Hadoop PageRank

Sohil Jain Ashutosh Bhargave

[**sohjain@iu.edu**](mailto:sohjain@iu.edu)[**ashubhar@indiana.edu**](mailto:ashubhar@indiana.edu)

**Goal**

This assignment provides an illustration of PageRank algorithms and Hadoop. We have implemented a parallel version of PageRank using the programming interfaces of the Hadoop MapReduce framework. Enclosed inside -

1. The source code of Hadoop PageRank
2. Technical report (**sohjain\_HadoopPageRank\_report.docx**) that contains:
   1. The description of the main steps and data flow in your program.
   2. The output file (**sohjain\_HadoopPageRank\_output.txt**) which contains the first 10 urls along with their ranks.

**Implementation of Hadoop Pagerank:**

We implemented pagerank with the help of Hadoop Mapreduce in iterative steps. The steps of the project are mentioned below,

A] *Converting input file*:

The purpose of this step is to convert the input file into a format which the mapreduce understands. The <key value> pair is generated in the 1st mapreduce job named “creategraph”. The Initial rank is calculated as 1/N where N is the total number of sourceURLs. The implementation of the above code is as follows,

The format of the input file is as given below ,

<SourceURL TargetURL1 TargetURL2>

0 4

1 2

2 0

3 14

4 34

5 0 4 6 18 60

6 2 16 92

7 4 16 130

8 0 56 174

9 224

The format of the output file is as given below, <key value>

<SourceURL InitialRank#TargetURL1#TargetURL2>

0 2.0E-4#4

1 2.0E-4#2

2 2.0E-4#0

3 2.0E-4#14

4 2.0E-4#34

5 2.0E-4#0#4#6#18#60

6 2.0E-4#2#16#92

B] *Calculating PageRank*:

The purpose of this step is to calculate the pagerank of all the inputURLs and output those to the Cleanup mapreduce task. The mapper gets the data as a <key, value> pair of the input URL’s and the rank and tagetURLs list together as values. These values are separated by # symbol after every targetURL.

Mapper Output format to reducer

Input to mapper

The format of the input file to the mapper is as given below , <key value>

<SourceURL InitialRank#TargetURL1#TargetURL2>

0 4.0E-8

0 2.0E-4

4 6.666666666666667E-5

5 #0#4#6#18#60

12 4.0E-8

12 #4#56#410

The format of the output file of reducer is as given below, <key value>

<SourceURL Rank > Or

<SourceURL #TargetURL1#TargetURL2>

0 0.10272345983353663#4

1 4.101717217578249E-5#2

2 0.0413952184222524#0

3 4.10171721757826E-5#14

4 0.15173069758712518#34

5 4.1017172171E-5#0#4#6#18#60e

During this stage the Mapper gives the output in two formats. Either it outputs as <SourceURL Rank> or <SourceURL #TargetURLlist>. The reducer when gets the file in that format it first aggregates the rank of all the SourceURL which are its input as keys and then the TargetURL is appended to the RankValue which is calculated. We have used the number of reducer tasks as 10 for the parallel implementation purpose. After the pagerank reducer calculates the pagerank value the output is again given to the mapper task based on the number of iterations we have specified in the initial step. This output is then given to the cleanup mapreduce task.

Thus,

Cleanup Reducer

Mapper

Reducer

Reducer to mapper if iteration is used.

The format of the input file to the mapper is as given below ,

<SourceURL InitialRank#TargetURL1#TargetURL2>

The format of the output file of reducer is as given below,

<SourceURL Rank >

Or

<SourceURL #TargetURL1#TargetURL2>

SOHIL CHANGE THIS . Inset the mapper output here

The format of the output file of cleanup reducer is as given below,

<SourceURL Rank >

C] *Cleanup*:

This is the 3rd and final job of the pagerank implementation. The purpose of this mapreduce is to create a final output file. The output file contains the proper format as sourceURL and the rank of the sourceURL.

**Compile and run your code**

**Usage:**

./compileAndExecHadoopPageRank.sh [PageRank Input File][Number of Urls][Number Of Iterations]

$ ./compileAndExecHadoopPageRank.sh PageRankDataGenerator/pagerank5000g50.input.0 5000 10

## Parallel MapReduce

## 

**Output**

The output file is submitted as HadoopPageRank/output/part-r-0000 file

The top 10 sorted URLs are as follows –

4 0.1206985439094576

34 0.10776863798237155

0 0.09651067430015867

20 0.07730804787564192

2 0.03690483528905177

146 0.03517334419309333

3424 0.030985948953611255

14 0.016459328448425656

16 0.011379122999302936

12 0.010968028739305422