

# Business Intelligence - Stockage structuré des données

Olivier Schwander <olivier.schwander@lip6.fr>

UPMC

# Données sur le disque

## Données hétérogènes

- ▶ Valeurs numériques
- ▶ Images
- ▶ Textes

## Gros volumes

- ▶ Giga-octets
- ▶ Tera-octets
- ▶ Peta-octets

# Exploitable par une machine

## Données structurées

- ▶ Interprétable par un programme
- ▶ Formats de fichiers

## Retrouver l'information

- ▶ Recherche rapide
- ▶ Filtrage selon des critères
- ▶ Indexation

## Fichiers plats

Le Costa Rica, officiellement appelée république du Costa Rica, en espagnol República de Costa Rica , est une république constitutionnelle unitaire d'Amérique centrale ayant un régime présidentiel.

La majeure partie de son territoire est situé sur l'isthme centraméricain, encadré par mer des Caraïbes à l'est et l'océan Pacifique à l'ouest et au sud, bordé au nord par le Nicaragua et au sud-est par le Panama, mais comprend également l'Île Cocos située dans l'océan Pacifique à plus de 500 kilomètres des côtes du pays. Elle a pour capitale San José, pour langue officielle l'espagnol et pour monnaie le colon. Sa devise est « ¡Vivan siempre el trabajo y la paz ! » (« Que vivent pour toujours le travail et la paix ! ») et son drapeau est constitué de cinq bandes horizontales respectivement bleue, blanche, rouge, blanche et bleue. Son hymne est Noble patria, tu hermosa bandera.

# Tableaux

0	2
1	4
2	8
3	32
4	52
5	666
6	999
7	1664

## Retrouver l'information

- ▶ Avec le numéro de la case

## Inconvénients

- ▶ Il faut connaître le numéro
- ▶ Taille fixe pour les données

## Clé-valeur

two1	2
two2	4
two3	8
factor1	32
factor2	52
beast1	666
beast2	999
product	1664

### Correspondance clé-valeur

- Accès à une valeur à l'aide d'une clé

### Comment faire ?

- Recherche linéaire : temps linéaire

### Avec un ordre sur les clés

- Dichotomie : temps logarithmique

# Tables de hachage

Figure

## Fonction de hachage

- ▶ Projection de n'importe quoi vers un espace de taille fixe
- ▶  $\{0,1\}^* \rightarrow \{0,1\}^d$  avec  $d$  choisi à l'avance
- ▶ Collision inévitables
- ▶ En moyenne : accès en temps constant
- ▶ Pire cas : accès linéaire

# Structures de données

## Files de priorité

- ▶ Récupérer l'élément avec la plus grande priorité
- ▶ Insérer un élément

## Arbres binaires de recherche

- ▶ Recherche et insertion en temps logarithmique

## kd-tree

- ▶ Dimension quelconque
- ▶ Arbre binaire
- ▶ Partitions de l'espace par des hyperplans parallèles aux axes



# Formats

## Sur le disque

- ▶ Structures de données pas seulement en mémoire

## Entrées/sorties

- ▶ Lire les données
- ▶ Écrire les données

## Contraintes

- ▶ Exploitable par une machine
- ▶ Bonus : exploitable par un humain
- ▶ Lecture facile ? Modification facile ? Création facile ?

## Comma Separated Values

```
PassengerId,Survived,Pclass,Name,Sex,Age,SibSp,Parch,Ticket,
1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,A/5 21171,7.25,
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",
3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,STON/O2. 310128
4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)",female,
5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.05,,S
6,0,3,"Moran, Mr. James",male,,0,0,330877,8.4583,,Q
7,0,1,"McCarthy, Mr. Timothy J",male,54,0,0,17463,51.8625,E4
8,0,3,"Palsson, Master. Gosta Leonard",male,2,3,1,349909,21
9,1,3,"Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)",fe
10,1,2,"Nasser, Mrs. Nicholas (Adele Achem)",female,14,1,0,2
11,1,3,"Sandstrom, Miss. Marguerite Rut",female,4,1,1,PP 954
12,1,1,"Bonnell, Miss. Elizabeth",female,58,0,0,113783,26.55
```

# JavaScript Object Notation

```
{  
  "titanic": [  
    { "PassengerId": 1, "Survived": 0, "Pclass": 3,  
      "Name": "Braund, Mr. Owen Harris", "Sex": "male", ...  
    },  
    { "PassengerId": 2, "Survived": 1, "Pclass": 1,  
      "Name": "Cumings, Mrs. John Bradley", "Sex": "female",  
      ...  
    }  
  ]  
}
```

# YAML

```
titanic:  
  - PassengerId: 1  
    Survived: 0  
    Pclass: 3  
    Name: Braund, Mr. Owen Harris  
    Sex: male  
  - PassengerId: 2  
    Survived: 1  
    Pclass: 1  
    Name: Cumings, Mrs. John Bradley  
    Sex: female
```

# XML

```
<dataset name="titanic">
  <passenger id="1">
    <Survived>0<Survived>
    <Pclass>3</Pclass>
    <Name>Braund, Mr. Owen Harris</Name>
    <Sex>male<Sex>
  </passenger>
  <passenger id="2">
    <Survived>1<Survived>
    <Pclass>1</Pclass>
    <Name>Cumings, Mrs. John Bradley</Name>
    <Sex>female<Sex>
  </passenger>
</dataset>
```

# HTML

```
<!doctype html>
<html>
  <head>
    <title>M2 Statistiques - Business Intelligence</title>
    <link href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/3.3.7/css/bootstrap.min.css">
  </head>
  <body>
    <h1 class="title">M2 Statistiques - Business Intelligence</h1>
    <h2 id="agenda">Emploi du temps</h2>
    <table class="table table-striped table-bordered table-hover">
      <tr>
        <td class="left">Mardi 5 janvier</td>
        <td class="left">14h - 17h</td>
        <td class="left">salle 1525-101</td>
      </tr>
    </table>
```

# Requêtes XML et HTML

## XPath

- ▶ Langage standardisé de requêtes XML et HTML
- ▶ `/dataset/passenger/` : tous les passagers
- ▶ `//Name` : tous les noms, peu importe la position
- ▶ `//passenger[@id=1]` : passager avec l'identifiant 1

## CSS

- ▶ Pour le HTML : surtout pour l'apparence des pages web, mais pas seulement
- ▶ `h1.title` : titre h1 avec la classe title
- ▶ `#agenda` : nœud avec l'identifiant agenda

# Numpy .npy

## Entête

- ▶ *Magic string* : [93, "N", "U", "M", "P", "Y"]
- ▶ Numéro de version du format : 2 octets
- ▶ Taille de l'entête : 2 octets
- ▶ Description du format numpy : (taille de l'entête) - 10 octets

## Données

- ▶ Données brutes du tableau

http:

[//docs.scipy.org/doc/numpy-dev/neps/npv-format.html](http://docs.scipy.org/doc/numpy-dev/neps/npv-format.html)



# Systèmes de fichiers

## Stockage des données

- ▶ Disque dur
- ▶ Mémoire flash

## Chemins de fichiers

- ▶ `C:\Windows\system\bsod.dll`
- ▶ `C:\Users\Toto\Documents\blah.docx`
- ▶ `/etc/fstab`
- ▶ `/home/toto/Documents/blah.odt`
- ▶ `/Users/Toto/stevejobs.pdf`

# Conception

## Arborescence

- ▶ Trouver un fichier : chemin à parcourir dans un arbre
- ▶ Chercher un fichier : parcourir toutes les branches possibles

## Contraintes

- ▶ Optimisé pour la lecture, pour l'écriture
- ▶ Pour les gros fichiers, les petits fichiers
- ▶ Les gros disques, les petits
- ▶ Les disques magnétiques, les mémoires flash

# Exemples de systèmes de fichiers

## Génériques

- ▶ DECTape : PDP (1964)
- ▶ FAT{8,12,16,32} : DOS, Windows (1977, 1980, 1984, 1996)
- ▶ ext{1,2,3,4} : Linux (1992, 1993, 1999, 2006)

## Gross volumes

- ▶ ZFS : Solaris (2004)

## Mémoire flash

- ▶ UBIFS : Linux (2008)
- ▶ exFAT : Windows (2006)
- ▶ F2FS : Linux (2012)

# Bases de données relationnelles

## Universel

- ▶ Domaines différents
- ▶ Données différentes
- ▶ Tailles différentes

## Avantages

- ▶ Garanties de sûreté sur les données
- ▶ Requêtes efficaces

## Contrainte

- ▶ Besoin d'une formalisation de la structure des données

# Orienté tables

Figure

Schéma

- Description des tables : types et noms des colonnes

# ACID

## Atomicité

- ▶ Une transaction se fait complètement ou pas du tout

## Cohérence

- ▶ Le système passe toujours d'un état valide à un autre

## Isolation

- ▶ Indépendance entre les transactions

## Durabilité

- ▶ Une transaction effectuée l'est de façon durable

# Langage de requêtes standardisé

## Structured Query Language

- ▶ Interopérable (à peu près)
- ▶ Langage déclaratif

## Insertion

- ▶ `INSERT INTO passengers (name, survived, sex, class)  
VALUES ("Braund, Mr. Owen Harris", 0, "male", 1);`

## Requête

- ▶ `SELECT (name, sex, class) FROM passengers WHERE  
survived = 1;`

# Algèbre relationnelle

## Relation

- ▶ Table
- ▶ Ensemble de n-uplets

## Opérations ensemblistes

- ▶ Projection : sélection de colonnes `SELECT`
- ▶ Sélection : sélection de lignes `WHERE`
- ▶ Jointure : lien entre deux tables `JOIN`



# Architecture client-serveur

## Serveur

- ▶ Stocke les données
- ▶ Reçoit et interprète les requêtes

## Client

- ▶ Application qui utilise la base de donnée

# Systèmes de gestion des bases de données

## MySQL

- ▶ Libre
- ▶ Très utilisé par des sites web

## PostgreSQL

- ▶ Libre
- ▶ Extensible
- ▶ Types de données évolués

## Oracle Database

- ▶ Propriétaire

## SQLite

- ▶ Libre
- ▶ Embarqué dans l'application
- ▶ Compact

# Passage à l'échelle

## Réplication des données

- ▶ Copies de la base

## Maître/esclaves

- ▶ Écriture sur le maître
- ▶ Propagation des modifications sur les esclaves
- ▶ Lecture sur les esclaves

# Not Only SQL

## Bases relationnelles pas toujours appropriées

- ▶ Schémas compliqués à concevoir
- ▶ Passage à l'échelle pas toujours bon
- ▶ Pas forcément besoin de requêtes compliquées
- ▶ Données faiblement structurées

## Évolution récente

- ▶ Pas de SQL
- ▶ Modèle plus simple pour les données

## Idées anciennes

- ▶ Stockage simple

# Orienté document

## Tableau associatif

- ▶ Pas de tables
- ▶ Passage à l'échelle facile

## Clé-valeur

- ▶ Identifiants pour les documents

## Document

- ▶ Format JSON ou autre

## Exemples de serveurs

### BigTable

- ▶ Google

### HBase

- ▶ Facebook

### BerkeleyDB

- ▶ 1994
- ▶ Bibliothèque embarquée

### MongoDB

- ▶ SourceForge.net

### CouchDB

- ▶ Ubuntu

### Redis

- ▶ Stockage en mémoire
- ▶ Cache

## Language de requête spécifique

Pas de langage commun

### Exemple avec MongoDB

```
db.inventory.find( { type: { $in: [ 'food', 'snacks' ] } } )  
db.inventory.find( { type: 'food', price: { $lt: 9.95 } } )  
db.inventory.find(  
  {  
    type: 'food',  
    $or: [ { qty: { $gt: 100 } }, { price: { $lt: 9.95 } } ]  
  }  
)
```

# Théorème CAP

Dans un système distribué

## Cohérence

- ▶ Tous les nœuds voient la même version

## Disponibilité

- ▶ Chaque requête obtient une réponse

## Résistance aux pannes

- ▶ Perdre un nœud ou un message ne bloque pas le système

**Théorème :** au plus deux propriétés sur les trois



## Passage à l'échelle

### Deux propriétés

- ▶ Disponibilité
- ▶ Résistance aux pannes

### Sacrifice

- ▶ Cohérence : des nœuds peuvent avoir de vieilles versions

### Table de hachage distribué

- ▶ Données réparties sur plusieurs nœuds
- ▶ Mécanismes pour trouver le nœud qui contient le document