

# Structures de données avancées :

*SDDS (structures de données distribuées et  
scalables)*

Pr ZEGOUR DJAMEL EDDINE  
Ecole Supérieure d'Informatique (ESI)  
[www.zegour.uniq.com](http://www.zegour.uniq.com)  
email: [d\\_zegour@esi.dz](mailto:d_zegour@esi.dz)

## Sdds

### *Systèmes distribués*

- Un ensemble de machines se partageant les données à stocker et la charge CPU.
- Les tâches de communication inter-machines sont transparentes à l'utilisateur,
- L'ensemble du système apparaît comme une seule machine virtuelle.

## Sdds

### *Architecture à mémoire partagée : Shared-Memory*

- Processeurs indépendant avec une mémoire commune
- Limité à une dizaine de processeurs ( La mémoire devient un goulot d'étranglement)
- L'ajout d'un nouveau processeur peut entraîner des modifications importantes

## Sdds

### *Architecture à disques partagés : Shared-Disk*

- Chaque processeur a sa mémoire privée et peut accéder à n'importe quel disque à travers un réseau d'interconnexion
- Chaque processeur peut ainsi copier des pages d'une base de données sur un disque partagé dans son propre cache disque.
- Nécessite une gestion des conflits d'accès aux même pages et l'implantation d'un protocole de gestion de cohérence des caches.
- La complexité de ces mécanismes limite les performances de cette architecture.

## Sdds

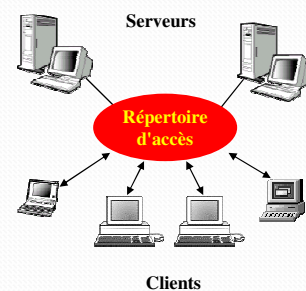
### *Architecture sans partage : Shared-Nothing*

- Chaque processeur dispose de sa propre mémoire locale et de son propre espace disque ( Chaque nœud apparaît comme un serveur de données local avec sa propre base de données)
- Cette architecture s'est développée grâce aux réseaux de PCs et de stations de travail qui permettent d'intégrer un grand nombre de processeurs (plusieurs centaines, voire des milliers de processeurs).
- Extensibilité et disponibilité très séduisantes.
- Minimiser la quantité de données transférées sur le réseau.
- Ces systèmes sont cependant difficiles à administrer et à programmer.

## Sdds

### *Structures de données classiques*

- Stockage sur un seul site  
→ Limite de taille
- Mécanisme de calcul d'adresse unique et centralisé  
→ un point d'accumulation
  - Limite sur les performances d'accès
  - Vulnérabilité aux pannes
  - Pas de Scalabilité



## Sdds

### *Structures de données classiques*

- Schémas de distribution classiques :
  - Partitionnement par intervalle
  - Partitionnement par hachage
  - Round-Robin
  - Etc...

## Sdds

### *SDDS : Scalable distributed data structures*

- Une nouvelle classe de structures de données pour des bases de données modernes
- Conçues spécifiquement pour les multi ordinateurs d'une grande taille (Données multimédia, géographiques, texte,... )
- Assure la Scalabilité (Grandissant rapidement en gardant les mêmes performances )
- Assure une haute disponibilité

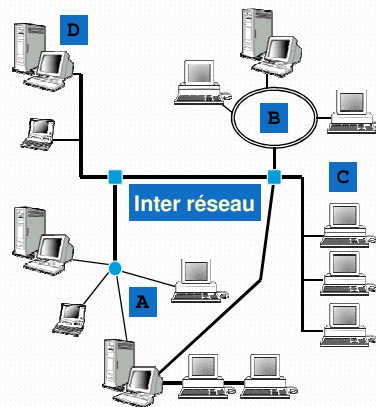
# Sdds

## *Multi ordinateurs*

- Une collection faiblement couplés d'ordinateurs :  
Station de travail, PC, CPU
- Interconnectés par un réseau haut-débit ( $\geq 100 \text{ Mb/s}$ ) :  
LAN (Local Area Network), WAN(Wide Area Network), ...
- Architecture à partage de rien avec passage de messages :  
Unicast, Multicast et Broadcast

# Sdds

## *Multi ordinateurs*



## Sdds

### *Multi ordinateurs / Avantages*

- Bon marchés ( les ordinateurs existent déjà )
- Puissance théorique de ressources en calcul et mémoires
- Impossible pour un super-ordinateur traditionnel :
  - HP laboratories (1500 WS) = 50 GO de RAM et quelques TO ( Téra octets) de disque. (1 TO = 1020 octets )
  - UC Berkeley Soda Hall (500 WS) = 64 GO de RAM et plus d'un TO de disque
- Possibilités de bases de données en RAM distribuée :

## Sdds

### *Multi ordinateurs / Avantages*

	RAM local	RAM distant <i>réseau gigabit</i>	RAM distant <i>Ethernet</i>	Disque local
Temps d'accès	100 nsecs	1 µsecs	100 µsecs	10 msec
En proportion	1 mn	10 mn	2 heures	8 jours

## Sdds

### *SDDS : Axiomes généraux*

- Les données sont stockées sur des serveurs et accédées par des clients autonomes
- Il n'y a pas de répertoire central d'accès
  - Image locale par client
  - Mécanisme de vérification et de redirection des requêtes
  - Mécanisme de mise à jour distribué des images des clients (IAM : Image Adjustment Message)
- Mises à jour asynchrones

## Sdds

### *SDDS : Contraintes*

- Contraintes de performance d'accès d'une SDDS
  - Nombre de messages sur le réseau par opération (indépendante des paramètres du réseau)
  - Convergence de vue d'un nouveau client et d'un client peu actif
- Contraintes de scalabilité
  - Prendre en charge n'importe quelle quantité de données
  - Maintenir les performances quand le volume de données stockées varie

## Sdds

### *SDDS : Contraintes*

- Contraintes de distribution
  - Une grande quantité de données
  - Traitement parallèle et distribué
- Contraintes de disponibilité
  - Assurer la continuité du fonctionnement 24 heures / 7 jours

## Sdds

### *SDDS : Typologie*

- Hachage : LH\*, DDH et EH\*
- Ordonnées : RP\*, CTH\*, DRT
- Multidimensionnelles : k-RP\*, k-DRT et IH\*
- Haute-disponibilité : LH\*<sub>m</sub>, LH\*<sub>s</sub>, LH\*<sub>g</sub>, LH\*<sub>RS</sub>, LH\*<sub>SA</sub>