CS 5300: Final Project Presentation

Horace Chan (hhc39), Kevin Chen (kfc35), Sweet Song (ss2249)

May 2, 2013

Agenda

Introduction

Solution Overview

Lingering Issues

Q & A

Problem Description

Fast Convergence PageRank in Hadoop

- compute PageRank for a reasonably large Web graph (685230 nodes, 7600595 edges)
- Blocked Matrix Multiplication instead of a direct node-by-node approach

Average Residual – Simple Page Rank

Iteration Number	Average Residual
0	2.338671828254877 7
1	0.3229799562618682 6
2	0.1918914214682807 3
3	0.0940770135730922 4
4	0.0628498339289581 7

Algorithm Overview

<nodeID, (pageRank numOuts (List: outNodes)>

↓ MAP

<nodeID, (incomingPR) OR (pageRank numOuts (List: outNodes)>

↓ REDUCE

<nodeID, (newPR numOuts (List: outNodes)>

Reducer receives either an incomingPR value (from a diff. node) to sum or "initial data”

Details

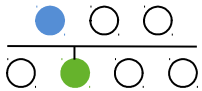
Mapper:

Parses the value and sends its rank to its outNodes or to the counter for no outNodes

Reducer:

Adds its residual to the counter (multiple the number by 10E12)

Output the files in the exact same format as mapper receives them



Average Residual – Block Page Rank

Iteration Number	Average Residual	Average Block Passes
0	2.816369003250558	20.0
1	0.03952213406068036	19.455882352941178
2	0.02515115893901318	14.411764705882353
3	0.011081515876333494	10.779411764705882
4	0.00476701952334252 7	8.08823529411764 7
5	0.001200520182011879 2	4.352941176470588
6	0.000635626823810983 1	2.676470588235294

Biggest Node in Block PageRank Values

Biggest Node in Block PageRank Values

TWENTY pagerank is 9.303777749E-4

TWENTY_ONE pagerank is 2.284187E-7

TWENTY_TWO pagerank is 2.379202E-7

TWENTY_THREE pagerank is 1.0786747E-6

TWENTY_FOUR pagerank is 5.7964084E-6

TWENTY_FIVE pagerank is 1.14296625E-5

TWENTY_SIX pagerank is 2.77469697E-5

TWENTY_SEVEN pagerank is 2.901975E-7

TWENTY_EIGHT pagerank is 2.3133899E-6

TWENTY_NINE pagerank is 9.623195E-7

THIRTY pagerank is 2.549672E-7

THIRTY_ONE pagerank is 5.626E-7

THIRTY_TWO pagerank is 1.91628686E-5

THIRTY_THREE pagerank is 2.2843913E-6

THIRTY_FOUR pagerank is 2.0010211E-6

THIRTY_FIVE pagerank is 1.44600731E-5

THIRTY_SIX pagerank is 2.393408E-7

THIRTY_SEVEN pagerank is 6.85389392E-5

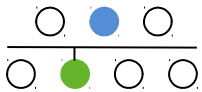
THIRTY_EIGHT pagerank is 4.0670731E-6

THIRTY_NINE pagerank is 7.15008E-7

Biggest Node in Block PageRank Values

FOURTY pagerank is 3.737095E-7	FIFTY pagerank is 0.0011249653912
FOURTY_ONE pagerank is 1.5324738E-6	FIFTY_ONE pagerank is 8.244486212E-4
FOURTY_TWO pagerank is 5.364979E-7	FIFTY_TWO pagerank is 0.0092611085217
FOURTY_THREE pagerank is 5.090524E-7	FIFTY_THREE pagerank is 0.0018054817275
FOURTY_FOUR pagerank is 3.513666E-7	FIFTY_FOUR pagerank is 0.0017413944329
FOURTY_FIVE pagerank is 5.215045E-7	FIFTY_FIVE pagerank is 2.189046E-7
FOURTY_SIX pagerank is 5.256554E-7	FIFTY_SIX pagerank is 1.13478455E-5
FOURTY_SEVEN pagerank is 1.63904432E-5	FIFTY_SEVEN pagerank is 3.907504533E-4
FOURTY_EIGHT pagerank is 7.767719E-7	FIFTY_EIGHT pagerank is 1.320183E-6
FOURTY_NINE pagerank is 8.198576E-7	FIFTY_NINE pagerank is 3.20079E-7

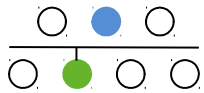
Block Page Rank



Solutions Overview

Biggest Node in Block PageRank Values

10327 1.8946085087207 83E-6
20372 5.1446551384154 15E-7
30628 3.0659370092288 054E-7
40644 2.8092757176422 52E-7
50461 2.9985814319868 72E-7
60840 2.1890460137472 094E-7
70590 3.0039254814045 086E-7
80117 2.1890460137472 094E-7
90496 8.6781473727114 81E-4
100500 2.6004293165929 734E-7
110566 2.0669036348835 56E-6
120944 4.2078945754737 91E-7
130998 2.3475209735695 868E-7
140573 6.2462763459647 1E-7
150952 2.1890460137472 094E-7
161331 2.1890460137472 094E-7
171153 5.6220502086367 44E-7
181513 3.5294555179394 39E-7
191624 5.0274474009116 57E-6
202003 4.1012316208535 67E-6
212382 2.6241452671835 857E-7
222761
0.0012296187423283694
232592 1.1623395590934 494E-4
242877 5.4693359769910 01E-7
252937 2.1890460137472 094E-7
263148 3.0489269801220 846E-7
273209 2.1890460137472 094E-7
283472 3.3682959383192 096E-7
293254 2.6241446587531 77E-7
303042 5.0335219320747 91E-6
313369 8.7191683943024 45E-7
323521 2.1890460137472 094E-7
333882 5.6099646786741 38E-7
343662 2.5594038513883 153E-7
353644 3.5951008172834 61E-7
363928 3.5564578445364 246E-7

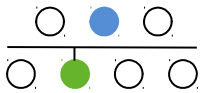


Block Page Rank

Solutions Overview

Biggest Node in Block PageRank Values

374235 3.840988942466275E-6
384553 4.123516215246791E-7
394928 6.704061996436505E-7
404711 2.47132284088853 1E-7
414616 3.1166957382743 27E-7
424746 1.3800460168039234E-6
434706 2.8142783774753487E-6
444488 3.8057415526741863E-6
454284 4.975162372075287E-7
464397 1.1191790566942556E-5
474195 9.8692036625713 46E-7
484049 5.827251748805271E-7
493967 2.1890460137472094E-7
503751 6.9165753040804 37E-6
514130 6.165920506325054E-7
524509 9.884126202737804E-4
534708 2.5730518846919 5E-5
545087 0.0026419954798941147
555466 0.0011913855931278384
565845 1.6342851456044 42E-6
576224 1.1053308062662258E-6
586603 9.740401207655695E-7
596584 4.368801127608822E-6
606366 4.267736144364642E-7
616147 6.12217506194149E-7
626447 1.4459894301271158E-5
636239 1.0720388107813924E-6
646021 3.1193905695897734E-7
655803 2.1890460137472094E-7
665665 9.9768923079453 33E-7
675447 1.1023447496272205E-6
685229 3.5859316143098615E-7



Algorithm Overview

<nodeID, (pageRank numOuts (List: outNodes)>

↓ MAP

<blockID, (typeOfValue ...)>

↓ REDUCE

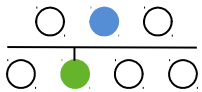
<nodeID, (newPR numOuts (List: outNodes)>

typeOfValue indicates whether the value is of initial data, an edge within the block, or a boundary PR for a certain node.

Details

Mapper:

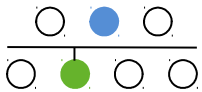
- Parses the value and sends the value to its node
- If it does have any out edges, add its pagerank to the counter for no out edges
- Calculates its outrank to each outNode
  - If its outNode is in the same block, send its nodeID to it.
  - Else send its outrank to that node's block.



Details

Reducer:

- Process all edges to create various structures and pageranks.
- Iterate at most 10 times or until the block residual is less than 0.01. (Keep the iteration average)
- Find the biggest nodeID in each block and pass it to its corresponding counter



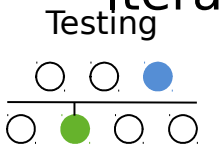
Algorithm Testing

Testing was incremental.

- 4 node sample graphs
- Hundreds of randomly generated nodes
- Counter/sanity checks
- Local Hadoop cluster for simple computations

& comparative

- Computed smaller graphs by hand
- Wrote another program for verification
- Iterative results vs Block results.



Problems?

None! :)

Questions?