

CS5300 Project2 Report

Xi He(xh243)
Yanjia Li(yl2493)
Long Ma(lm675)

Code Functionality & Structure

1. Filter file

The followings are the parameters in terms selecting unique dataset for the node graph

NetID: xh243

rejectMin: 0.3078

rejectLimit: 0.3178

total edges: 7600595

rejected edges: 76370

Number of edges actually selected: 7524225

The input file format: Each line→ NodeID PageRank_Value OutLinks1 OutLink2 ...

2. Simple Computation

(1) Overview

1. Jobs: MapReduce passes. 5 passes in total so it does not converge.
2. Map(read line by line from file)
 - a. In: <LineNumofFile(longWritable), LineText(Text)>
 - b. out: <NodeID(longWritable), Data(Text)>
3. Reduce(write line by line to file)
 - a. in: <NodeID(longWritable), List<Data>(Text)>
 - b. out: <NodeID(longWritable), NodeInfoLine(Text)>
 - c.

(2) Mapper

1. Mapper reads the data line by line
2. For each line of data/each node, Mapper sends two types of data (1) splitted PR value to each outlink node (2) and also send its own node info line(marked with flag so reducer can distinguish).

(3) Reducer

1. Reducer gets PR value flowing from other nodes for one node and its own node info line (distinguish the two type of data by reading flag in the string)
2. Reducer calculate new PR value for its own node and generate data input for next round. The output is exactly same as the input file in every round except the pagerank value is updated.
3. Increment the residual counter so the job can calculate the average residual.

Note: running the code requires to start from PageRank.java and it requires a argument which is a part of the input file location. In our case:

/Users/BboyKellen/Documents/workspace/project2/input/iter

3. Blocked Computation - Jacobi

(1) Overview

1. Jobs: MapReduce passes. AvgResidual<0.001 to break
2. Map(read line by line from file)
 - c. In: <LineNumofFile(longWritable), LineText(Text)>
 - d. out: <BlockID(longWritable), Data(Text)>
3. Reduce(write line by line to file)
 - d. in: <BlockID(longWritable), List<Data>(Text)>
 - e. out: <NodeID(longWritable), NodeInfoLine(Text)>

(2) Mapper

1. Read text file line by line.
2. Each line/node contains {NodeID, PR, Outlinks}
ex: "123 0.234 45 86 90"
3. For each line / each node
 - a. send entire line<flag, line> ex: "NodeInfo 123 0.234 45 86 90"
send to block u
 - b. send BE(edges in same block): {flag, source node id, destination node id,

source node degree} ex: "BE 32 43 3"

$u \rightarrow v$ (send to block of v)

- c. send BC(boundaries) {flag, destination node id, pageRank shared with it}.
ex: "BC 34 0.02 "

$u \rightarrow v$ (send to block of v)

(3) Reducer

1. Sort List<Data> for one block, put in the following HashMaps
 - a. **degree_map**<nodeID, degree>
 - b. **pageRank_map** && **new_pageRank_map** <nodeID, PR>
 - c. **boundary_map** <nodeID, List<PR_shared>>
 - d. **edge_map** <nodeID, List<inLinks> >
 - e. **Others**: forNextPass map<nodeID, NodeInfoText> && *startMap* (same as pageRank_map)
2. Run iterations (max 20 times)
 - a. **edge_map**: add PR shared to each desti node to new_pageRank_map
 - b. **boundary_map**: add PR shared to each desti node from other blocks
 - c. **new_pageRank_map**: $\text{sum} * 0.85 + 0.15 / \text{Num_OF_NODES}$
 - d. Update **pageRank_map**; calculate avgResidual<0.01?break;keep iterate
3. Write To file
 - a. Update Node line Info with new PR in forNextPass map. Write to file on the fly. The output is exactly same as the input file in every round except the pagerank value is updated.
 - b. Sum the residuals of all nodes in block (start-end)/end, add to counter. Job driver will calculate average residual after one map-reduce pass job to determine if it converges or not.

Note: running the code requires to start from PageRank_block.java and it requires a argument which is a part of the input file location. In our case:
/Users/BboyKellen/Documents/workspace/project2/input/iter

Results of Simple computation

The followings are the result from running in local computer and AWS EMR. The results from local and AWS are the same. We ran 5 passes and it's still far away from convergence

1. Run at local computer

Output from local computer:

The average residual of iter0 is 2.338917429928637

The average residual of iter1 is 0.32286126213096333

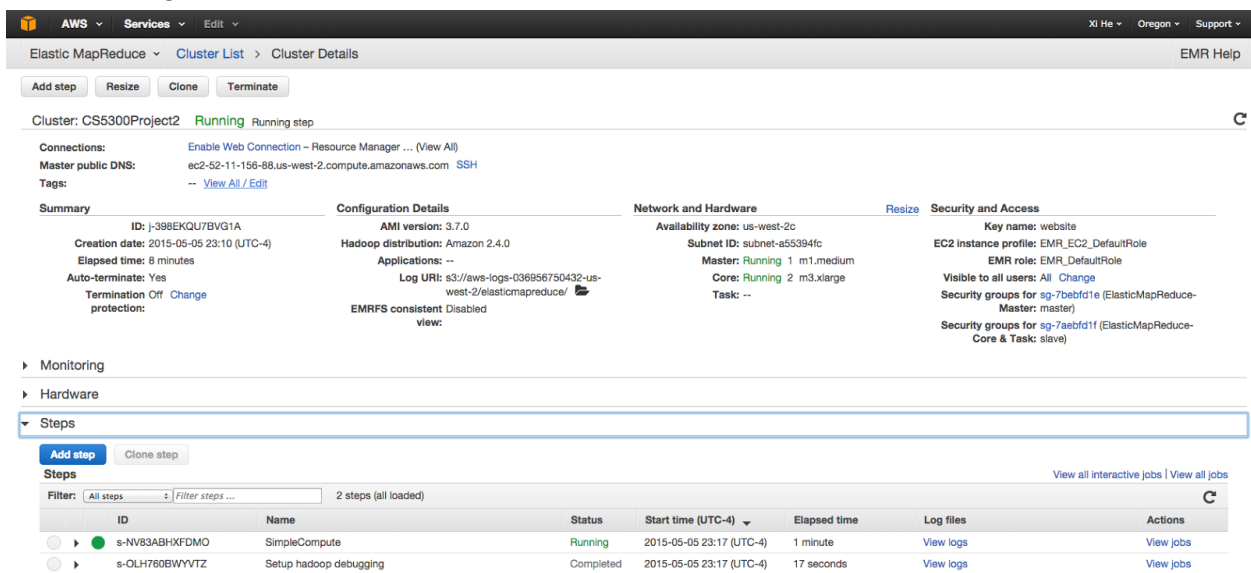
The average residual of iter2 is 0.19209467914422892

The average residual of iter3 is 0.09412065993899858

The average residual of iter4 is 0.06281029263167112

2. Run from Amazon AWS EMR

Using one master node and two slave nodes



The screenshot displays the AWS Management Console for an Elastic MapReduce cluster. The cluster name is CS5300Project2, and its state is Running. The console shows various configuration details, including the AMI version (3.7.0), Hadoop distribution (Amazon 2.4.0), and the number of master and core nodes (1 m1.medium master and 2 m3.xlarge cores). The 'Steps' section is expanded, showing a list of two steps: 'SimpleCompute' (Status: Running) and 'Setup hadoop debugging' (Status: Completed). The 'SimpleCompute' step is highlighted with a green dot, indicating it is currently running.

ID	Name	Status	Start time (UTC-4)	Elapsed time	Log files	Actions
s-NV83ABH-XFDMO	SimpleCompute	Running	2015-05-05 23:17 (UTC-4)	1 minute	View logs	View jobs
s-OLH760BWVYTZ	Setup hadoop debugging	Completed	2015-05-05 23:17 (UTC-4)	17 seconds	View logs	View jobs

Figure 1. Simple Computation Task Running

Cluster: CS5300Project2 Terminated Steps completed

Connections: --
Master public DNS: ec2-52-11-156-88.us-west-2.compute.amazonaws.com [SSH](#)
Tags: --

Summary	Configuration Details	Network and Hardware	Security and Access
ID: j-398EKQU7BVG1A Creation date: 2015-05-05 23:10 (UTC-4) End date: 2015-05-05 23:24 (UTC-4) Elapsed time: 14 minutes Auto-terminate: Yes Termination protection: Off	AMI version: 3.7.0 Hadoop distribution: Amazon 2.4.0 Applications: -- Log URI: s3://aws-logs-036956750432-us-west-2/elasticmapreduce/ EMRFS consistent view: Disabled	Availability zone: us-west-2c Subnet ID: subnet-a55394fc Master: Terminated 1 m1.medium Core: Terminated 2 m3.xlarge Task: --	Key name: website EC2 instance profile: EMR_EC2_DefaultRole EMR role: EMR_DefaultRole Visible to all users: All Change Security groups for sg-7bebfd1e (ElasticMapReduce-Master: master) Security groups for sg-7aebfd1f (ElasticMapReduce-Core & Task: slave)

► Monitoring

► Hardware

▼ Steps

[Add step](#) [Clone step](#)

Steps [View all interactive jobs](#) [View all jobs](#)

Filter: 1 Filter steps ... 2 steps (all loaded)

ID	Name	Status	Start time (UTC-4)	Elapsed time	Log files	Actions
s-NV83ABH-XFDMO	SimpleCompute	Completed	2015-05-05 23:17 (UTC-4)	5 minutes	View logs	View jobs
s-OLH760BWVYTZ	Setup hadoop debugging	Completed	2015-05-05 23:17 (UTC-4)	17 seconds	View logs	View jobs

Figure 2. Simple Computation Task Completed in 5 minutes

Output from EMR:

The average residual of iter0 is 2.338917430016199

The average residual of iter1 is 0.322861262145557

The average residual of iter2 is 0.19209467914422892

The average residual of iter3 is 0.09412065993899858

The average residual of iter4 is 0.06281029263167112

Results of the Blocked computation (Jacobi)

It takes 6 map-reduce passes to coverage(Jacobi). The followings are the result from running in local computer and AWS EMR. The results from local and AWS are exactly the same. The reducer max iteration limit is set to be **20**

1. Run at local computer

Average Residual:

The average residual of iter1 is **2.8138796845584695**

The average residual of iter2 is **0.03858919129343432**

The average residual of iter3 is **0.023914335347255664**

The average residual of iter4 is **0.009912790318579162**

The average residual of iter5 is **0.0038443647388468104**

The average residual of iter6 is **9.667192621455569E-4**

Average Iterations:

The average reducer iterations of pass 1 is 16.014705882352942

The average reducer iterations of pass 2 is 7.897058823529412

The average reducer iterations of pass 3 is 5.897058823529412

The average reducer iterations of pass 4 is 3.926470588235294

The average reducer iterations of pass 5 is 2.5

The average reducer iterations of pass 6 is 1.338235294117647

2. Run from Amazon AWS EMR

The screenshot displays the AWS EMR console for a cluster named 'CS5300Project2'. The cluster is in the 'Terminated' state, and all steps are completed. The console shows various configuration details, including the AMI version (3.7.0), Hadoop distribution (Amazon 2.4.0), and network settings. The 'Steps' tab is expanded, showing a list of steps with columns for ID, Name, Status, Start time, Elapsed time, Log files, and Actions. Two steps are listed: 's-1BJYNH80G8KJUS' (wc5) and 's-24VNB8B38UE10' (Setup hadoop debugging), both with a status of 'Completed'.

ID	Name	Status	Start time (UTC-4)	Elapsed time	Log files	Actions
s-1BJYNH80G8KJUS	wc5	Completed	2015-05-05 22:36 (UTC-4)	7 minutes	controller syslog stderr stdout	View jobs
s-24VNB8B38UE10	Setup hadoop debugging	Completed	2015-05-05 22:35 (UTC-4)	15 seconds	View logs	View jobs

Figure 3. Blocked Computation Task Completed in 7 minutes

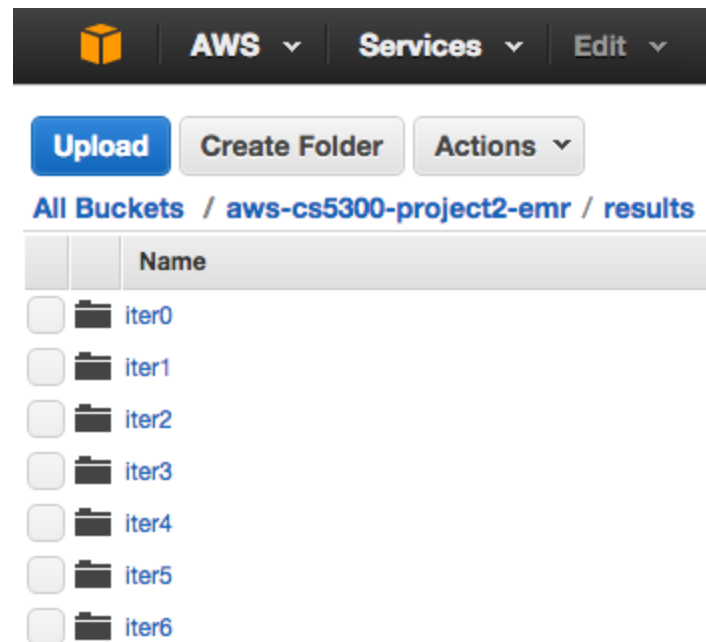


Figure 4. S3 iteration folders (6 pass)

Note: *iter0* is input folder, *iter1*- *iter6* are the output folders

Average Residual:

<Map_Reduce> The average residual of iter1 is 2.8138796845584695
 <Map_Reduce> The average residual of iter2 is 0.03858919129343432
 <Map_Reduce> The average residual of iter3 is 0.023914335347255664
 <Map_Reduce> The average residual of iter4 is 0.009912790318579162
 <Map_Reduce> The average residual of iter5 is 0.0038443647388468104
 <Map_Reduce> The average residual of iter6 is 9.667192621455569E-4

The following in the PageRank values for the two lowest-numbered Nodes in each Block for Jacobi method. This is when it converged which is after the 6th iteration.

Block 0: nodeID 0 PageRank val: 1.7523507706503166E-5
 Block 0: nodeID 1 PageRank val: 5.247302353180255E-5
 Block 1: nodeID 10328 PageRank val: 4.320536703455484E-7
 Block 1: nodeID 10329 PageRank val: 2.4343234927539884E-7
 Block 2: nodeID 20373 PageRank val: 2.1890460137472089E-7
 Block 2: nodeID 20374 PageRank val: 2.1890460137472089E-7
 Block 3: nodeID 30629 PageRank val: 2.65351916910355E-7
 Block 3: nodeID 30630 PageRank val: 2.1890460137472089E-7
 Block 4: nodeID 40645 PageRank val: 2.4785173536526804E-5
 Block 4: nodeID 40646 PageRank val: 2.4785173536526804E-5

Block 5: nodeID 50462 PageRank val: 0.004977779022854708
Block 5: nodeID 50463 PageRank val: 0.005057745184055878
Block 6: nodeID 60841 PageRank val: 1.2146979118951305E-5
Block 6: nodeID 60842 PageRank val: 3.108408989160212E-6
Block 7: nodeID 70591 PageRank val: 2.4785173536526804E-5
Block 7: nodeID 70592 PageRank val: 2.4785173536526804E-5
Block 8: nodeID 80118 PageRank val: 5.258927811254045E-7
Block 8: nodeID 80119 PageRank val: 9.729016450819983E-7
Block 9: nodeID 90497 PageRank val: 9.128502044150412E-7
Block 9: nodeID 90498 PageRank val: 1.1314892274058739E-5
Block 10: nodeID 100501 PageRank val: 1.0375565648745236E-6
Block 10: nodeID 100502 PageRank val: 1.2190843200423103E-6
Block 11: nodeID 110567 PageRank val: 0.0010554128069618766
Block 11: nodeID 110568 PageRank val: 9.644197267211728E-5
Block 12: nodeID 120945 PageRank val: 8.090945096697029E-7
Block 12: nodeID 120946 PageRank val: 6.691256160272805E-7
Block 13: nodeID 130999 PageRank val: 2.287043002976799E-7
Block 13: nodeID 131000 PageRank val: 2.287043002976799E-7
Block 14: nodeID 140574 PageRank val: 0.0010429502553185982
Block 14: nodeID 140575 PageRank val: 2.287043002976799E-7
Block 15: nodeID 150953 PageRank val: 9.950038995034244E-4
Block 15: nodeID 150954 PageRank val: 0.001796355564433476
Block 16: nodeID 161332 PageRank val: 2.2854548267780417E-7
Block 16: nodeID 161333 PageRank val: 2.2854548267780417E-7
Block 17: nodeID 171154 PageRank val: 2.2937950464802463E-7
Block 17: nodeID 171155 PageRank val: 2.2937950464802463E-7
Block 18: nodeID 181514 PageRank val: 2.245912327480114E-7
Block 18: nodeID 181515 PageRank val: 7.432491902512471E-4
Block 19: nodeID 191625 PageRank val: 2.587950552753508E-4
Block 19: nodeID 191626 PageRank val: 2.1890460137472089E-7
Block 20: nodeID 202004 PageRank val: 9.108084685489672E-7
Block 20: nodeID 202005 PageRank val: 9.659526838870594E-7
Block 21: nodeID 212383 PageRank val: 0.0011315697473620518
Block 21: nodeID 212384 PageRank val: 2.1890460137472089E-7
Block 22: nodeID 222762 PageRank val: 2.287301988613208E-7
Block 22: nodeID 222763 PageRank val: 2.287301988613208E-7
Block 23: nodeID 232593 PageRank val: 2.3668267144597314E-7
Block 23: nodeID 232594 PageRank val: 2.968574935685045E-7
Block 24: nodeID 242878 PageRank val: 1.087991708726062E-6
Block 24: nodeID 242879 PageRank val: 5.809708071903946E-7
Block 25: nodeID 252938 PageRank val: 5.971063617589379E-6
Block 25: nodeID 252939 PageRank val: 5.445771785664395E-7
Block 26: nodeID 263149 PageRank val: 9.764425522370657E-6
Block 26: nodeID 263150 PageRank val: 1.7950395696361192E-6
Block 27: nodeID 273210 PageRank val: 2.7561792566085423E-5
Block 27: nodeID 273211 PageRank val: 2.4476685721546168E-6
Block 28: nodeID 283473 PageRank val: 2.8760046520703846E-7
Block 28: nodeID 283474 PageRank val: 2.8760046520703846E-7
Block 29: nodeID 293255 PageRank val: 2.727146044193484E-6
Block 29: nodeID 293256 PageRank val: 1.1149665154001324E-5
Block 30: nodeID 303043 PageRank val: 9.767189507287338E-7
Block 30: nodeID 303044 PageRank val: 5.16957509354777E-6
Block 31: nodeID 313370 PageRank val: 2.549039249713656E-7
Block 31: nodeID 313371 PageRank val: 2.3423949100377943E-5
Block 32: nodeID 323522 PageRank val: 5.414059849650272E-7
Block 32: nodeID 323523 PageRank val: 1.7597247306889524E-6
Block 33: nodeID 333883 PageRank val: 1.7659301588113937E-5

Block 33: nodeID 333884 PageRank val: 4.067090383612633E-6
Block 34: nodeID 343663 PageRank val: 2.443580369096522E-6
Block 34: nodeID 343664 PageRank val: 3.664019253241096E-7
Block 35: nodeID 353645 PageRank val: 1.994186166918186E-6
Block 35: nodeID 353646 PageRank val: 9.25275723270379E-7
Block 36: nodeID 363929 PageRank val: 1.5068473807499254E-5
Block 36: nodeID 363930 PageRank val: 4.6063642198925323E-7
Block 37: nodeID 374236 PageRank val: 2.39331041910047E-7
Block 37: nodeID 374237 PageRank val: 9.282129689120674E-6
Block 38: nodeID 384554 PageRank val: 6.582568430548817E-5
Block 38: nodeID 384555 PageRank val: 5.294252279764473E-6
Block 39: nodeID 394929 PageRank val: 3.95704187266802E-6
Block 39: nodeID 394930 PageRank val: 2.5961696861641865E-6
Block 40: nodeID 404712 PageRank val: 6.892586413844494E-7
Block 40: nodeID 404713 PageRank val: 8.030218198680796E-7
Block 41: nodeID 414617 PageRank val: 4.883586314017715E-7
Block 41: nodeID 414618 PageRank val: 3.7004729544201926E-7
Block 42: nodeID 424747 PageRank val: 2.530613162138082E-6
Block 42: nodeID 424748 PageRank val: 3.5026288102322745E-6
Block 43: nodeID 434707 PageRank val: 5.281027023093166E-7
Block 43: nodeID 434708 PageRank val: 3.2808462434874405E-6
Block 44: nodeID 444489 PageRank val: 5.069018773975154E-7
Block 44: nodeID 444490 PageRank val: 2.4030404467089753E-7
Block 45: nodeID 454285 PageRank val: 3.5183009441460406E-7
Block 45: nodeID 454286 PageRank val: 3.186396811408006E-7
Block 46: nodeID 464398 PageRank val: 5.294432868591476E-7
Block 46: nodeID 464399 PageRank val: 5.294432868591476E-7
Block 47: nodeID 474196 PageRank val: 5.249324563858699E-7
Block 47: nodeID 474197 PageRank val: 8.997104298039208E-7
Block 48: nodeID 484050 PageRank val: 1.6334529172060845E-5
Block 48: nodeID 484051 PageRank val: 1.2353797504037491E-5
Block 49: nodeID 493968 PageRank val: 8.264413346754528E-7
Block 49: nodeID 493969 PageRank val: 1.3861962650324838E-6
Block 50: nodeID 503752 PageRank val: 7.909762200078046E-7
Block 50: nodeID 503753 PageRank val: 7.909762200078046E-7
Block 51: nodeID 514131 PageRank val: 0.0011086330010622722
Block 51: nodeID 514132 PageRank val: 2.6297314402703114E-5
Block 52: nodeID 524510 PageRank val: 8.902046034406036E-4
Block 52: nodeID 524511 PageRank val: 6.581354509556912E-5
Block 53: nodeID 534709 PageRank val: 0.009100999807326645
Block 53: nodeID 534710 PageRank val: 4.136521163106482E-6
Block 54: nodeID 545088 PageRank val: 0.0017604983452414321
Block 54: nodeID 545089 PageRank val: 3.256691020403494E-5
Block 55: nodeID 555467 PageRank val: 0.0018000826033106473
Block 55: nodeID 555468 PageRank val: 7.477788539966055E-7
Block 56: nodeID 565846 PageRank val: 2.1890460137472089E-7
Block 56: nodeID 565847 PageRank val: 2.1890460137472089E-7
Block 57: nodeID 576225 PageRank val: 1.1623800019976774E-5
Block 57: nodeID 576226 PageRank val: 2.1890460137472089E-7
Block 58: nodeID 586604 PageRank val: 3.905993033379477E-4
Block 58: nodeID 586605 PageRank val: 4.1579489769031345E-7
Block 59: nodeID 596585 PageRank val: 1.3328991029953962E-6
Block 59: nodeID 596586 PageRank val: 1.3861606038888671E-6
Block 60: nodeID 606367 PageRank val: 3.1553275902627255E-7
Block 60: nodeID 606368 PageRank val: 2.24754247314111227E-7
Block 61: nodeID 616148 PageRank val: 9.839328759630283E-7
Block 61: nodeID 616149 PageRank val: 4.282200854154952E-7

Block 62: nodeID 626448 PageRank val: 2.1890460137472089E-7
Block 62: nodeID 626449 PageRank val: 2.5808132575326E-7
Block 63: nodeID 636240 PageRank val: 2.2894929629334115E-7
Block 63: nodeID 636241 PageRank val: 2.287043002976799E-7
Block 64: nodeID 646022 PageRank val: 5.68202306226931E-7
Block 64: nodeID 646023 PageRank val: 4.109379868053116E-7
Block 65: nodeID 655804 PageRank val: 2.2894929629334115E-7
Block 65: nodeID 655805 PageRank val: 2.1890460137472089E-7
Block 66: nodeID 665666 PageRank val: 2.5611838360842345E-7
Block 66: nodeID 665667 PageRank val: 4.0855816343692305E-7
Block 67: nodeID 675448 PageRank val: 2.1890460137472089E-7
Block 67: nodeID 675449 PageRank val: 4.897094180747991E-6

7.1 Extra Credit (Jacobi vs Gauss-Seidel)

Jacobi was implemented in the previous section, the Gauss-Seidel is implemented here. The reducer iteration max limit is also set to be **20**. The pagerank hashmap is updated as soon as a node's new pagerank is calculated. There is only one hashmap holding pagerank value in Gauss-Seidel rather than a old pagerank map and a new pagerank map in Jacobi.

1. **Jacobi** (More reducer iterations but less mapreduce passes)

Average Residual:

The average residual of iter1 is **2.8138796845584695**

The average residual of iter2 is **0.03858919129343432**

The average residual of iter3 is **0.023914335347255664**

The average residual of iter4 is **0.009912790318579162**

The average residual of iter5 is **0.0038443647388468104**

The average residual of iter6 is **9.667192621455569E-4**

Average Iterations:

The average reducer iterations of pass 1 is 16.014705882352942

The average reducer iterations of pass 2 is 7.897058823529412

The average reducer iterations of pass 3 is 5.897058823529412

The average reducer iterations of pass 4 is 3.926470588235294

The average reducer iterations of pass 5 is 2.5

The average reducer iterations of pass 6 is 1.338235294117647

2. **Gauss-Seidel** (Less reducer iterations but more mapreduce passes)

Average Residual:

The average residual of iter1 is **2.8152492448374997**

The average residual of iter2 is **0.038988288983261093**

The average residual of iter3 is **0.02517357601097442**

The average residual of iter4 is **0.01106794108547495**

The average residual of iter5 is **0.004904624709951403**

The average residual of iter6 is **0.0020033862644659455**

The average residual of iter7 is **8.19530099382689E-4**

Average Iterations:

The average iterations of pass 1 is 9.632352941176471

The average iterations of pass 2 is 5.235294117647059

The average iterations of pass 3 is 4.529411764705882

The average iterations of pass 4 is 3.3088235294117645

The average iterations of pass 5 is 2.338235294117647

The average iterations of pass 6 is 1.7205882352941178

The average iterations of **pass 7** is 1.2647058823529411

Comparison

In terms of convergence, we can see the Jacobi method takes more iterations per round but end up with less mapreduce passes(6 passes) whereas Gauss-Seidel takes less iterations per round but end up with more mapreduce passes(7 passes). The overall running time of them are close.

The following in the PageRank values for the two lowest-numbered Nodes in each Block for Gauss-Seidel method. This is when it converged which is after the 7th iteration. It's pretty close to Jacobi's result.

```
Block 0: nodeID 0 PageRank val: 1.5782898801159986E-5
Block 0: nodeID 1 PageRank val: 5.2720914620846545E-5
Block 1: nodeID 10328 PageRank val: 4.325765143451551E-7
Block 1: nodeID 10329 PageRank val: 2.434155550586E-7
Block 2: nodeID 20373 PageRank val: 2.1890460137472089E-7
Block 2: nodeID 20374 PageRank val: 2.1890460137472089E-7
Block 3: nodeID 30629 PageRank val: 2.653383232668444E-7
Block 3: nodeID 30630 PageRank val: 2.1890460137472089E-7
Block 4: nodeID 40645 PageRank val: 2.2959135049179293E-5
Block 4: nodeID 40646 PageRank val: 2.2959135049179293E-5
Block 5: nodeID 50462 PageRank val: 0.0050159924185534475
Block 5: nodeID 50463 PageRank val: 0.005097633299781845
Block 6: nodeID 60841 PageRank val: 1.1923698367858318E-5
Block 6: nodeID 60842 PageRank val: 3.1397174942293887E-6
Block 7: nodeID 70591 PageRank val: 2.2959135049179293E-5
Block 7: nodeID 70592 PageRank val: 2.2959135049179293E-5
Block 8: nodeID 80118 PageRank val: 5.146996892463159E-7
Block 8: nodeID 80119 PageRank val: 9.521944251056844E-7
Block 9: nodeID 90497 PageRank val: 9.127883992613444E-7
Block 9: nodeID 90498 PageRank val: 1.1313592557558346E-5
Block 10: nodeID 100501 PageRank val: 1.0361223378696294E-6
Block 10: nodeID 100502 PageRank val: 1.2173641114169807E-6
Block 11: nodeID 110567 PageRank val: 0.0010516288831408007
Block 11: nodeID 110568 PageRank val: 9.6166780442772E-5
Block 12: nodeID 120945 PageRank val: 8.064977268954758E-7
Block 12: nodeID 120946 PageRank val: 6.669552254254303E-7
Block 13: nodeID 130999 PageRank val: 2.2870453920156296E-7
Block 13: nodeID 131000 PageRank val: 2.2870453920156296E-7
Block 14: nodeID 140574 PageRank val: 0.0010378671514086038
Block 14: nodeID 140575 PageRank val: 2.2870450879510112E-7
Block 15: nodeID 150953 PageRank val: 9.90043833201171E-4
Block 15: nodeID 150954 PageRank val: 0.0018571146408043549
Block 16: nodeID 161332 PageRank val: 2.285454338334325E-7
Block 16: nodeID 161333 PageRank val: 2.285454338334325E-7
Block 17: nodeID 171154 PageRank val: 2.293818643276435E-7
Block 17: nodeID 171155 PageRank val: 2.293818643276435E-7
Block 18: nodeID 181514 PageRank val: 2.2458510118274566E-7
```

Block 18: nodeID 181515 PageRank val: 7.432371592442118E-4
Block 19: nodeID 191625 PageRank val: 2.5842752196153734E-4
Block 19: nodeID 191626 PageRank val: 2.1890460137472089E-7
Block 20: nodeID 202004 PageRank val: 9.108074761498904E-7
Block 20: nodeID 202005 PageRank val: 9.659516212255917E-7
Block 21: nodeID 212383 PageRank val: 0.001124381916100436
Block 21: nodeID 212384 PageRank val: 2.1890460137472089E-7
Block 22: nodeID 222762 PageRank val: 2.2874753819214744E-7
Block 22: nodeID 222763 PageRank val: 2.2874753819214744E-7
Block 23: nodeID 232593 PageRank val: 2.3681562645069306E-7
Block 23: nodeID 232594 PageRank val: 2.9686136461521697E-7
Block 24: nodeID 242878 PageRank val: 1.0903299845028207E-6
Block 24: nodeID 242879 PageRank val: 5.8603143668696E-7
Block 25: nodeID 252938 PageRank val: 5.935475857585854E-6
Block 25: nodeID 252939 PageRank val: 5.440350531160757E-7
Block 26: nodeID 263149 PageRank val: 1.1553359418577865E-5
Block 26: nodeID 263150 PageRank val: 2.131079188833501E-6
Block 27: nodeID 273210 PageRank val: 2.7556667764519612E-5
Block 27: nodeID 273211 PageRank val: 2.4472036852167013E-6
Block 28: nodeID 283473 PageRank val: 2.867584488920419E-7
Block 28: nodeID 283474 PageRank val: 2.867584488920419E-7
Block 29: nodeID 293255 PageRank val: 2.84865094988016E-6
Block 29: nodeID 293256 PageRank val: 1.1313300376693858E-5
Block 30: nodeID 303043 PageRank val: 9.737606351245214E-7
Block 30: nodeID 303044 PageRank val: 5.1462314078678395E-6
Block 31: nodeID 313370 PageRank val: 2.5490442699304666E-7
Block 31: nodeID 313371 PageRank val: 2.351429535378025E-5
Block 32: nodeID 323522 PageRank val: 5.419417567077877E-7
Block 32: nodeID 323523 PageRank val: 1.8093897769044232E-6
Block 33: nodeID 333883 PageRank val: 1.7662389550747438E-5
Block 33: nodeID 333884 PageRank val: 4.058467178774166E-6
Block 34: nodeID 343663 PageRank val: 2.4287430622166933E-6
Block 34: nodeID 343664 PageRank val: 3.6510075533000286E-7
Block 35: nodeID 353645 PageRank val: 1.9982937693797115E-6
Block 35: nodeID 353646 PageRank val: 9.252465000334598E-7
Block 36: nodeID 363929 PageRank val: 1.535961063375759E-5
Block 36: nodeID 363930 PageRank val: 4.601054930244595E-7
Block 37: nodeID 374236 PageRank val: 2.393451808618723E-7
Block 37: nodeID 374237 PageRank val: 9.28216032308689E-6
Block 38: nodeID 384554 PageRank val: 6.594938286680731E-5
Block 38: nodeID 384555 PageRank val: 5.314993277446196E-6
Block 39: nodeID 394929 PageRank val: 3.978542246637196E-6
Block 39: nodeID 394930 PageRank val: 2.6099102180074173E-6
Block 40: nodeID 404712 PageRank val: 6.961205095461194E-7
Block 40: nodeID 404713 PageRank val: 8.099187266156739E-7
Block 41: nodeID 414617 PageRank val: 4.883586019164162E-7
Block 41: nodeID 414618 PageRank val: 3.7004729540842107E-7
Block 42: nodeID 424747 PageRank val: 2.7494193159329024E-6
Block 42: nodeID 424748 PageRank val: 3.7304014152940386E-6
Block 43: nodeID 434707 PageRank val: 5.312891058793679E-7
Block 43: nodeID 434708 PageRank val: 3.3157499085139157E-6
Block 44: nodeID 444489 PageRank val: 5.070707073729396E-7
Block 44: nodeID 444490 PageRank val: 2.404774057311931E-7
Block 45: nodeID 454285 PageRank val: 3.518852158019706E-7
Block 45: nodeID 454286 PageRank val: 3.1860541251861254E-7
Block 46: nodeID 464398 PageRank val: 4.996720868017624E-7
Block 46: nodeID 464399 PageRank val: 4.996720868017624E-7

Block 47: nodeID 474196 PageRank val: 5.278713752645789E-7
Block 47: nodeID 474197 PageRank val: 9.023194472690076E-7
Block 48: nodeID 484050 PageRank val: 1.6394231394245317E-5
Block 48: nodeID 484051 PageRank val: 1.2382296263468657E-5
Block 49: nodeID 493968 PageRank val: 8.498701134687434E-7
Block 49: nodeID 493969 PageRank val: 1.3859316141899654E-6
Block 50: nodeID 503752 PageRank val: 7.90970669706643E-7
Block 50: nodeID 503753 PageRank val: 7.909706690704464E-7
Block 51: nodeID 514131 PageRank val: 0.0011101722122781386
Block 51: nodeID 514132 PageRank val: 2.6701194633872852E-5
Block 52: nodeID 524510 PageRank val: 9.082105059461759E-4
Block 52: nodeID 524511 PageRank val: 6.843799745830406E-5
Block 53: nodeID 534709 PageRank val: 0.009196604641083865
Block 53: nodeID 534710 PageRank val: 4.3700265539087695E-6
Block 54: nodeID 545088 PageRank val: 0.001755234027054937
Block 54: nodeID 545089 PageRank val: 3.2652576840433335E-5
Block 55: nodeID 555467 PageRank val: 0.0018768147159274595
Block 55: nodeID 555468 PageRank val: 7.477211489451847E-7
Block 56: nodeID 565846 PageRank val: 2.1890460137472089E-7
Block 56: nodeID 565847 PageRank val: 2.1890460137472089E-7
Block 57: nodeID 576225 PageRank val: 1.1115525290103064E-5
Block 57: nodeID 576226 PageRank val: 2.1890460137472089E-7
Block 58: nodeID 586604 PageRank val: 3.928673781506888E-4
Block 58: nodeID 586605 PageRank val: 4.237879563285573E-7
Block 59: nodeID 596585 PageRank val: 1.3321100267573348E-6
Block 59: nodeID 596586 PageRank val: 1.386596127855315E-6
Block 60: nodeID 606367 PageRank val: 3.15322430776406E-7
Block 60: nodeID 606368 PageRank val: 2.2473121150863272E-7
Block 61: nodeID 616148 PageRank val: 9.834797036133579E-7
Block 61: nodeID 616149 PageRank val: 4.2841645720807876E-7
Block 62: nodeID 626448 PageRank val: 2.1890460137472089E-7
Block 62: nodeID 626449 PageRank val: 2.5808065260019963E-7
Block 63: nodeID 636240 PageRank val: 2.2894947581394387E-7
Block 63: nodeID 636241 PageRank val: 2.2870453920156296E-7
Block 64: nodeID 646022 PageRank val: 5.682011894539081E-7
Block 64: nodeID 646023 PageRank val: 4.109371624461026E-7
Block 65: nodeID 655804 PageRank val: 2.2894947581394387E-7
Block 65: nodeID 655805 PageRank val: 2.1890460137472089E-7
Block 66: nodeID 665666 PageRank val: 2.5611838360842345E-7
Block 66: nodeID 665667 PageRank val: 4.040346029043178E-7
Block 67: nodeID 675448 PageRank val: 2.1890460137472089E-7
Block 67: nodeID 675449 PageRank val: 4.903792949206062E-6

7.2 Extra Credit (Random Block Partition)

We are a group of 3 students so this part is not required for us.

By replacing the `getBlockID_ofNode` function in Jacobi block computation method in the following, we generate block partition in random instead. This way has high probability of being a bad partition with fairly uniform sized blocks.

```
static long getBlockID_ofNode(Long nodeID)
{
    return nodeID.hashCode() % GlobalVals.NUM_OF_BLOCKS;
}
```

The result of random block partition doesn't look good since the average residual of it is still fairly high within the first 5 rounds compared with Jacobi with METIS partition. The convergence is slow and will take many more passes before convergence.

Random Block Partition

The average residual of iter1 is **2.339524908133036**

The average residual of iter2 is **0.32230001109116646**

The average residual of iter3 is **0.191251995782438**

The average residual of iter4 is **0.09351859153860748**

The average residual of iter5 is **0.06199881518614188**

Jacobi with METIS partition

The average residual of iter1 is **2.8138796845584695**

The average residual of iter2 is **0.03858919129343432**

The average residual of iter3 is **0.023914335347255664**

The average residual of iter4 is **0.009912790318579162**

The average residual of iter5 is **0.0038443647388468104**

The average residual of iter6 is **9.667192621455569E-4**