




 CentraleSupélec  


Big Data : Informatique pour les données et calculs massifs

4 – Schémas élémentaires de parallélisation

Stéphane Vialle

 université PARIS-SACLAY ÉCOLE DOCTORALE Sciences et technologies de l'information et de la communication (STIC)   ALSACE CHAMPAGNE-ARDENNE LORRAINE

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

 CentraleSupélec

1 - Outils de synchronisation

- *Verrous*
- *Sémaphores*
- *Variables conditionnelles*

} Autre cours

- **Barrières** sur fin de tâches
- **Barrières** génériques

CentraleSupélec

1 - Outils de synchronisation

Barrières sur fin de tâches

- Chaque tâche arrivant sur une barrière se bloque
- La dernière du groupe à se bloquer débloque toutes les tâches
- Concept de « groupe de tâches » associé à une barrière

Sub-tasks creation

Sub-Task Id 1 Sub-Task Id 2 Sub-Task Id 3

Barrier on the end of subtasks

Commandes de type « join(Task Id) »

- Dans n'importe quel ordre
- Ou bien un « join([task Id]) »

CentraleSupélec

1 - Outils de synchronisation

Barrières génériques

Ex : Calculs itératifs & Synchronisation à chaque itération

```

1 // Main task code
2 Data_t InTab1[N];
3 Data_t InOutTab2[N];
4 Data_t OutTab3[N];
5 ... // - Create and launch SubTasks

```

```

1 // SubTask code
2 int Me = .... // Ranking each task
3 ....
4 // - read InTab1, write into InOutTab2
5 f(InTab1, InOutTab2, N/3*Me, N/3*(Me+1))
6 // - Wait all tasks have finished to write into InOutTab2
7 Barrier(TaskGroupId)
8 // - read InOutTab2, write into OutTab3
9 g(InOutTab2, OutTab3, N/3*Me, N/3*(Me+1))
10 ....

```

Barrière : pour être sûr que le tableau InOutTab2[] a été entièrement généré avant d'être exploité

1 - Outils de synchronisation

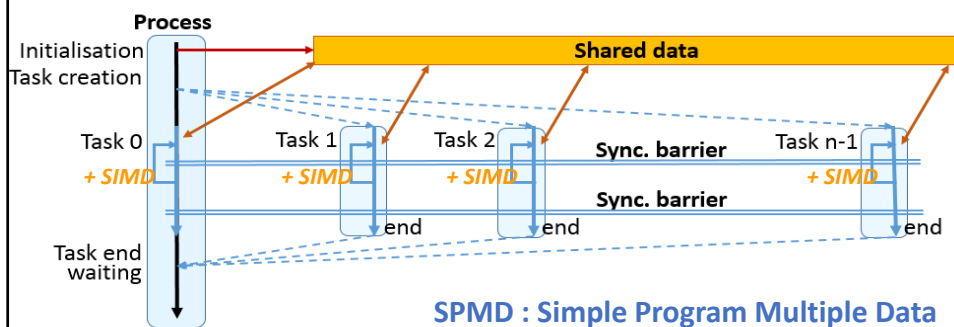
Mise en œuvre de barrières génériques

- Assez simple et rapide en mémoire partagée (dans un PC)
→ sémaphores, PAS d'attente active...
- Plus complexe et plus couteux en mémoire distribué (clusters)
→ échanges de messages entre PC

Traverser une barrière de synchronisation à toujours un coût (même si les tâches terminent en même temps)

→ En mettre le moins possible !

2 – Schéma SPMD (HPC)



- 1 seul programme répété dans toutes les tâches
- Toutes les tâches s'exécutent en parallèle mais sur des données différentes
- Synchronisation seulement sur des barrières
- Divergences possibles entre les tâches, avec respect des barrières

*Rmq : + code SIMD dans chaque tâche (HPC)
(SPMD + Single Instruction Multiple Data ...)*

CentraleSupélec

2 – Schéma SPMD (HPC)

SPMD (Simple Program Multiple Data) Embarrassingly Parallel

- 1 seul programme répété dans toutes les tâches
- Toutes les tâches s'exécutent en parallèle mais sur des données différentes

CentraleSupélec

3 – Schéma Map-Reduce (Big Data)

Résumé du fonctionnement

← Data collection and filtering
← Data redistribution (grouped by identical key data)
← Data gathering (into data lists) & computation of group features

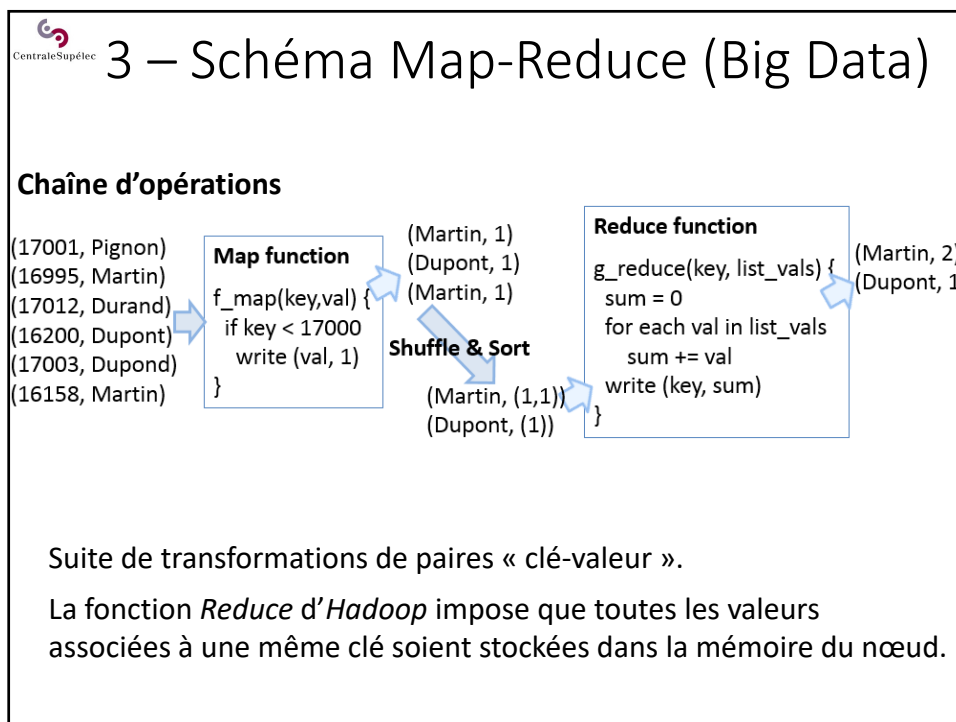
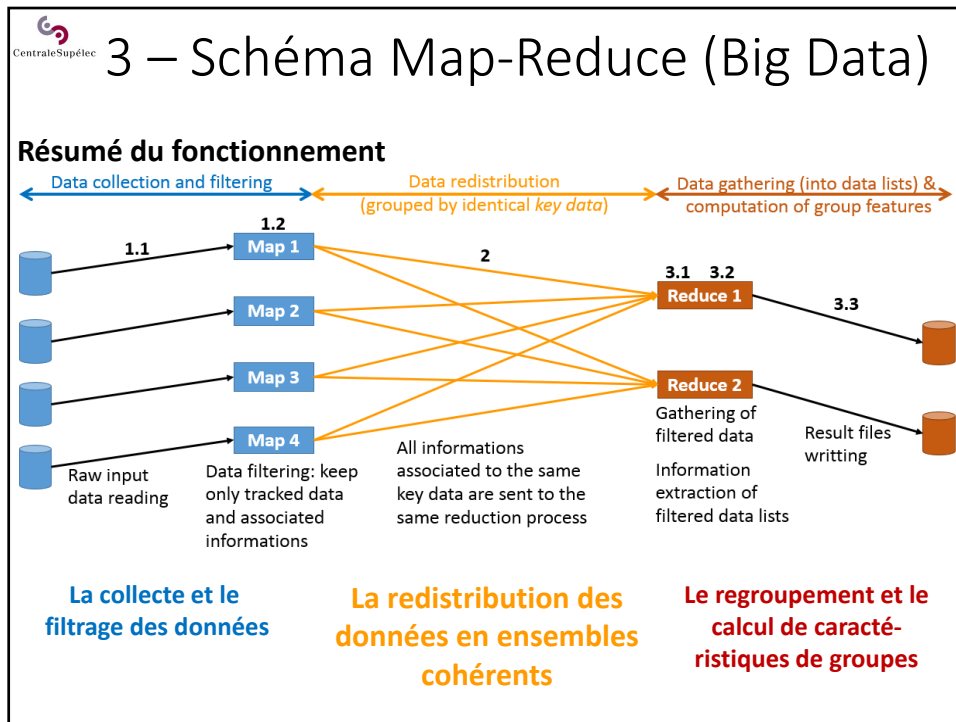
1.1 Raw input data reading
 1.2 Map 1
 Map 2
 Map 3
 Map 4
 Data filtering: keep only tracked data and associated informations

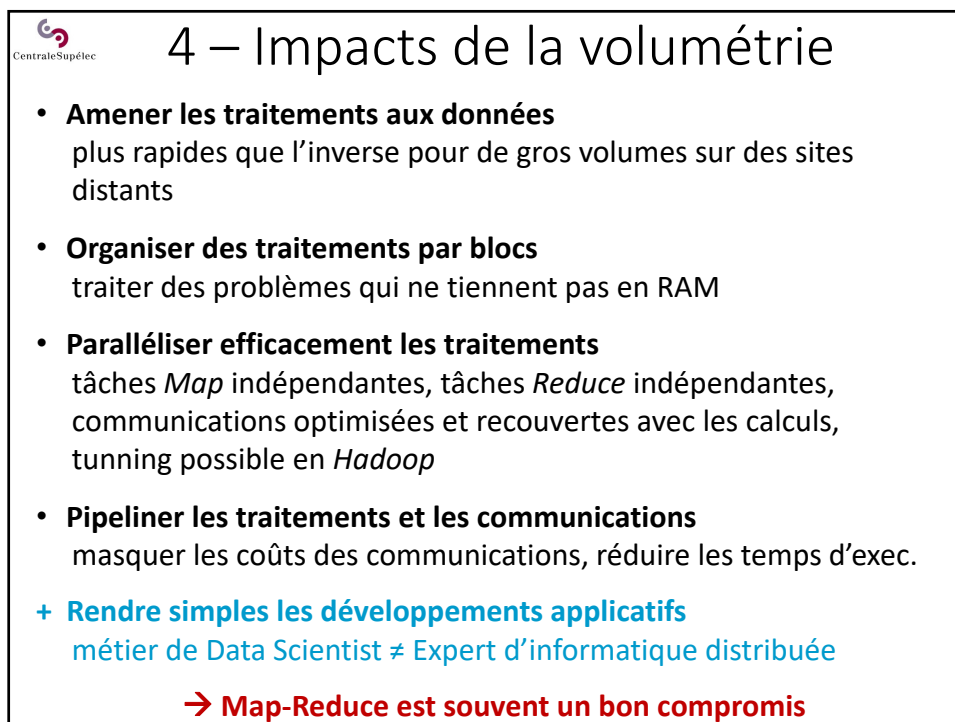
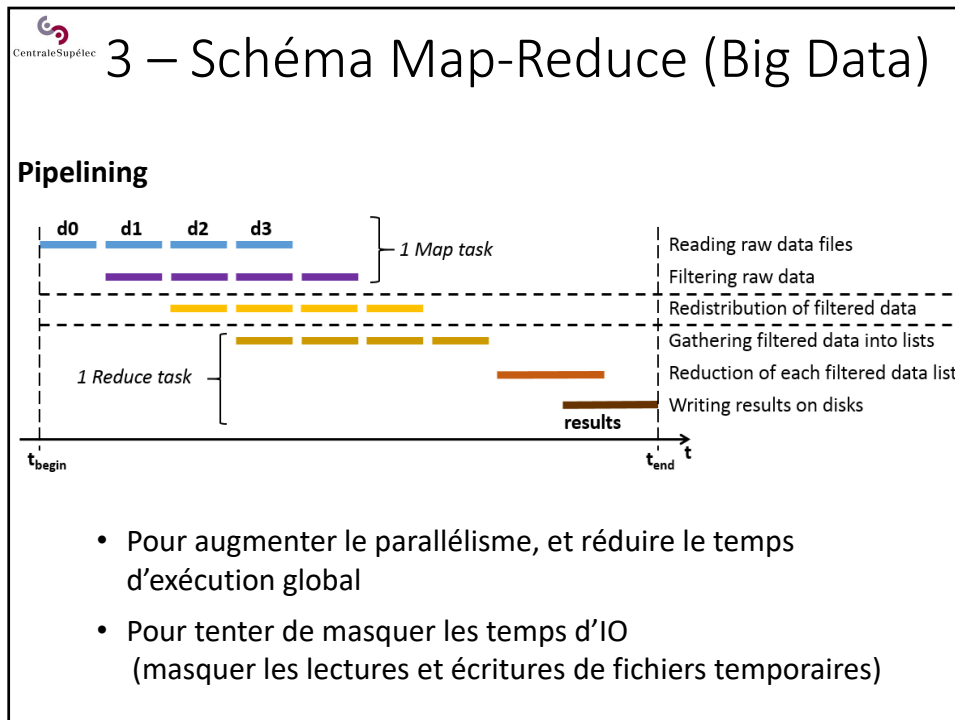
2 All informations associated to the same key data are sent to the same reduction process

3.1 3.2 Reduce 1
 Reduce 2
 Gathering of filtered data
 Information extraction of filtered data lists

3.3 Result files writing

Map → Shuffle & Sort → Reduce
À base de paires (clé – valeur(s))





Schémas élémentaires de parallélisation

