UNIVERSITÀ DI PISA

Artificial Intelligence and Data Engineering

Cloud Computing

PageRank

Project Documentation

TEAM MEMBERS:
Daniele Cioffo
Edoardo Fazzari
Federica Baldi
Mirco Ramo

Academic Year: 2020/2021

Contents

1 — Introduction

1.1 Description

Each chapter will contain a section describing the performance.

2 — PageRank

2.1 Introduction

In this section a description of the MapReduce implementation of *Page Rank* is given. The algorithm is carried out in **three distinct steps** due to the size of the inputted data:

1. *Graph Construction* phase:
2. *Page Rank Computation* phase:
3. *Sorting* phase:

A cleanup function should be taken in consideration in each step in order to taper down the memory usage as much as possible.

2.1.1 Dangling Node (Dead Ends)

2.1.2 Spider Traps

2.2 First Phase: Graph Construction

DOES NOT CONSIDER NODE WITHOUT OUTGOING LINKS

In the **conf** object it is contained the number of pages used, the solution tallies it without the use of MapReduce. If the file is huge this is not the smartest solution. The number of pages can be count using MapReduce and the use of a Combiner.

Algorithm 1 Graph Construction Mapper

```
1: procedure MAP(key k, page p)
2:     title ← getTitle(d)
3:     list<OutgoingLink> l ← getOutgoingLink(d)
4:     if l.length > 0 then
5:         for all OutgoingLink link in l do
6:             EMIT(title, link)
```

Algorithm 2 Graph Construction Reducer

```
1: procedure REDUCE(title t, edges [e_1, e_2, ...], numberOfPages n)
2:     initialPageRank ← 1/n
3:     edgesString ← edges.toString()
4:     EMIT(title, {initialPageRank, edgesString})
```

2.3 Second Phase: PageRank Estimation

Compute the value iterating till the pagerank of the nodes doesn't change, in the sol the iteration is fixed to 10 (the real algorithm doesn't work in this sense).

In order to check if the values are the same or not a comparison between the previous output and the new one is required. **Note**: a diff should be enough.

For performance we can initiate a variable for counting the number of iterations done.

2.4 Third Phase: Sorting

Algorithm 3 PageRank Computation Mapper

```
1: procedure MAP(key k, page p)
2:     title ← getTitle(d)
3:     list<OutgoingLink> l ← getOutgoingLink(d)
4:     if l.length > 0 then
5:         for all OutgoingLink link in l do
6:             EMIT(title, link)
```

3 — Hadoop Implementation

4 — Spark Implementation in Java

5 — Spark Implementation in Python