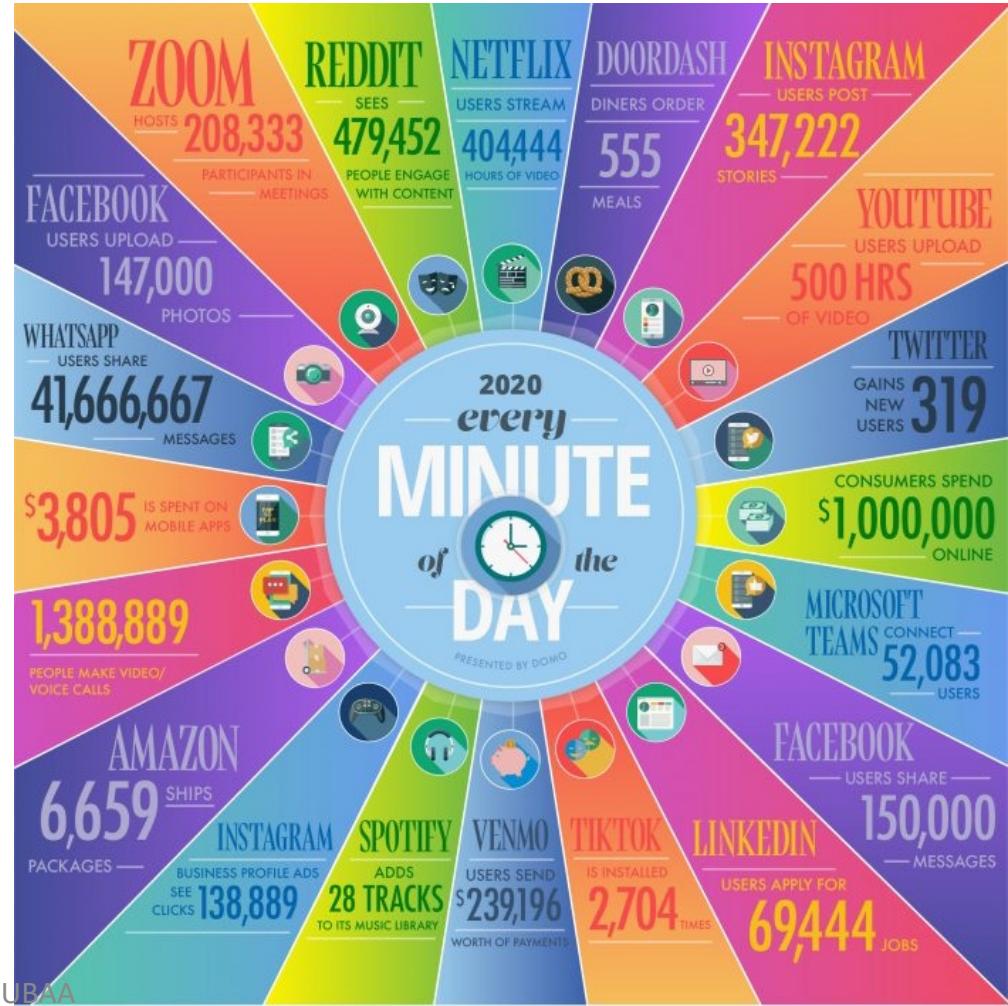


Big Data

Chapitre 1: Introduction générale

Contexte

- Quantité de données générées chaque minute
- Quelques chiffres étonnants...

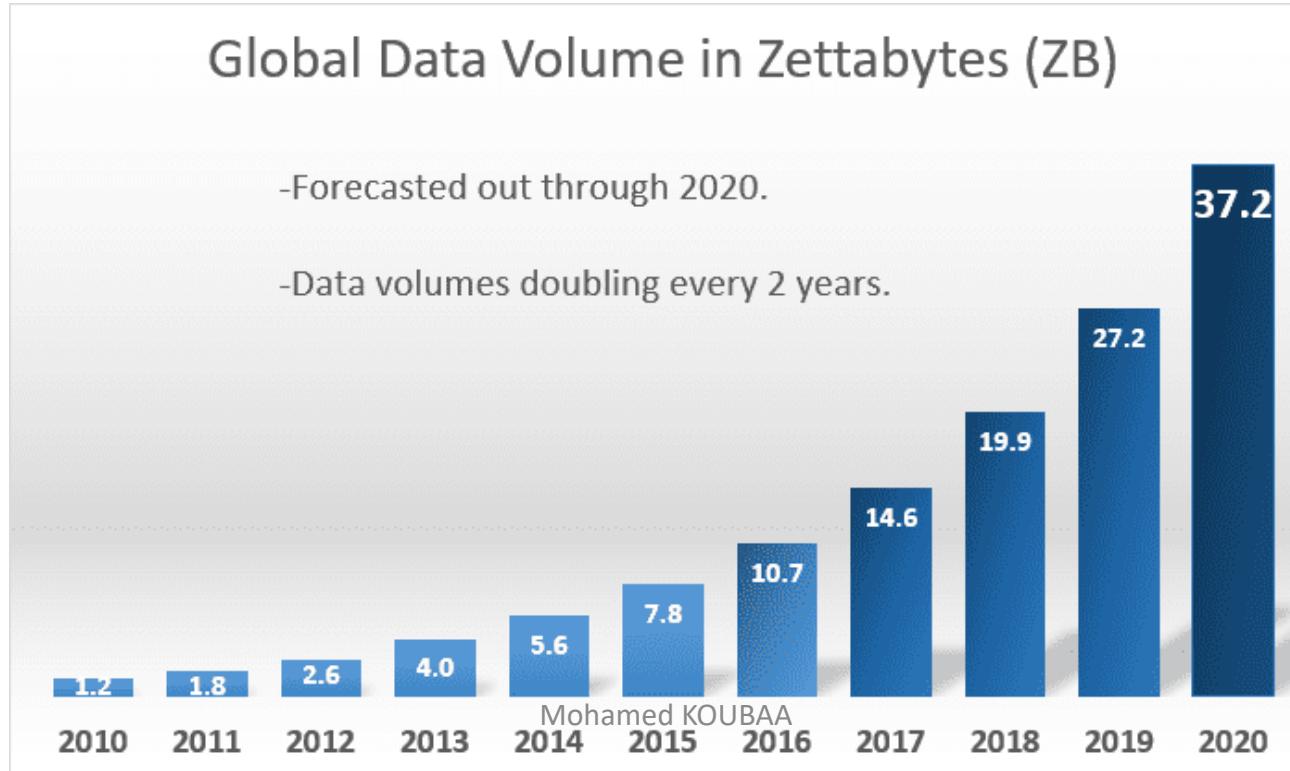


Big Data, pourquoi?

- D'où vient ce concept du “big data”?
- Est-ce seulement le “petit” data qui est devenu “big”?
 - Simplement plus de data?
- Quelques pistes:
 - **Explosion de la disponibilité des données**
 - **Augmentation de la capacité de stockage**
 - **Augmentation de la capacité d'analyse**

Disponibilité des données

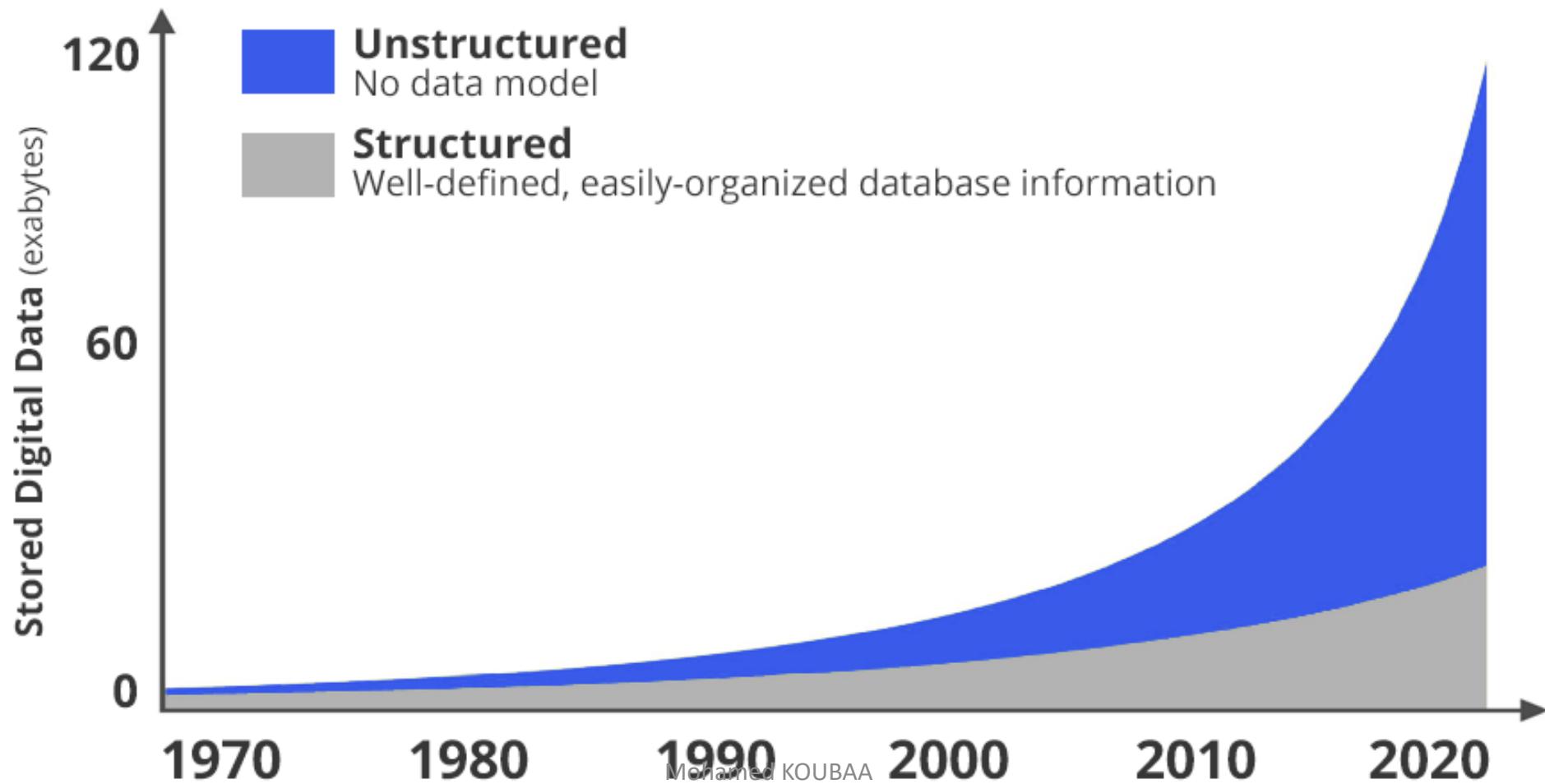
- La quantité de données digitales double tous les deux ans.
- En d'autres termes, on a produit autant de données digitales ces 2 dernières années que tout ce qui a été produit auparavant.



Disponibilité des données

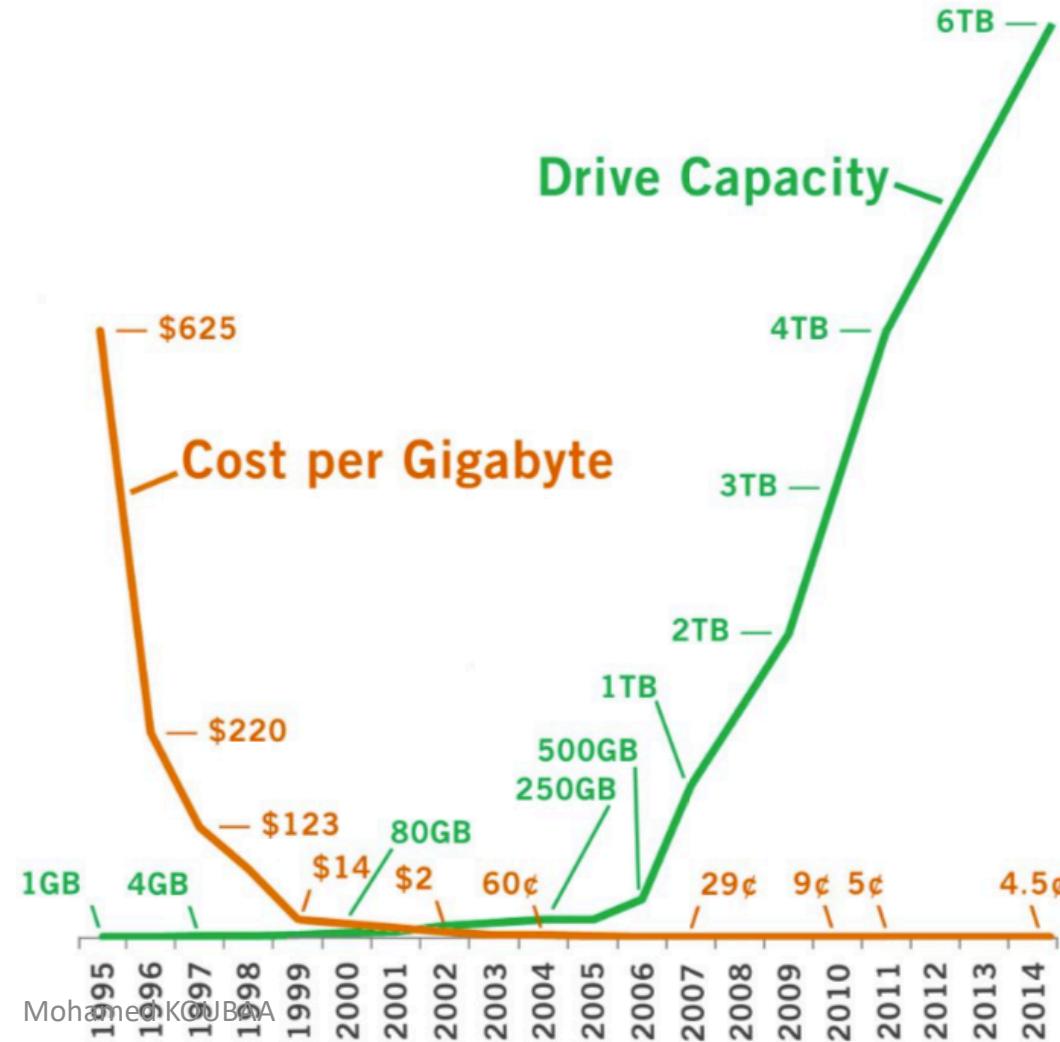
- 90% des données générées sont non structurées
- Source:
 - Capteurs utilisés pour collecter les informations climatiques
 - Messages sur les réseaux sociaux
 - Images numériques et vidéos publiées en ligne
 - Enregistrements transactionnels d'achat en ligne
 - Signaux GPS de téléphones mobiles
 - ...
- Données appellées Big Data ou données massives

Disponibilité des données



Capacité de stockage

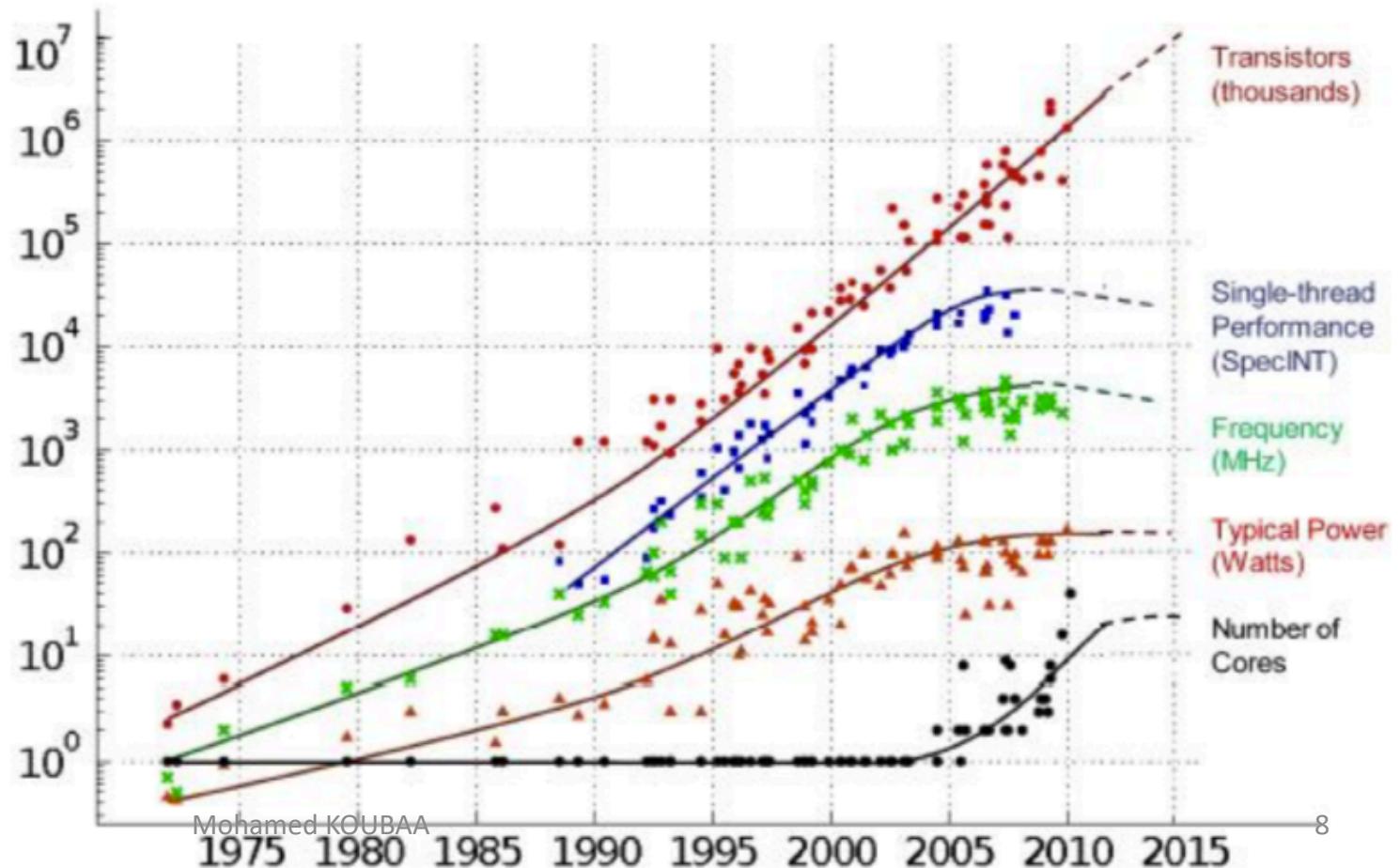
- Entre 2000 et 2006, la capacité des disques a augmenté par 10x alors que le prix par Gb a chuté du même ratio
- Une augmentation de 100x à prix constant



Capacité d'analyse

- La loi de moore en action depuis 35 ans
- Plus récemment, la capacité d'analyse augmente grâce à l'ajout de coeurs dans les unités centrales

35 Years of Microprocessor Trend Data



Big Data, pourquoi?

- Augmentation exponentielle de la quantité de données non structurées
 - Email, chat, blog, web, musique, photo, vidéo, etc.
- Augmentation de la capacité de stockage et d'analyse
 - L'utilisation de plusieurs machines en parallèle devient accessible
- Les technologies existantes ne sont pas conçues pour ingérer ces données
 - Base de données relationnelles (tabulaires), mainframes, tableurs (Excel), etc.
- De “nouvelles” technologies et techniques d’analyse sont nécessaires
 - “Google File System” - Google 2003
 - “MapReduce: Simplified Data Processing on Large Clusters” - Google, 2004
 - Hadoop: circa 2006
- D'où le “Big Data”: pas strictement plus de data...

Applications

- Santé
- Education
 - Mooc (Massive Open Online Course)
- Commerce de détail
 - Amazon, wallmart
- Biologie
 - Génomique
- Science, recherche fondamentale
- Machine learning, Deep Learning
- Recommendation
 - Netflix, Hopper
- Urbanisme
- Gouvernement
- Média
 - Journalisme de données
- Fraude (détection/ prévention)

Big Data

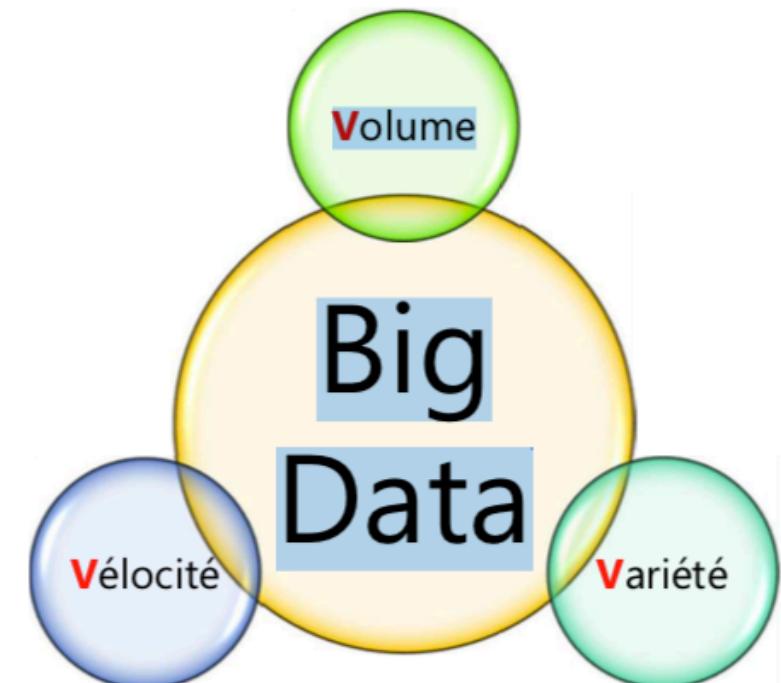
- Définition:

*“Le Big Data (ou méga données) représente les collections de données caractérisées par un **volume**, une **vélocité** et une **variété** si grands que leurs transformations en valeur utilisable requiert l’utilisation de technologies et de méthodes analytiques spécifiques”*

Big Data 3V

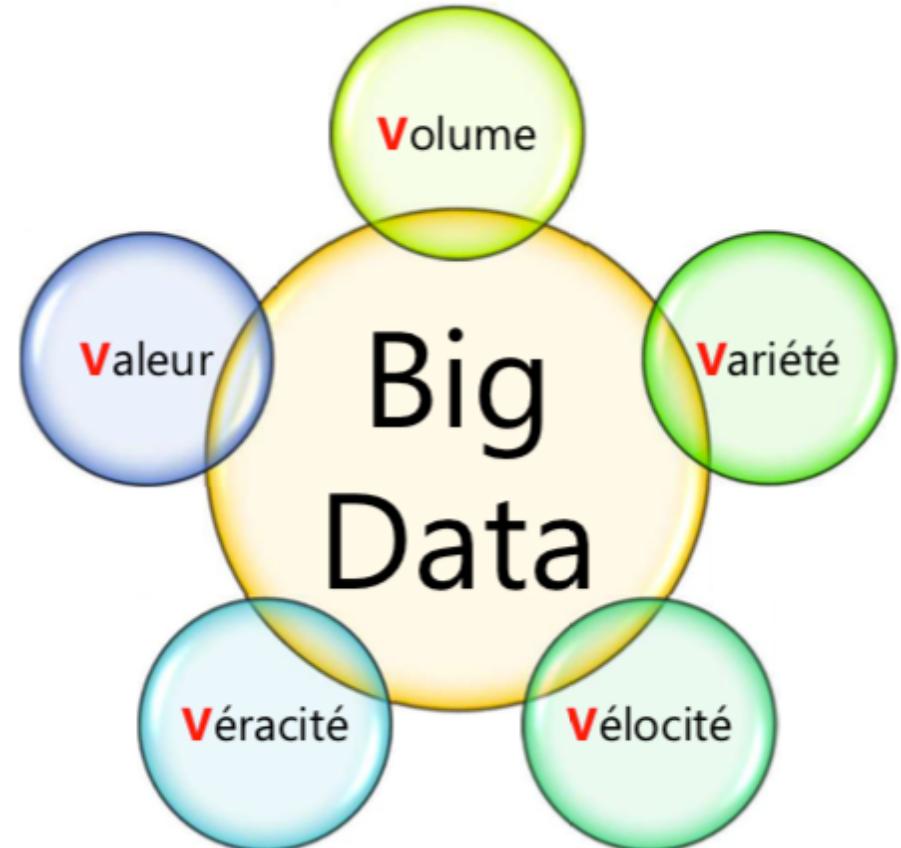
- **Volume :**
 - Quantité de données (L'ordre des PO)
- **Variété :**
 - Différents formats de données (Structurées: 20% ou non structurées : 80%)
 - Texte, CSV, XML, JSON, Binaires, BDR, etc ...
- **Vélocité :**
 - Fréquence de l'arrivée des données
 - Exemple : (Twiter)

Chaque seconde environ 5 900 tweets sont expédiés sur le site de micro-blogging Twitter. Cela représente 504 millions de tweets par jour ou 184 milliards par an.
Cette masse d'information vient alimenter le flot d'informations ("big data") publiée par l'humanité chaque jour sur internet.

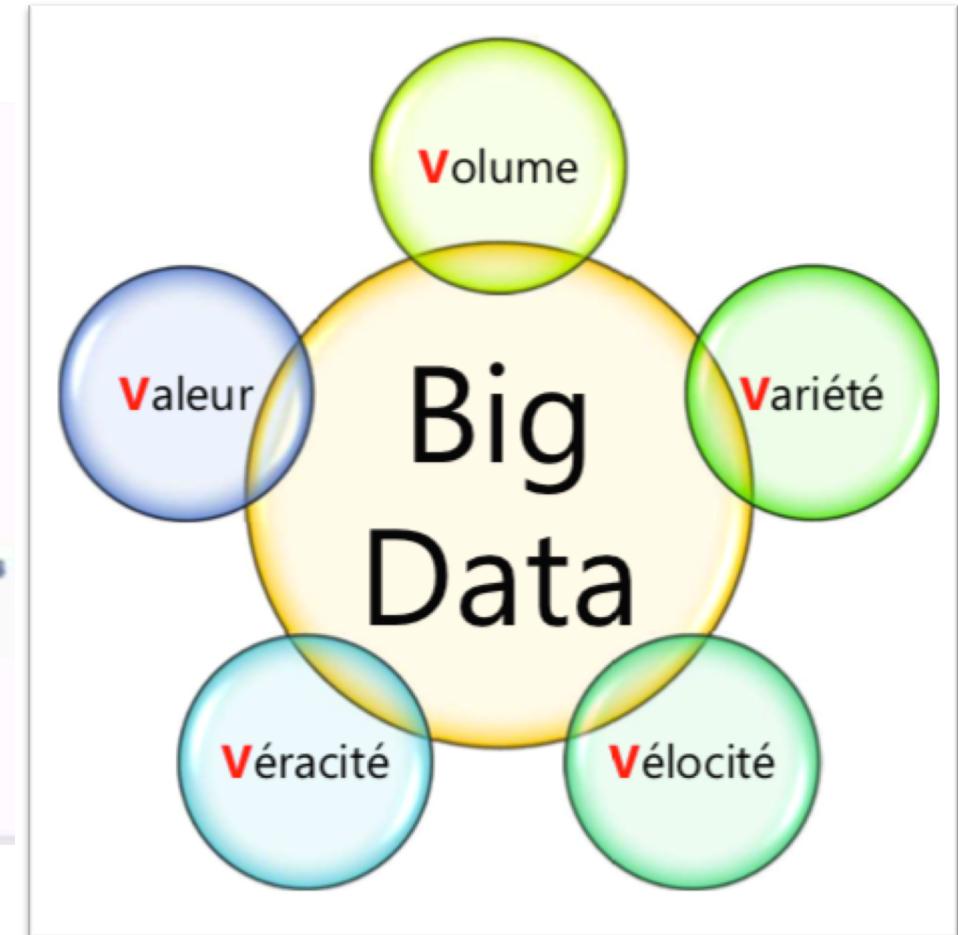
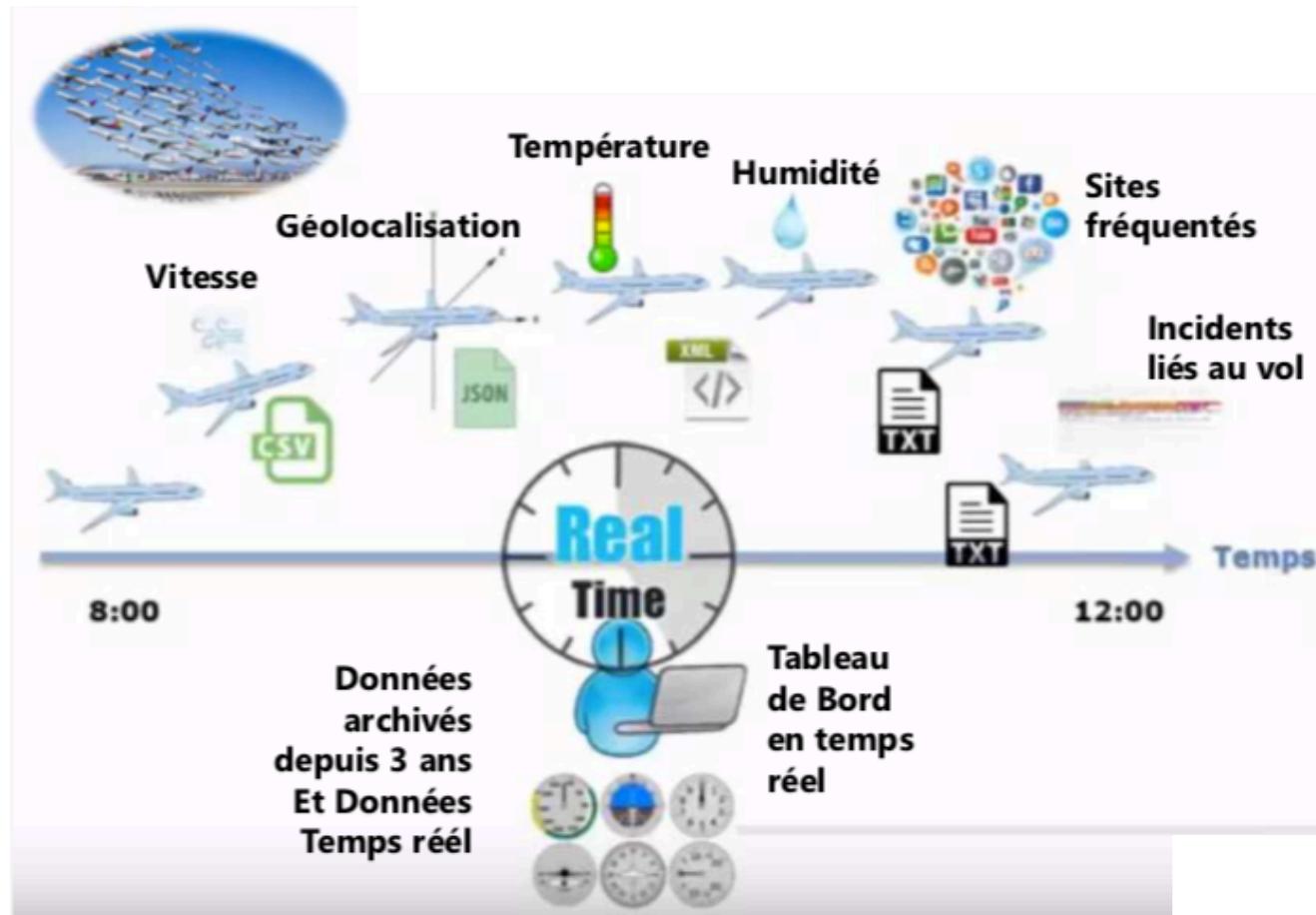


Big Data 5V

- **Volume**
- **Variété**
- **Vélocité** : Fréquence de l'arrivée des données
- **Véracité**
 - Fiabilité et la crédibilité des données collectées
(Sources Fiable)
- **Valeur**
 - Profit et connaissances que l'on peut extraire de ces données
 - Transformer les données en valeurs

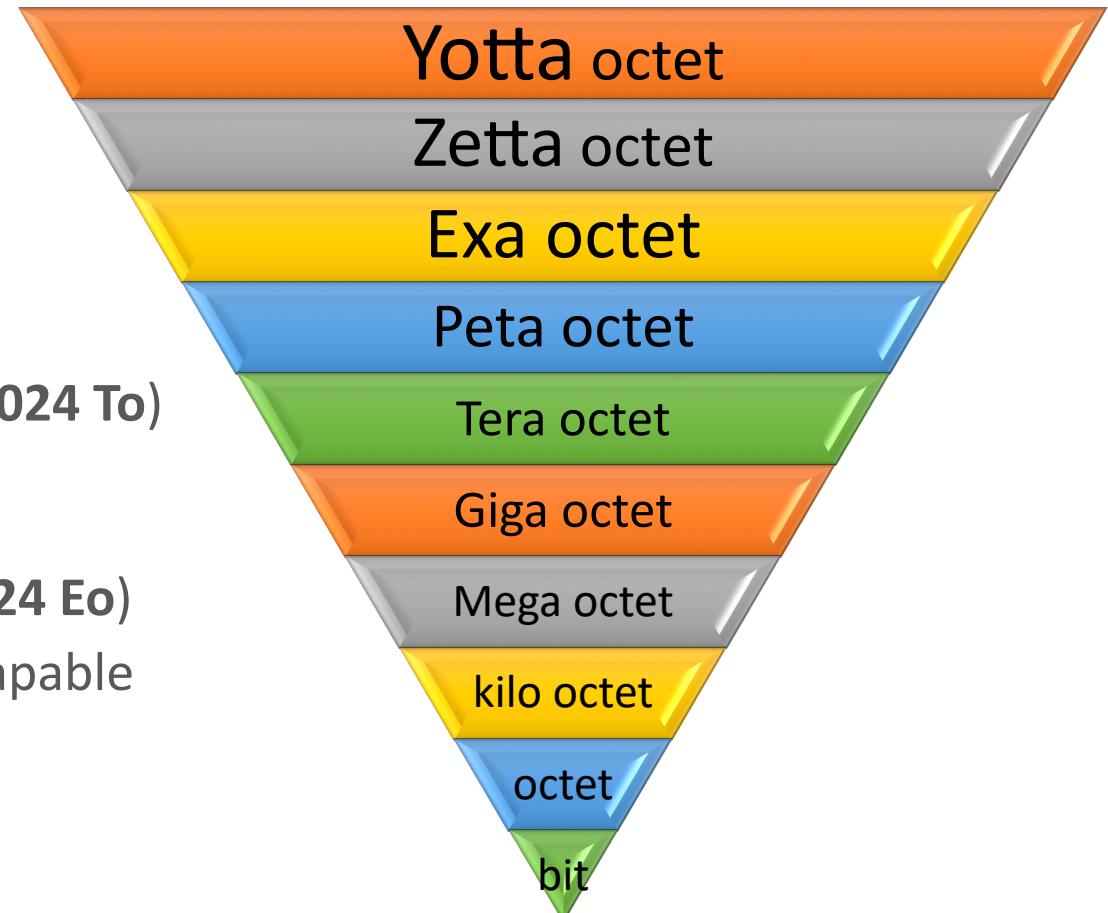


Exemple de problème: Compagnie aérienne



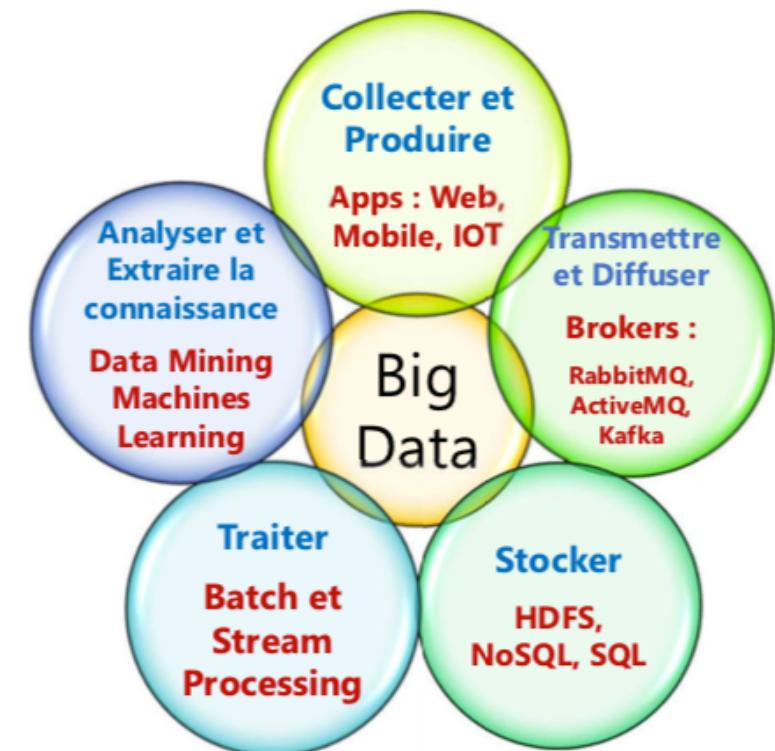
Ordre de mesure de stockage

- 1 caractère => 1 octet
- 1 Page de texte => 30 ko (**1ko=1024 o**)
- 1 Morceau de musique => 5Mo (**1Mo=1024 ko**)
- 1 Film de 2H => 1Go (**1Go=1024 Mo**)
- 6 Millions de Livres => 1 To (**1To=1024 Go**)
- 2 Milliard de Photos Rés Moyenne => 1 Po (**1 Po=1024 To**)
- Toutes les infos produites jusqu'à 2003 => 5 ExaO
(1Eo=1024 Po)
- Données produites en 2011 => 1,8 ZetaO (**1Zo=1024 Eo**)
- En 2013 le plus grand data center au monde est capable de traiter 1 YottaO (**1Yo=1024 Zo**)



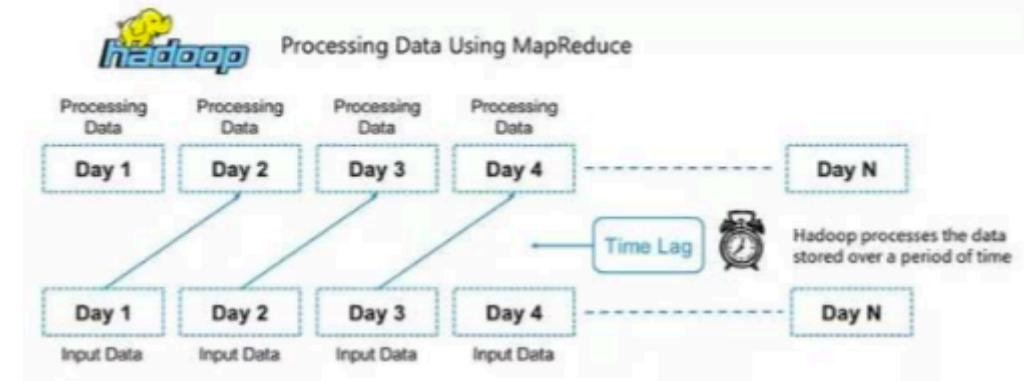
Besoins fonctionnels

- Explosion de la quantité de données générées par:
 - Les applications: réseaux sociaux, applications web mobiles, objets connectés, logs etc...
- Il est nécessaire de chercher les moyens qui permettent:
 - **Transmettre et diffuser les données entre les appli distribuées**
(Brokers : RabbitMQ, ActiveMQ, Kafka)
 - **Stocker et sécuriser les données d'une manière distribuée**
(Hadoop HDFS, NoSQL : Cassandra, MongoDb, Hbase, Elastic Search, etc..)
 - **Traiter et Analyser les données d'une manière distribuée en vue d'en extraire la connaissance pour des prises de décisions**
 - Big Data Processing : Batch Processing (Map Reduce, Spark) et Stream Processing) (Spark, Kafka Stream, Flink, Storm, Samza)
 - Data Mining, Machines Learning (TensorFlow, DeepLearning4J, Weka, etc...) pour l'extraction de la connaissance
 - **Analyser et Visualiser les indicateurs de prises de décisions**
 - Big Data visualisation Tools



Batch Processing/ Stream processing

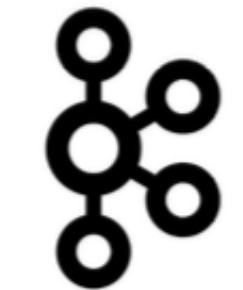
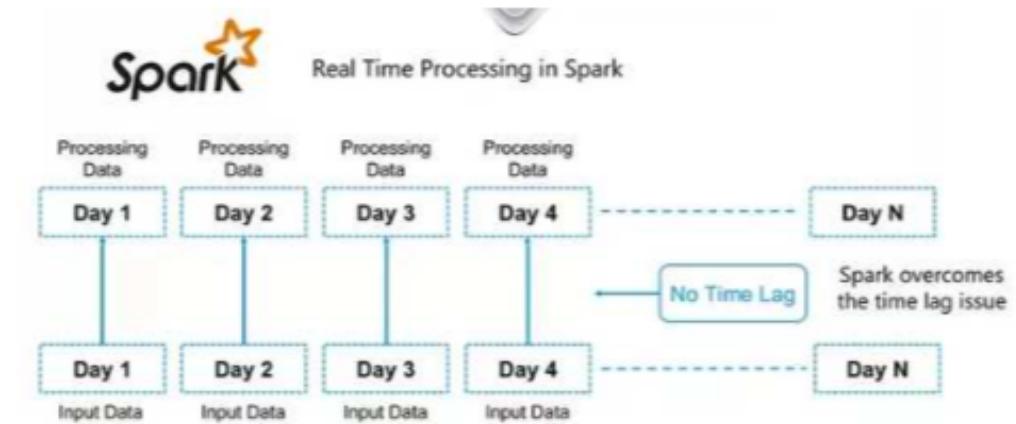
- **Batch Processing** (Traitement par lots):
 - Traitement de blocs de données déjà stockés sur une période donnée.
 - Par exemple, traiter toutes les transactions effectuées par une entreprise financière en une semaine.
 - Ces données contiennent des millions d'enregistrements pour chaque jour pouvant être stockés sous forme de fichiers textes (CSV) ou d'enregistrements stockées dans HDFS, SGBD SQL, NoSQL, etc.
 - Exemple de Framework:
 - MapReduce
 - Spark



Batch Processing/ Stream processing

- **StreamProcessing** (Traitement de flux):

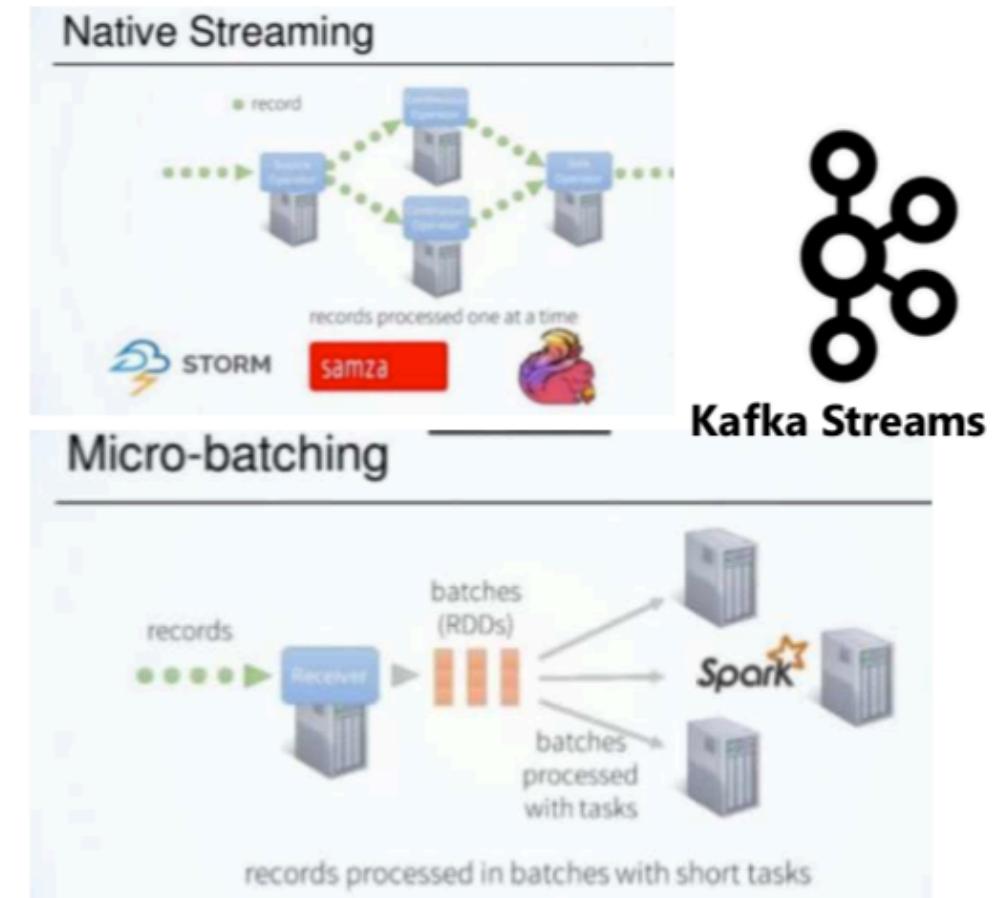
- Contrairement au traitement par lots où les données sont liées avec un début et une fin
- Le Stream Processing est destiné au traitement de flux de données sans fin arrivant en temps réel de façon continue pendant des jours, des mois, des années et à jamais.
- Le traitement de flux nous permet de traiter les données en **temps réel**
- Le traitement de flux permet d'introduire des données dans des outils d'analyse dès qu'elles sont générées et d'obtenir des résultats d'analyse instantanés.



Kafka Streams

Batch Processing/ Stream processing

- **Stream processing** : 2 approches pour mettre en place un Framework Streaming:
 - **Native Streaming (Real Time Processing)**
 - Chaque enregistrement entrant est traité dès son arrivée, sans attendre les autres.
 - Exemples: Storm, Flink, Kafka Streams, Samza.
 - **Micro Batch Processing (Micro Batching)**
 - Les enregistrements entrants toutes les quelques secondes sont mis en lots, puis traités en un seul mini-lot avec un délai de quelques secondes.
 - Exemples: Spark Streaming, Storm-Trident.



Ecosystème du Big Data

L'écosystème du Big data s'est élargi avec beaucoup d'outils comme :

- **Stream Processing :**
 - **Apache Storm** : Framework de calcul et de traitement distribué de flux de données (Stream Processing)
 - **Apache Flink** : Framework de calcul et de traitement distribué de flux distribués (Stream Processing)
 - **Apache Spark** : Framework de traitement distribué de données Big data (Alternative de Map Reduce) et de Stream Processing
 - **Apache Kafka Streams** : Plateforme de Streaming de données en temps réel entre les applications distribuées et Système de Messagerie applicative
- **Apache Zookeeper** : Système de gestion de configuration des systèmes distribués pour assurer la coordination entre les nœuds.
- **SQBD NoSQL :**
 - **Apache Hbase** : SGBD NoSQL, distribué (Stockage structuré pour les grandes tables)
 - **MangoDB** : SGBD NoSQL (Not Only SQL) : Les données sont stockées d'une manière distribuée dans plusieurs nœuds sous forme de documents au format JSON
 - **Cassandra** : est un système de gestion de base de données (SGBD) de type NoSQLconçu pour gérer des quantités massives de données sur un grand nombre de serveurs, assurant une haute disponibilité en éliminant les point de défaillance unique
 - **ElasticSearch** : Moteur de recherche distribué et multi-entité à travers une interface REST.
- **Hazelcast** : Cache mémoire distribué, SGBD NoSQL en mémoire, Système de Messagerie applicative

Ecosystème du Big Data

- **Apache Pig:** Plateforme de haut niveau de création d'applications MapReduce (Langage Pig Latin qui ressemble à SQL) au lieu d'écrire du code Java.
- **Apache Hive:** Infrastructure d'entrepot de données pour l'analyse et le requêtage avec un langage proche de SQL
- **Apache Phoenix:** Moteur de Base de donnée relationnel qui repose sur Hbase
- **Apache Impala:** Moteur de requêtes SQL de cloudera pour un système basé sur HDFS et Hbase
- **Apache Flume :** Système de collecte et d'analyse des fichiers logs
- **Apache Sqoop:** Outils sur ligne de commandes pour transférer les données entre les SGBD relationnels et Hadoop.
- **Apache Oozie :** outil d'ordonnancement des flux de Hadoop