Design Discussion

PageRank Algorithm pseudo code-

map(key, values)
{
delta = value rertieved from context
totalNoOfPages = value retrieved from context
alpha = 0.15
for each value in values
- Split value to get pageName, outlinkNames, pageRank

if pageRank = -1.0
// setting value of pageRank=-1.0 after pre-processing. For first iteration pageRank is →
- pageRank = 1/totalNoOfPages

else
- pageRank = pageRank // from value

if delta != 0 // dangling nodes found
- pageRank += (1- alpha) * (delta/totalNoOfPages)

if outlinkName = empty
- emit (“dummy”, pageRank)

else
- emit (“outlinkName”, pageRank)

- obj = outlinkNames, pageRank/outlinkList.size()

emit(pageName, obj)
}

reduce(key, values)
{
alpha = 0.15
totalNoOfPages = value retrieved from context
- if key = “dummy”
  - for each value in values
    - s += pageRank
    - delta set to context

- else
    - for each value in values
        - Split value to get pageName, outlinkNames, pageRank
        - pageRank = (alpha/totalNoOfPages) + (1-alpha) * (s)

obj = outlinkNames + pageRank

emit (key, obj)
}

I. Pre-processing of my program followed the exact same steps as expected by the professor. Apart from basic parsing, I also removed duplicate self nodes and duplicate outlink values.

II. The types of PageRank problems given were -
Type 1: Has a separate phase to compute δ, passes to MR job which updates pagerank. This is iterating over 10 times which is an overhead, hence not the most optimal solution.

Type 2: Old values of dangling nodes is sent to each reducer which is lot of transfer between map and reduce.

Type 3: Here my reducer receives all old ranks of dangling nodes. It computes δ which is sent to map of next iteration which updates pageranks. The amount of data transfer is not as significant as 2 or doesn't need separate method to calculate δ. Yes, there is an extra overhead but it is only of map phase for 10 iterations. Hence seems to be an optimal approach.

I have followed the pseudo code to a large extent. Couple of changes incorporated are for resolving dangling node issue-

- For all nodes obtained in map phase when delta (obtained from previous reduce call) is not zero, I distribute the delta to all pages and update pagerank
- When dangling node is found I emit dummy with pagerank value which is sent to one reducer since key is same and aggregated to get total dangling nodes
- For reduce phase, I calculate total number of dangling nodes and add the delta obtained to all pages.

- Instead of running iterations 10 times, I have to run it 10.5 times i.e. 1 extra map phase to ensure last delta calculated is updated to all pages.

III. I have used the the same pseudo code as provided by professor for Top-100

Iteration	**Map → Reduce (in bytes)**	**Reduce → S3 (in Bytes)**
5 Slaves		
Pre- Processing	1104110485	1133403455
PageRank 1	1531939796	1181367841
PageRank 2	1998795979	1181370277
PageRank 3	2000703659	1181360345
PageRank 4	2000519733	1181351279
PageRank 5	2000489429	1181352390
PageRank 6	2000552554	1181344203
PageRank 7	2000652946	1181358263
PageRank 8	2000228981	1181351819
PageRank 9	2000678192	1181351914
PageRank 10	2000475443	1181356043
Delta	774384787	1183113061
Top - 100	56948	3115
10 Slaves		
Pre- Processing	1127796816	1133403455
PageRank 1	1568359221	1181351887
PageRank 2	2044012687	1181350500
PageRank 3	2045386134	1181342612

PageRank 4	2045165934	1181337087
PageRank 5	2046019747	1181345005
PageRank 6	2045829908	1181344362
PageRank 7	2045818099	1181338460
PageRank 8	2045889040	1181331066
PageRank 9	2045949497	1181338192
PageRank 10	2045822497	1181333185
Delta	787359753	1183096676
Top - 100	61291	3106

By and large the amount of data remains constant with several mapreduce iterations which is the expectation since total number of pages are same hence data transfer remains same amount. I

Performance Comparison

	Preprocessing (mm:ss)	PageRank (mm:ss)	Top-K (mm:ss)
10 Slaves	19:03	17:18	0:39
5 Slaves	34:34	28:18	0:50

The amount of time necessary to run when number of slaves increase reduces significantly which is what our expectation is. As an observation, speedup is maximum for pre-processing and least top-K. Reason for the same could be because there is very little computation which needs to be done in Top-K over Pre-Processing. Top-K ends up having only 1 reducer, hence it does not matter to a large extent if number of slaves are 5 or 10.
Difference between pre-processing and pagerank speed up is because pagerank uses more of data transfer between mapreduce calls which is seen through the shuffled records which are significantly greater than pre-processing. Hence time is used more in data transfer (on different systems) than computations on a single system

Wiktionary and Wikimedia are in top page ranks. These are non-profit organizations which are giving out a lot of data. They have links to one another which technically is a self link. This is not a good representation of page-rank since it should depend on external inlinks and outlinks.

Top - 100 Page Ranks for EMR (10 slaves)

United_States_09d4	0.0026229336
2006	0.0012285069
United_Kingdom_5ad7	0.0012031491
Biography	9.82E-04
2005	9.17E-04
England	8.80E-04
Canada	8.56E-04
Geographic_coordinate_system	7.72E-04
France	7.25E-04
2004	7.20E-04
Australia	6.80E-04
Germany	6.54E-04
2003	5.87E-04
India	5.83E-04
Japan	5.83E-04
Internet_Movie_Database_7ea7	5.34E-04
Europe	5.09E-04
Record_label	4.91E-04
2001	4.87E-04

2002	4.83E-04
World_War_II_d045	4.78E-04
Population_density	4.70E-04
Music_genre	4.67E-04
2000	4.65E-04
Italy	4.46E-04
Wiktionary	4.36E-04
Wikimedia_Commons_7b57	4.35E-04
London	4.35E-04
English_language	4.18E-04
1999	4.06E-04
Spain	3.63E-04
1998	3.56E-04
Russia	3.44E-04
1997	3.37E-04
Television	3.36E-04
New_York_City_1428	3.35E-04
Football_(soccer)	3.26E-04
1996	3.24E-04
Census	3.24E-04
Scotland	3.22E-04
1995	3.10E-04
China	3.09E-04
Population	3.04E-04

Square_mile	3.04E-04
Scientific_classification	3.04E-04
California	3.02E-04
1994	2.91E-04
Sweden	2.88E-04
Public_domain	2.87E-04
Film	2.86E-04
Record_producer	2.84E-04
New_Zealand_2311	2.83E-04
New_York_3da4	2.79E-04
Netherlands	2.77E-04
Marriage	2.76E-04
1993	2.75E-04
United_States_Census_Bureau_2c85	2.75E-04
1991	2.72E-04
1990	2.68E-04
1992	2.66E-04
Politician	2.65E-04
Album	2.61E-04
Latin	2.60E-04
Actor	2.58E-04
Ireland	2.58E-04
Per_capita_income	2.56E-04
Studio_album	2.52E-04

Poverty_line	2.51E-04
Km²	2.50E-04
1989	2.47E-04
Norway	2.41E-04
Website	2.39E-04
1980	2.35E-04
Animal	2.29E-04
Area	2.29E-04
1986	2.27E-04
Personal_name	2.26E-04
Poland	2.26E-04
Brazil	2.26E-04
1985	2.24E-04
1987	2.23E-04
1983	2.22E-04
1982	2.21E-04
1981	2.19E-04
French_language	2.19E-04
1979	2.19E-04
1984	2.19E-04
World_War_I_9429	2.19E-04
1988	2.19E-04
Paris	2.18E-04
1974	2.18E-04

Mexico	2.16E-04
19th_century	2.12E-04
1970	2.11E-04
January_1	2.11E-04
USA_f75d	2.11E-04
1975	2.09E-04
1976	2.08E-04
Africa	2.08E-04
South_Africa_1287	2.07E-04

Top - K for Local Run

United_States_09d4	0.005189009
Wikimedia_Commons_7b57	0.0048067665
Country	0.0039402847
England	0.0027524814
Water	0.0026878096
Animal	0.0025540876
City	0.0025108241
United_Kingdom_5ad7	0.0023586471
Germany	0.0023504017
Earth	0.0023247349
France	0.0023236079

Europe	0.002038097
Wiktionary	0.0017538842
English_language	0.0017496771
Government	0.0017323447
Computer	0.0017168405
India	0.0017131709
Money	0.0016673837
Japan	0.0015516906
Plant	0.0015235595
Italy	0.0015074331
Canada	0.0014814073
Spain	0.0014711237
Food	0.0014246868
Human	0.001412097
China	0.0013967151
People	0.0013822485
Australia	0.0013298542
Asia	0.0012844362
Capital_(city)	0.0012742684
Television	0.0012649972
Sun	0.0012602101
Number	0.0012432362
State	0.0012403757
Sound	0.0012352117

Science	0.0012325432
Mathematics	0.0012310566
Metal	0.0011923046
Year	0.0011770926
2004	0.0011733573
Language	0.0011501659
Russia	0.0011461818
Wikipedia	0.0011233303
Religion	0.0010985667
19th_century	0.0010965391
Music	0.0010874313
Scotland	0.0010548007
20th_century	0.001053705
Greece	0.0010492227
Latin	0.0010298606
London	0.0010273554
Greek_language	0.0010043573
Energy	9.99E-04
World	9.86E-04
Centuries	9.76E-04
Culture	9.45E-04
History	9.36E-04
Liquid	9.15E-04
Netherlands	9.06E-04

Planet	9.05E-04
Light	9.02E-04
Society	9.01E-04
Atom	8.90E-04
Wikimedia_Foundation_83d9	8.88E-04
Scientist	8.88E-04
Image	8.88E-04
Law	8.86E-04
Geography	8.79E-04
List_of_decades	8.79E-04
Uniform_Resource_Locator_1b4e	8.62E-04
Africa	8.61E-04
Turkey	8.45E-04
Inhabitant	8.30E-04
Capital_city	8.23E-04
Plural	8.22E-04
Electricity	8.14E-04
Poland	7.97E-04
Building	7.97E-04
Car	7.95E-04
Sweden	7.92E-04
Book	7.91E-04
Biology	7.87E-04
War	7.71E-04

Chemical_element	7.68E-04
God	7.61E-04
North_America_e7c4	7.56E-04
September_7	7.55E-04
Website	7.46E-04
Nation	7.43E-04
Politics	7.40E-04
2006	7.33E-04
Fish	7.32E-04
Species	7.31E-04
Mammal	7.22E-04
Island	7.18E-04
Portugal	7.17E-04
Gas	7.16E-04
River	7.12E-04
Switzerland	7.06E-04
World_War_II_d045	7.02E-04