

Introduction aux systèmes répartis

Prof. Babahenini Mohamed Chaouki

Département d'Informatique
Université Mohamed Khider - Biskra

Introduction aux systèmes répartis

1. Introduction
2. Définitions
3. Middleware
4. Caractéristiques
5. Domaines d'application et types
6. Problèmes

Systèmes centralisés

- Tout est localisé sur la même machine
 - 1 processeur : une horloge commune
 - 1 mémoire centrale : un espace d'adressage commun
 - 1 système d'exploitation
- Gestion centralisée des ressources
- Etat global du système facilement reconstituable
- Accès local aux ressources
- Les décisions pour l'objectif du système sont prises dans un mécanisme central et transmises aux composants exécutifs.

Emergence du réparti

- Evolution technologique
 - machines:
de plus en plus performantes avec une baisse des prix
 - Équipement réseau:
de plus en plus rapides
- Les progrès technologiques des ordinateurs et le haut débit permettent aux différentes applications sur des ordinateurs distincts (et distants) de coopérer pour effectuer des tâches coordonnées.

Introduction

Pourquoi un système réparti ?

- 1. Partage des ressources (données, applications, périphériques chers). Optimisation de leur utilisation.
- 2. Tolérance aux pannes (fiabilité, disponibilité).
- 3. Adaptation de la structure d'un système à celle des applications (géographique ou fonctionnelle).
- 4. Interconnexion de machines dédiées (ex : MVS-CICS + PC windows).
- 5. Besoin de communication et de partage d'information.
- 6. Concurrence, parallélisme $\Rightarrow$ efficacité.
- 7. Prix des processeurs de petite puissance inférieur à ceux de grande puissance $\Rightarrow$ raisons économiques. (partage des ressources, construire à partir de l'existant).
- 8. Flexibilité, facilité d'extension du système (matériels, logiciels). Sauvegarde de l'existant.

Définition d'un système réparti

Un système réparti ?

Un ensemble d'ordinateurs indépendants qui apparaît à ses utilisateurs comme un seul système cohérent [Tanenbaum et van Steen, 2007]

Vue utilisateur

- C'est une définition possible, qui rend compte d'une vue orientée utilisateur
- On peut aussi l'appeler la définition informaticienne d'un système
- Du point de vue de l'ingénierie, est une sorte de définition a posteriori

Définition d'un système réparti

Un système distribué est constitué d'une multiplicité de composants

- Entités de calcul indépendantes / autonomes (ordinateurs, microprocesseurs, . . .)
- Aucune hypothèse sur leur nature individuelle, leur structure, leur comportement, . . .
- Hétérogénéité

Un système distribué peut être vu comme un seul système cohérent

- Selon le point de vue de l'utilisateur ou de l'ingénieur, ou les deux
- Besoin de cohérence sur la multiplicité et l'hétérogénéité

Définition du système distribué : Quelques remarques

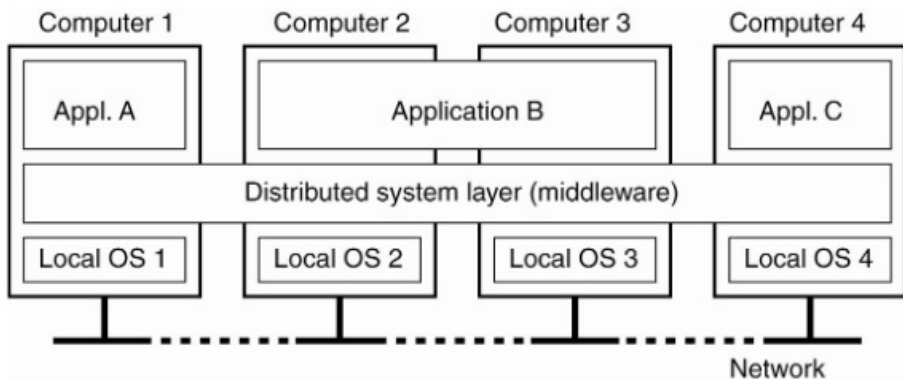
Un système réparti ?

Un ensemble d'entités de calcul autonomes conçues comme un seul système cohérent par son concepteur.

Vue ingénieur

- C'est une autre définition possible, qui rend compte d'une vision constructive, orientée design
- Nous pouvons également l'appeler la définition d'ingénieur informatique d'un système distribué
- Du point de vue de l'ingénierie, est une sorte de définition à priori

Architecture d'un système réparti



Un système distribué organisé en middleware. La couche middleware s'étend sur plusieurs machines et offre à chaque application la même interface [Tanenbaum et van Steen, 2007]

Middleware: Principes

Collaboration et hétérogénéité

Solution par séparation

- La couche middleware permet une interaction significative entre les composants distribués autonomes
 - des problèmes de communication comme la syntaxe, la sémantique, ...
- La couche middleware masque les différences de technologie, de structure, de comportement, ...
 - fournit une interface partagée commune pour les applications et les composants, comme les systèmes d'exploitation

Middleware: Interlogiciel

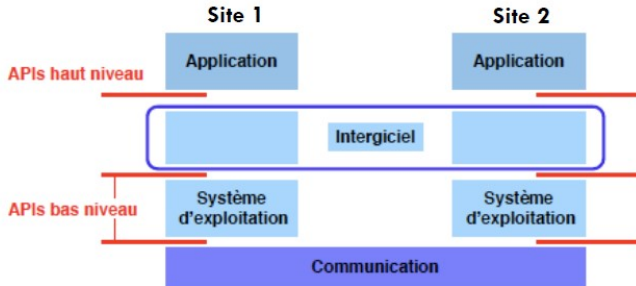
Problème

Dans un système réparti, l'interface fournie par les systèmes d'exploitation et de communication est encore trop complexe pour être utilisée directement par les applications:

- Hétérogénéité (matérielle et logicielle)
- Complexité des mécanismes (bas niveau)
- Nécessité de gérer la répartition

Solution

- Introduire une couche logicielle intermédiaire (répartie) entre les niveaux bas (systèmes et communication) et le niveau haut (applications) : c'est l'intergiciel (Middleware en anglais)
- L'intergiciel joue un rôle analogue à celui d'un "super-système d'exploitation" pour un système réparti

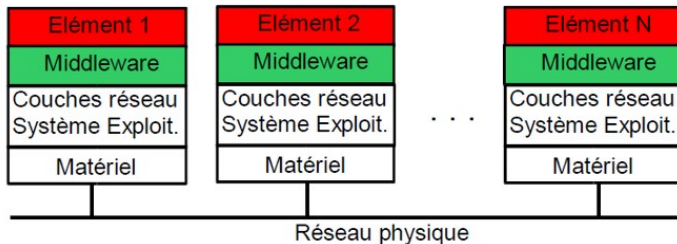


Définition middleware

- Un middleware ou intergiciel ou élément du milieu est l'ensemble des couches réseau et services logiciel qui permettent le dialogue entre les différents composants d'une application répartie.
- Gartner Group définit le middleware comme une interface de communication universelle entre processus.
- Représente l'élément le plus important de tout système réparti.

Fonctions du middleware

- Masquer l'hétérogénéité (des machines, systèmes, protocoles de communication)
- Fournir, ce qui permettra de masquer la complexité des échanges et de faciliter le développement d'une application répartie
- Rendre la répartition aussi invisible (transparente) que possible
- Fournir des services répartis d'usage courant



Services du middleware

- Conversion
 - permet la communication entre machines mettant en œuvre des formats différents de données
 - prise en charge par la FAP (Format And Protocol)
- Nommage
 - permet d'identifier la machine serveur sur laquelle est localisé le service demandé afin d'en déduire le chemin d'accès.
 - fait, souvent, appel aux services d'un annuaire.
- Sécurité
 - permet de garantir la confidentialité et la sécurité des données à l'aide de mécanismes d'authentification et de cryptage des informations
- Communication
 - permet la transmission des données entre les deux systèmes

Middleware: Communication - Exemple

- Exemple basique de protocole
 - Une entité envoie des données à une deuxième entité
 - La deuxième entité envoie un acquittement pour prévenir qu'elle a bien reçue les données
 - Mais si utilise un réseau non fiable ou aux temps de transmission non bornés
 - Comment gérer la perte d'un paquet de données ?
 - Comment gérer la perte d'un acquittement ?
 - Comment gérer qu'un message peut arriver avant un autre alors qu'il a été émis après ?

Middleware: Modèles d'interaction

- Les éléments distribués interagissent, communiquent entre eux selon plusieurs modèles possibles
 - Client/serveur, Diffusion de messages, Mémoire partagée, Pair à pair, ...
- Abstraction/primitive de communication basique
 - Envoi de message d'un élément vers un autre élément
 - A partir d'envois de messages, peut construire les protocoles de communication correspondant à un modèle d'interaction
- Rôle des messages
 - Données échangées entre les éléments : Demande de requête, Résultat d'une requête, Donnée de toute nature, ...
- Gestion, contrôle des protocoles
 - Acquittement : message bien reçu
 - Synchronisation, coordination ...

Caractéristiques d'un système réparti

Plusieurs horloges non synchronisées + communications asynchrones $\Rightarrow$ plus d'état global facilement calculable et présence d'indéterminisme logique.

- Activités parallèles: Composants autonomes exécutant des tâches concurrentes
- Absence d'état global
 - Pas de référence spatiale commune à cause de l'absence (dans la majorité des cas) d'une mémoire partagée
 - Pas de référence temporelle commune à cause de l'existence de plusieurs processeurs ayant chacun sa propre horloge
- Existence d'un réseau
 - non géré par le système d'exploitation
 - le comportement du système réparti dépend de l'état du réseau

Ce qu'offre un système réparti

Transparence 1

- Propriété fondamentale : Tout cacher à l'utilisateur
 - La répartition doit être non perceptible : une ressource distante est accédée comme une ressource locale
 - Cacher l'architecture et le fonctionnement du système réparti
- L'ISO définit plusieurs transparences
 - accès, localisation, concurrence, réplication, mobilité, panne, performance, échelle

1. Transparence à la localisation $\Rightarrow$ désignation. L'utilisateur ignore la situation géographique des ressources. Transparence à la migration.

2. Transparence d'accès. L'utilisateur accède à une ressource locale ou distante d'une façon identique.

3. Transparence à l'hétérogénéité $\Rightarrow$ interopérabilité L'utilisateur n'a pas à se soucier des différences matérielles ou logicielles des ressources qu'il utilise.

Ce qu'offre un système réparti

Transparence 2

Sources d'hétérogénéité: machines (architecture matérielle), systèmes d'exploitation, langages de programmation, protocoles de communications.

4. Transparence aux pannes (réseaux, machines, logiciels). Les pannes et réincarnations sont cachées à l'utilisateur. Transparence à la réplication. Un système réparti doit pouvoir tolérer la panne des machines : (Une machine tombe en panne, Une machine envoie des informations erronées, Une machine n'est plus atteignable (problème réseau)).

5. Transparence à l'extension des ressources. Extension ou réduction du système sans occasionner de gêne pour l'utilisateur (sauf performance).

Ce qu'offre un système réparti

Scalability (dimensionnement)

- Le système doit fonctionner efficacement lors du passage à l'échelle
- Le système doit rester performant lorsque le nombre d'utilisateurs ou de ressources augmente
 - Performance: La complexité de l'algorithme ne doit pas dépasser un certain seuil lorsque la taille des données augmente.
 - Eviter les goulots d'étranglement: Utiliser des algorithmes décentralisés

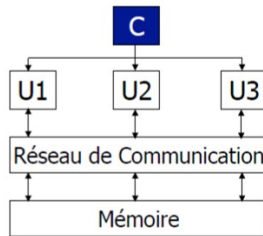
Concurrence

- Offrir et gérer les accès concurrent aux ressources partagées:
 - Garantir un ordonnancement "juste" et performant
 - Eviter les inter blocages

Systèmes distribués vs parallèles

Systèmes Parallèles.

1. Une machine multiprocesseurs avec un environnement du type SIMD (tous les processeurs exécutent le même programme et ont une vision uniforme de l'état global du système).
2. Extensible à un réseau de machines asynchrones fortement couplées

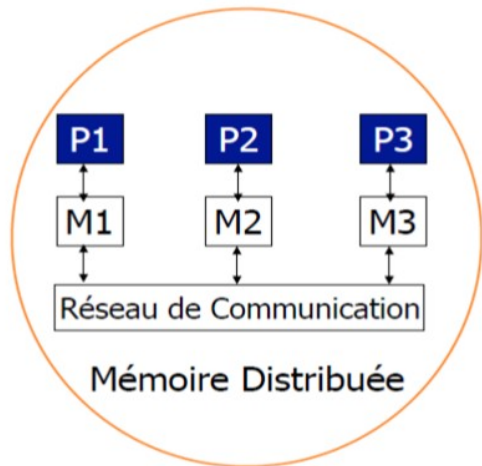


C : Unité de Contrôle
U : Unité de Calcul

Systèmes distribués vs parallèles

Systèmes distribués.

1. processus indépendants sur des machines distinctes
2. et communiquant par échange de messages asynchrones (en général, des réseaux faiblement couplés).



Domaines 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
- Centres de documentation, bibliothèques
 - Recherche, navigation, visualisation multimédia
- Systèmes d'aide à la formation

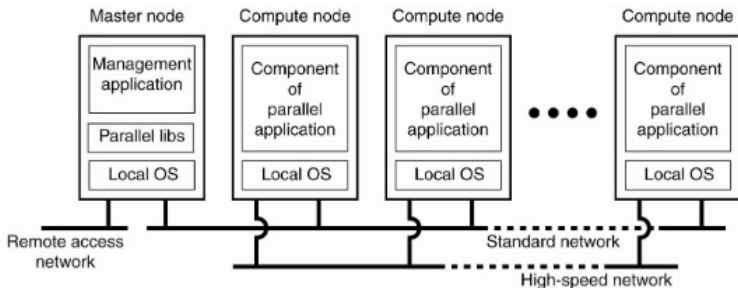
Types de systèmes répartis

- Systèmes informatiques distribués
 - Utilisé pour les tâches de calcul haute performance
 - Systèmes de cluster et Cloud computing
 - Systèmes de calcul en grille
- Systèmes d'information distribués
 - Systèmes principalement pour la gestion et l'intégration des fonctions commerciales
 - Systèmes de traitement des transactions
 - Intégration d'applications d'entreprise
- Systèmes pervasive (ou Ubiquitous)
 - Systèmes mobiles et embarqués
 - Systèmes domestiques
 - Réseaux de capteurs

Types de systèmes répartis

Systèmes informatiques distribués : Systèmes informatiques en grappe (cluster)

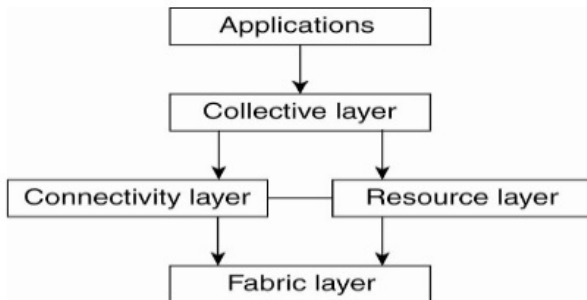
- Collection de PC similaires, étroitement connectés, exécutant tous le même système d'exploitation
- Un exemple de système informatique en cluster.



Types de systèmes répartis

Systèmes informatiques distribués : Systèmes informatiques en grappe (cluster)

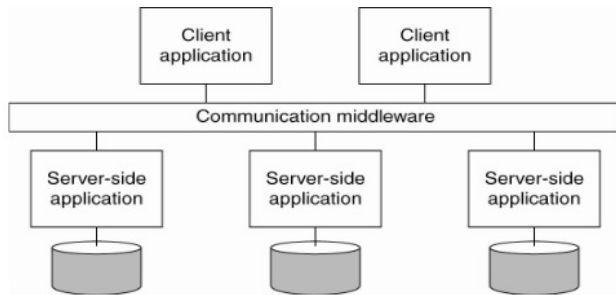
- Association de systèmes informatiques autonomes et hétérogènes (HW, OS, ...), plusieurs domaines adm
- Une architecture en couches pour les systèmes informatiques en grille.



Types de systèmes répartis

Systèmes d'information distribués: Intégration d'applications d'entreprise

- Permettre aux applications existantes d'échanger directement des informations à l'aide d'un middleware de communication
- Middleware en tant que facilitateur de communication dans l'intégration d'applications d'entreprise.

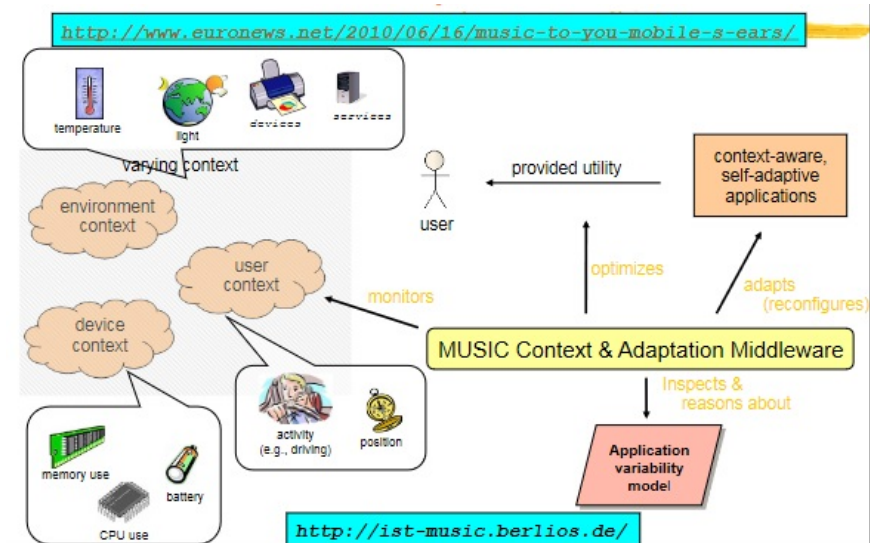


Types de systèmes répartis

Systèmes ubiquitaires distribués: Distributed pervasive system

- un système informatique disponible partout au sein d'une entreprise ou en tant que système grand public qui apparaît le même partout avec les mêmes fonctionnalités mais qui fonctionne à partir de la puissance de calcul, du stockage et des emplacements distants.
- consistent à exploiter l'intégration croissante des services et des (petits/minuscules) appareils informatiques dans notre monde physique quotidien.
- Il aurait des contrôles et une sécurité conformes aux réglementations locales, mais gérerait cela sur l'ensemble du réseau.
- Les appareils mobiles dans les systèmes ubiquitaires distribués découvrent l'environnement (ses services) et s'y établissent le mieux possible.
- Exigences pour les applications omniprésentes : Accepter les changements contextuels, Encourager la composition ad hoc et dynamique, Reconnaître le partage.

Exemple d'un système ubiquitaire



Où sont les problèmes ?

- Y 'a t-il un algorithme pour détecter la terminaison distribuée d 'une application ?
- Comment modéliser et exprimer les algorithmes distribués? Comment les prouver ?
- IComment les analyser, calculer leur complexité, les classifier etc. . . ?
- Difficulté de l'algorithmique distribuée / centralisé:
 - Pas de connaissance de l'état global
 - Absence de temps universel (ou horloge globale)
 - Non déterminisme (lié souvent au problème du synchronisme)
 - Et surtout pas de modèle "universel" et standard pour l'algorithmique distribuée