GIF-4104 - TP 5

Équipe 1 : Vincent Commin & Louis Leenart

1 Introduction

Pour ce dernier TP, nous avons eu à implémenter l'algorithme de *PageRank* en utilisant le moteur *Apache Spark*. L'algorithme consiste à associer un rang à chaque élément d'un graphe orienté. Dans notre cas, les noeuds du graphe représentent des URL, et les arêtes sont les références vers d'autres pages. Concernant l'implémentation, Spark support de nombreux langages et nous avons choisi Java pour sa simplicité d'utilisation ainsi que notre maîtrise du langage.

2 Notre approche

Pour implémenter cet algorithme, nous avons suivi la procédure suivante :

```
LECTURE fichier
CONSTRUCTION rdd A PARTIR de fichier

POUR iteration ALLANT DE 0 A 100 FAIRE
    CALCUL rang
FIN POUR

AFFICHE lien ET rang
```

Plus précisément, le calcul du rang pour chacun des liens se faire de la manière suivante :

```
// un élément = id du lien, liste des id de lien qu'il référence
liens : LISTE COUPLE (id, LISTE id)
rangs : LISTE COUPLE (id, rang)
damping_factor : ENTIER

contributions : LISTE COUPLE (id, contribution)
contributions = join(liens, ranks).map(s -> {
    s.gauche.map(n -> {
        n, s.droit / len(s.gauche)
    })
})
rangs = contributions.reduce(+).map(s -> {
    s * damping_factor + 0.15
})
```

Après un nombre significatif d'intérations, le contenu de ‘rangs‘ contient les couples ‘lien : rang‘, en sachant que plus le rang tends vers 0, moins il est considéré comme intéressant, et plus il est élevé, plus il est considéré comme pertinant.

3 Machine utilisée pour les tests de performance

Modèle	intel i7-8550U
Architecture	x86_64
OS	Archlinux
Fréquence CPU	3.4GHz
Cœurs (physique / logique)	4 / 8
Ram	16 Go, 2400 MT/s
Java	OpenJDK 11.0.15
Spark Java	3.2.0