

Big Data

Daniel Aloise

`daniel.aloise@polymtl.ca`

Le « *Big* » du *Big Data*

- *Big Data* est un terme à la mode qui dénote l'analyse de données massives.
- La quantité de données nécessaire pour être qualifiée de « massive » augmente avec le temps.
- Les données massives se distinguent des données qui rentrent en mémoire ou qui peuvent être analysées par un seul ordinateur.
- L'analyse de données massive requiert des infrastructures à grande échelle et robustes :
 - *Robust Cloud Computing*



Le « *Big* » du *Big Data*

En 2020, chaque seconde, il y a environ :

- +8 000 tweets envoyés
- +84 000 vidéos YouTube regardées
- +82 000 requêtes Google émises
- +2 900 000 mails envoyés

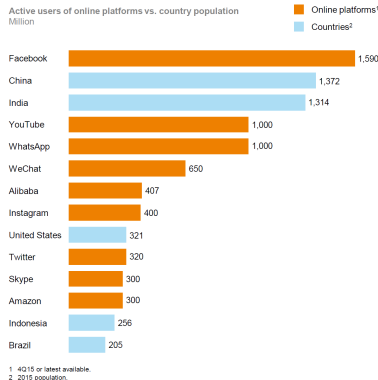
[Internet Live Stats](#)

Le « *Big* » du *Big Data*

En 2020, chaque seconde, il y a environ :

- +8 000 tweets envoyés
- +84 000 vidéos YouTube regardées
- +82 000 requêtes Google émises
- +2 900 000 mails envoyés

Internet Live Stats



Source : McKinsey global

Ces nombres augmentent rapidement !

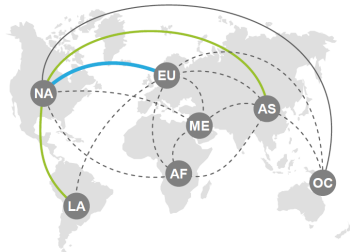
Le « *Big* » du *Big Data*

Used cross-border bandwidth

Regions	NA United States and Canada	EU Europe	AS Asia	LA Latin America	ME Middle East	AF Africa	OC Oceania
Bandwidth Gigabits per second (Gbps)	----	—	—	—	—	—	—
	<50	50–100	100–500	500–1,000	1,000–5,000	5,000–20,000	>20,000

2005¹

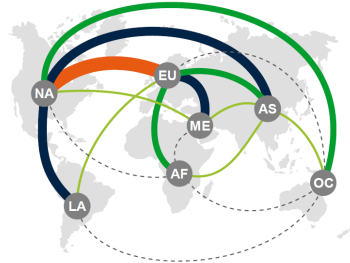
100% = 4.7 Terabits per second (Tbps)



2014

100% = 211.3 Tbps

45x larger



SOURCE: TeleGeography; McKinsey Global Institute analysis

Le « *Big* » du *Big Data*

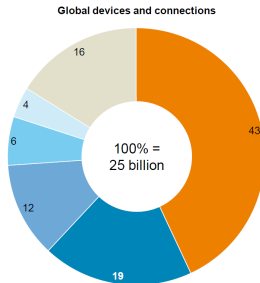
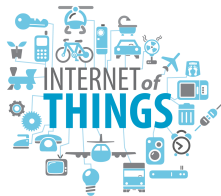


Le « *Big* » du *Big Data*

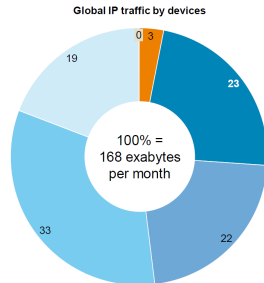
By 2019, machine-to-machine connections are expected to account for more than 40 percent of global devices and connections

Connections, 2019

- Machine-to-machine (M2M)
- Smartphones
- TVs
- PCs
- Tablets
- Other



SOURCE: Cisco, McKinsey Global Institute analysis



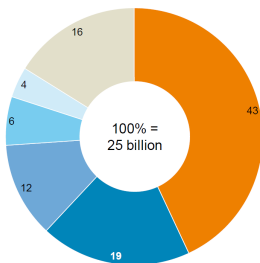
Le « *Big* » du *Big Data*

By 2019, machine-to-machine connections are expected to account for more than 40 percent of global devices and connections

Connections, 2019

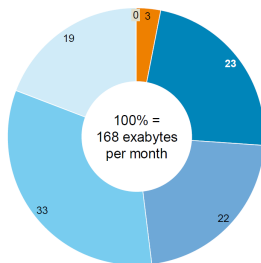
Machine-to-machine (M2M)	Smartphones	TVs	PCs	Tablets	Other
43	19	12	6	4	16

Global devices and connections



SOURCE: Cisco, McKinsey Global Institute analysis

Global IP traffic by devices



- IBM estime que plus de 90% des données mondiales ont été générées au cours de ces deux dernières années.

Les 3 « V » - Volume

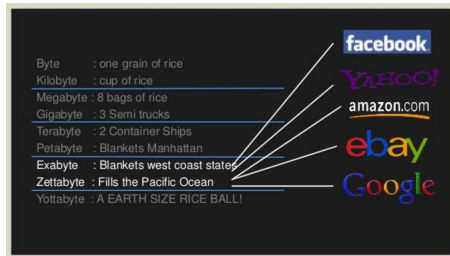
Volume

- Infrastructures de stockage sophistiquées.
- Algorithmes linéaires en temps.
- Comment se représenter la taille des données ?

Les 3 « V » - Volume

Volume

- Infrastructures de stockage sophistiquées.
- Algorithmes linéaires en temps.
- Comment se représenter la taille des données ?
- Supposons qu'un octet est un grain de riz, alors :



Source : Skiena, 2017

Les 3 « V » - Variété

Variété Aujourd'hui, les sources de données sont très hétérogènes :

- Technique d'intégration ad hoc¹.
- Les algorithmes d'analyse sont très différents selon le type des données (textuelle, vidéo, audio, etc.).

1. ad hoc est une locution latine qui signifie « pour cela ». (Wikipedia)

Les 3 « V » - Vitesse

Vitesse Systèmes en **temps réel** (*live*) et collection des données en **continu** (*always on*).

- Infrastructures sophistiquées pour collecter, indexer, récupérer, et visualiser les données.
- Les utilisateurs souhaitent accéder aux dernières données en temps réel.
- Ingénierie et technologie.


Big Data ou Bad Data ?

- Les données massives peuvent être une ressource exceptionnelle.
- Cependant, les données massives sont particulièrement sujettes aux limitations et biais tels que :
- **Participation non-représentative**
 - Les données de n'importe quel réseau social ne reflètent pas les idées des personnes qui ne l'utilisent pas.
 - Par exemple : Instagram (jeunes), The Wall Street Journal (riche), etc.

Big Data ou Bad Data ?

- Données générées par des machines
 - Une armée de critiques (*reviewers*) écrit chaque jour de faux commentaires.
 - De nombreux agents numériques (*bots*) écrivent et consomment massivement des tweets et autres textes !
 - 90% des mails envoyés sont des pourriels (*spams*).
 - Par conséquent, les données vous mentent peut-être !
 - Twitter estime que 23 M de ces utilisateurs actifs sont des agents numériques (août 2014) !

Exemple : marché boursier




PennyStocksBy
@PennyStocksBy

Follow

\$NGTF Breaking News: Record Sales on Amazon. ow.ly/cpU430eMC6r **\$AMZN**
\$COST **\$TSLA** **\$DRYS** **\$ENCC** **\$MVES**
\$AEXE **\$BMXI** **\$BDCI** **\$BITC** **\$ECEZ**

3:00 PM - 30 Aug 2017

200 Retweets 1 Like

 200
  1


TipeDaily
@TipeDaily

Follow

 149
  4


SmallCapMarketPlace
@SmallCapMarketP

Follow

 266
  4


PennyStockAlerts
@PennyStockAlerts

Follow

 276
  2

Chavoshi et al. 2019

Big Data ou Bad Data ?

- Trop de redondance
 - La majorité des données correspondent à des objets déjà connus.
 - La déduplication, c'est-à-dire la suppression des doublons, est une étape essentielle de beaucoup d'analyses.
 - Par exemple, utiliser des photos prises sur internet pour identifier des bâtiments :



Exploiter le stockage hiérarchique

- Les algorithmes d'analyse des données massives sont souvent limités par le **stockage** ou le **débit** plutôt que par la **puissance de calcul**.
- Il faut 30 minutes pour lire 1 To depuis un HDD, et 5 minutes depuis un SSD (PCIe Gen4).
- L'infrastructure est importante !
 - La performance dépend plus de la gestion des données que de la qualité des algorithmes.

Gestion des données massives

- La latence suit généralement une remise de volume
 - le premier accès aux données est plus coûteux que les accès suivants.
- Organiser les calculs pour tenir compte de cette remise, notamment en :
 - analysant les données sous la forme d'un flot (*stream*),
 - pensant « gros fichier » plutôt que « dossier ».
 - compressant les données, si possible.

Paradigmes modernes de calculs

- Ordinateurs individuels :
 - Téléphone ≈ 0.005 TFLOPS
 - Ordinateur central (mainframe) 143 000 TFLOPS

Paradigmes modernes de calculs

- Ordinateurs individuels :
 - Téléphone ≈ 0.005 TFLOPS
 - Ordinateur central (mainframe) 143 000 TFLOPS
- Matériel spécialisé :
 - Se concentre sur un sous-ensemble d'opérations
 - Les GPUs, par exemple, NVIDIA A100 :
 - 19 TFLOPS (FP32)
 - 156 TFLOPS (TF32 Tensor Core)
 - 312 TFLOPS (TF32 Tensor Core + Sparsity)

Paradigmes modernes de calculs

- Ordinateurs individuels :
 - Téléphone ≈ 0.005 TFLOPS
 - Ordinateur central (mainframe) 143 000 TFLOPS
- Matériel spécialisé :
 - Se concentre sur un sous-ensemble d'opérations
 - Les GPUs, par exemple, NVIDIA A100 :
 - 19 TFLOPS (FP32)
 - 156 TFLOPS (TF32 Tensor Core)
 - 312 TFLOPS (TF32 Tensor Core + Sparsity)
- Système distribué :
 - De nombreux ordinateurs « peu puissants » qui travaillent ensemble.
 - Peut atteindre 100 000 TFLOPS

Calcul distribué

- Diviser le problème en sous-problème plus simple à résoudre :
 - ① Les ordinateurs (peu puissant) résolvent simultanément un sous-problème chacun.
 - ② Combiner les solutions des sous-problèmes pour résoudre le problème initial.

Exemple

- Vous devez compter le nombre de parcmètres à Montréal.
 - Approche centralisée (1 ordinateur) :
 - 1 marathonien parcourt toute la ville et compte les parcmètres.
 - 1 système de comptage automatique à partir d'images satellites.
 - Approche distribuée (beaucoup d'ordinateurs) :
 - 1,000 personnes parcourent une petite zone géographique et comptent les parcmètres.
 - Une fois terminé, chacun envoie son rapport au QG.



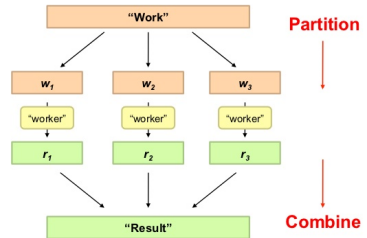
Parallélisme de donnée

- La meilleure (peut-être unique) façon d'exploiter le parallélisme pour traiter les données massives.
- **Problème :**
 - Les données sont trop massives.
 - Envoyer les données sur le réseau prend du temps.
- **Solution :**
 - Rapprocher les calculs des données
 - Traiter les données séquentiellement ; les recherches sont coûteuses.
 - Stocker les données plusieurs fois pour augmenter la fiabilité.

Un algorithme de *Big Data* classique

- Itérer sur un grand nombre d'enregistrements
- Extraire de chaque enregistrement quelque chose d'intérêt
- Mélanger et trier les résultats intermédiaires
- Agréger les résultats intermédiaires
- Générer la sortie finale

Divide and Conquer



Les difficultés de la parallélisation

- Comment assigner les sous-problèmes aux travailleurs (*workers*) ?
- Que faire s'il y a plus de sous-problèmes que de travailleurs ?
- Que faire si les travailleurs doivent s'échanger des résultats intermédiaires ?
- Comment agréger les résultats partiels ?
- Comment savoir si tous les travailleurs ont fini ?
- Que faire si un travailleur ne répond plus ?
- ...

MapReduce

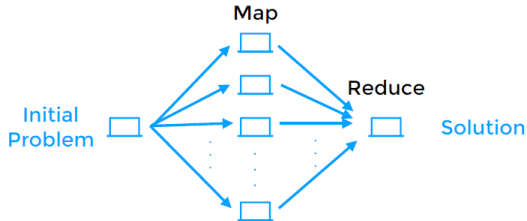
- **MapReduce**, le paradigme de Google pour faire des calculs distribués, s'est aujourd'hui largement répandu grâce à son implémentation libre de droits (**Apache**) **Hadoop** qui offre :
 - Un modèle de programmation parallèle simple
 - Une mise à l'échelle à des milliers d'ordinateurs simples.
 - Une tolérance aux pannes grâce à de la redondance.

Les composants de Hadoop

- Le noyau d'Hadoop est constitué de deux systèmes :
 - **MapReduce** : paradigme de traitement des données massives de manière parallèle/distribuée (*fault-tolerant, scheduler, execution*)
 - **HDFS** (Hadoop Distributed File System) : un système de fichiers distribué (*fault-tolerant, high-bandwidth, high-availability*)
- Plus d'information sur l'infrastructure dans le cours de F. Khomh [LOG 8415 - Concepts avancés en infonuagique](#)

Map et Reduce

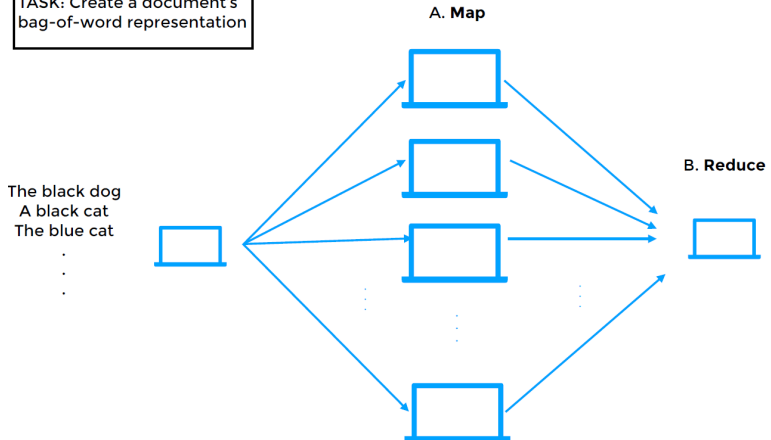
- Deux types de tâches :
 - **Map** : résoudre un sous-problème
 - **Reduce** : agréger les résultats des sous problèmes



Charlin, 2017

Example

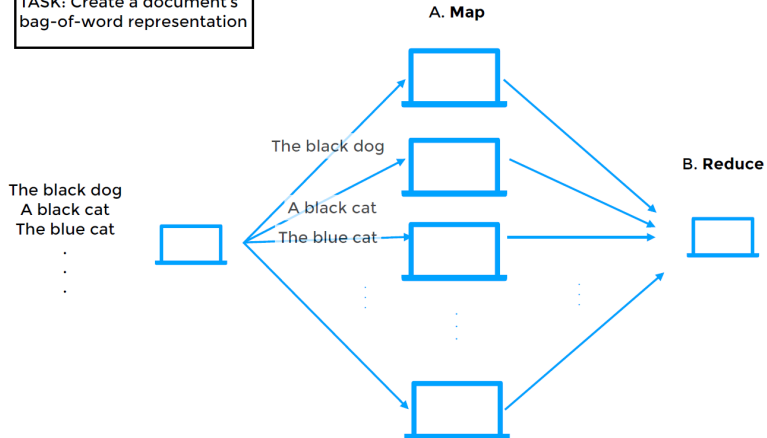
TASK: Create a document's
bag-of-words representation



Charlin, 2017

Example

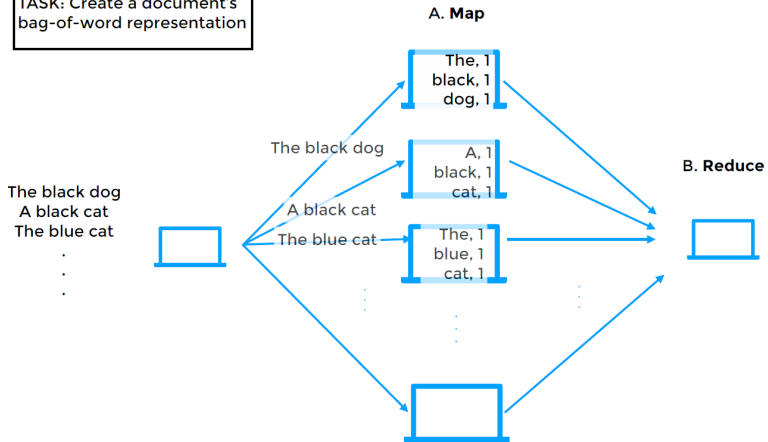
TASK: Create a document's bag-of-words representation



Charlin, 2017

Example

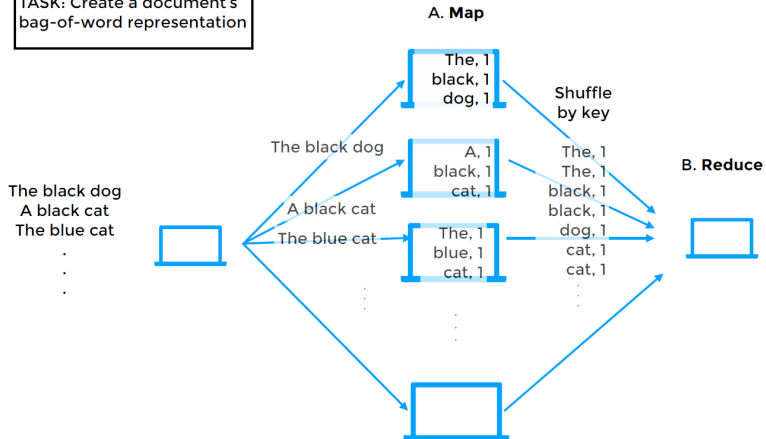
TASK: Create a document's bag-of-words representation



Charlin, 2017

Example

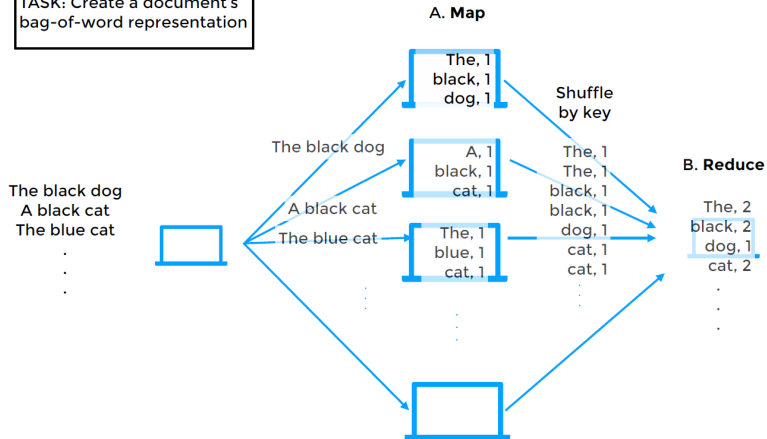
TASK: Create a document's bag-of-words representation



Charlin, 2017

Example

TASK: Create a document's bag-of-words representation



Charlin, 2017

Example

```
Map(String docid, String text):  
    for each word w in text:  
        Emit(w, 1);
```

```
Reduce(String term, Iterator<Int> values):  
    int sum = 0;  
    for each v in values:  
        sum += v;  
    Emit(term, sum);
```

MapReduce

- En général, il y a plus de sous-problèmes que de machines disponibles.
- Temps \approx linéairement proportionnel au nombre de machines
- Si une machine crash, il suffit de recalculer le sous-problème associé
- Les données sont lues depuis le disque au début et écrites à la fin

Exercice

Implémenter les fonctions Map et Reduce qui calculent pour de larges fichiers d'entiers :

- ① La plus grande valeur
- ② La valeur moyenne

Exercice 1

Pour la + grande valeur

```
map(file_id, iterator numbers){  
  
    max=INTEGER.MIN_VALUE  
    while(numbers.hasNext()):  
        num=numbers.next()  
        if(num>max):  
            max=num  
    end while  
    emit('max',max)  
}
```

```
reduce(key, iterator max_values){  
  
    max=INTEGER.MIN_VALUE  
    while(max_values.hasNext()):  
        num=max_values.next()  
        if(num>max):  
            max=num  
    end while  
    emit('overall_max',max)  
}
```

Exercice 2

```
map(file_id,iterator numbers){
```

```
  sum=0
```

```
  count=0
```

```
  while(numbers.hasNext()):
```

```
    num=numbers.next()
```

```
    sum+=num
```

```
    count+=1
```

```
  end while
```

```
  emit('avg',(sum,count))
```

```
}
```

Valeur
moyenne
MAP

Exercice 2

```
reduce(key, iterator sum_count_tuples){  
  
    sum=0  
    count=0  
    while(sum_count_tuples.hasNext()):  
        sum_i,count_i=sum_count_tuples.next()  
        sum=sum+sum_i  
        count=count+count_i  
    end while  
    emit('overall_avg',(sum/count))  
}
```

Valeur moy. REDUCE

En résumé

- Les calculs distribués sont utiles :
 - pour les données massives,
 - pour accélérer les calculs.
- Les *frameworks* actuel tel que Spark permettent d'implémenter facilement des modèles et algorithmes de *machine learning*.
- Des calculs plus rapides en décomposant les problèmes en plusieurs sous-problèmes identiques.
- Nécessite quand même de l'implémentation.

Services de *cloud computing*

- Les plateformes comme Amazon AWS, Google Cloud et Microsoft Azure rendent facile la location d'un grand nombre de machines pendant une période courte.
- Le coût est difficile à évaluer, car plusieurs facteurs rentrent en compte : processeurs, cartes graphiques, et mémoire vive, mais aussi le stockage à long terme et la bande passante.

Services de *cloud computing*

- Il est possible de réduire les coûts pour certains usages :
 - Les *spot instances* permettent de payer uniquement le temps où les instances sont utilisées.
 - Les *reserved instances* permettent de réserver des instances pour une longue période.
- Ressources gratuites pour les chercheurs au Canada :

compute | calcul
canada | canada



Implications éthiques et sociétales du *Big Data*

- Transparence et propriété des données.
- Biais des modèles.
- Préserver la sécurité des données massives.
- Préserver l'anonymat dans les données agrégées.
- Une discussion approfondie peut être trouvée dans le livre de Skiena, 2017.

Transparence et propriété des données

- Est-ce que votre organisation suit les bonnes pratiques de stockage et d'utilisation des données ?
- Dans quelle mesure les utilisateurs possèdent-ils les données qu'ils ont générées ?
- Est-ce que les erreurs peuvent se propager ? Existe-t-il des mécanismes de correction ?
- Est-ce que la provenance des données est préservée ?

Biais des modèles

- Les algorithmes de *machine learning* héritent des biais des données d'apprentissage :
 - Will your search engine show better job opportunities to men than women ?
 - Are predatory ads shown to poor people ?
 - Do news filters reinforce political polarization ?

Préserver la sécurité des données massives

- Il y a une responsabilité éthique à encrypter et supprimer les données pour éviter les failles de sécurité :
 - Demander à 100 millions d'utilisateurs de changer leur mot de passe nécessite 190 « hommes-années » (*man-year*).
 - Les adresses, identifiants, et mots de passe divulgués persistent des années.

Préserver l'anonymat dans les données agrégées

- Les utilisateurs sont souvent identifiables même si les noms, adresses, et identifiants sont supprimés.
 - Le moteur de recherche AOL pas si anonyme que ça
- Les *data scientists* doivent aspirer à être responsables.