Fast Convergence Map Reduce

Group Members:

Batu Inal (bi49)
Gaurav Keswani (gk368)
Tanvi Mehta (tmm259)

0. Compiling the Java Project and Running it on EMR:

We tested if our project compiles using javac and need the utils package to be compiled before any of the other packages. We have included the sample command we used to compile the project for your reference:
`

Format:
javac -cp <':' separated path to all jars> <path to utils package> <' ' separated path to every package>

We have included the jars we have used in the top level directory under 'jars' and have included an example command that can be run from the top level directory (proj2) of the project:

Example Command:

```
javac -cp jars/commons-cli-1.3.1.jar:jars/log4j-1.2.17.jar:jars/hadoop-common-2.7.2.jar:jars/hadoop-mapreduce-client-core-2.7.2.jar src/utils/*.java src/blocked_pagerank/*.java src/simple_pagerank/*.java src/gaussseidel_pagerank/*.java
```

Example Command's for EMR:

1) Jacobi Page Rank: `src/blocked_pagerank/BlockPageRank s3n://cs5300s16-gk368-tmm259-emr/input/Blocked_PageRank.txt s3n://cs5300s16-gk368-tmm259-emr/output/`

2) Gauss Seidel Page Rank: `src/gaussseidel_pagerank/GaussSeidelBlockPageRank s3n://cs5300s16-gk368-tmm259-emr/input/Blocked_PageRank.txt s3n://cs5300s16-gk368-tmm259-emr/output/`

where the 2 arguments are - the pre processed file and the temporary directory where we store the outputs in between map reduce passes

NOTE: The output folder needs to be empty when the EMR is run since there might be a clash in the temporary files names that are generated.

1. Input File Generation

Source File: https://s3.amazonaws.com/cs5300s16-bi49-tmm259-gk368-p2/edges.txt

1) Put edges.txt in the '/Data' directory along with the python files
   - Use simple_parser.py file for Simple Page Rank
   - Use blocked_parser.py file for Blocked_PageRank
2) Run $ python <python_file>.py
3) We then store the filtered vertices list in 94_edges.txt and the pre processed data in a file named Simple_PageRank.txt or Blocked_PageRank.txt depending on which python file was run

2. Filtered Data Statistics

0) netid:bi49 rejectMin: 0.846 rejectLimit: 0.856
1) Total Edges in New Set: 7524549 (Total: 7600595)
2) Filtered In Percentage: 0.989994730676

3. Input File Format

3.1 Simple PageRank MapReduce Tasks

Source File: https://s3.amazonaws.com/cs5300s16-bi49-tmm259-gk368-p2/input/Simple_PageRank.txt

For the Simple PageRank computation, we have a file called Simple_PageRank.txt which is created from edges.txt via simple_parser.py. It outputs a new format of <source_node> <dest_node> <PageRank Value> <Number of Outgoing Edges> where delimiter is space ' '.

3.2 Blocked PageRank MapReduce Tasks

Source File: https://s3.amazonaws.com/cs5300s16-bi49-tmm259-gk368-p2/Blocked_PageRank.txt

For the Blocked PageRank computation, we have a file called Blocked_PageRank.txt which is created from edges.txt via blocked_parser.py. It outputs a new format of <src_node>$<PageRank Value>$[comma separated list of out going edges] where delimiter is '$'

3.3 Gauss Seidel PageRank MapReduce Tasks

Source File: https://s3.amazonaws.com/cs5300s16-bi49-tmm259-gk368-p2/Blocked_PageRank.txt

For the Gauss Seidel PageRank computation, we have the same file and input file format as the blocked page rank described above.

4. Simple PageRank

Our Simple PageRank utilizes the counters so runs in 5 MapReduce tasks. Our Main, Mapper and Reducer Classes can be summarized as below:

File Format: <src_node> <dest_node> <PR_val> <number_of_outgoing_edges>

1. Source File: Simple_PageRank.java

Main: Sets our mapper and reducer classes as NodeMapper and NodeResidualReducer respectively. Sets output key, value as LongWritable and Text. Iterates for 5 map-reduce tasks, creates a folder output name with respect to the second argument provided (in e.g if its ../output it would create folders output0, output1 etc.). Our main runs for 5 loops only as we utilize the global counter, by collecting each nodes residual from their reducer and dividing it by the number of total nodes in the system, to find the average (along with a normalization constant of 10^8 to maintain precision of floating number, as counters are only types of Long). The class finally displays the average change in residual across 685230 nodes in the system.

2. Source File: NodeMapper.java

Mapper: Has key value of LongWritable, Text and spits out LongWritable, Text for the reducer. Parses the input file for the source node, destination node, page-rank value and number of outgoing edges for each line of the file. For each of the destination nodes it sends the calculated PR value from the current source node to the destination node (by dividing current page rank value by number of outgoing edges) with a special initial tag of "OTHERS". For each of the destination nodes it also sends metadata to itself to be able to generate the file for the next round. This message starts with the tag "ME" and has the metadata of the current PageRank Value, number of outgoing edges from this node and the destination node sending this information to the current source node.

3. Source File: NodeResidualReducer.java

Reducer: The reducer takes input key as a LongWritable referring to node id and values as list of Text. The reducer goes through the list of Text, and parses the text according to it's initial tag which was assigned as either "ME" or "OTHERS" by the mapper. If it's the "OTHERS" tag, it means the incoming message is the incoming page rank value from another node for this round, so it sums up all these values to be able to calculate it's new page rank value. If it's the "ME" tag, it is information that is to be used to generate the next input file. So, it creates a list of all the nodes the current node sends page rank values to, gets the previous page rank value (for residual change calculations) and the number of out-going edges from this node. The new page rank value is calculate according to the Surfer Model by summing up; 15 percent of the time we jump to a random node(0.15/685230), otherwise; 85 percent of the time its the PageRank value we collected(0.85*received_PR_value) from incoming edges. We calculate our residual by finding the difference between our old and new page rank values and dividing this value by our new calculated page rank value. We also utilize the global counters to be able to report from our main class the average change in residual across all nodes. This is done by sending the residual value, first by de-normalizing it (by multiplying it with a factor of 10^8) so that we can have the float/double precision we need. The last step generates the file for the last round.

5. Jacobi Blocked PageRank

Our Blocked PageRank utilizes the counters and converges in 6 map reducer iterations. Our Main, Mapper and Reducer Classes can be summarized as below:

Source File: BlockPageRank.java

Main: The main class creates the map reduce jobs and sets the mapper and reducer to BPRMapper.java and BPRReducer.java respectively. It continues to generate map reduce jobs until the average residual errors across all reduce tasks is less than the specified threshold (0.001). The main method uses counter RESIDUAL_ERROR to keep track of the residual errors across reduce tasks.
Furthermore, the main method keeps track of the number of iterations for each block and displays this information and the average iterations for all blocks after the job is complete and converged.

Source File: BPRMapper.java

Mapper: The mapper expects input to be in the format nodeID$pageRank$edges (where edges is a comma separated list of edges from the nodeID). The mapper breaks this string up and then checks gets the blockID for the source nodeID.

We emit a message with the PR identifier for the source node with its blockID as key and PR$nodeID$pageRank$outgoing_edges as the value.

Furthermore, for each edge in the list of edges, we check if the edge is part of the same block as its source. If so, we emit a message with BE identifier for the source node with the edge's blockID as key and BE$sourceID$destinationID as value. If the edge is not in the block as its source, we emit a message with the BC identifier with the edge's blockID as key and BC$sourceID$destinationID$PR_factor as value.

Source File: BPRReducer.java

Reducer: The reducer gets messages from all the nodes that belong to a particular blockID. If the message has a PR identifier, we add the node to a list of nodes that belong to this block. If the message has a BE identifier, we add this node to the blockEdges list and in case of a BC identifier, we add the node and its corresponding page rank factor to a map called boundaryConditions.

The nodes are iterated on and the page ranks are updated until either the residual error is less than the threshold OR the number of iterations is more than 20. Each iteration through the nodes calculated the new page rank for each node by taking into account its blockEdges and boundary conditions. The residual error is calculated by getting the difference between the oldPR and newPR and dividing it by the newPR. **The oldPR in this case is always the page rank from the last iteration over the block.**

Then the reducer emits messages for each node with its updated page rank.

6. Gauss Seidel PageRank (Extra Credit: 7.1)

Our Gauss Seidel PageRank utilizes the counters and converges in 7 map reducer iterations. Our Main, Mapper and Reducer Classes can be summarized as below:

Source File: GaussSeidelPageRank.java: Same as BlockPageRank.java

Source File: GaussSeidelReducer.java

The logic for the reducer is similar to Jacobi's except for the following two changes:

- For the page rank calculation of node v, with an incoming edge from u, we use $PR^{(t+1)}(u)$ if available, else we use $PR^{(t)}(u)$ compared to Jacobi which always used $PR^{(t)}(u)$
- The calculation of the internal residual error uses the PR values of the last iteration rather than using the original values of the current reducer run as in the case of Jacobi.

7. Random Partitioning (Extra Credit: 7.2)

We implemented the random partitioning by having a flag in the getBlockFromNodeID method where if the flag is set to true, instead of partitioning the blocks according to METIS, we return random blocks for the nodes using Java's internet hashcode() method. It was evident that the convergence was much slower than the METIS partitioning and we had to manually stop the execution after 11 map reduce passed. The results for the same have been attached below in the results section.

8. Results

8.1 Simple PageRank

Average Residual Errors in Each MapReduce Pass

```
    0) Average Residual Error's Updated Value is: 2.3270748
        - Node: 0 PageRank Value: 1.5114821464219157E-5
        - Node: 1 PageRank Value: 5.267111562630828E-5
    1) Average Residual Error's Updated Value is: 0.32670537
        - Node: 0 PageRank Value: 6.447241803516533E-6
        - Node: 1 PageRank Value: 4.979968738336481E-5
    2) Average Residual Error's Updated Value is: 0.19784908
        - Node: 0 PageRank Value: 2.9470947991453282E-5
        - Node: 1 PageRank Value: 6.455414771622583E-5
    3) Average Residual Error's Updated Value is: 0.10070821
        - Node: 0 PageRank Value: 1.7665185465037863E-5
        - Node: 1 PageRank Value: 6.058341533818806E-5
    4) Average Residual Error's Updated Value is: 0.069692
        - Node: 0 PageRank Value: 1.6466880835062683E-5
        - Node: 1 PageRank Value: 6.0120305704489935E-5
```

8.2 Jacobi Blocked PageRank

Average Residual Errors in Each MapReduce Pass

```
    0) Average residual error after iteration 0: 2.806003712352941
    1) Average residual error after iteration 1: 0.038160
    2) Average residual error after iteration 2: 0.023614
    3) Average residual error after iteration 3: 0.009774
    4) Average residual error after iteration 4: 0.003799
    5) Average residual error after iteration 5: 0.000974
```

Average number of iterations per block

```
    Average of Block 0: 5.833 iterations
    Average of Block 1: 6.167 iterations
    Average of Block 2: 9.833 iterations
    Average of Block 3: 5.333 iterations
    Average of Block 4: 6.667 iterations
    Average of Block 5: 7.833 iterations
    Average of Block 6: 5.833 iterations
    Average of Block 7: 5.667 iterations
    Average of Block 8: 6.833 iterations
    Average of Block 9: 6.000 iterations
    Average of Block 10: 9.500 iterations
    Average of Block 11: 6.667 iterations
```

```
Average of Block 12: 4.667 iterations
Average of Block 13: 6.000 iterations
Average of Block 14: 2.667 iterations
Average of Block 15: 3.000 iterations
Average of Block 16: 2.667 iterations
Average of Block 17: 6.167 iterations
Average of Block 18: 6.500 iterations
Average of Block 19: 7.500 iterations
Average of Block 20: 10.167 iterations
Average of Block 21: 6.333 iterations
Average of Block 22: 6.500 iterations
Average of Block 23: 4.833 iterations
Average of Block 24: 4.667 iterations
Average of Block 25: 6.333 iterations
Average of Block 26: 6.500 iterations
Average of Block 27: 4.667 iterations
Average of Block 28: 8.167 iterations
Average of Block 29: 10.833 iterations
Average of Block 30: 11.000 iterations
Average of Block 31: 10.833 iterations
Average of Block 32: 14.000 iterations
Average of Block 33: 9.000 iterations
Average of Block 34: 6.500 iterations
Average of Block 35: 6.833 iterations
Average of Block 36: 5.500 iterations
Average of Block 37: 7.667 iterations
Average of Block 38: 6.167 iterations
Average of Block 39: 4.500 iterations
Average of Block 40: 5.333 iterations
Average of Block 41: 5.667 iterations
Average of Block 42: 7.500 iterations
Average of Block 43: 7.000 iterations
Average of Block 44: 3.500 iterations
Average of Block 45: 4.667 iterations
Average of Block 46: 6.333 iterations
Average of Block 47: 6.167 iterations
Average of Block 48: 6.333 iterations
Average of Block 49: 11.000 iterations
Average of Block 50: 5.333 iterations
Average of Block 51: 4.167 iterations
Average of Block 52: 9.333 iterations
Average of Block 53: 6.167 iterations
Average of Block 54: 4.333 iterations
Average of Block 55: 3.667 iterations
Average of Block 56: 3.333 iterations
Average of Block 57: 7.333 iterations
Average of Block 58: 4.333 iterations
Average of Block 59: 4.000 iterations
Average of Block 60: 5.167 iterations
Average of Block 61: 5.167 iterations
Average of Block 62: 4.833 iterations
Average of Block 63: 4.833 iterations
Average of Block 64: 4.833 iterations
Average of Block 65: 3.167 iterations
Average of Block 66: 5.667 iterations
Average of Block 67: 4.500 iterations
```

Total average iterations for all blocks using Jacobi: 6.264705882352941

Page Rank values for 1st 2 nodes of each block

```
Block: 0  Node ID: 0        Page Rank: 1.4501976620847241E-5
Block: 0  Node ID: 1        Page Rank: 5.304988327015483E-5
Block: 1  Node ID: 10328    Page Rank: 4.335996368731741E-7
Block: 1  Node ID: 10329    Page Rank: 2.434979690420696E-7
Block: 2  Node ID: 20373    Page Rank: 2.1890460137472089E-7
Block: 2  Node ID: 20374    Page Rank: 2.1890460137472089E-7
Block: 3  Node ID: 30629    Page Rank: 2.640904435172594E-7
Block: 3  Node ID: 30630    Page Rank: 2.1890460137472089E-7
Block: 4  Node ID: 40645    Page Rank: 2.196810711016847E-5
Block: 4  Node ID: 40646    Page Rank: 2.196810711016847E-5
Block: 5  Node ID: 50462    Page Rank: 0.005275099083757872
Block: 5  Node ID: 50463    Page Rank: 0.005405415123097596
Block: 6  Node ID: 60841    Page Rank: 1.1596771093409895E-5
Block: 6  Node ID: 60842    Page Rank: 3.037398067297396E-6
Block: 7  Node ID: 70591    Page Rank: 2.196810711016847E-5
Block: 7  Node ID: 70592    Page Rank: 2.196810711016847E-5
Block: 8  Node ID: 80119    Page Rank: 9.869337490665268E-7
Block: 8  Node ID: 80120    Page Rank: 8.131089311003053E-7
Block: 9  Node ID: 90497    Page Rank: 9.120241436381751E-7
Block: 9  Node ID: 90498    Page Rank: 1.1525029552936084E-5
Block: 10 Node ID: 100501 Page Rank: 1.0205406348191643E-6
Block: 10 Node ID: 100502 Page Rank: 1.2418792487553472E-6
Block: 11 Node ID: 110567 Page Rank: 0.0010396194525063383
Block: 11 Node ID: 110568 Page Rank: 9.919193091272177E-5
Block: 12 Node ID: 120945 Page Rank: 8.307805323883279E-7
Block: 12 Node ID: 120946 Page Rank: 6.908492903925219E-7
Block: 13 Node ID: 130999 Page Rank: 2.287235965730013E-7
Block: 13 Node ID: 131000 Page Rank: 2.287235965730013E-7
Block: 14 Node ID: 140574 Page Rank: 0.001077193299134865
```

```
Block: 14 Node ID: 140575 Page Rank: 2.284770330971243E-7
Block: 15 Node ID: 150953 Page Rank: 0.0010211590355305976
Block: 15 Node ID: 150954 Page Rank: 0.0021743861513896213
Block: 16 Node ID: 161332 Page Rank: 2.2855318884396009E-7
Block: 16 Node ID: 161333 Page Rank: 2.2855318884396009E-7
Block: 17 Node ID: 171154 Page Rank: 2.2913728279051707E-7
Block: 17 Node ID: 171155 Page Rank: 2.2913728279051707E-7
Block: 18 Node ID: 181514 Page Rank: 2.243351457729522E-7
Block: 18 Node ID: 181515 Page Rank: 7.439638278820458E-4
Block: 19 Node ID: 191625 Page Rank: 2.8411445048157644E-4
Block: 19 Node ID: 191626 Page Rank: 2.1890460137472089E-7
Block: 20 Node ID: 202004 Page Rank: 9.306676236193025E-7
Block: 20 Node ID: 202005 Page Rank: 9.869906188223314E-7
Block: 21 Node ID: 212383 Page Rank: 0.0012588017532741552
Block: 21 Node ID: 212384 Page Rank: 2.1890460137472089E-7
Block: 22 Node ID: 222762 Page Rank: 2.2866523638471948E-7
Block: 22 Node ID: 222763 Page Rank: 2.2866523638471948E-7
Block: 23 Node ID: 232593 Page Rank: 2.375365289754462E-7
Block: 23 Node ID: 232594 Page Rank: 2.7304686261131964E-7
Block: 24 Node ID: 242878 Page Rank: 1.0668835123188052E-6
Block: 24 Node ID: 242879 Page Rank: 6.04600483532989E-7
Block: 25 Node ID: 252938 Page Rank: 5.974164666019328E-6
Block: 25 Node ID: 252939 Page Rank: 5.791544228401522E-7
Block: 26 Node ID: 263149 Page Rank: 1.634935961717381E-5
Block: 26 Node ID: 263150 Page Rank: 2.0548216977075466E-6
Block: 27 Node ID: 273210 Page Rank: 2.7911792882284616E-5
Block: 27 Node ID: 273211 Page Rank: 2.4800604217906782E-6
Block: 28 Node ID: 283492 Page Rank: 2.888562448931899E-7
Block: 28 Node ID: 283493 Page Rank: 1.6459267462281303E-6
Block: 29 Node ID: 293255 Page Rank: 2.7188645773866672E-6
Block: 29 Node ID: 293256 Page Rank: 1.1879905246906389E-5
Block: 30 Node ID: 303043 Page Rank: 1.0663251510698539E-6
Block: 30 Node ID: 303044 Page Rank: 5.69222498307594E-6
Block: 31 Node ID: 313370 Page Rank: 2.5489449669483937E-7
Block: 31 Node ID: 313371 Page Rank: 2.3620165628958788E-5
Block: 32 Node ID: 323522 Page Rank: 5.680689456521207E-7
Block: 32 Node ID: 323523 Page Rank: 1.9353377707744125E-6
Block: 33 Node ID: 333883 Page Rank: 2.029147251484531E-5
Block: 33 Node ID: 333884 Page Rank: 4.470138773166331E-6
Block: 34 Node ID: 343663 Page Rank: 2.305122426090921E-6
Block: 34 Node ID: 343664 Page Rank: 3.65208248880505E-7
Block: 35 Node ID: 353646 Page Rank: 9.501756169587024E-7
Block: 35 Node ID: 353647 Page Rank: 4.349352047603532E-6
Block: 36 Node ID: 363929 Page Rank: 1.4341831377757323E-5
Block: 36 Node ID: 363930 Page Rank: 4.6631381699461374E-7
```

```
Block: 37 Node ID: 374236 Page Rank: 2.3922929314199124E-7
Block: 37 Node ID: 374237 Page Rank: 9.239766141969E-6
Block: 38 Node ID: 384554 Page Rank: 6.439945419388612E-5
Block: 38 Node ID: 384555 Page Rank: 5.191493953007018E-6
Block: 39 Node ID: 394929 Page Rank: 4.156326681721551E-6
Block: 39 Node ID: 394930 Page Rank: 2.5768285486480347E-6
Block: 40 Node ID: 404712 Page Rank: 7.067209909300036E-7
Block: 40 Node ID: 404713 Page Rank: 8.130217019372084E-7
Block: 41 Node ID: 414617 Page Rank: 3.7004729544201937E-7
Block: 41 Node ID: 414618 Page Rank: 3.7004729544201937E-7
Block: 42 Node ID: 424747 Page Rank: 1.939713903234723E-6
Block: 42 Node ID: 424748 Page Rank: 4.384596433774146E-6
Block: 43 Node ID: 434707 Page Rank: 5.206327911971795E-7
Block: 43 Node ID: 434708 Page Rank: 3.2744380473563057E-6
Block: 44 Node ID: 444489 Page Rank: 4.6570084330838154E-7
Block: 44 Node ID: 444490 Page Rank: 2.39738603497882E-7
Block: 45 Node ID: 454285 Page Rank: 3.52491595718365E-7
Block: 45 Node ID: 454286 Page Rank: 3.1863897659284026E-7
Block: 46 Node ID: 464398 Page Rank: 4.61386842360818E-7
Block: 46 Node ID: 464399 Page Rank: 4.61386842360818E-7
Block: 47 Node ID: 474196 Page Rank: 5.314476401712595E-7
Block: 47 Node ID: 474197 Page Rank: 8.415449894584053E-7
Block: 48 Node ID: 484050 Page Rank: 1.6528024939317757E-5
Block: 48 Node ID: 484051 Page Rank: 1.2433581165133424E-5
Block: 49 Node ID: 493968 Page Rank: 7.963571935557455E-7
Block: 49 Node ID: 493969 Page Rank: 1.3892482698481373E-6
Block: 50 Node ID: 503752 Page Rank: 7.909787420081567E-7
Block: 50 Node ID: 503753 Page Rank: 7.909787420081567E-7
Block: 51 Node ID: 514131 Page Rank: 0.0011160251402292317
Block: 51 Node ID: 514132 Page Rank: 2.6672426609245564E-5
Block: 52 Node ID: 524510 Page Rank: 8.605976340386568E-4
Block: 52 Node ID: 524511 Page Rank: 6.512111894525243E-5
Block: 53 Node ID: 534709 Page Rank: 0.009138915151778039
Block: 53 Node ID: 534710 Page Rank: 4.536463399365877E-6
Block: 54 Node ID: 545088 Page Rank: 0.0017749830453795683
Block: 54 Node ID: 545089 Page Rank: 3.397687134778407E-5
Block: 55 Node ID: 555467 Page Rank: 0.0018386815309452452
Block: 55 Node ID: 555468 Page Rank: 7.557064071015638E-7
Block: 56 Node ID: 565846 Page Rank: 2.1890460137472089E-7
Block: 56 Node ID: 565847 Page Rank: 2.1890460137472089E-7
Block: 57 Node ID: 576225 Page Rank: 1.1532032116006564E-5
Block: 57 Node ID: 576226 Page Rank: 2.1890460137472089E-7
Block: 58 Node ID: 586604 Page Rank: 3.906874508350674E-4
Block: 58 Node ID: 586605 Page Rank: 4.2889635180936506E-7
Block: 59 Node ID: 596585 Page Rank: 1.3315659953126192E-6
```

```
Block: 59 Node ID: 596586 Page Rank: 1.3878058043161398E-6
Block: 60 Node ID: 606367 Page Rank: 3.166709959851151E-7
Block: 60 Node ID: 606368 Page Rank: 2.2492825528646362E-7
Block: 61 Node ID: 616148 Page Rank: 9.830758777624129E-7
Block: 61 Node ID: 616149 Page Rank: 4.312231139098447E-7
Block: 62 Node ID: 626448 Page Rank: 2.1890460137472089E-7
Block: 62 Node ID: 626449 Page Rank: 2.5839554187923826E-7
Block: 63 Node ID: 636240 Page Rank: 2.286815160382686E-7
Block: 63 Node ID: 636241 Page Rank: 2.287235965730013E-7
Block: 64 Node ID: 646022 Page Rank: 5.682029671406778E-7
Block: 64 Node ID: 646023 Page Rank: 4.1093847629995806E-7
Block: 65 Node ID: 655804 Page Rank: 2.286815160382686E-7
Block: 65 Node ID: 655805 Page Rank: 2.1890460137472089E-7
Block: 66 Node ID: 665666 Page Rank: 2.5611838360842345E-7
Block: 66 Node ID: 665667 Page Rank: 4.0142481981186924E-7
Block: 67 Node ID: 675448 Page Rank: 2.1890460137472089E-7
Block: 67 Node ID: 675449 Page Rank: 4.905029211237358E-6
```

8.3 Gauss Seidel Blocked PageRank (Extra Credit: 7.1)

Average Residual Errors in Each MapReduce Pass:

```
0) Average residual error after iteration 0: 2.807087
1) Average residual error after iteration 1: 0.038735
2) Average residual error after iteration 2: 0.024959
3) Average residual error after iteration 3: 0.010955
4) Average residual error after iteration 4: 0.004942
5) Average residual error after iteration 5: 0.001870
6) Average residual error after iteration 6: 0.000808
```

Average number of iterations per block:

```
Average of Block 0: 3.429 iterations
Average of Block 1: 3.714 iterations
Average of Block 2: 7.143 iterations
Average of Block 3: 3.714 iterations
Average of Block 4: 4.571 iterations
Average of Block 5: 5.000 iterations
Average of Block 6: 4.000 iterations
Average of Block 7: 4.143 iterations
Average of Block 8: 4.429 iterations
Average of Block 9: 3.857 iterations
Average of Block 10: 6.143 iterations
Average of Block 11: 4.571 iterations
Average of Block 12: 3.143 iterations
Average of Block 13: 3.571 iterations
Average of Block 14: 2.143 iterations
Average of Block 15: 2.286 iterations
Average of Block 16: 2.143 iterations
Average of Block 17: 3.714 iterations
Average of Block 18: 4.000 iterations
Average of Block 19: 4.857 iterations
Average of Block 20: 6.714 iterations
Average of Block 21: 4.429 iterations
Average of Block 22: 4.429 iterations
Average of Block 23: 2.857 iterations
Average of Block 24: 2.714 iterations
Average of Block 25: 3.857 iterations
Average of Block 26: 3.714 iterations
Average of Block 27: 3.286 iterations
Average of Block 28: 4.714 iterations
Average of Block 29: 7.000 iterations
Average of Block 30: 6.286 iterations
Average of Block 31: 6.714 iterations
Average of Block 32: 8.429 iterations
Average of Block 33: 5.429 iterations
Average of Block 34: 3.714 iterations
Average of Block 35: 3.429 iterations
Average of Block 36: 2.857 iterations
Average of Block 37: 4.143 iterations
Average of Block 38: 3.286 iterations
Average of Block 39: 2.714 iterations
Average of Block 40: 3.143 iterations
Average of Block 41: 3.429 iterations
Average of Block 42: 4.857 iterations
Average of Block 43: 4.429 iterations
Average of Block 44: 2.571 iterations
Average of Block 45: 3.143 iterations
Average of Block 46: 3.857 iterations
Average of Block 47: 3.714 iterations
Average of Block 48: 3.857 iterations
Average of Block 49: 6.143 iterations
Average of Block 50: 3.571 iterations
Average of Block 51: 2.714 iterations
Average of Block 52: 6.429 iterations
Average of Block 53: 4.000 iterations
Average of Block 54: 2.857 iterations
Average of Block 55: 2.429 iterations
Average of Block 56: 2.143 iterations
Average of Block 57: 5.714 iterations
Average of Block 58: 3.286 iterations
Average of Block 59: 3.143 iterations
Average of Block 60: 3.857 iterations
Average of Block 61: 3.714 iterations
Average of Block 62: 3.000 iterations
Average of Block 63: 3.000 iterations
Average of Block 64: 3.286 iterations
Average of Block 65: 2.286 iterations
Average of Block 66: 3.571 iterations
Average of Block 67: 2.571 iterations
```

Total average iterations for all blocks using Gauss Seidel: 4.0
Thus, we can see that although Gauss Seidel took one extra map reduce pass than Jacobi, the average number of iterations in each block was 4.0 compared to Jacobi's 6.26

Page Rank values for 1st 2 nodes of each block

```
Block: 0  Node ID: 0          Page Rank: 1.7418820697488106E-5
Block: 0  Node ID: 1          Page Rank: 5.368988766652773E-5
Block: 1  Node ID: 10328     Page Rank: 4.3345194914898106E-7
Block: 1  Node ID: 10329     Page Rank: 2.434668784931631E-7
Block: 2  Node ID: 20373     Page Rank: 2.1890460137472089E-7
Block: 2  Node ID: 20374     Page Rank: 2.1890460137472089E-7
Block: 3  Node ID: 30629     Page Rank: 2.640717992316986E-7
Block: 3  Node ID: 30630     Page Rank: 2.1890460137472089E-7
Block: 4  Node ID: 40645     Page Rank: 2.4764551628535827E-5
Block: 4  Node ID: 40646     Page Rank: 2.4764551628535827E-5
Block: 5  Node ID: 50462     Page Rank: 0.004906257060053242
Block: 5  Node ID: 50463     Page Rank: 0.005003030010759752
Block: 6  Node ID: 60841     Page Rank: 1.2276011942130702E-5
Block: 6  Node ID: 60842     Page Rank: 3.0206296531885608E-6
Block: 7  Node ID: 70591     Page Rank: 2.4764551628535827E-5
Block: 7  Node ID: 70592     Page Rank: 2.4764551628535827E-5
Block: 8  Node ID: 80119     Page Rank: 9.880210968785311E-7
Block: 8  Node ID: 80120     Page Rank: 8.140047685544295E-7
Block: 9  Node ID: 90497     Page Rank: 9.119837132268266E-7
Block: 9  Node ID: 90498     Page Rank: 1.1518316258715307E-5
Block: 10 Node ID: 100501 Page Rank: 1.0191821506892705E-6
Block: 10 Node ID: 100502 Page Rank: 1.240220497744371E-6
Block: 11 Node ID: 110567 Page Rank: 0.0010492458566464407
Block: 11 Node ID: 110568 Page Rank: 9.508000345312261E-5
Block: 12 Node ID: 120945 Page Rank: 8.293756260949714E-7
Block: 12 Node ID: 120946 Page Rank: 6.89347728846674E-7
Block: 13 Node ID: 130999 Page Rank: 2.2872338869916999E-7
Block: 13 Node ID: 131000 Page Rank: 2.2872338869916999E-7
Block: 14 Node ID: 140574 Page Rank: 0.0010483584088687022
```

```
Block: 14 Node ID: 140575 Page Rank: 2.284768591382912E-7
Block: 15 Node ID: 150953 Page Rank: 9.985101933938764E-4
Block: 15 Node ID: 150954 Page Rank: 0.001874182154397404
Block: 16 Node ID: 161332 Page Rank: 2.2855236804758796E-7
Block: 16 Node ID: 161333 Page Rank: 2.2855236804758796E-7
Block: 17 Node ID: 171154 Page Rank: 2.2913676014364196E-7
Block: 17 Node ID: 171155 Page Rank: 2.2913676014364196E-7
Block: 18 Node ID: 181514 Page Rank: 2.2433024011131508E-7
Block: 18 Node ID: 181515 Page Rank: 7.439687358462343E-4
Block: 19 Node ID: 191625 Page Rank: 2.5026025084721977E-4
Block: 19 Node ID: 191626 Page Rank: 2.1890460137472089E-7
Block: 20 Node ID: 202004 Page Rank: 9.277155243600278E-7
Block: 20 Node ID: 202005 Page Rank: 9.837518283133615E-7
Block: 21 Node ID: 212383 Page Rank: 0.001155693362718149
Block: 21 Node ID: 212384 Page Rank: 2.1890460137472089E-7
Block: 22 Node ID: 222762 Page Rank: 2.2867415193583587E-7
Block: 22 Node ID: 222763 Page Rank: 2.2867415193583587E-7
Block: 23 Node ID: 232593 Page Rank: 2.374120727653179E-7
Block: 23 Node ID: 232594 Page Rank: 2.7313089319310654E-7
Block: 24 Node ID: 242878 Page Rank: 1.0808251907661102E-6
Block: 24 Node ID: 242879 Page Rank: 6.027722973457049E-7
Block: 25 Node ID: 252938 Page Rank: 6.0893757602767145E-6
Block: 25 Node ID: 252939 Page Rank: 5.791483050608226E-7
Block: 26 Node ID: 263149 Page Rank: 1.0167005371841199E-5
Block: 26 Node ID: 263150 Page Rank: 2.6181821466083933E-6
Block: 27 Node ID: 273210 Page Rank: 2.7901823524925153E-5
Block: 27 Node ID: 273211 Page Rank: 2.478146638056596E-6
Block: 28 Node ID: 283492 Page Rank: 2.879261613332309E-7
Block: 28 Node ID: 283493 Page Rank: 1.6275080764587365E-6
Block: 29 Node ID: 293255 Page Rank: 2.6708051901489244E-6
Block: 29 Node ID: 293256 Page Rank: 1.1885322029259977E-5
Block: 30 Node ID: 303043 Page Rank: 9.681611469629746E-7
Block: 30 Node ID: 303044 Page Rank: 5.402617203081908E-6
Block: 31 Node ID: 313370 Page Rank: 2.549046826091592E-7
Block: 31 Node ID: 313371 Page Rank: 2.3172000195308E-5
Block: 32 Node ID: 323522 Page Rank: 5.751948779015665E-7
Block: 32 Node ID: 323523 Page Rank: 1.732524320694183E-6
Block: 33 Node ID: 333883 Page Rank: 2.0152592934679934E-5
Block: 33 Node ID: 333884 Page Rank: 4.4685167464471005E-6
Block: 34 Node ID: 343663 Page Rank: 2.320607134167965E-6
Block: 34 Node ID: 343664 Page Rank: 3.665650647075475E-7
Block: 35 Node ID: 353646 Page Rank: 9.501532999879425E-7
Block: 35 Node ID: 353647 Page Rank: 4.345805957503835E-6
Block: 36 Node ID: 363929 Page Rank: 1.4993053668982614E-5
Block: 36 Node ID: 363930 Page Rank: 4.655230983228377E-7
```

```
Block: 37 Node ID: 374236 Page Rank: 2.392838134115712E-7
Block: 37 Node ID: 374237 Page Rank: 9.240257627687299E-6
Block: 38 Node ID: 384554 Page Rank: 6.441933395665749E-5
Block: 38 Node ID: 384555 Page Rank: 5.196762225298254E-6
Block: 39 Node ID: 394929 Page Rank: 4.178141810009121E-6
Block: 39 Node ID: 394930 Page Rank: 2.5792740037608845E-6
Block: 40 Node ID: 404712 Page Rank: 7.159434190385312E-7
Block: 40 Node ID: 404713 Page Rank: 8.274565075574724E-7
Block: 41 Node ID: 414617 Page Rank: 3.7004729540842604E-7
Block: 41 Node ID: 414618 Page Rank: 3.7004729540842604E-7
Block: 42 Node ID: 424747 Page Rank: 2.0820838369552297E-6
Block: 42 Node ID: 424748 Page Rank: 4.84441316159305E-6
Block: 43 Node ID: 434707 Page Rank: 5.293936013417592E-7
Block: 43 Node ID: 434708 Page Rank: 3.306576106678182E-6
Block: 44 Node ID: 444489 Page Rank: 4.696239245450406E-7
Block: 44 Node ID: 444490 Page Rank: 2.399140927359464E-7
Block: 45 Node ID: 454285 Page Rank: 3.52320597289748E-7
Block: 45 Node ID: 454286 Page Rank: 3.187287706068162E-7
Block: 46 Node ID: 464398 Page Rank: 5.400632950761645E-7
Block: 46 Node ID: 464399 Page Rank: 5.400632950761645E-7
Block: 47 Node ID: 474196 Page Rank: 5.352175098114307E-7
Block: 47 Node ID: 474197 Page Rank: 8.434260424816179E-7
Block: 48 Node ID: 484050 Page Rank: 1.6534554345449957E-5
Block: 48 Node ID: 484051 Page Rank: 1.2516882401460374E-5
Block: 49 Node ID: 493968 Page Rank: 8.186164286007328E-7
Block: 49 Node ID: 493969 Page Rank: 1.3894340244170792E-6
Block: 50 Node ID: 503752 Page Rank: 7.909706726560407E-7
Block: 50 Node ID: 503753 Page Rank: 7.909706712035854E-7
Block: 51 Node ID: 514131 Page Rank: 0.00111628914113486
Block: 51 Node ID: 514132 Page Rank: 2.709219400565983E-5
Block: 52 Node ID: 524510 Page Rank: 9.019809790281143E-4
Block: 52 Node ID: 524511 Page Rank: 6.808388487729795E-5
Block: 53 Node ID: 534709 Page Rank: 0.008829070486716903
Block: 53 Node ID: 534710 Page Rank: 4.105114940872206E-6
Block: 54 Node ID: 545088 Page Rank: 0.001782197231380873
Block: 54 Node ID: 545089 Page Rank: 3.388263008301343E-5
Block: 55 Node ID: 555467 Page Rank: 0.0017953104807013212
Block: 55 Node ID: 555468 Page Rank: 7.555917415288712E-7
Block: 56 Node ID: 565846 Page Rank: 2.1890460137472089E-7
Block: 56 Node ID: 565847 Page Rank: 2.1890460137472089E-7
Block: 57 Node ID: 576225 Page Rank: 1.1663281248755078E-5
Block: 57 Node ID: 576226 Page Rank: 2.1890460137472089E-7
Block: 58 Node ID: 586604 Page Rank: 3.9363819737011995E-4
Block: 58 Node ID: 586605 Page Rank: 4.1454811757877424E-7
Block: 59 Node ID: 596585 Page Rank: 1.332197711863217E-6
```

```
Block: 59 Node ID: 596586 Page Rank: 1.3867159790260577E-6
Block: 60 Node ID: 606367 Page Rank: 3.1568179784848956E-7
Block: 60 Node ID: 606368 Page Rank: 2.2486747977852568E-7
Block: 61 Node ID: 616148 Page Rank: 9.825119565210007E-7
Block: 61 Node ID: 616149 Page Rank: 4.3120886830125656E-7
Block: 62 Node ID: 626448 Page Rank: 2.1890460137472089E-7
Block: 62 Node ID: 626449 Page Rank: 2.583975154388138E-7
Block: 63 Node ID: 636240 Page Rank: 2.286813093588265E-7
Block: 63 Node ID: 636241 Page Rank: 2.2872338869916999E-7
Block: 64 Node ID: 646022 Page Rank: 5.682011831311591E-7
Block: 64 Node ID: 646023 Page Rank: 4.1093715500757444E-7
Block: 65 Node ID: 655804 Page Rank: 2.286813093588265E-7
Block: 65 Node ID: 655805 Page Rank: 2.1890460137472089E-7
Block: 66 Node ID: 665666 Page Rank: 2.5611838360842345E-7
Block: 66 Node ID: 665667 Page Rank: 4.019244991522316E-7
Block: 67 Node ID: 675448 Page Rank: 2.1890460137472089E-7
Block: 67 Node ID: 675449 Page Rank: 4.902819624188416E-6
```

8.4 Blocked PageRank With Random Partitioning (Extra Credit: 7.2)

Average Residual Errors in Each MapReduce Pass

```
0) Average residual error after iteration 0: 2.338398
1) Average residual error after iteration 1: 0.320930
2) Average residual error after iteration 2: 0.190345
3) Average residual error after iteration 3: 0.092883
4) Average residual error after iteration 4: 0.061320
5) Average residual error after iteration 5: 0.033192
6) Average residual error after iteration 6: 0.026360
7) Average residual error after iteration 7: 0.016122
8) Average residual error after iteration 8: 0.013767
9) Average residual error after iteration 9: 0.009408
10) Average residual error after iteration 10: 0.008029
```

The blocked page rank method with random partitioning did not converge even after 11 map reduce iterations and we had to manually stop the program.