

Systemes distribués

Fabien Coelho

MINES ParisTech, PSL Research University

18 mai 2020

Définition de systèmes distribués

A collection of independent computers that appears to its users as a single coherent system. (Tanenbaum)

Une collection d'ordinateurs indépendants qui apparaît à ses utilisateurs comme un seul système cohérent.

NoSQL

Not Only SQL

- modèle de données ?
- théories sous jacentes ?
- langage de requête ?
- navigation (jointures) ?
- redondance, normalisation ?
- performance ?

Key-Value (et sous variantes)

Redis FoundationDB Couchbase Memcached Voldemort. . .

Document Store (souvent du JSON, éventuellement du XML)

CouchDB, MongoDB, Elasticsearch Postgres. . .

Object DB . . .

Graph . . .

Tabular . . .

Tuple store . . .

Blockchains . . .

. . .

Key-Value Store

Key	Value
K1	AAA,BBB,CCC
K2	AAA,BBB
K3	AAA,DDD
K4	AAA,2,01/01/2015
K5	3,ZZZ,5623

<https://commons.wikimedia.org/wiki/File:KeyValue.PNG>

```
{
  "orders": [
    {
      "orderno": "748745375",
      "date": "June 30, 2088 1:54:23 AM",
      "trackingno": "TN0039291",
      "custid": "11045",
      "customer": [
        {
          "custid": "11045",
          "fname": "Sue",
          "lname": "Hatfield",
          "address": "1409 Silver Street",
          "city": "Ashland",
          "state": "NE",
          "zip": "68003"
        }
      ]
    }
  ]
}
```

<https://static.goanywhere.com/images/tutorials/read-json/ExampleJSON2.png>

Données C

$K \rightarrow V$

```
"calvin" : [ ("maths", 5.0), ("physics", 3.0) ]
"susie"  : [ ("maths", 8.0), ("physics", 9.0) ]
"hobbes" : [ ("maths", 2.0), ("physics", 1.0) ]
```

Map = filtrage

$\forall i \in C : \text{map}(i) = \text{list}(K' \rightarrow V')$

```
for (k,v) in C: for (m,n) in v: emit (m: (n, 1))
"maths" : (5.0, 1), "physics" : (3.0, 1), ...
```

Reduce = combinaison

$(K' \rightarrow \text{list}(V')) \rightarrow (K', V')$

```
for (k : (v1, v2)): (k : (v1[0] + v2[0], v1[1] + v2[1]))
"maths" : (15.0, 3), "physics" : (13.0, 3)
```

Équivalent en SQL

```
SELECT matière, SUM(note), COUNT(*)  
FROM Notes  
GROUP BY 1;
```

- complexité d'écriture de *MapReduce*
- implémentation avec nombreuses communications
- développement de langages frontaux en pseudo-SQL
- compilés en *MapReduce*

Théorème CAP sur les systèmes distribués

Choose 2 out of 3

consistency données identiques sur les nœuds
eventually consistent

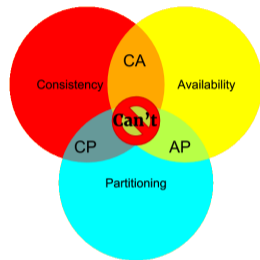
availability disponibilité si panne nœuds

partition tolerance disponibilité si panne réseau

Postgres peut être CA ; S. D. souvent CP ou AP

conjecture *Eric Brewer, UC Berkeley, 2000*

théorème *Seth Gilbert et Nancy Lynch, MIT, 2002*



mangoresearch.co