

CentraleSupélec

Big Data : Informatique pour les données et calculs massifs

2 – Terminologie des architectures matérielles et logicielles distribuées

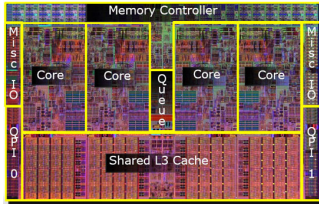
Stéphane Vialle

universit  PARIS-SACLAY Sciences et technologies de l'information et de la communication (STIC) RISEGrid GrandEst

Stephane.Vialle@centralesupelec.fr
<http://www.metz.supelec.fr/~vialle>

CentraleSupélec

1 – Composants mat riels



Memory Controller

Core Core Queue Core Core

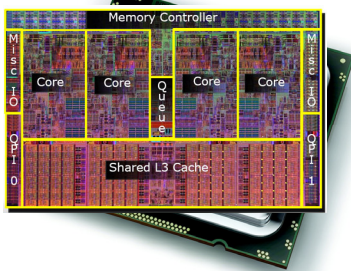
Shared L3 Cache

Computing cores

2

CentraleSupélec

1 – Composants mat riels



Memory Controller

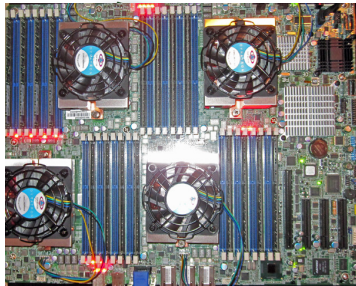
Core Core Queue Core Core

Shared L3 Cache

Computing cores
Multi-core processor

3

1 – Composants matériels

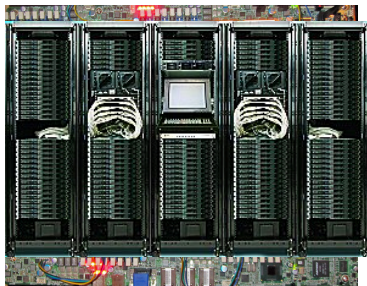


- Computing cores
- Multi-core processor
- Multi-core PC/node

SMP → NUMA

4

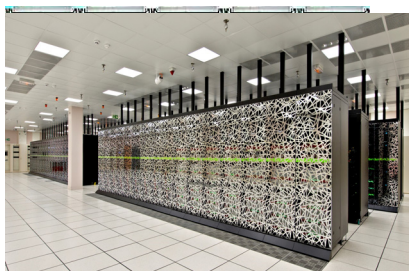
1 – Composants matériels



- Computing cores
- Multi-core processor
- Multi-core PC/node
- Multi-core PC cluster

5

1 – Composants matériels



- Computing cores
- Multi-core processor
- Multi-core PC/node
- Multi-core PC cluster
- Super-Computer

6

1 – Composants matériels

Fat Tree interconnect

- Computing cores
- Multi-core processor
- Multi-core PC/node
- Multi-core PC cluster
- Super-Computer
- + hardware accelerators

1 – Composants matériels

Calcul / Stockage
Approche traditionnelle

Baie de stockage **Baie de calcul**

Réseau rapide pour toutes les E/S

Haute capacité
Haute fiabilité

Pas/peu de disques

1 – Composants matériels

Calcul / Stockage
Approche Cluster de PC

Baie de stockage **Baie de calcul**

Réseau rapide pour l'archivage

Haute capacité
Haute fiabilité

PC std
Disques locaux std

1 – Composants matériels

Calcul / Stockage
Approche Big Data (Hadoop)

Grande baie de stockage-calcul

PC std
GROS disques locaux std

2 – Composants logiciels

Espace mémoire du processus (et de ses threads)

code

stack

thread x

thread y

thread z

code

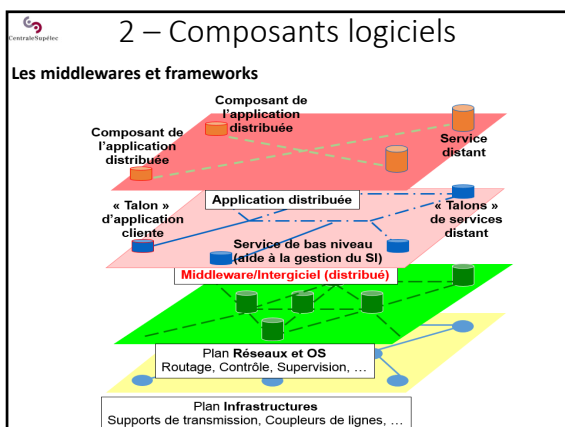
Pile et code du processus : le « thread principal » du processus

- **Processus** : un programme en cours d'exécution
- **Threads** : un sous-programme actif
- **Tâches** : une entité active (thread ou processus)

2 – Composants logiciels

Les outils et protocoles de synchronisation de tâches

- Outils de bas niveau
verrous, sémaphores, variables conditionnelles, barrières de synchronisation, moniteurs...
- Protocoles classiques
mutex !!, producteurs-consommateurs, lecteurs-rédacteurs...
la « mutex » est la pire solution : elle re-séquentialise les tâches, elle « tue la performance » !
- Modèles de parallélisation
pgms *embarrassingly/happily* parallel
SPMD/MPMD...
...Map-Reduce



2 – Composants matériels/logiciels

Cluster vs clouds

- Cher
- Performant
- Confidentialité supposée sans pb...

Google, Amazon, Microsoft

- Mutualisés
- Bon marché
- Moyennement performant
- Cloud HPC !
 - Une tranche de cluster HPC
 - Plus cher....

3 – Problématique du déploiement

Architecture logicielle distribuée

Architecture matérielle distribuée

- Recherche de performance
Où mettre quoi ?
- Mise en œuvre de la tolérance aux pannes
Stratégie ? Mécanismes ?
- Capacité de passage à l'échelle ?
Déploiement extensible ?
- Maîtrise des coûts d'exécution
Choix du matériel ?


Terminologie des architectures
matérielles et logicielles distribuées