



--Chapitre 2 : Traitement des données massives avec Hadoop--

Asma KERKENI asma.kerkeni@gmail.com

Objectifs

2

- Au terme de ce chapitre, vous serez capable de:
 - Expliquer le principe de localité de données
 - Décrire la vue d'ensemble et architecture(s) d'Hadoop
 - Décrire le fonctionnement de HDFS
 - Ecrire des algorithmes Map-Reduce
 - Comprendre la gestion de ressource avec YARN

Motivation

3

Too much data



Not enough compute power, storage or infrastructure



TeenClips.com

#7013

service@teenclips.com

BigData

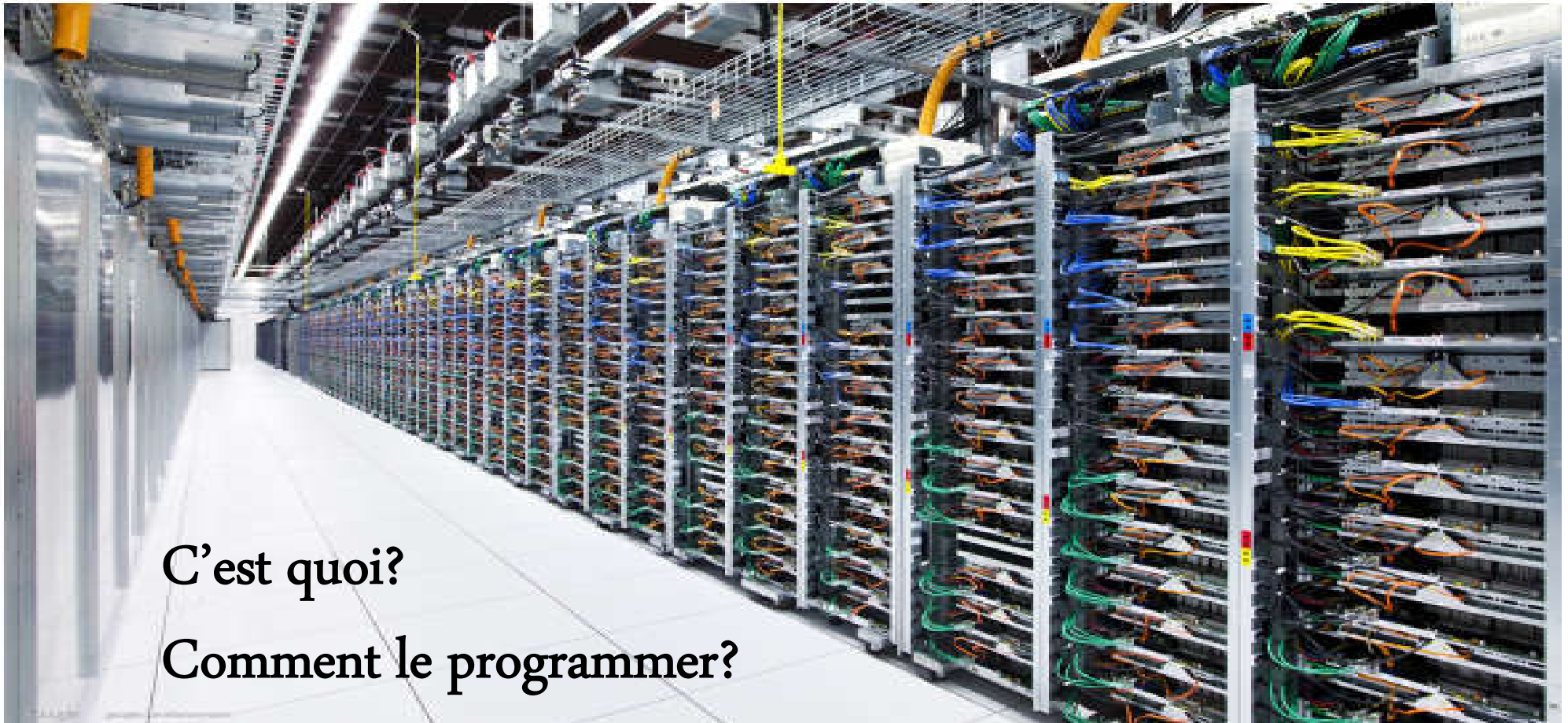
Plan

4

- Contexte d'apparition
- Présentation d'Hadoop
- Ecosystème d'Hadoop
- Système de fichiers distribué d'Hadoop : HDFS
- Hadoop Map-Reduce
- Allocation et gestion de ressources avec Yarn
- API JAVA d'Hadoop

5

Contexte



C'est quoi?

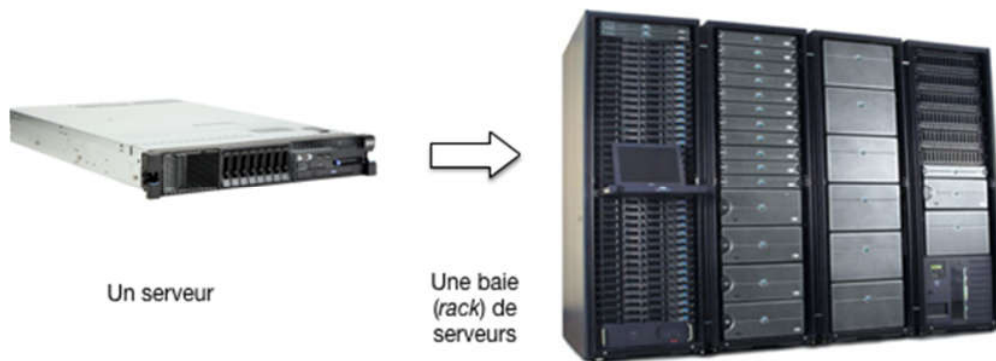
Comment le programmer?

Distribution des données et traitements

- Centre de données -

6

- Dans un centre de données, les serveurs sont empilés dans des **baies (Rack)** équipées pour leur fournir l'alimentation, la connexion réseau vers la grappe de serveur, la ventilation.
- Une baie contient environ 40 serveurs. **Un centre de données** contient quelques centaines de baies. Ordre de grandeur : **quelques milliers de serveurs par centre**.
- Combien des centres chez Google, Facebook, Amazon ... ? Des millions de serveurs.



BigData

Data Center

Distribution des données et traitements

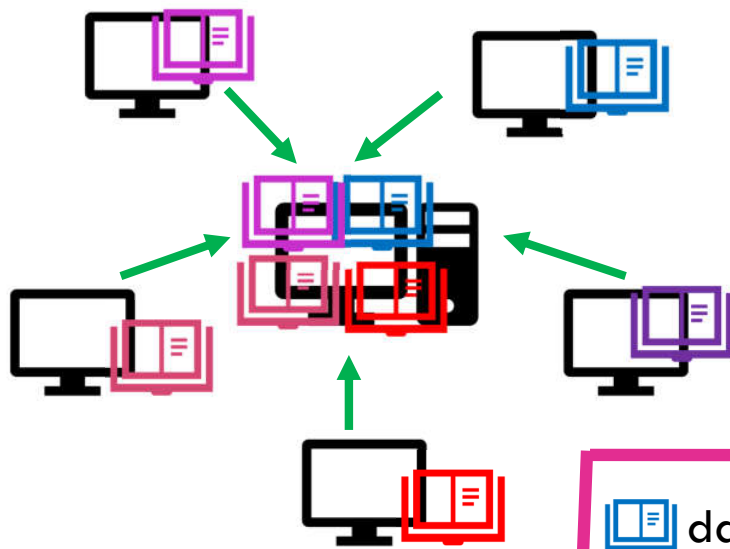
- Principe du *Data Locality* -

7

- C'est toujours l'accès aux données qui coute cher, pas le calcul lui-même une fois les données dans le cœur de calcul.
- Principe de *data locality* ou proximité des données.

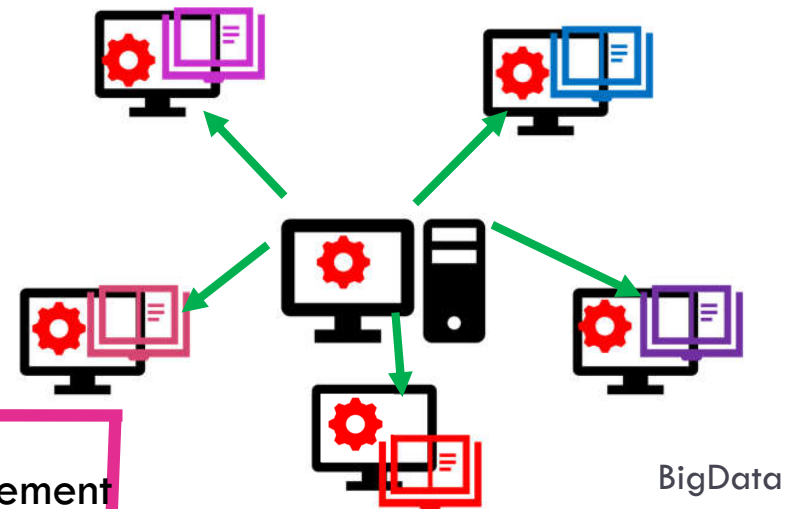
Classique:

Ramener les données au serveur pour les traiter



Big Data:

Amener les codes de traitements aux données
data locality



Distribution des données et traitements

-Besoins du Big Data-

8

- **Besoins:** Frameworks dédiés prenant en charge les enjeux du calcul distribué :
- **Optimisation des transferts disques et réseau** en limitant les déplacements de données
- **Tolérance aux pannes** (*fault tolerance*).
- **Scalabilité** pour permettre d'adapter la puissance au besoin (*scalability*)

Distribution des données et traitements

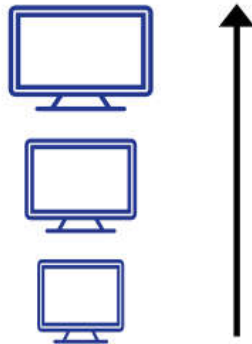
-Scalabilité-

9

□ **Scalabilité:** C'est la capacité d'un système à gérer des quantités croissantes de données, sans diminution significative de ses performances

VERTICAL SCALING

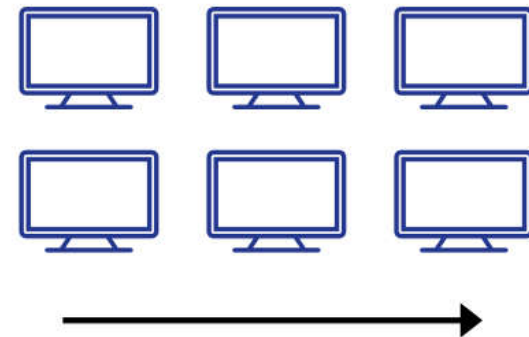
Increase size of instance
(RAM, CPU etc.)



- Plus facile de maintenir une seule machine.
- Control centralisé sur les données et les calculs.

HORIZONTAL SCALING

(Add more instances)



- Mise à niveau illimitée de la puissance de calcul d'un système
- Tolérance aux pannes