Report - Assignment 3

Name: Chandrika Sharma
Git-hub: https://github.ccs.neu.edu/chandrika2311/MapReduce-CS-6240
HW = #3

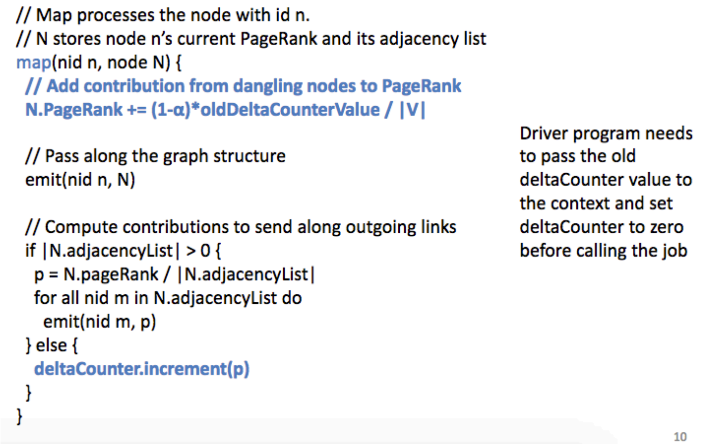
Design Discussion:

//Job1:

1) Pre-Processing:

```
Map (Object, value){
Parser(Object)
// Used Hash Set instead of lists used in Parser
//Added code to change all occurrences of "&" by "&amp;"

if (the node has a non-empty adjacency list):
        loop over adjacency list:
                //Handling the case where a page might point to a page which is not
                  present in the collection- dangling node case 2
                emit (adjacency-list-link-name, empty adjacency-list)

        emit (Page-Name, Adjacency-List)
//Dangling node case 1
else:
        emit (Page-Names, empty-Adjacency-List )
}
//---------------------------------------------------------------------------------------------------
reduce (page-name, adjacency-list){
        emit (Page-Name, Adjacency-List)
//To track the number of nodes in total
        Number-Of-Nodes-in-Graph-Counter ++
}

//---------------------------------------------------------------------------------------------------

//Following runs in a map only job after the first Job1(Mentioned in assignment) finishes.
Map2(Object, value) {
        Initial-Page-Rank = 1/ Number-Of-Nodes-in-Graph-Counter
        Loop over all the records in the file:
                append the value of (Initial-Page-Rank)

}
```

2)Page Ranking:
Map: Professor’s Slide for Module 6 in Extra Material.

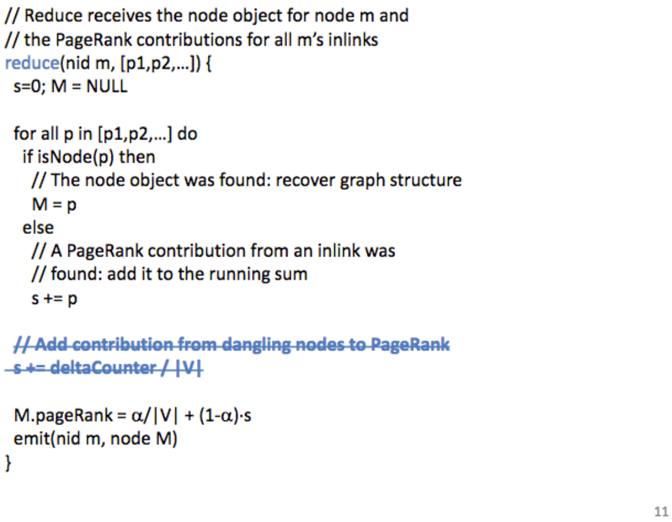


```
// Map processes the node with id n.
// N stores node n’s current PageRank and its adjacency list
map(nid n, node N) {
  // Add contribution from dangling nodes to PageRank
  N.PageRank += (1-α)*oldDeltaCounterValue / |V|

  // Pass along the graph structure
  emit(nid n, N)

  // Compute contributions to send along outgoing links
  if |N.adjacencyList| > 0 {
    p = N.pageRank / |N.adjacencyList|
    for all nid m in N.adjacencyList do
      emit(nid m, p)
  } else {
    deltaCounter.increment(p)
  }
}
```

Driver program needs
to pass the old
deltaCounter value to
the context and set
deltaCounter to zero
before calling the job

10

Reduce: Professor’s Slide for Module 6 in Extra Material.



```
// Reduce receives the node object for node m and
// the PageRank contributions for all m’s inlinks
reduce(nid m, [p1,p2,…]) {
  s=0; M = NULL

  for all p in [p1,p2,…] do
    if isNode(p) then
      // The node object was found: recover graph structure
      M = p
    else
      // A PageRank contribution from an inlink was
      // found: add it to the running sum
      s += p

  ~~// Add contribution from dangling nodes to PageRank~~
  ~~s += deltaCounter / |V|~~

  M.pageRank = α/|V| + (1-α)·s
  emit(nid m, node M)
}
```

11

3)Top-K Computation: Refer to Module 5 for Top-K computation with Order Inversion.

To analyze if the values are converging we can see the difference in the delta counter value from one iteration to another. As iterations increase, the delta in the delta counter values stabilizes.

Performance Comparison:

11 m4large machines time taken	6 m4 large machines time taken	Speedup
Pre-processing		
1223292	2023520	1.739039385
Page Ranking		
1489918	877633	1.69
Top 100		
22429	35489	1.582281867

Speedup on increasing the workers from 6 to 11 is approximately 1.7
I expected a speedup of slightly less than 2 and was able to see the same.
All the processes show a good speedup in my computation of page rank.

Data Transferred:

6 m4. Large machines	
Map-Reducer bytes	Reducers-HDFS
Map output bytes=3516733382	Bytes Written=1154446369
Map output bytes=3518704422	Bytes Written=1154395643
Map output bytes=3520664722	Bytes Written=1154393456
Map output bytes=3520042494	Bytes Written=1154382227
Map output bytes=3521894515	Bytes Written=1154381343
Map output bytes=3521365298	Bytes Written=1154361050
Map output bytes=3520751954	Bytes Written=1154348773
Map output bytes=3521204504	Bytes Written=1154369279
Map output bytes=3520573539	Bytes Written=1154363756
Map output bytes=3519351478	Bytes Written=1154359513

11 m4. Large machines	
Map-Reducer bytes	Reducers-HDFS
Map output bytes=3516733382	Bytes Written=1154446906
Map output bytes=3518674341	Bytes Written=1154399470
Map output bytes=3520488356	Bytes Written=1154381344

Map output bytes=3521343521	Bytes Written=1154380286
Map output bytes=3520758925	Bytes Written=1154381386
Map output bytes=3518722651	Bytes Written=1154376471
Map output bytes=3521105451	Bytes Written=1154366839
Map output bytes=3521271348	Bytes Written=1154376892
Map output bytes=3521959631	Bytes Written=1154360314
Map output bytes=3520696827	Bytes Written=1154350195

Data Transfer numbers: Ideally the number of bytes written from Map to reduce should be same in each of the 10 iterations, I am getting different values for them as I have used String as transferring bytes, I am appending the page rank to the string in each iteration. Since the page rank values are changing in each iteration there is a minor difference in the bytes transferred.

Are the data transfer bytes changing on changing the number of workers?
No, there is no significant changes in the bytes transferring from map-reduce or reduce-hdfs on increasing the number of workers

The outputs for both the local and the remote runs on the simple and big data seem correct as the pages like United_States_09d4 and Wikimedia_Commons_7b57 have appeared to have the most page ranks and they are the pages. These are pages which are referred to by man many pages in wiki documents.

DETAILED CALCULATIONS FROM ALL THE INDIVIDUAL FILES IN THE LOGS:

Description	**11 m4. Large machines(ms)**			
Preprocessing time	Launch Time	Finish Time	Time Taken(ms)	TOTAL TIMES(ms)
Preprocessing				
Time(ms)	1.51958E+12	1.51958E+12	1188072	1223292
Time(ms)	1.51958E+12	1.51958E+12	35220	
Page Rank				
1st	1.51958E+12	1.51958E+12	81223	877633
2nd	1.51958E+12	1.51958E+12	109818	
3rd	1.51958E+12	1.51958E+12	88691	
4th	1.51958E+12	1.51958E+12	87532	
5th	1.51958E+12	1.51958E+12	83904	
6th	1.51958E+12	1.51958E+12	81182	
7th	1.51958E+12	1.51958E+12	83554	
8th	1.51958E+12	1.51958E+12	84227	
9th	1.51958E+12	1.51958E+12	88535	
10th	1.51958E+12	1.51958E+12	88967	
top-100 pages				
	1.51958E+12	1.51958E+12	22429	22429

Description	6 m4. large machines(ms)			
Preprocessing time	Launch Time	Finish Time	Time Taken(ms)	TOTAL TIME(ms)
Preprocessing				
1st	1.51959E+12	1.51959E+12	2066104	2023520
2nd	1.51959E+12	1.51959E+12	57416	
Page Rank				
1st	1.51959E+12	1.51959E+12	147191	1489918
2nd	1.51959E+12	1.51959E+12	149552	
3rd	1.51959E+12	1.51959E+12	149349	
4th	1.51959E+12	1.51959E+12	147578	
5th	1.51959E+12	1.51959E+12	149141	
6th	1.51959E+12	1.51959E+12	150304	
7th	1.51959E+12	1.51959E+12	147744	
8th	1.51959E+12	1.51959E+12	150353	

9th	1.51959E+12	1.51959E+12	147977	
10th	1.51959E+12	1.51959E+12	150729	
top-100 pages				
	1.51959E+12	1.51959E+12	35489	35489