MapReduce

Nicolas Dugué
nicolas.dugue@univ-orleans.fr

M2 MIAGE
Systèmes d'information répartis

Plan

Introduction

Les données

- Facebook, Twitter, LinkedIn
- Des appels téléphoniques
- Des clients et leurs achats
- Des utilisateurs d'internet
- Des emails
- Des trajets en Velib
- Des vidéos de surveillance
- Des données scientifiques

Introduction

Un exemple : Twitter

- $>$ 500 millions d'utilisateurs recensés en 2012
- Des milliards d'abonnement entre utilisateurs
- un milliard de tweets tous les deux jours et demi
- Utilisation de hashtags, mentions, urls

$\rightarrow$ Données complexes (textes, liens entre utilisateurs, tags) + volume immense : Big Data

Introduction

Un exemple : Twitter

- $>$ 500 millions d'utilisateurs recensés en 2012
- Des milliards d'abonnement entre utilisateurs
- un milliard de tweets tous les deux jours et demi
- Utilisation de hashtags, mentions, urls

$\rightarrow$ Données complexes (textes, liens entre utilisateurs, tags) + volume immense : Big Data

Stocker ces données

Usama Fayyad (Yahoo) : principale utilisation d'Hadoop
Appliances Teradata couteuses
Apache Hadoop et son HDFS : OpenSource

Introduction

Un exemple : Twitter

- $>$ 500 millions d'utilisateurs recensés en 2012
- Des milliards d'abonnement entre utilisateurs
- un milliard de tweets tous les deux jours et demi
- Utilisation de hashtags, mentions, urls

$\rightarrow$ Données complexes (textes, liens entre utilisateurs, tags) + volume immense : Big Data

Fouiller ces données

Algorithmes peu complexes
Epurer les données
Paralléliser le traitement des données : le modèle MapReduce

Plan

MapReduce

Les implémentations

- Octopy ou Mincemeat python
- Phoenix C++ pour multicoeurs
- Mars pour GPU
- MongoDB
- Apache Hadoop et son HDFS

MapReduce

Programmation MapReduce

Modèle de programmation très contraint :
Map et **Reduce** $\rightarrow$ Issu de la programmation fonctionnelle (Caml, Haskell, Lisp)

Pourquoi MapReduce ?

Traiter de gros volumes de données !
Avec Apache Hadoop :

- Parallélisation automatique des opération Map et Reduce
- Equilibrage de charge
- Tolérance aux pannes

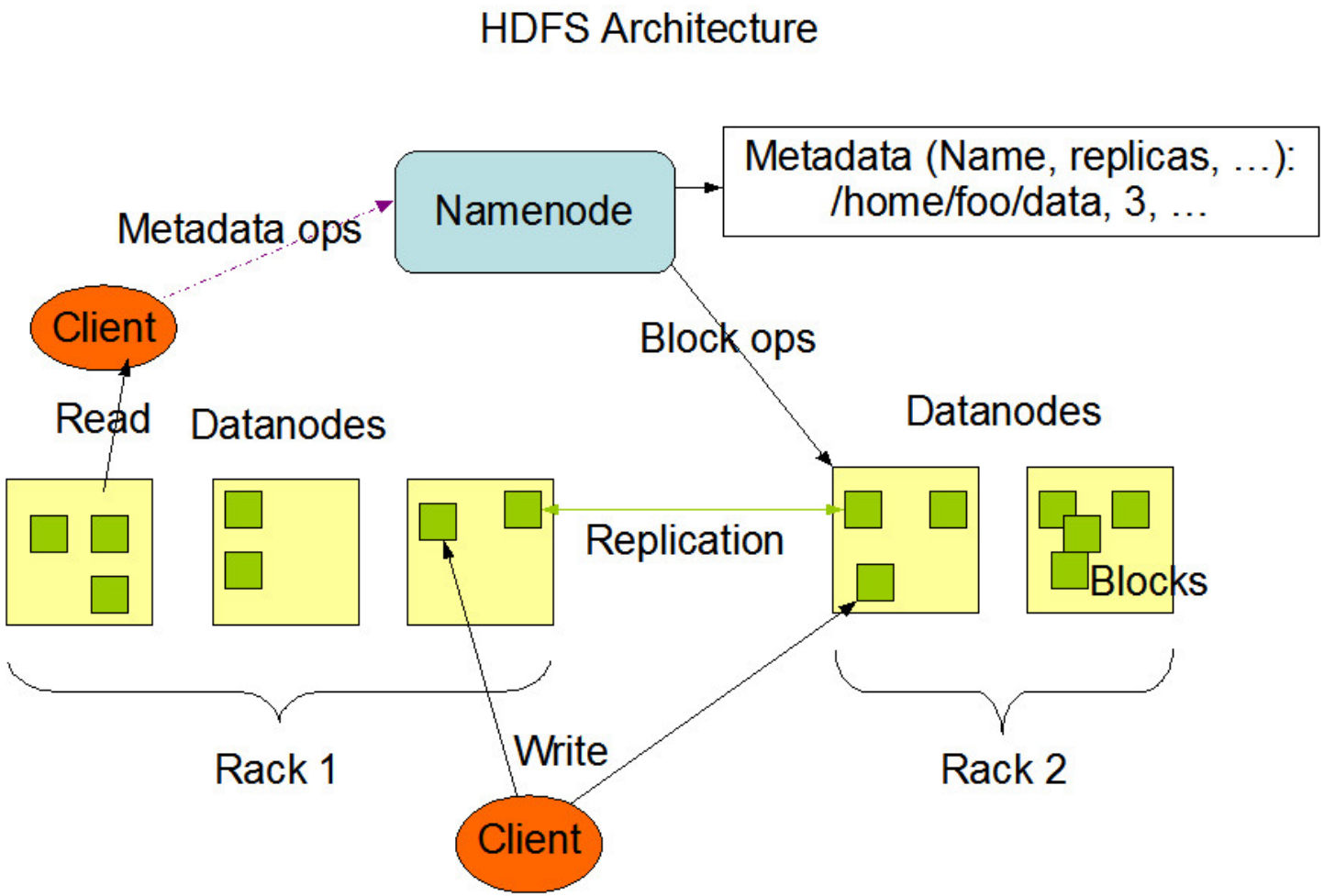
$\rightarrow$ Scalabilité horizontale

MapReduce et le Cloud

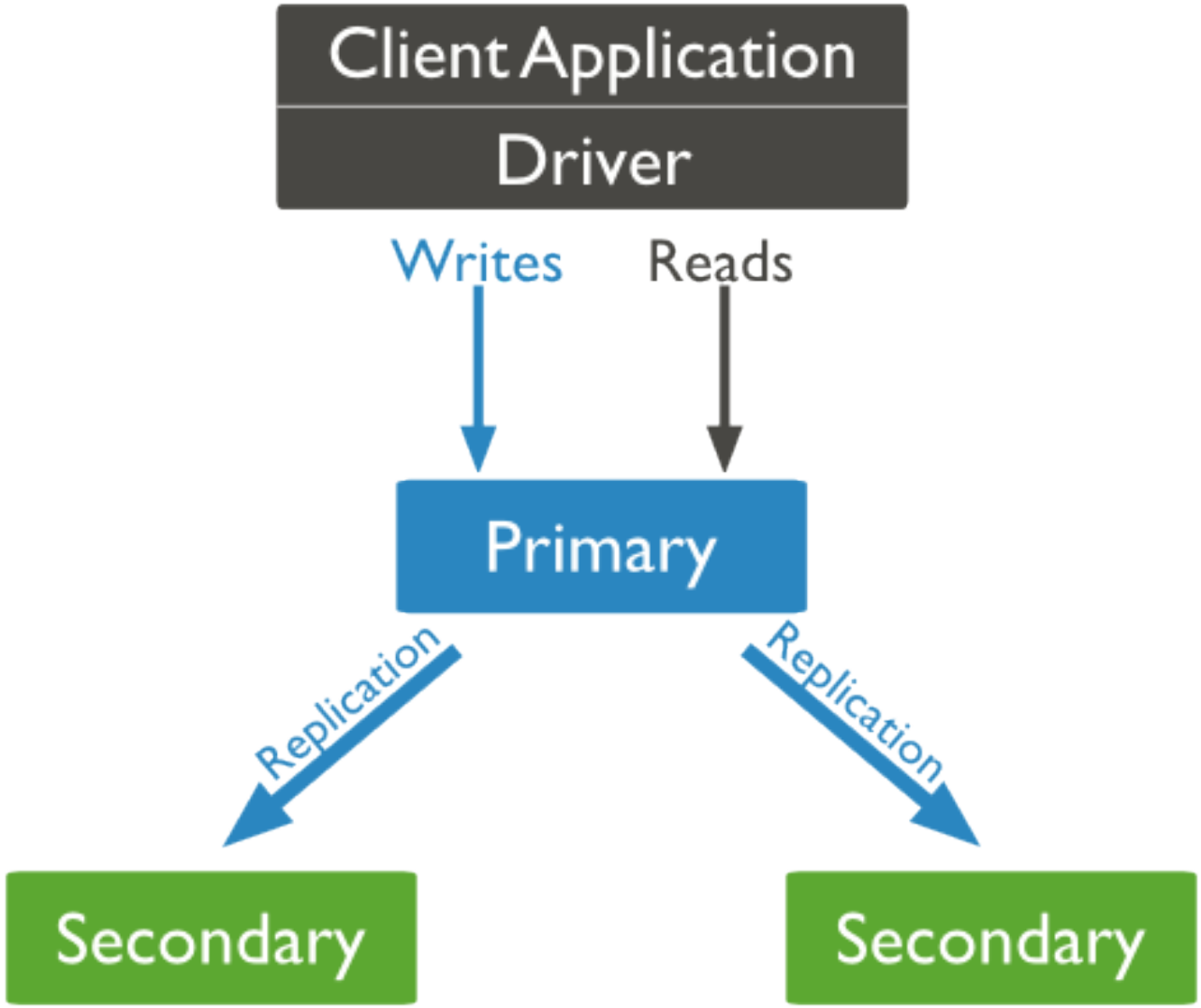
Adapté au Cloud

- Nombre de Mapper et de Reducer potentiellement illimités
- Parallélisme automatique
- Amazon Elastic MapReduce

Le stockage - HDFS



Le stockage - MongoDB



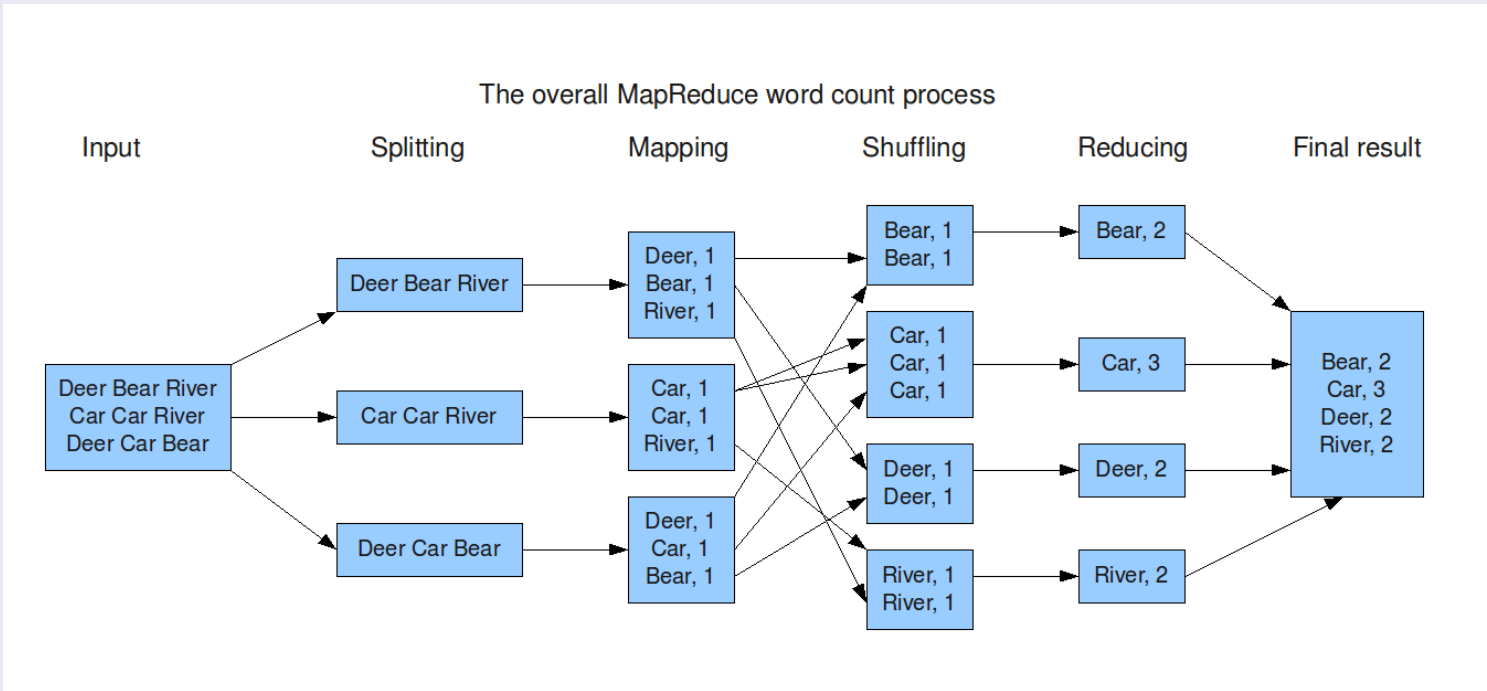
Le modèle MapReduce

5 étapes

1. Lecture des données
2. Map : pour chaque élément des données, appliquer une fonction qui retourne un couple (clé,valeur)
3. Trier les couples selon leurs clés
4. Reduce: agréger, résumer, filtrer ou transformer les données
5. Ecrire le résultat

Le modèle MapReduce

Wordcount



MapReduce en Natif

Du Javascript

```
db.tweets_sentiment.mapReduce(
function() { emit(KEY, VALUE); },
function (key, values) { return RESULT },
{
query : { },
out : "RESULT_NAME"
}
)
```

Stockage du résultat

Résultat stocké dans db.RESULT_NAME

Plan

MapReduce pour fouiller des tweets

Exemple : compter la taille moyenne des tweets

```
db.tweets_sentiment.mapReduce(
function() { emit(this.sentiment, this.tweet.length);
},
function (key, values) { return Array.avg(values)
},
{ out : "avg_char_per_sentiment" }
)
```

Stockage du résultat

```
> db.avg_char_per_sentiment.find()
{ "_id" : 0, "value" : 68.82639589889486 }
{ "_id" : 4, "value" : 66.30062250406955 }
```

Exemple 2 : Fréquence des mots

```
db.tweets_sentiment.mapReduce(
function() {
var tab = this.tweet.split(" ");
for (var i = 0; i < tab.length; i++) {
emit(tab[i], 1);
}
},
function (key, values) { return Array.sum(values) },
{ query : { sentiment : 0 },
out : "word_frequency_0" }
)
```

Les plus fréquents

```
> db.word_frequency_0.find().sort({value : -1})
{ "_id" : "", "value" : 8582 }
```

Les tweets : des sacs de mots

Soit W le dictionnaire de mots au moins une fois dans un ensemble de tweets T.
La représentation sac de mots d'un tweet $t_j \in T$ est un vecteur de poids $(w_{1j}, ..., w_{|W|j})$ où w_{ij} est la fréquence d'apprition du mot w_i du dictionnaire dans le tweet t_j

Les tweets : des sacs de mots

Soit W le dictionnaire de mots au moins une fois dans un ensemble de tweets T.
La représentation sac de mots d'un tweet $t_j \in T$ est un vecteur de poids $(w_{1j}, ..., w_{|W|j})$ où w_{ij} est la fréquence d'apprition du mot w_i du dictionnaire dans le tweet t_j

Algo Data mining

- K plus proches voisins
- Naive Bayes

Plan

La recommandation

Objectif système de recommandation

Présenter des contenus susceptibles d'intéresser l'utilisateur

Exemples

- Amazon : suggestion de produits ;
- Last.fm : suggestion de groupes/chansons ;
- Facebook : suggestion d'amis ;

La recommandation

Construire un profil utilisateur

- Pages, Objets, chansons : tracer les habitudes de l'utilisateur
- Demander à l'utilisateur d'évaluer
- Demander à l'utilisateur de créer des listes de préférences
- Utiliser le réseau social de l'utilisateur
- Demander des informations personnelles
- Utiliser des tags

Systèmes de Filtrage collaboratifs utilisateurs

- Qu'est ce qui est proche de ce que j'ai aimé ?
- Qu'aiment les profils proches du mien ?
- Qu'aiment mes amis ?

La recommandation

Systèmes de Filtrage collaboratifs utilisateurs

Trouver utilisateurs avec un profil proche de l'utilisateur
Calculer une liste de recommandations

La recommandation

Suggérer des amis

A $\rightarrow$ B C D
B $\rightarrow$ A C D
C $\rightarrow$ A B
D $\rightarrow$ A B

La recommandation

Suggérer des amis

A $\rightarrow$ B C D
B $\rightarrow$ A C D
C $\rightarrow$ A B
D $\rightarrow$ A B

For map(A $\rightarrow$ B C D)

(A B) $\rightarrow$ B C D
(A C) $\rightarrow$ B C D
(A D) $\rightarrow$ B C D

La recommandation

Suggérer des amis

A $\rightarrow$ B C D
B $\rightarrow$ A C D
C $\rightarrow$ A B
D $\rightarrow$ A B

For map(A $\rightarrow$ B C D)

(A B) $\rightarrow$ B C D
(A C) $\rightarrow$ B C D
(A D) $\rightarrow$ B C D

For map(B $\rightarrow$ A C D)

(A B) $\rightarrow$ A C D
(B C) $\rightarrow$ A C D
(B D) $\rightarrow$ A C D

La recommandation

Suggérer des amis

A $\rightarrow$ B C D
B $\rightarrow$ A C D
C $\rightarrow$ A B
D $\rightarrow$ A B

For map(C $\rightarrow$ A B)

(A C) $\rightarrow$ A B
(B C) $\rightarrow$ A B

La recommandation

Suggérer des amis

A $\rightarrow$ B C D
B $\rightarrow$ A C D
C $\rightarrow$ A B
D $\rightarrow$ A B

For map(C $\rightarrow$ A B)

(A C) $\rightarrow$ A B
(B C) $\rightarrow$ A B

For map(D $\rightarrow$ A B)

(A D) $\rightarrow$ A B
(B D) $\rightarrow$ A B

La recommandation

Suggérer des amis

A → B C D
B → A C D
C → A B
D → A B

Tri par clé

(A B) → (A C D) (B C D)
(A C) → (A B) (B C)
(A D) → (A B) (B C)
(B C) → (A B D) (A C D)
(B D) → (A B C) (A C D)
(C D) → (A B C) (A B D)

La recommandation

Suggérer des amis

A $\rightarrow$ B C D
B $\rightarrow$ A C D
C $\rightarrow$ A B
D $\rightarrow$ A B

Reduce

(A B) $\rightarrow$ (C D)
(A C) $\rightarrow$ (B)
(A D) $\rightarrow$ (B)
(B C) $\rightarrow$ (A D)
(B D) $\rightarrow$ (A C)
(C D) $\rightarrow$ (A B)

Plan

Big data

Big data

Les terriens ont en moyenne :
un sein et un testicule

Big data

- Récolter de gands volumes de données
- Stocker les données
- Paralléliser les traitements

Big data

- Récolter de gands volumes de données
- Stocker les données
- Paralléliser les traitements
- **Interpréter les données**