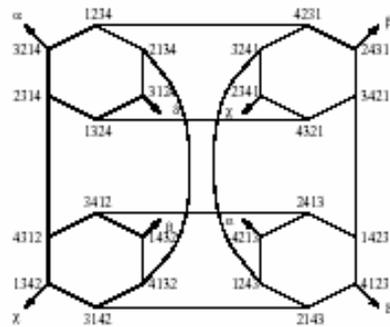
Communication en réseau commuté

Un 3-star graphe



***n*-star graphe** : Un *n*-star graphe est un graphe $G_n = (S, A)$ où
- S est $n!$ sommets dont chacun correspond à une permutation de n
- A est ensemble de tous les arcs (u,v) tels que $u, v \in S$ et $v = P_j(u)$, $1 < j \leq n$

Un 4-star graphe



Pj : soit $s_1, s_2, \dots, s_j, \dots, s_n$ est une permutation de n . On appelle $P_j(s_1, s_2, \dots, s_n)$ la permutation $(s_j, s_2, \dots, s_{j-1}, s_1, s_{j+1}, \dots, s_n)$

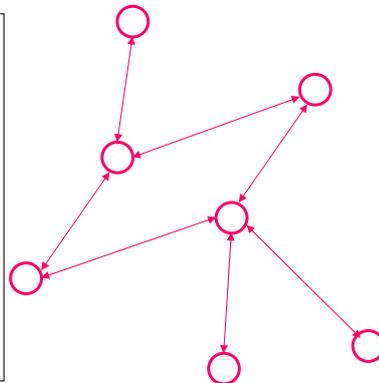
I. États de l'art

2. Communication en réseaux

Communication en réseau commuté

Pair-à-Pair (Peer-to-Peer) –P2P :

- Réseau commuté évolutif et aléatoire.
- Chaque nœud peut être lié à un certain nombre de nœuds déterminés dynamiquement.
- C'est la configuration la plus générale des réseaux commutés : les précédents réseaux sont des cas particuliers de P2P
- Plusieurs modes de routage selon la configuration choisie.
- Dans le cas le plus général, on utilise la technique de « diffusion par le voisinage », appelée « inondation » qui accompagne quelques mécanismes d'optimisation



I. États de l'art

3. Structures logiques

- Principes de normalisation
 - Importance des normes
 - OSF (consortium) : modèle DCE
 - OMG (consortium) : ORB, services
 - ODP (ISO-CCETT) : modélisation, ingénierie
 - standards spécifiques à un domaine technique : exemple de POSC
 - Mécanismes pour la construction et la réutilisation
 - Assemblage de protocoles à partir d'éléments génériques (exemples : protocoles de communication, journalisation, etc)
 - Mise en œuvre de systèmes génériques sur micro-noyaux
 - Utilisation de techniques à objets pour la structuration des systèmes et applications

I. États de l'art

3. Structures logiques

- Évolution des architectures

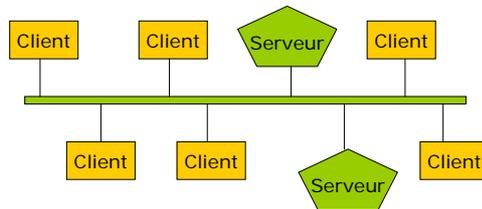


I. États de l'art

3. Structures logiques

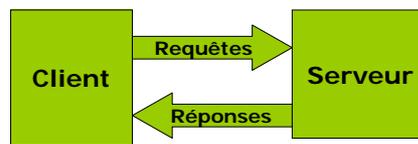
Aspect physique

- machines clients et serveurs physiquement identifiées
- indépendance des systèmes
- liaison statique ou dynamique



Aspect logique

- modularité
- abstraction (séparation entre interface et réalisation)
- modèle sémantique connu (RPC)
- traitement local du parallélisme chez le client et le serveur



I. États de l'art

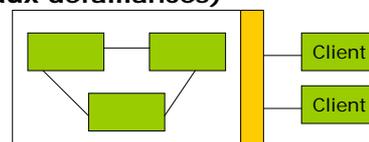
3. Structures logiques

▣ Serveurs coopérants

- **Réalisation d'un modèle abstrait à serveur unique**
 - ▣ L'interface dissimule la réalisation coopérative

■ Motivations

- ▣ Disponibilité (tolérance aux défaillances)
- ▣ Efficacité
 - Traitement parallèle
 - Proximité physique



■ Problèmes

- ▣ Protocole de communication interne
 - "pairs" ou maître-esclave
- ▣ Reprise après panne (détection, élection)
- ▣ Cohérence des informations dupliquées

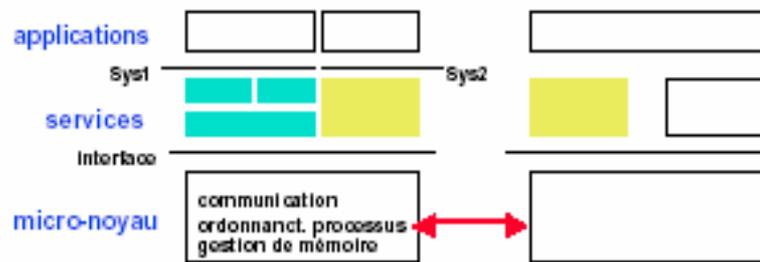
Interface unique

I. États de l'art

3. Structures logiques

□ Micro-noyau

- Interface standard pour les fonctions de base
- Base pour le support de services de systèmes
- Coexistence possible de plusieurs systèmes



I. États de l'art

3. Structures logiques

□ Micro-noyau : avantages

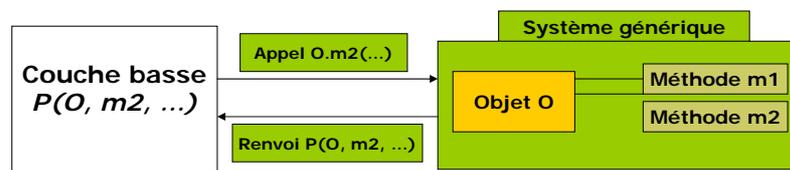
- Noyau de base simple et de taille réduite
 - Facile à comprendre, mettre au point et maintenir
- Opérations primitives efficaces
 - Processus légers ("threads")
 - Mémoire virtuelle avec couplage ("mappeurs" programmables)
 - Communication rapide et protégée
- Services modulaires réutilisables
- Interfaces publiques ouvertes (noyau, services)
 - Réalisation pouvant rester "propriétaire"

I. États de l'art

3. Structures logiques

□ Sous systèmes génériques

- Chaque sous-système générique encapsule certaines fonctions qui sont accessibles par les couches basses (call up)
- Utilisation du modèles à objets (simulation de l'héritage)



I. États de l'art

3. Structures logiques

□ Sous systèmes génériques

- **Intérêt : exploiter la modularité en vue de la réutilisation**
 - Isolation des fonctions : la couche basse connaît le minimum sur la couche haute
- **Une technique de structuration : l'appel ascendant ("upcall") équivaut au passage d'une procédure en paramètre**

□ Exemple :

- **cacher la connaissance de la structure interne d'un objet (accessible par appel d'une méthode de l'objet)**
- **procédure de sauvegarde pour reprise**

I. États de l'art

4. Problématique

□ Généralité

Un système distribué est un système qui s'exécute sur un ensemble de machines sans mémoire partagée, mais que pourtant l'utilisateur voit comme une seule et unique machine

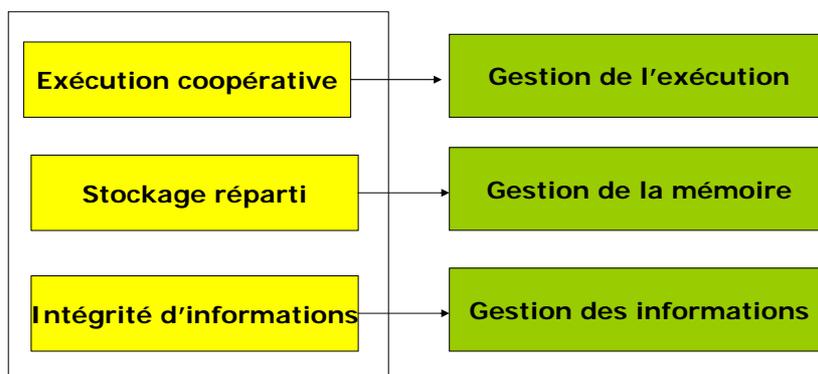
Caractéristiques

- possédant un mécanisme de communication interprocessus unique et global permettant la dialogue entre deux processus quelconques
- possédant un système de protection unique et global
- possédant un mécanisme de gestion de processus unique
- possédant un ensemble des appels système unique disponible sur toutes les machines
- possédant un noyau de système d'exploitation identique implanté sur toutes les machines
- possédant un mécanisme de gestion de mémoire identique

I. États de l'art

4. Problématique

□ Généralité

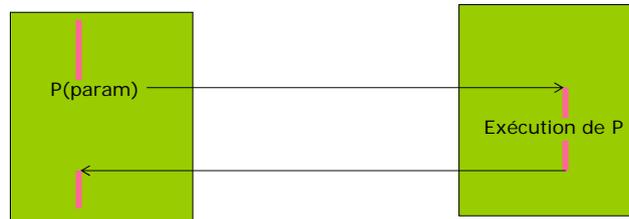


I. États de l'art

4. Problématique

□ Gestion d'exécution : modèles de communication

- Modèle asynchrone (primitif) :
 - Primitives SEND(m,P) et RECEIVE(P):m
- Modèle synchrone (RPC) :



I. États de l'art

4. Problématique

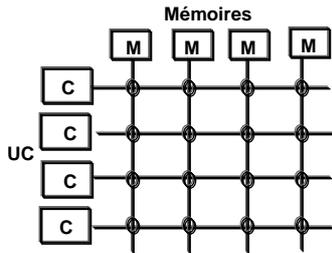
□ Gestion d'exécution : Processus-mémoire

- Processus lourds : une entité d'exécution autonome disposant d'un espace de mémoire virtuel propre (contexte)
 - Intérêt : unité d'exécution indépendante
- Processus légers : plusieurs processus partagent un espace virtuel de mémoire prédéfini
 - Intérêt : communication facile, commutation rapide, parallélisme moins coûteux, multi-processeur virtuel

I. États de l'art

4. Problématique

- Gestion d'exécution : Processus-mémoire
 - Accès aux mémoires distribuées (grille)



Technique de matrice de commutation (Crossbar Switch)

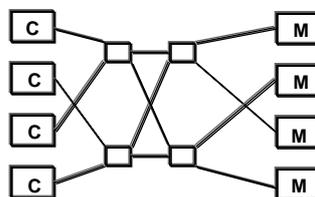
- les UC et mémoires sont reliées par une matrice de commutation
- à chaque l'intersection, le noeud de commutation (crosspoint switch) peut-être ouvert ou fermé
- quand une UC veut accéder à un module de mémoire elle ferme temporairement le noeud de commutation correspondant
- si plusieurs UC veulent accéder au même module, une file d'attente est nécessaire

Inconvénient
 le nombre des noeuds de commutation nécessaires : il faut n^2 de noeuds de commutation pour rélier n UC aux n module de mémoire

I. États de l'art

4. Problématique

- Gestion d'exécution : Processus-mémoire
 - Accès aux mémoires distribuées (multi-étages)



Principe du réseau multiétage

- utilisation des commutateurs 2x2 : 2 entrées et deux sorties
- chaque commutateur peut relier à n'importe quelle entrée et à n'importe quelle sortie
- pour relier n UC à n mémoires, il nécessite n étages dont chacun contient $\log_2 n$ commutateurs 2x2
- le nombre nécessaire de commutateurs est : $n \times \log_2 n$

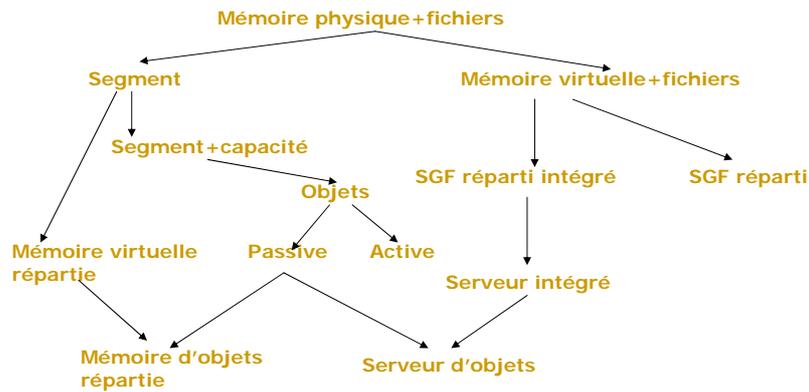
Inconvénient : Temps de propagation donc coût

Si $n = 1024$, on a besoin 10 étages. Avec les UC de 50Mhz, le cycle de calcul est de 20ns, une requête mémoire traverse 20 étages (allé/retour) en 20 ns si le temps de commutation est de 1ns. On doit avoir 10 240 commutateurs à 1ns !!!

I. États de l'art

4. Problématique

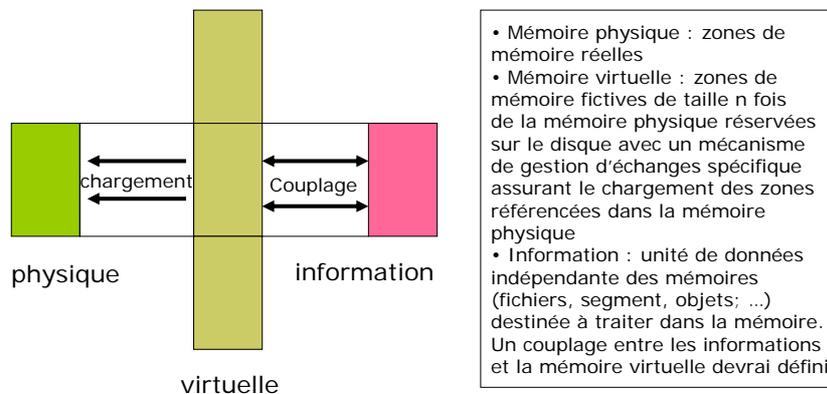
□ Gestion de la mémoire



I. États de l'art

4. Problématique

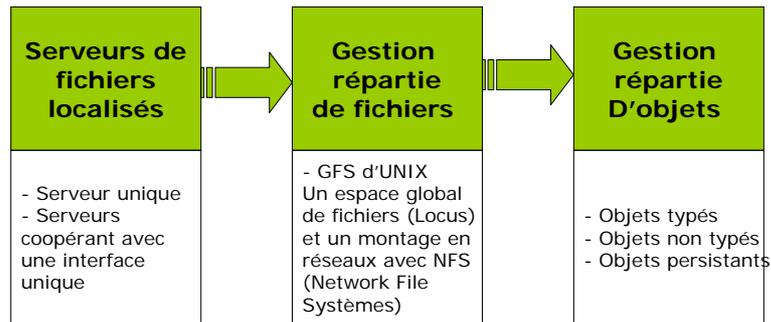
□ Gestion de la mémoire : Mémoires physique, virtuelle et information



I. États de l'art

4. Problématique

□ Gestion des informations



I. États de l'art

4. Problématique

□ Gestion des informations : accès au données

Séparation des fonctions de partage et de stockage

■ Stockage

- Conservation permanente, fiabilité, disponibilité
- Archivage à très long terme (hiérarchies)
- Espace partagé + espace privé
- Enregistrement définitif des transactions

■ Partage

- Accès efficace
- Accès concurrent, mémoire de travail

I. États de l'art

4. Problématique

□ Gestion des informations: Persistance définie par transitivité d'accès (depuis racine persistante prédéfinie)

- Interêt :
 - supprimer la dualité fichiers-variables (requêtes-traitements dans le SGBD)
 - survie des informations après défaillance matérielle ou logicielle
- Problèmes :
 - représentation multiple des références inter-objets (mémoire/disque)
 - sauvegarde d'un état cohérent (transactions atomiques)
 - Ramassage des miettes
 - degré de connaissance de la structure interne par le système
 - lien avec langages

I. États de l'art

4. Problématique

□ Schéma global

Etat global
Temps virtuel

services de base
algorithmes répartis

Services de base

- Temps physique
- Désignation
- Mécanismes d'appel
- Gestion des exécutions
- Gestion des mémoires
- Gestion des informations

Calculs distribués

- Exclusion mutuelle
- Diffusion fiable
- Élection

Observations

- Interblocage
- Terminaison
- Points de reprise

I. États de l'art

5. Développement et déploiement

□ Consigne de développement

- Duplication pour augmenter disponibilité
- Exploiter informations invariantes
- Délais de garde (incertitude sur état distant)
- Caches pour exploiter la localité de référence
- Utilisation d'indications (si valide, gain de temps ; sinon, détection assurée)
 - combinaison des 2 derniers : indicateurs en cache
- Utilisation d'un mécanisme standard d'appel distant
 - appel synchrone + parallélisme local
- Compromis entre disponibilité (rapide) et cohérence
 - la cohérence faible est souvent suffisante
- Ne faire confiance qu'aux machines physiquement protégées
- Utilisation d'algorithmes repartis standard (prouvés)
- Actions atomiques et diffusion fiable

I. États de l'art

5. Développement et déploiement

□ Facteurs du développement

- **Évolution de la demande et des moyens**
 - **Besoins des utilisateurs**
 - amélioration des applications existantes
 - applications nouvelles
 - **Progrès technologique**
 - informatique
 - communications
- **Changements de perspective**
 - Globalisation des systèmes d'information
 - Effet de taille
 - Intégration

I. États de l'art

5. Développement et déploiement

□ Facteurs technologiques

- **Évolution des machines**
 - Processeurs RISC rapides (horloge à 200MHz)
 - Capacité d'adressage (64 bits)
 - Capacité des disques (Terabytes >> Teraflops)
- **Évolution des réseaux**
 - Débit (vers le Gbit/s) - ATM pour réseaux locaux
 - Connectivité (accès partout) + usagers mobiles
 - Connaissance de l'heure physique exacte + localisation (GPS)
- **Évolution des systèmes**
 - Systèmes ouverts (standards)
 - Micro-noyaux et systèmes spécialisés
 - Grande mémoire physique (cache)
 - Fiabilité

I. États de l'art

5. Développement et déploiement

□ Facteurs technologiques : exemples

- **Extension du domaine de faisabilité**
 - Grand espace d'adressage + réseaux rapides
 - mémoire virtuelle répartie
 - partage d'informations complexes avec degré de cohérence spécifié
 - Vitesse CPU + disponibilité temps global
 - regain d'intérêt pour les méthodes synchrones
 - Grands disques + réseaux rapides + connectivité
 - base d'information globale
- **Rapprochement de thèmes de recherche**
 - systèmes répartis - systèmes parallèles
 - systèmes répartis - bases de données - langages de programmation

I. États de l'art

5. Développement et déploiement

- Facteurs de taille
 - **Un "grand" système réparti ?**
 - Nombre d'entités
 - Nombre de composants (machines, réseaux, ...)
 - Nombre d'utilisateurs
 - Nombre (taille, complexité) des informations conservées
 - Étendue géographique
 - Nombre d'organisations responsables
 - **Propriétés recherchées : capacité de croissance**
 - algorithmes (localisation, recherche d'information, communication) maîtrise de la complexité

I. États de l'art

5. Développement et déploiement

- Facteurs de taille : Propriétés du système influencées par la taille
 - La composition "instantanée" du système n'est pas connue
 - Les informations ne sont pas cohérentes
 - Le système est hétérogène
 - Il y a au moins un sous-système en panne ou inaccessible
 - Les entités (machines, usagers, informations) sont mobiles
 - Le système évolue en permanence

I. États de l'art

5. Développement et déploiement

□ Facteurs de taille : problèmes

- Désignation
 - Décentralisation du service ; abandon des identificateurs universels
 - Usage intensif des caches
 - Réorganisation de l'espace des noms
- Maîtrise des performances quantitatives et qualitatives
 - Débits, temps de réponse, ...; qualité de service
- Disponibilité
 - Réplication des composants critiques
- Sécurité
 - Petit nombre de composants critiques
 - Authentification par cryptographie ; capacités
- Administration
 - Complexité, hétérogénéité

I. États de l'art

5. Développement et déploiement

□ Facteurs d'intégration

- Évolution des "métiers" de l'informatique
 - Demande de solutions globales par les utilisateurs
 - intégration des domaines
 - Systèmes ouverts
 - recherche de compétitivité dans les couches hautes & applications
- Maturation de la discipline informatique
 - Exemples de rapprochements :
 - Systèmes - Langages - Bases de données (aussi influence de l'IA)
 - Systèmes répartis – Systèmes parallèles
 - Conséquences :
 - Infrastructures de systèmes pour BD
 - Utilisation de techniques de BD dans les systèmes
 - Langages & systèmes à objets persistants

I. États de l'art

5. Développement et déploiement

□ Critère de transparence

Même mode d'utilisation que celui d'un système centralisé en temps partagé : niveau d'utilisateur et niveau de programme

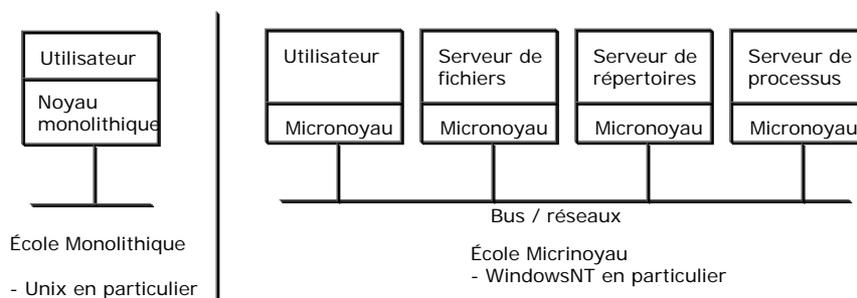
Types de transparence	Signification
à l'emplacement	L'utilisateur ne connaît pas où sont situées les ressources
à la migration	Les ressources peuvent être déplacées sans modification de leur nom
à la duplication	L'utilisateur ne connaît pas le nombre de copies existantes
à la concurrence	Plusieurs utilisateurs peuvent partager en même temps les mêmes ressources
au parallélisme	Des tâches peuvent être exécutées en parallèle sans que les utilisateurs le sachent

I. États de l'art

5. Développement et déploiement

□ Critère de souplesse

La facilité de modification de configuration et d'extension



I. États de l'art

5. Développement et déploiement

□ Critère de fiabilité

Disponibilité	La disponibilité est la fraction de temps pendant laquelle le système est utilisable : <ul style="list-style-type: none">-limiter de nombre des composants critiques-dupliquer les parties clés des composants logiciels et matériels (redondance)-maintenir la cohérence des copies (contradiction!!!)
Sécurité	Les ressources doivent être protégées contre des utilisations abusives et malveillantes. En particulier le problème de piratage des données sur le réseau de communication
Tolérance aux pannes	Le système doit être conçu pour masquer les pannes aux utilisateurs. La panne de certains serveurs (ou leur réintégration dans le système après la réparation) ne doit pas perturber l'utilisation du système en terme de fonctionnalité (NFS sans état par exemple)

I. États de l'art

5. Développement et déploiement

□ Critère de performance

Critères	<ul style="list-style-type: none">-Minimiser le temps de réponse-Maximiser le débit (nombre de travaux/h)-Maximiser le taux d'utilisation du système-Maximiser le taux d'utilisation de la bande de passante
Problèmes	<ul style="list-style-type: none">- La communication est en général assez lente dans les systèmes distribués- Le mécanisme de reprise sur panne consomme beaucoup de temps
Solutions	<ul style="list-style-type: none">- minimiser les échanges de message- maximiser des granules grosses (gros grains) et éviter des grains fins pour les calcul à distance- réduire le champ d'application des mécanismes de reprises sur pannes

I. États de l'art

5. Développement et déploiement

□ Critère de dimensionnement

Critères	Minimiser les goulots d'étranglement potentiels
Problèmes	- Composants centralisés - exemple : Un seul serveur de courrier pour tous les utilisateurs - Données centralisées – exemple : Un seul annuaire en ligne - Algorithmes centralisés – exemple : Avoir un routage qui nécessite une connaissance totale du réseau
Solutions	Utilisation des algorithmes distribués : - les programmes exécutables sur les postes clients - aucune machine n'a une information complète sur l'état du système - les machines prennent des décisions à partir des seules informations locales disponibles - la panne d'une machine n'empêche pas l'algorithme de fonctionner - on ne fait pas l'hypothèse de l'existence d'une horloge globale