

Partie 1 Fondements de Systèmes Distribués

11

1. Définition d'un SD
2. Exemples de SDs
3. caractéristiques des SDs
4. Difficultés dans les SDs
5. Modèles de SDs

12

12

Quelque mots clés

Réseaux, Ubiquitous, omniprésente pervasive, Internet ,cloud, computing as a utility, services, location aware, user generated content, web, 3G, 4G, 3G, bluetooth, réseaux, protocoles, mobile computing, personal area, wireless network WPAN , PAN,

13

1. SD :Définitions(1)

Définition1 [Tanenbaum]: *Un ensemble d'ordinateurs indépendants qui apparaît à un utilisateur comme un système unique et cohérent : Les machines sont autonomes et les utilisateurs ont l'impression d'utiliser un seul système.*

Définition2 Ensemble composé d'éléments (fonctions de traitement (processeurs), de stockage (mémoire), de relation avec le monde extérieur (capteurs, actionneurs)) reliés par un système de communication.

Définition3 Les différents éléments du système ne fonctionnent pas indépendamment mais collaborent à une ou plusieurs tâches communes.

14

1. SD :Définitions(3)

Machines reliées par un réseau (où la distance n'a aucun impact).

Dans les SDs, les **programmes** qui s'exécutent sont en **concurrence** (exemple : accès multiples aux mêmes ressources mais la capacité du SD à gérer les ressources partagées est renforcée par l'ajout de nouvelles composantes.)

Les programmes qui partagent les ressources **coopèrent**: ils **coordonnent** leurs actions en échangeant des messages (problèmes de synchronisation: absence d'une horloge globale)

15

1. SD :Définitions(4) Centralisé vs Distribué

L'architecture **centralisée** supporte un **noyau central** l'architecture distribuée supporte **plusieurs**.

Le coût de l'architecture varie suivant le domaine.

Si les périphériques ne sont pas utilisés à plein temps (par exemple, une imprimante), l'architecture centralisée est plus économique (car on économise en se basant sur le fait que tous les périphériques ne seront jamais tous utilisés en même temps).

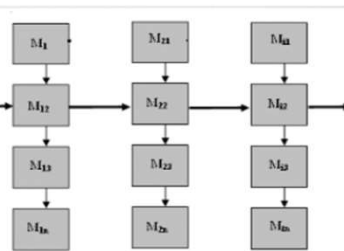
Dans le cas contraire (carte vidéo, réseau de PC), c'est l'architecture distribuée la plus économique (un gros ordinateur coûte plus cher que 10 petits ordinateurs mais est 10 fois plus puissant).

16

1. SD :Définitions(4) parallèle vs distribué

Les ordinateurs parallèles sont des machines qui comportent une architecture parallèle, constituée de plusieurs processeurs identiques, ou non, qui concourent au traitement d'une application.

La performance d'une architecture parallèle est la combinaison des performances de ses ressources et de leur agencement (Latence, débit).



17

17

1. SD :Définitions(5) parallèle vs distribué

Systèmes parallèles	Systèmes répartis	Systèmes d'exploitation de réseaux
Processeurs	Sites	Ressources
Homogènes	Hétérogènes	Hétérogènes
Partage ou non de mémoire	Mémoires individuelles	Mémoires individuelles
Couplage fort	Couplage faible	Couplage faible
Topologie du réseau d'interconnexion	Réseau LAN + WAN	Réseau LAN
Les users sont au courant de la multiplicité des Processeurs	Les users ont l'impression d'être dans un système centralisé	Les users sont au courant de la multiplicité des Machines

18

2. Exemples de SDs (1) SDs populaires

- Le web : World WideWeb.
- Systèmes de **messaging électronique**
- Moteurs de recherche: **google** search engine, yahoo, **bing**..
- Réseaux sociaux**: FB, myspace, instagram, tweeter
- Online gaming** :
- Plateformes de **paiement et de e-commerce**: Amazon, ebay, **Pay-Pal**..

19

2. Exemples de SDs (1) SDs populaires

- Systèmes de géo-localisation ..**GPS**, traffic management intelligent vehicles,
 - web based map services **MapQuest**, **GoogleMaps**
 - Environnements de e-learning (virtual learning, distance learning, community based learning) : FUN, EDX, UDEMY..
 - Grille: technologie de partage pour la e-science
 - Les clouds AWS(Amazon), AZUR(Microsoft),.
 - Systemes de surveillance pour la météorologie
- Internet**

20

2. Exemples de SDs (2): les Systèmes de messagerie

Enoncé : Considérant un système informatique constitué d'un ensemble de stations reliées entre elles par un moyen de communication. On veut implémenter un système de messagerie tel qu'un usager peut émettre ou retirer des messages de n'importe quelle station.

Implémentations possibles :

- Architecture **centralisée** ;
- Architecture **décentralisée - ou répartie** ;
- Architecture **mixte**.

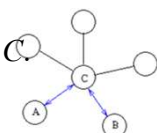
21

21

2. Exemples de SDs (2): les Systèmes de messagerie

Architecture centralisée

- Tous les courriers sont stockés sur *C* (station centrale).
- 1 boîte aux lettres sur *C* pour chaque usager (*A*,*B*,...).
- 1 opération = 1 transfert d'informations (émission/retrait).
- ☐ Volume de stockage important sur *C*.
- ☐ Disponibilité du service = disponibilité de *C*.

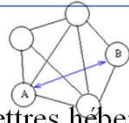


22

22

2. Exemples de SDs (2): les Systèmes de messagerie

Architecture Répartie



Si Dépôt ou retrait dans une boîte aux lettres hébergée sur la station de rattachement **Alors** Retrait d'un message sur station de rattachement = opération locale.

Sinon Si Dépôt ou retrait sur une autre station **Alors** le volume de stockage global est réparti sur l'ensemble des stations, l'accès à un espace usager donné nécessite la mise en œuvre d'un dictionnaire Usager/Station-de-rattachement.

23

23

2. Exemples de SDs (2): les Systèmes de messagerie

Une **disponibilité accrue** est exigée à l'architecture **Répartie**

-**Panne d'une station** : empêche la réception des messages (**uniquement**) pour les usagers de cette station.

-Possibilité d'avoir **des copies** du répertoire sur d'autres machines.

Une implémentation en **architecture mixte** (données et traitements répartis avec contrôle centralisé) est également possible

24

24

2. Exemples de SDs (3): Le Web

-**Dépôt de documents** (textuels, audio-visuel, Information et connaissances..)

Renforcé par l'évolution des :

-**moteurs de recherche** (google, Yahoo,..)

-**User generated content** : Youtube, Wikipedia, Flickr, Blogs..

-Réseaux sociaux FB, Twitter, Myspace..

-**Streaming content** : netflix

25

2. Exemples de SDs(4) Les moteurs de recherche

•La recherche sur le Web est une industrie de croissance majeure au cours de la dernière décennie.

•La **tâche d'un moteur de recherche Web** est d'**indexer** l'intégralité du **contenu** du **World Wide Web**, englobant une large gamme de styles d'informations, y compris les pages web, les sources multimédia et des livres (scannés).

26

2. Exemples de SDs(4) Les moteurs de recherche

- Les moteurs de recherche analysent l'ensemble du contenu Web et portent ensuite un traitement sophistiqué sur cette énorme base de données : cette tâche représente elle-même un défi majeur pour la conception de SDs.
- Google, le leader du marché de la technologie de recherche sur Internet, a consacré beaucoup d'efforts à la conception d'une infrastructure sophistiquée du système distribué pour soutenir la recherche (et en fait d'autres applications et services Google tels que Google Earth).

27

2. Exemples de SDs(4) Les moteurs de recherche

Les points forts de l'infrastructure Google sont entre autres:

- une **infrastructure physique sous-jacente** composée d'un très grand nombre de machines connectées situés dans des datacentres du monde entier;
- un **SGF distribué** conçu pour prendre en charge des fichiers très volumineux et fortement optimisé pour le style d'utilisation requis par la recherche et d'autres applications Google (en particulier la lecture de fichiers à des taux élevés et soutenus).

28

2. Exemples de SDs(4) Les moteurs de recherche

Parmi les points forts de l'infrastructure Google, nous citons aussi:

- un système de stockage distribué structuré associé qui offre un accès rapide à de très grands ensembles de données;
- un service de verrouillage qui offre des **fonctions système distribuées** (verrouillage distribué, accord distribué...);
- un **modèle de programmation** prenant en charge la gestion de **très grands calculs parallèles et distribués** dans l'infrastructure physique sous-jacente.

29

2. Exemples de SDs(5) Les MMOGs

Les jeux massivement multijoueurs en ligne: **Massively Multiplayer Online games MMOG** offrent une expérience immersive par laquelle des utilisateurs (centaines de milliers de joueurs simultanés en ligne) interagissent via Internet avec un monde virtuel persistant.

30

2. Exemples de SDs(5) Les MMOGs

Le MMOG est un jeu vidéo qui se définit traditionnellement par **3 critères** cumulatifs :

- l'**univers** n'est accessible que par un **réseau** ;
- l'**univers** est **persistant**, c'est-à-dire qu'il existe tout le temps, que des joueurs y soient connectés ou non ;
- l'**univers** est **accessible** à un très grand nombre de joueurs **simultanément**.

31

2. Exemples de SDs(5) Les MMOGs

L'ingénierie des MMOG représente des défis :

- Préserver l'expérience utilisateur du jeu afin d'assurer délais de réponse rapides.
- Propager en temps réel des événements aux nombreux acteurs et maintenir d'une vision cohérente du monde partagé.

32

2. Exemples de SDs(5) Les MMOGs

- Ressources distribuées** vues comme produits de base ou **utilitaires**.
- Ressources fournies par les **fournisseurs de services** (ressources), (comme Amazon, Google, Azure de Microsoft,..) sont **louées** efficacement plutôt que **détenues par l'utilisateur final**.
- Ce modèle s'applique à la fois aux **ressources physiques** et aux **services logiques**:

33

2. Exemples de SDs(6) Les Clouds

Les **clouds** (nuages) sont généralement implémentés sur les machines du cluster pour fournir l'échelle nécessaire et la performance requise par ces services.

Un **cluster** (groupement) est un ensemble de machines interconnectées qui coopèrent étroitement pour fournir une capacité informatique unique et intégrée haute performance

34

2. Exemples de SDs(6) Les Clouds

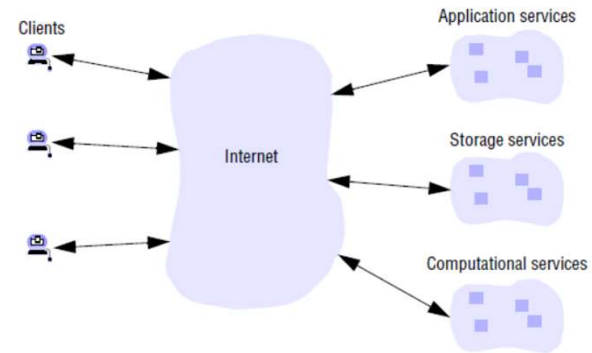
-Les **ressources physiques en cloud**: (stockage, traitement) sont mises à la disposition des machines en réseau, éliminant ainsi la nécessité de posséder ces mêmes ressources.

La **virtualisation du SE** est une technologie clé pour cette approche: les services sont souvent offerts par un nœud virtuel plutôt que physique.

35

2. Exemples de SDs(6) Les Clouds

Cloud computing



36

2. Exemples de SDs(6) Les Clouds

Grâce à ce modèle, un utilisateur peut:

- Effectuer un stockage à distance de fichiers (données multimédia, sauvegardes,...).
- Louer des nœuds de calcul (besoins informatiques de base, calcul distribué,...).
- Accéder à distance à des centres de données sophistiqués ou bien à une infrastructure informatique précise.

37

2. Exemples de SDs(6) Les Clouds

• **Les services logiciels (Software services) en cloud:**

Des gammes de services sont offertes avec des formules de location efficaces (messagerie électronique, calendriers distribués...)

Exemple : Google, regroupe une gamme de services pro sous la bannière Google Apps [www.google.com].

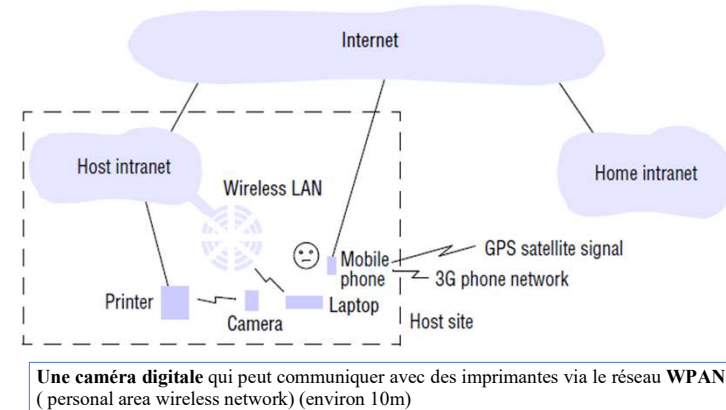
Ce développement est rendu possible par les normes convenues pour les services logiciels (exemple : les normes pour les services Web).

38

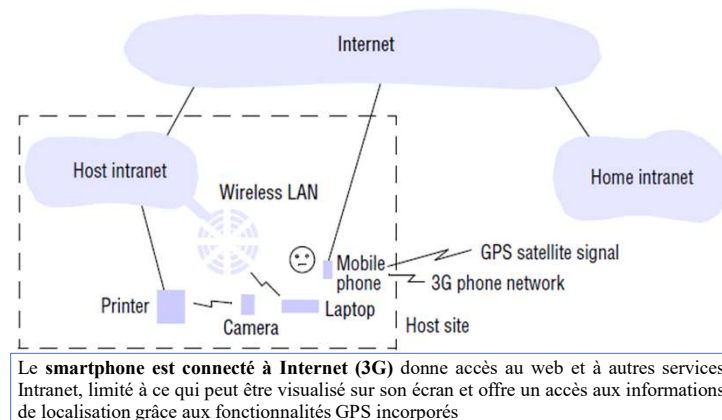
Nouvelles technologies et tendances:

- Emergence de la technologie de l'informatique **omniprésente (ubiquitous)**
- Mobilité des utilisateurs : **Mobile computing**
- **Location aware computing** et context aware computing
- services multimédias et **User generated content**
- Les services et le **computing as a utility**

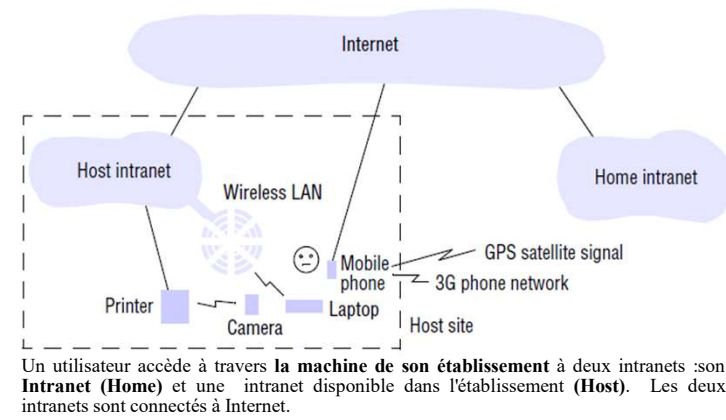
39



40



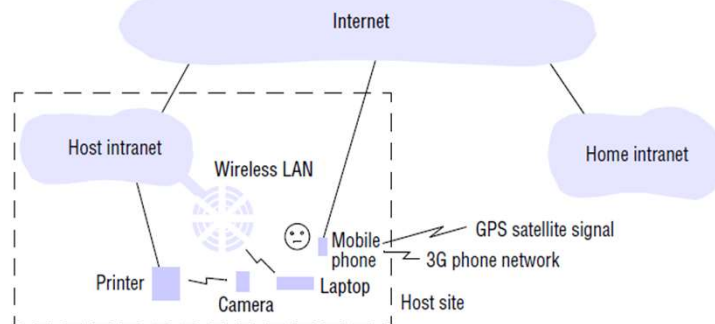
41



42

2. Exemples de SDs(7) Système Très distribué: Internet

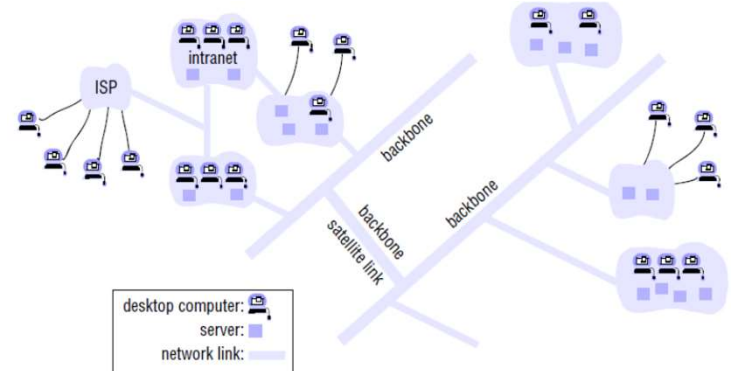
L'utilisateur a accès aux trois formes de connexion sans fil :



Son ordinateur portable peut se connecter au réseau local sans fil (s'étendant sur quelques centaines de mètres). Il se connecte au reste du réseau local à travers une passerelle ou un point d'accès.

43

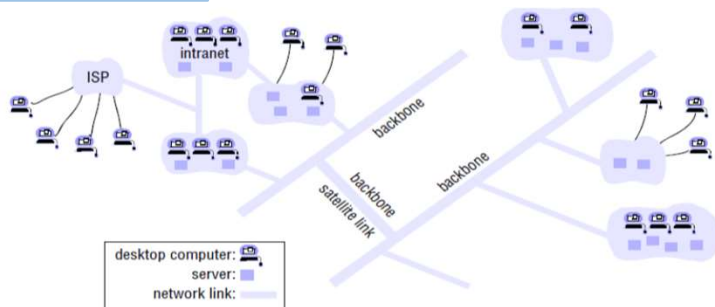
2. Exemples de SDs(8) Internet : réseau de réseaux



collection d'intranets : d'entreprises et d'organisations typiquement protégées de Firewalls (parefeu: qui protège en contrôlant les flux de messages entrants et sortants).

44

2. Exemples de SDs(8) Internet : réseau de réseaux



Les (ISP) sont les entreprises qui fournissent des liens large bande et d'autres types de connexion à des utilisateurs individuels et à de petites organisations..

Les intranets sont reliés entre eux par des backbones.

Un réseau fédérateur est une liaison réseau à forte capacité de transmission, utilisant des connexions par satellite, des câbles à fibres optiques et d'autres circuits à large bande passante.

45

3. Caractéristiques des SDs (1)

On demande à un SD d'avoir les caractéristiques suivantes:

- Transparence des répartitions
- Transparence d'accès
- Transparence de location/relocation
- Transparence de réplication
- Ouverture,
- Mise à l'échelle (Scalabilité),
- Tolérance aux pannes
- Sécurité

46

46

3. Caractéristiques des SDs (2) Transparence des Répartitions

.Uniformité des **accès locaux et distants**.

.Le **taux de distribution** de la puissance de calcul, des périphériques et des bases de données dépend de différentes considérations stratégiques, humaines et professionnelles

47

47

3. Caractéristiques des SDs (3): Transparence d' accès

.Représentation réelle des données cachée

.Méthodes d'accès aux données qui cachent les différences entre les architectures

Exemple : différentes conventions de nommage des fichiers liées à l'usage de différents SGFs.

48

48

3. Caractéristiques des SDs (4): Transparence [Loc/Reloc]ation

.Séparation physique entre machines et différences matériels/logiciels sont invisibles par l'utilisateur.

.Localisation des ressources non perceptible (nom logique ex: URL <http://www.labri.fr/>)

.Migration des ressources possible sans interférence avec la localisation physique (ex. transférer un objet uniquement par son nom logique sans modification de ce nom et sans modification de l'environnement d'un utilisateur).

49

49

3. caractéristiques des SDs (5): Transparence de Réplifications

.**Concurrence d'accès aux ressources non perceptible** (ex. accès à un même fichier ou une table dans une base de données: exemple de solution : mécanisme de verrou ou de Transaction)

.**Invisibilité du parallélisme offert par l'environnement d'exécution.**

.**Tolérance aux pannes** permettant à un utilisateur de ne pas s'interrompre (ou même se rendre compte) en cas de pannes de ressources (**répliquées**).

50

50

3. Caractéristiques des SDs (1) Ouverture

- Services** offerts selon des **règles standards** qui décrivent la syntaxe et la sémantique de ces services (Interfaces publiées, ex. IDL)
- Interopérabilité des matériels** (fournisseurs différents)
- Portabilité**
- Flexibilité** (facilité d'utilisation et de configuration)
- Extensibilité** (aux ajouts et MAJ de composants sans en affecter les autres)

51

51

3. caractéristiques des SDs (7): Mise à l'échelle (scalability)

Un SD doit fonctionner efficacement dans **différentes échelles**:

- Deux postes** de travail et un serveur de fichiers.
- Un réseau local** avec **plusieurs centaines** de postes de travail et serveurs de fichiers.
- Plusieurs réseaux locaux** reliés pour former Internet (**millions de machines** reliés à Internet)

52

52

3. caractéristiques des SDs (8): Tolérance aux pannes

Le système doit pouvoir **fonctionner** (au moins de façon dégradée) **même en cas de défaillance de certains de ses éléments**, c'est dû aux mécanismes de

- Détection de pannes** (difficulté et même impossibilité de détection pour certains systèmes, suspicion de machines) exemple : connexion par un navigateur à un serveur distant qui répond pas.
- Correction d'erreurs** (de fichiers/messages corrompus).
- Reprise sur pannes** (tech. de journalisation dans les BD₃₃)

53

3. caractéristiques des SDs (9): Sécurité

Le système doit pouvoir **résister** à des **attaques contre sa sécurité** (**violation** de la **confidentialité**, de l'**intégrité**, **usage indu de ressources**, **déni de service**)

- Ainsi, dans un SD on doit **garantir** :
- La **confidentialité** (authentification)
- L'**intégrité** (protection contre les falsification et les corruptions)
- La **disponibilité** (accès aux ressources) **Exemple**: le commerce électronique, les banques en ligne.

54

54

4. Difficultés rencontrées par les SDs(1)

Parmi les difficultés, nous comptons:

- ♦ **Asynchronisme du système de communication** (pas de borne supérieure stricte pour le temps de transmission d'un message)

→ **Conséquence** : difficulté de détecter les défaillances

55

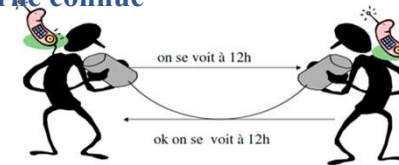
55

4. Difficultés rencontrées par les SDs(2)

Rappelons : la communication Synchrones :

Durée de transmission d'un message d'un nœud à un autre bornée et **borne connue**

Durée d'exécution d'une action interne d'un processus bornée et **borne connue**



56

56

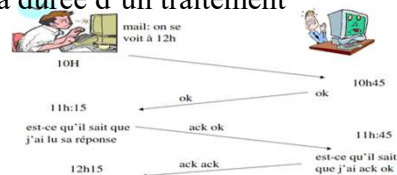
4. Difficultés rencontrées par les SDs(3)

Rappelons : la communication asynchrone :

Borne inconnue sur la transmission d'un message

Borne inconnue sur la dérive des horloges

Borne inconnue sur la durée d'un traitement



57

57

4. Difficultés rencontrées par les SDs(4)

Parmi les difficultés, nous comptons aussi:

- ♦ **Grande échelle** (nombre de composants, d'utilisateurs, dispersion géographique)

→ **Conséquence** : la capacité de croissance (scalability) est une propriété importante, mais difficile à réaliser

Malgré ces difficultés, des grands systèmes répartis existent et sont largement utilisés : le DNS, le web..

58

4. Difficultés rencontrés par les SDs(5)

• **Dynamisme** (la composition du système change en permanence)

→ **Conséquences** : Difficulté de **définir un état global**
Difficulté d'**administrer le système**

59

4. Difficultés rencontrés par les SDs(6)

Modes d'utilisation très variés: les composants des SDs sont soumis à de larges variations de la charge de travail -

Exemple1 : on accède à plusieurs pages web plusieurs millions de fois par jour. Pendant ce temps, certaines parties du système peuvent être déconnectées ou mal connectées.

Exemple2 : lorsque les ordinateurs mobiles sont inclus dans un système. Certaines applications ont des exigences particulières pour une large bande passante de communication et une faible latence (les applications multimédia...)

60

5. Modèles de SDs (1)

• la conception des systèmes distribués peut être décrite selon trois façons importantes et complémentaires :

- **Modèles physiques**
- **Modèles architecturaux**
- **Modèles fondamentaux**

61

5. Modèles de SDs (2): Les modèles physiques

Les modèles physiques considèrent les nœuds informatiques (ordinateurs et autres dispositifs) qui constituent le SD et leur interconnectivité (réseaux), sans détails de technologies spécifiques. Ces nœuds communiquent et ne coordonnent leurs actions qu'en transmettant des messages. C'est en fait le modèle minimum du SD.

• Au-delà de ce modèle de référence, nous pouvons identifier de façon utile trois générations de SDs

62

5. Modèles de SDs (2): Les modèles physiques

Premiers systèmes répartis : 1970 -1980: réponse à l'émergence de la technologie des réseaux locaux. Ces systèmes comprennent généralement entre 10 et 100 nœuds interconnectés par un réseau local, avec une connectivité Internet limitée et une petite gamme de services tels que les imprimantes locales partagées, les serveurs de fichiers, le courrier électronique et le transfert de fichiers sur Internet.

63

5. Modèles de SDs (2): Les modèles physiques

Systèmes distribués à grande échelle (Internet) à partir de 1990 l'infrastructure physique est un ensemble extensible de nœuds interconnectés par un réseau de réseaux (Internet). Ces systèmes exploitent l'infrastructure offerte par Internet pour devenir vraiment global.

(Hétérogénéité en termes de réseaux, d'architecture informatique, des systèmes d'exploitation, des langues employées et des équipes de développement impliquées.)

-->Cela a mis l'accent sur les normes ouvertes et les technologies de middleware associées (CORBA, services Web.)

64

5. Modèles de SDs (2): Les modèles physiques

Les systèmes distribués contemporains: noeuds non statiques (gardant un seul emplacement physique pour des périodes prolongées), ni discrets (forcément indépendants des autres entités physiques) grâce à :

-L'émergence de **l'informatique mobile** a conduit à des modèles physiques où les noeuds tels que les ordinateurs portables ou les smartphones peuvent se déplacer dans le système.

-L'émergence de **l'informatique omniprésente (ubiquitous)**

-L'émergence **du cloud computing**

Les ultralarge-scale (ULS) distributed systems : systèmes distribués de systèmes

65

5. Modèles de SDs (2): Les modèles physiques

Distributed systems:	Early	Internet-scale	Contemporary
<i>Scale</i>	Small	Large	Ultra-large
<i>Heterogeneity</i>	Limited (typically relatively homogenous configurations)	Significant in terms of platforms, languages and middleware	Added dimensions introduced including radically different styles of architecture
<i>Openness</i>	Not a priority	Significant priority with range of standards introduced	Major research challenge with existing standards not yet able to embrace complex systems
<i>Quality of service</i>	In its infancy	Significant priority with range of services introduced	Major research challenge with existing services not yet able to embrace complex systems

66

5. Modèles de SDs(3): Les modèles architecturaux

- **Les modèles architecturaux** décrivent un système en termes de calcul et tâches de communication réalisées par ses éléments. Les éléments étant des ordinateurs individuels soutenus par des interconnexions réseau.

- **Client-serveur** et **peer-to-peer** sont les deux formes les plus communes des modèle architecturaux pour les systèmes distribués.

Autres approches : les **objets distribués**, les **composants distribués**, ..

67

5. Modèles de SDs(3): Les modèles architecturaux

Entités communicantes:

Nœuds | processus | Objets | composants | services

Paradigmes de communication

Communication interprocessus | invocation à distance |

Communication indirecte.

68

5. Modèles de SDs(3): Les modèles architecturaux

Communication Interprocessus : communication relativement bas niveau de SDs (primitives de transmission de messages, programmation par socket...

Invocation à distance (le + courant dans les SDs) : techniques basées sur l' échange dans les deux sens entre entités communicantes (en modèle client-serveur) et conduit à l'appel d'une opération, d'une procédure ou d'un procédé distant: **Protocoles Reply response, RPC, RMI**,.

69

5. Modèles de SDs(3): Les modèles architecturaux

La **communication indirecte**, se fait à travers une troisième entité, permettant un fort degré de découplage entre les expéditeurs et les destinataires.

- Les expéditeurs n'ont pas besoin de savoir à qui ils envoient (découplage spatial).
- Les expéditeurs et destinataires n'ont pas besoin d'exister en même temps (découplage temporel).

Exemples : Group Communication| publish-subscribe systems | message queues | tuple spaces| distributed shared memories

70

Chapitre 1. Caractérisation des SDs		5. Modèles de SDs(3): Les modèles architecturaux		
Communicating entities (what is communicating)		Communication paradigms (how they communicate)		
System-oriented entities	Problem- oriented entities	Interprocess communication	Remote invocation	Indirect communication
Nodes	Objects	Message passing	Request- reply	Group communication
Processes	Components	Sockets	RPC	Publish-subscribe
	Web services	Multicast	RMI	Message queues
				Tuple spaces
				DSM

71

Chapitre 1.
Caractérisation
des SDs

5.Modèles de SDs(4):
Les modèles fondamentaux

Les modèles fondamentaux adoptent une perspective abstraite afin de décrire les solutions aux problèmes individuels des éléments appartenant aux systèmes distribués.

72

Chapitre 1.
Caractérisation
des SDs

5.Modèles de SDs(4):
Les modèles fondamentaux

-Les modèles d'interaction s'intéressent à la structure et au séquençage de la communication entre les éléments du système;

-Les modèles d'échec qui considèrent les manières dont un système peut échouer ou plutôt comment assurer qu'il puisse fonctionner correctement

-Les modèles de sécurité qui considèrent comment le système est protégé contre les tentatives pour contourner son fonctionnement correct ou de voler ses données

73

Chapitre 1.
Caractérisation
des SDs

5.Modèles de SDs(4):
Les modèles fondamentaux

Exemple : **modèle d'échec**:

3 grands types de fautes ou pannes

- Franches : le processus ne fait plus rien
- Par omission : des messages sont perdus ou non délivrés
- Arbitraires ou byzantines : le processus renvoie des valeurs fausses et/ou fait « n'importe quoi »: Cas de fautes les plus complexes à gérer. Les autres fautes peuvent être considérées comme des cas particuliers des fautes byzantines

74