



## HW 13.1: Spark implementation of basic PageRank

Write a basic Spark implementation of the iterative PageRank algorithm that takes sparse adjacency lists as input.

Make sure that your implementation utilizes teleportation (1-damping/the number of nodes in the network), and further, distributes the mass of dangling nodes with each iteration so that the output of each iteration is correctly normalized (sums to 1).

[NOTE: The PageRank algorithm assumes that a random surfer (walker), starting from a random web page, chooses the next page to which it will move by clicking at random, with probability  $d$ , one of the hyperlinks in the current page. This probability is represented by a so-called ‘damping factor’  $d$ , where  $d \in (0, 1)$ . Otherwise, with probability  $(1 - d)$ , the surfer jumps to any web page in the network. If a page is a dangling end, meaning it has no outgoing hyperlinks, the random surfer selects an arbitrary web page from a uniform distribution and “teleports” to that page]

In your Spark solution, please use broadcast variables and caching to make sure your code is as efficient as possible.

As you build your code, use the test data

s3://ucb-mids-mls-networks/PageRank-test.txt Or under the Data Subfolder for HW7 on Dropbox with the same file name. (On Dropbox <https://www.dropbox.com/sh/2c0k5adwz36lkcw/AAAAsjQfF9uHfv-X9mCqr9wa?dl=0> (<https://www.dropbox.com/sh/2c0k5adwz36lkcw/AAAAsjQfF9uHfv-X9mCqr9wa?dl=0>))

with teleportation parameter set to 0.15 (1-d, where  $d$ , the damping factor is set to 0.85), and crosscheck your work with the true result, displayed in the first image in the Wikipedia article:

<https://en.wikipedia.org/wiki/PageRank> (<https://en.wikipedia.org/wiki/PageRank>)

and here for reference are the corresponding PageRank probabilities:

A,0.033 B,0.384 C,0.343 D,0.039 E,0.081 F,0.039 G,0.016 H,0.016 I,0.016 J,0.016 K,0.016

**Run this experiment locally first. Report the local configuration that you used and how long in minutes and seconds it takes to complete your job.**

**Repeat this experiment on AWS. Report the AWS cluster configuration that you used and how long in minutes and seconds it takes to complete your job. (in your notebook, cat the cluster config file)**

## Instructions for AWS

- 1) start a cluster in EMR UI (make sure have spark 1.6.1/or earlier if not available installed)
- 2) make sure you have ssh/pem setup (in the last step in the UI) and your security group allows ssh inbound connection
- 3) once cluster is started, ssh to cluster, `ssh hadoop@ec2-52-91-127-197.compute-1.amazonaws.com -i hw13.pem` run `sudo pip install ipython jupyter`
- 4) in the cluster console, run `PYSPARK_DRIVER_PYTHON=jupyter`  
`PYSPARK_DRIVER_PYTHON_OPTS="notebook --no-browser --port=7777" pyspark`
- 5) in your local computer, forward the port `ssh -i hw13.pem -N -f -L localhost:7776:localhost:7777 hadoop@public-dns`
- 6) then open the browser and navigate to `localhost:7776`.
- 7) then you are good to go

In [4]:

```

#All credit to Ron cordell for this implementation
import re

#read line so it can be parallelized into spark
def line_splitter(line):
    node, adj_list = re.split('\t',line.strip()) #split
    node = node.strip('')
    neighbors = eval(adj_list) #render as dict
    node_list = []
    node_list.append((node, neighbors.keys()))
    for neighbor in neighbors:
        node_list.append((neighbor, []))
    return node_list

#helper function to update the pagerank
def adjustRank(rank, mass):
    adj_rank = 0.0
    if rank is not None:
        adj_rank = rank
    return d*adj_rank + d*mass/n + t #formula

# damping parameter
d = 0.85

D = sc.textFile("PageRank-test.txt") #insert file here to run

graph = D.flatMap(lambda line: line_splitter(line)).reduceByKey(lambda
bda a,b:a+b).cache()

# compute the number of nodes
n = graph.count()

# compute teleportation factor
t = (1.0-d)/n

# prime the pump with the initial page rank for each node = 1/n
adj_list = graph.map(lambda (node, outlinks): (node, (1.0/n, outlinks)))

for i in range(0,30):
    dangling_mass = adj_list.filter(lambda x: len(x[1][1])==0).map(
(lambda x: x[1][0])).reduce(lambda x,y:x+y)

    distributed_mass = adj_list.filter(lambda (node, (rank,outlinks): len(outlinks) > 0)\
        .map(lambda (node, (rank, outlinks)): (rank/len(outlinks),
outlinks))\
        .flatMapValues(lambda x:x)\
        .map(lambda (rank, outlink): (outlink, rank))\
        .reduceByKey(lambda x,y: x+y)

    adj_list=graph.leftOuterJoin(distributed_mass)\

```

```
        .map(lambda (node, (outlinks, rank)):(node, (rank,outlinks)))\
        .map(lambda (node, (rank, outlinks)):(node, (adjustRank(rank, dangling_mass), outlinks)) )

for node in adj_list.sortBy(lambda x: -x[1][0]).collect():
    print node[0], node[1][0]
```

```
B 0.383410412554
C 0.343378600107
E 0.0808856932689
D 0.0390870921233
F 0.0390870921233
A 0.0327814931824
G 0.0161694790207
I 0.0161694790207
K 0.0161694790207
H 0.0161694790207
J 0.0161694790207
```

On AWS I used 6 m3.xlarge nodes which took 43.2308270931 for 30 iterations

## HW 13.2: Applying PageRank to the Wikipedia hyperlinks network

Run your Spark PageRank implementation on the Wikipedia dataset for 10 iterations, and display the top 100 ranked nodes (with  $\alpha = 0.85$ ).

Run your PageRank implementation on the Wikipedia dataset for 50 iterations, and display the top 100 ranked nodes (with teleportation factor of 0.15). Plot the pagerank values for the top 100 pages resulting from the 50 iterations run. Then plot the pagerank values for the same 100 pages that resulted from the 10 iterations run. Comment on your findings. Have the top 100 ranked pages changed? Have the pagerank values changed? Explain.

Report the AWS cluster configuration that you used and how long in minutes and seconds it takes to complete your job.

NOTE: ===== English Wikipedia hyperlink network.data ===== The dataset is available via Dropbox at:

<https://www.dropbox.com/sh/2c0k5adwz36lkcw/AAAKsjQfF9uHfv-X9mCqr9wa?dl=0>  
[\(https://www.dropbox.com/sh/2c0k5adwz36lkcw/AAAKsjQfF9uHfv-X9mCqr9wa?dl=0\)](https://www.dropbox.com/sh/2c0k5adwz36lkcw/AAAKsjQfF9uHfv-X9mCqr9wa?dl=0)

on S3 at s3://ucb-mids-mls-networks/wikipedia/ -- s3://ucb-mids-mls-networks/wikipedia/all-pages-indexed-out.txt # Graph -- s3://ucb-mids-mls-networks/wikipedia/indices.txt # Page titles and page Ids

The dataset is built from the Sept. 2015 XML snapshot of English Wikipedia. For this directed network, a link between articles:

A -> B

is defined by the existence of a hyperlink in A pointing to B. This network also exists in the indexed format:

Data: s3://ucb-mids-mls-networks/wikipedia/all-pages-indexed-out.txt Data: s3://ucb-mids-mls-networks/wikipedia/all-pages-indexed-in.txt Data: s3://ucb-mids-mls-networks/wikipedia/indices.txt

but has an index with more detailed data:

(article name) \t (index) \t (in degree) \t (out degree)

In the dictionary, target nodes are keys, link weights are values . Here, a weight indicates the number of time a page links to another. However, for the sake of this assignment, treat this an unweighted network, and set all weights to 1 upon data input.

**10 r3.xlarge****Wikipedia 10 took: 1182.26167488 seconds****Wikipedia 50 took: Process took: 4909.77044916 seconds**

Wiki 10: 13455888 0.00143798119916 1184351 0.000658251945105 4695850 0.000629974450417  
5051368 0.000565939050801 1384888 0.00044507004801 6113490 0.000439619212504 2437837  
0.000439441328896 7902219 0.00043737364256 13425865 0.000425377992977 6076759  
0.000421589333108 4196067 0.000416677324754 6172466 0.000392502737321 14112583  
0.000378696768898 10390714 0.000357441833747 15164193 0.000339380924455 3191491  
0.000333818776096 6416278 0.000324284641442 6237129 0.000323688786565 7835160  
0.000322179443011 1516699 0.0003198989965 13725487 0.000308716462502 9276255  
0.00030565697608 7576704 0.000304530902473 10469541 0.000299306030787 5154210  
0.00029382115752 12836211 0.000280065925591 7990491 0.000278579812351 4198751  
0.00026445377167 2797855 0.000259527582694 11253108 0.000256916522151 9386580  
0.000252902351918 3603527 0.000251586860595 12074312 0.000247074225265 3069099  
0.000245401340598 14881689 0.000242623575376 2155467 0.000241229337963 1441065  
0.000235164413051 14503460 0.000229901982328 2396749 0.000217083966457 3191268  
0.000212449231613 10566120 0.000212042270391 11147327 0.000208264621275 2614581  
0.00020821301672 1637982 0.00020463498912 11245362 0.000200221093362 12430985  
0.000200183731881 9355455 0.00019374766018 10527224 0.000189360730011 14112408  
0.000187349470231 2614578 0.000185369494635 9391762 0.000184862463639 6172167  
0.000184579657789 8697871 0.000184216787849 981395 0.000182309018089 6171937  
0.000176715345303 5490435 0.000176367442048 11582765 0.000170392829913 14725161  
0.000167468441412 9562547 0.000164712124816 12067030 0.000164601303706 994890  
0.000163200340599 9394907 0.000158007914422 9997298 0.000157883033536 13280859  
0.000156536146392 10345830 0.000155516376098 4978429 0.000152983458437 12447593  
0.000152377403763 8019937 0.000150460071734 11148415 0.000147187978269 13432150  
0.000145361511179 4344962 0.000144515680409 1175360 0.000140122326465 12038331  
0.000139143487726 14565507 0.000136935090125 4624519 0.00013570017848 1523975  
0.000134052440417 14981725 0.000133283278987 13328060 0.00013252405361 1332806  
0.000128724206236 10399499 0.000128403691278 14963657 0.000127906141328 2826544  
0.000126494655262 2578813 0.000125795531863 1813634 0.000124978709254 1575979  
0.000124846716592 2778099 0.00012179799776 13853369 0.000118476377302 9924814  
0.00011845264948 4568647 0.000113986387573 9742161 0.000112399194205 12785678  
0.00011224883786 7467127 0.000112149478495 3328327 0.000111834932093 14727077  
0.000111541221372 10246542 0.000111460357921 3591832 0.000111396766917 5274313  
0.000111331922407 14709489 0.000110798226712 3973000 0.000110663936005 15070394  
0.000110524921697



Wiki 50: 13455888 0.00146154857879 1184351 0.000666013855615 4695850 0.000639672606533  
5051368 0.000574762875979 1384888 0.000450120670195 2437837 0.000446666570933 6113490  
0.000444629709428 7902219 0.000443875381845 13425865 0.000433138481704 6076759  
0.000427704719316 4196067 0.000423413581476 6172466 0.000397823296841 14112583  
0.000385482971376 10390714 0.000362663786257 15164193 0.000343585296234 3191491  
0.000338047422597 6416278 0.00032921787358 6237129 0.000328992181466 7835160  
0.000326199737819 1516699 0.000325108339111 13725487 0.000312680149329 9276255  
0.000309567354633 7576704 0.000307978908698 10469541 0.000303118343378 5154210  
0.00029754579874 12836211 0.000286034825153 7990491 0.000283617796922 4198751  
0.000269051323154 2797855 0.000264011972564 11253108 0.00026098273596 9386580  
0.000257695338763 3603527 0.000254969161082 12074312 0.000251020166063 3069099  
0.000248673948023 14881689 0.000245362759899 2155467 0.00024471811602 1441065  
0.0002386465625 14503460 0.000233302359678 2396749 0.000220630523434 3191268  
0.000214954172738 10566120 0.000214543272604 2614581 0.000211201668378 11147327  
0.000211185630694 1637982 0.000207030418227 12430985 0.000203300592195 11245362  
0.000202528755713 9355455 0.000197012613097 10527224 0.000191389653574 14112408  
0.000190781940738 9391762 0.000188169955943 2614578 0.000188020711274 8697871  
0.000187042464627 6172167 0.000186731465231 981395 0.000185227371084 6171937  
0.000178748154784 5490435 0.000178311959398 11582765 0.000173347300826 14725161  
0.00016948266271 12067030 0.000167650594693 9562547 0.000167213540803 994890  
0.000165398876541 9997298 0.000160694946602 9394907 0.000160522254514 13280859  
0.000159005418943 10345830 0.000157616963123 4978429 0.000155270639931 12447593  
0.000154928957148 8019937 0.000153288852635 11148415 0.000148833514713 13432150  
0.000147855764335 4344962 0.000147109546466 1175360 0.000141843191136 12038331  
0.000141298246852 14565507 0.000139065677208 4624519 0.000137645583292 1523975  
0.000136245282789 14981725 0.000134895063036 13328060 0.00013474183704 1332806  
0.000130692301864 10399499 0.00013020541465 14963657 0.000130036735847 2578813  
0.000128410980262 2826544 0.000128203747864 1575979 0.000127322338903 1813634  
0.000127152522854 2778099 0.000124107584482 13853369 0.000120935161169 9924814  
0.000120241485718 4568647 0.000115778325695 12785678 0.000114506611887 7467127  
0.000114472347202 9742161 0.000114300899384 3328327 0.000113592887172 10246542  
0.000113264541758 3591832 0.000113234971214 5274313 0.000113192001912 14727077  
0.000112910401522 14709489 0.000112415650593 5908108 0.000112186213975 3973000  
0.000112119387579

```
In [3]: IDs10 = []
values10 = []
IDs50 = []
values50 = []
with open('Wiki10.txt', 'r') as myfile:
    lines = myfile.readlines()
    for line in lines:
        line = line.split()
        IDs10.append(line[0])
        values10.append(float(line[1]))

with open('Wiki50.txt', 'r') as myfile:
    lines = myfile.readlines()
    for line in lines:
        line = line.split()
        IDs50.append(line[0])
        values50.append(float(line[1]))

print "10 iterations" + '\t' + '\t' + '\t' + '\t' + '\t' + ' 50 iterations'

for i in range(100):
    print "ID: " + str(IDs10[i]) + " value: " + str(values10[i]) +
    '\t' + '\t' + "ID: " + str(IDs50[i]) + " value: " + str(values50[i])
```



10 iterations

ID: 13455888 value: 0.00143798119916  
e: 0.00146154857879  
ID: 1184351 value: 0.000658251945105  
0.000666013855615  
ID: 4695850 value: 0.000629974450417  
0.000639672606533  
ID: 5051368 value: 0.000565939050801  
0.000574762875979  
ID: 1384888 value: 0.00044507004801  
0.000450120670195  
ID: 6113490 value: 0.000439619212504  
0.000446666570933  
ID: 2437837 value: 0.000439441328896  
0.000444629709428  
ID: 7902219 value: 0.00043737364256  
0.000443875381845  
ID: 13425865 value: 0.000425377992977  
e: 0.000433138481704  
ID: 6076759 value: 0.000421589333108  
0.000427704719316  
ID: 4196067 value: 0.000416677324754  
0.000423413581476  
ID: 6172466 value: 0.000392502737321  
0.000397823296841  
ID: 14112583 value: 0.000378696768898  
e: 0.000385482971376  
ID: 10390714 value: 0.000357441833747  
e: 0.000362663786257  
ID: 15164193 value: 0.000339380924455  
e: 0.000343585296234  
ID: 3191491 value: 0.000333818776096  
0.000338047422597  
ID: 6416278 value: 0.000324284641442  
0.00032921787358  
ID: 6237129 value: 0.000323688786565  
0.000328992181466  
ID: 7835160 value: 0.000322179443011  
0.000326199737819  
ID: 1516699 value: 0.0003198989965  
0.000325108339111  
ID: 13725487 value: 0.000308716462502  
e: 0.000312680149329  
ID: 9276255 value: 0.00030565697608  
0.000309567354633  
ID: 7576704 value: 0.000304530902473  
0.000307978908698  
ID: 10469541 value: 0.000299306030787  
e: 0.000303118343378  
ID: 5154210 value: 0.00029382115752  
0.00029754579874  
ID: 12836211 value: 0.000280065925591  
e: 0.000286034825153  
ID: 7990491 value: 0.000278579812351

50 iterations

ID: 13455888 valu  
ID: 1184351 value:  
ID: 4695850 value:  
ID: 5051368 value:  
ID: 1384888 value:  
ID: 2437837 value:  
ID: 6113490 value:  
ID: 7902219 value:  
ID: 13425865 valu  
ID: 6076759 value:  
ID: 4196067 value:  
ID: 6172466 value:  
ID: 14112583 valu  
ID: 10390714 valu  
ID: 15164193 valu  
ID: 3191491 value:  
ID: 6416278 value:  
ID: 6237129 value:  
ID: 7835160 value:  
ID: 1516699 value:  
ID: 13725487 valu  
ID: 9276255 value:  
ID: 7576704 value:  
ID: 10469541 valu  
ID: 5154210 value:  
ID: 12836211 valu  
ID: 7990491 value:

0.000283617796922	
ID: 4198751 value: 0.00026445377167	ID: 4198751 value:
0.000269051323154	
ID: 2797855 value: 0.000259527582694	ID: 2797855 value:
0.000264011972564	
ID: 11253108 value: 0.000256916522151	ID: 11253108 valu
e: 0.00026098273596	
ID: 9386580 value: 0.000252902351918	ID: 9386580 value:
0.000257695338763	
ID: 3603527 value: 0.000251586860595	ID: 3603527 value:
0.000254969161082	
ID: 12074312 value: 0.000247074225265	ID: 12074312 valu
e: 0.000251020166063	
ID: 3069099 value: 0.000245401340598	ID: 3069099 value:
0.000248673948023	
ID: 14881689 value: 0.000242623575376	ID: 14881689 valu
e: 0.000245362759899	
ID: 2155467 value: 0.000241229337963	ID: 2155467 value:
0.00024471811602	
ID: 1441065 value: 0.000235164413051	ID: 1441065 value:
0.0002386465625	
ID: 14503460 value: 0.000229901982328	ID: 14503460 valu
e: 0.000233302359678	
ID: 2396749 value: 0.000217083966457	ID: 2396749 value:
0.000220630523434	
ID: 3191268 value: 0.000212449231613	ID: 3191268 value:
0.000214954172738	
ID: 10566120 value: 0.000212042270391	ID: 10566120 valu
e: 0.000214543272604	
ID: 11147327 value: 0.000208264621275	ID: 2614581 value:
0.000211201668378	
ID: 2614581 value: 0.00020821301672	ID: 11147327 valu
e: 0.000211185630694	
ID: 1637982 value: 0.00020463498912	ID: 1637982 value:
0.000207030418227	
ID: 11245362 value: 0.000200221093362	ID: 12430985 valu
e: 0.000203300592195	
ID: 12430985 value: 0.000200183731881	ID: 11245362 valu
e: 0.000202528755713	
ID: 9355455 value: 0.00019374766018	ID: 9355455 value:
0.000197012613097	
ID: 10527224 value: 0.000189360730011	ID: 10527224 valu
e: 0.000191389653574	
ID: 14112408 value: 0.000187349470231	ID: 14112408 valu
e: 0.000190781940738	
ID: 2614578 value: 0.000185369494635	ID: 9391762 value:
0.000188169955943	
ID: 9391762 value: 0.000184862463639	ID: 2614578 value:
0.000188020711274	
ID: 6172167 value: 0.000184579657789	ID: 8697871 value:
0.000187042464627	
ID: 8697871 value: 0.000184216787849	ID: 6172167 value:
0.000186731465231	
ID: 981395 value: 0.000182309018089	ID: 981395 value:

0.000185227371084	
ID: 6171937 value: 0.000176715345303	ID: 6171937 value:
0.000178748154784	
ID: 5490435 value: 0.000176367442048	ID: 5490435 value:
0.000178311959398	
ID: 11582765 value: 0.000170392829913	ID: 11582765 valu
e: 0.000173347300826	
ID: 14725161 value: 0.000167468441412	ID: 14725161 valu
e: 0.00016948266271	
ID: 9562547 value: 0.000164712124816	ID: 12067030 valu
e: 0.000167650594693	
ID: 12067030 value: 0.000164601303706	ID: 9562547 value:
0.000167213540803	
ID: 994890 value: 0.000163200340599	ID: 994890 value:
0.000165398876541	
ID: 9394907 value: 0.000158007914422	ID: 9997298 value:
0.000160694946602	
ID: 9997298 value: 0.000157883033536	ID: 9394907 value:
0.000160522254514	
ID: 13280859 value: 0.000156536146392	ID: 13280859 valu
e: 0.000159005418943	
ID: 10345830 value: 0.000155516376098	ID: 10345830 valu
e: 0.000157616963123	
ID: 4978429 value: 0.000152983458437	ID: 4978429 value:
0.000155270639931	
ID: 12447593 value: 0.000152377403763	ID: 12447593 valu
e: 0.000154928957148	
ID: 8019937 value: 0.000150460071734	ID: 8019937 value:
0.000153288852635	
ID: 11148415 value: 0.000147187978269	ID: 11148415 valu
e: 0.000148833514713	
ID: 13432150 value: 0.000145361511179	ID: 13432150 valu
e: 0.000147855764335	
ID: 4344962 value: 0.000144515680409	ID: 4344962 value:
0.000147109546466	
ID: 1175360 value: 0.000140122326465	ID: 1175360 value:
0.000141843191136	
ID: 12038331 value: 0.000139143487726	ID: 12038331 valu
e: 0.000141298246852	
ID: 14565507 value: 0.000136935090125	ID: 14565507 valu
e: 0.000139065677208	
ID: 4624519 value: 0.00013570017848	ID: 4624519 value:
0.000137645583292	
ID: 1523975 value: 0.000134052440417	ID: 1523975 value:
0.000136245282789	
ID: 14981725 value: 0.000133283278987	ID: 14981725 valu
e: 0.000134895063036	
ID: 13328060 value: 0.00013252405361	ID: 13328060 valu
e: 0.00013474183704	
ID: 1332806 value: 0.000128724206236	ID: 1332806 value:
0.000130692301864	
ID: 10399499 value: 0.000128403691278	ID: 10399499 valu
e: 0.00013020541465	
ID: 14963657 value: 0.000127906141328	ID: 14963657 valu

e: 0.000130036735847	
ID: 2826544 value: 0.000126494655262	ID: 2578813 value:
0.000128410980262	
ID: 2578813 value: 0.000125795531863	ID: 2826544 value:
0.000128203747864	
ID: 1813634 value: 0.000124978709254	ID: 1575979 value:
0.000127322338903	
ID: 1575979 value: 0.000124846716592	ID: 1813634 value:
0.000127152522854	
ID: 2778099 value: 0.00012179799776	ID: 2778099 value:
0.000124107584482	
ID: 13853369 value: 0.000118476377302	ID: 13853369 valu
e: 0.000120935161169	
ID: 9924814 value: 0.00011845264948	ID: 9924814 value:
0.000120241485718	
ID: 4568647 value: 0.000113986387573	ID: 4568647 value:
0.000115778325695	
ID: 9742161 value: 0.000112399194205	ID: 12785678 valu
e: 0.000114506611887	
ID: 12785678 value: 0.00011224883786	ID: 7467127 value:
0.000114472347202	
ID: 7467127 value: 0.000112149478495	ID: 9742161 value:
0.000114300899384	
ID: 3328327 value: 0.000111834932093	ID: 3328327 value:
0.000113592887172	
ID: 14727077 value: 0.000111541221372	ID: 10246542 valu
e: 0.000113264541758	
ID: 10246542 value: 0.000111460357921	ID: 3591832 value:
0.000113234971214	
ID: 3591832 value: 0.000111396766917	ID: 5274313 value:
0.000113192001912	
ID: 5274313 value: 0.000111331922407	ID: 14727077 valu
e: 0.000112910401522	
ID: 14709489 value: 0.000110798226712	ID: 14709489 valu
e: 0.000112415650593	
ID: 3973000 value: 0.000110663936005	ID: 5908108 value:
0.000112186213975	
ID: 15070394 value: 0.000110524921697	ID: 3973000 value:
0.000112119387579	

The majority of the top IDs haven't really changed but a few of the bottom ones have swapped places by several ranks. Also note the pagerank values of the 50 iterations tend to be larger than the 10.

## NB: for an alternative implementation of HW 1-2 please see these links:

[https://github.com/dunmireg/HW13/blob/master/HW13\\_HD\\_localnotebook.ipynb](https://github.com/dunmireg/HW13/blob/master/HW13_HD_localnotebook.ipynb)  
([https://github.com/dunmireg/HW13/blob/master/HW13\\_HD\\_localnotebook.ipynb](https://github.com/dunmireg/HW13/blob/master/HW13_HD_localnotebook.ipynb))

[https://github.com/dunmireg/HW13/blob/master/HW13\\_HD\\_clusternotebook.ipynb](https://github.com/dunmireg/HW13/blob/master/HW13_HD_clusternotebook.ipynb)  
([https://github.com/dunmireg/HW13/blob/master/HW13\\_HD\\_clusternotebook.ipynb](https://github.com/dunmireg/HW13/blob/master/HW13_HD_clusternotebook.ipynb))

### HW 13.3: Spark GraphX versus your implementation of PageRank

Run the Spark GraphX PageRank implementation on the Wikipedia dataset for 10 iterations, and display the top 100 ranked nodes (with  $\alpha = 0.85$ ).

Run your PageRank implementation on the Wikipedia dataset for 50 iterations, and display the top 100 ranked nodes (with teleportation factor of 0.15). Have the top 100 ranked pages changed? Comment on your findings. Plot both 100 curves.

Report the AWS cluster configuration that you used and how long in minutes and seconds it takes to complete this job.

Put the runtime results of HW13.2 and HW13.3 in a tabular format (with rows corresponding to implementation and columns corresponding to experiment setup (10 iterations, 50 iterations)). Discuss the run times and explaining the differences.

Plot the pagerank values for the top 100 pages resulting from the 50 iterations run (using GraphX). Then plot the pagerank values for the same 100 pages that resulted from the 50 iterations run of your homegrown pagerank implementation. Comment on your findings. Have the top 100 ranked pages changed? Have the pagerank values changed? Explain.

Using a 6 node r3.xlarge cluster **10** iterations took: **644 seconds**

and **50** iterations took **1337 seconds**



## For a second alternative implementation of Homework 1 and 2, and our implementation of Question 3 please see:

<https://www.zeppelinhub.com/viewer/notebooks/aHR0cHM6Ly9yYXcuZ2l0aHVidXNlcmNvbnRlbnQuY29tL>  
(<https://www.zeppelinhub.com/viewer/notebooks/aHR0cHM6Ly9yYXcuZ2l0aHVidXNlcmNvbnRlbnQuY29tL>)

<https://www.zeppelinhub.com/viewer/notebooks/aHR0cHM6Ly9yYXcuZ2l0aHVidXNlcmNvbnRlbnQuY29tL>  
(<https://www.zeppelinhub.com/viewer/notebooks/aHR0cHM6Ly9yYXcuZ2l0aHVidXNlcmNvbnRlbnQuY29tL>)

Implementation	10 Iterations	50 Iterations	Cluster
Personal PageRank	1182.26 seconds	4909.77 seconds	10 r3.xlarge
GraphX	644 seconds	1337 seconds	6 r3.xlarge

The graphX implemenation is significantly faster than our homegrown version. This includes the fact that the cluster was almost twice as big for the personal pagerank algorithm. We suspect this is due to optimization.

The GraphX implementation was almost twice as fast at 10 iterations but nearly 4 times faster for the 50 iterations. This is a significant speed boost.

```
In [15]: IDsWiki = []
valuesWiki = []
IDsGraph = []
valuesGraph = []

with open('Wiki50.txt', 'r') as myfile:
    lines = myfile.readlines()
    for line in lines:
        line = line.split()
        IDsWiki.append(line[0])
        valuesWiki.append(float(line[1]))

with open('GraphX50.txt', 'r') as myfile:
    lines = myfile.readlines()
    for line in lines:
        line = line.strip().split(',')
        IDsGraph.append(line[0])
        valuesGraph.append(float(line[1]))

# for i in range(100):
#     if IDsWiki[i] != IDsGraph[i]:
#         print "problem at " + str(i)

print "Homegrown PageRank" + '\t' + '\t' + '\t' + '\t' + ' GraphX'

for i in range(100):
    print "ID: " + str(IDsWiki[i]) + " value: " + str(valuesWiki[i]) + '\t' + '\t' + "ID: " + str(IDsGraph[i]) + " value: " + str(valuesGraph[i])
```



## Homegrown PageRank

ID: 13455888 value: 0.00146154857879  
e: 6247.50602457  
ID: 1184351 value: 0.000666013855615  
2846.92399367  
ID: 4695850 value: 0.000639672606533  
2734.33065298  
ID: 5051368 value: 0.000574762875979  
2456.8690847  
ID: 1384888 value: 0.000450120670195  
1924.07262049  
ID: 2437837 value: 0.00044666570933  
1909.31188175  
ID: 6113490 value: 0.000444629709428  
1900.60117924  
ID: 7902219 value: 0.000443875381845  
1897.38011819  
ID: 13425865 value: 0.000433138481704  
e: 1851.48637305  
ID: 6076759 value: 0.000427704719316  
1828.25671853  
ID: 4196067 value: 0.000423413581476  
1809.91476025  
ID: 6172466 value: 0.000397823296841  
1700.52589431  
ID: 14112583 value: 0.000385482971376  
e: 1647.77882807  
ID: 10390714 value: 0.000362663786257  
e: 1550.2349427  
ID: 15164193 value: 0.000343585296234  
e: 1468.68052658  
ID: 3191491 value: 0.000338047422597  
1445.00877162  
ID: 6416278 value: 0.00032921787358  
1407.26757821  
ID: 6237129 value: 0.000328992181466  
1406.30332127  
ID: 7835160 value: 0.000326199737819  
1394.36454786  
ID: 1516699 value: 0.000325108339111  
1389.70145266  
ID: 13725487 value: 0.000312680149329  
e: 1336.57491714  
ID: 9276255 value: 0.000309567354633  
1323.26824076  
ID: 7576704 value: 0.000307978908698  
1316.4775797  
ID: 10469541 value: 0.000303118343378  
e: 1295.70214958  
ID: 5154210 value: 0.00029754579874  
1271.88183708  
ID: 12836211 value: 0.000286034825153  
e: 1222.68102579  
ID: 7990491 value: 0.000283617796922

## GraphX

ID: 13455888 valu  
ID: 1184351 value:  
ID: 4695850 value:  
ID: 5051368 value:  
ID: 1384888 value:  
ID: 2437837 value:  
ID: 6113490 value:  
ID: 7902219 value:  
ID: 13425865 valu  
ID: 6076759 value:  
ID: 4196067 value:  
ID: 6172466 value:  
ID: 14112583 valu  
ID: 10390714 valu  
ID: 15164193 valu  
ID: 3191491 value:  
ID: 6416278 value:  
ID: 6237129 value:  
ID: 7835160 value:  
ID: 1516699 value:  
ID: 13725487 valu  
ID: 9276255 value:  
ID: 7576704 value:  
ID: 10469541 valu  
ID: 5154210 value:  
ID: 12836211 valu  
ID: 7990491 value:

1212.34762351	ID: 4198751 value: 0.000269051323154	ID: 4198751 value:
1150.08191968	ID: 2797855 value: 0.000264011972564	ID: 2797855 value:
1128.54073542	ID: 11253108 value: 0.00026098273596	ID: 11253108 value:
e: 1115.59140905	ID: 9386580 value: 0.000257695338763	ID: 9386580 value:
1101.54036467	ID: 3603527 value: 0.000254969161082	ID: 3603527 value:
1089.88479554	ID: 12074312 value: 0.000251020166063	ID: 12074312 value:
e: 1073.00558945	ID: 3069099 value: 0.000248673948023	ID: 3069099 value:
1062.9753781	ID: 14881689 value: 0.000245362759899	ID: 14881689 value:
e: 1048.82041894	ID: 2155467 value: 0.00024471811602	ID: 2155467 value:
1046.06644683	ID: 1441065 value: 0.0002386465625	ID: 1441065 value:
ID: 1441065 value: 0.0002386465625	ID: 14503460 value: 0.000233302359678	ID: 14503460 value:
1020.11319674	e: 997.268930793	ID: 2396749 value:
ID: 14503460 value: 0.000233302359678	ID: 2396749 value: 0.000220630523434	ID: 3191268 value:
e: 997.268930793	ID: 3191268 value: 0.000214954172738	ID: 10566120 value:
ID: 2396749 value: 0.000220630523434	918.837100077	ID: 2614581 value:
943.10273484	ID: 10566120 value: 0.000214543272604	ID: 11147327 value:
ID: 3191268 value: 0.000214954172738	e: 917.080680832	ID: 1637982 value:
918.837100077	ID: 2614581 value: 0.000211201668378	ID: 12430985 value:
ID: 10566120 value: 0.000214543272604	902.797879358	ID: 11245362 value:
e: 917.080680832	ID: 11147327 value: 0.000211185630694	ID: 9355455 value:
ID: 2614581 value: 0.000211201668378	e: 902.729122567	ID: 10527224 value:
902.797879358	ID: 1637982 value: 0.000207030418227	ID: 14112408 value:
ID: 11147327 value: 0.000211185630694	884.966388419	ID: 9391762 value:
e: 902.729122567	ID: 12430985 value: 0.000203300592195	ID: 2614578 value:
ID: 1637982 value: 0.000207030418227	e: 869.024440309	ID: 8697871 value:
884.966388419	ID: 11245362 value: 0.000202528755713	ID: 6172167 value:
ID: 12430985 value: 0.000203300592195	e: 865.723576161	ID: 981395 value:
e: 869.024440309	ID: 9355455 value: 0.000197012613097	
ID: 11245362 value: 0.000202528755713	842.146345585	
e: 865.723576161	ID: 10527224 value: 0.000191389653574	
ID: 9355455 value: 0.000197012613097	e: 818.10826412	
842.146345585	ID: 14112408 value: 0.000190781940738	
ID: 10527224 value: 0.000191389653574	e: 815.513302371	
e: 818.10826412	ID: 9391762 value: 0.000188169955943	
ID: 14112408 value: 0.000190781940738	804.347929584	
e: 815.513302371	ID: 2614578 value: 0.000188020711274	
ID: 9391762 value: 0.000188169955943	803.709073933	
804.347929584	ID: 8697871 value: 0.000187042464627	
ID: 2614578 value: 0.000188020711274	799.527683902	
803.709073933	ID: 6172167 value: 0.000186731465231	
ID: 8697871 value: 0.000187042464627	798.197046081	
799.527683902	ID: 981395 value: 0.000185227371084	
ID: 6172167 value: 0.000186731465231		
798.197046081		
ID: 981395 value: 0.000185227371084		

791.769392538	
ID: 6171937 value: 0.000178748154784	ID: 6171937 value:
764.071728701	
ID: 5490435 value: 0.000178311959398	ID: 5490435 value:
762.206878711	
ID: 11582765 value: 0.000173347300826	ID: 11582765 valu
e: 740.987055347	
ID: 14725161 value: 0.00016948266271	ID: 14725161 valu
e: 724.465856433	
ID: 12067030 value: 0.000167650594693	ID: 12067030 valu
e: 716.636603108	
ID: 9562547 value: 0.000167213540803	ID: 9562547 value:
714.767428631	
ID: 994890 value: 0.000165398876541	ID: 994890 value:
707.009909547	
ID: 9997298 value: 0.000160694946602	ID: 9997298 value:
686.903793907	
ID: 9394907 value: 0.000160522254514	ID: 9394907 value:
686.165063732	
ID: 13280859 value: 0.000159005418943	ID: 13280859 valu
e: 679.681242586	
ID: 10345830 value: 0.000157616963123	ID: 10345830 valu
e: 673.745703606	
ID: 4978429 value: 0.000155270639931	ID: 4978429 value:
663.716385872	
ID: 12447593 value: 0.000154928957148	ID: 12447593 valu
e: 662.256350626	
ID: 8019937 value: 0.000153288852635	ID: 8019937 value:
655.246068531	
ID: 11148415 value: 0.000148833514713	ID: 11148415 valu
e: 636.199317031	
ID: 13432150 value: 0.000147855764335	ID: 13432150 valu
e: 632.021562568	
ID: 4344962 value: 0.000147109546466	ID: 4344962 value:
628.831991509	
ID: 1175360 value: 0.000141843191136	ID: 1175360 value:
606.319062589	
ID: 12038331 value: 0.000141298246852	ID: 12038331 valu
e: 603.99042163	
ID: 14565507 value: 0.000139065677208	ID: 14565507 valu
e: 594.447184069	
ID: 4624519 value: 0.000137645583292	ID: 4624519 value:
588.376820632	
ID: 1523975 value: 0.000136245282789	ID: 1523975 value:
582.391380026	
ID: 14981725 value: 0.000134895063036	ID: 14981725 valu
e: 576.618688587	
ID: 13328060 value: 0.00013474183704	ID: 13328060 valu
e: 575.964848259	
ID: 1332806 value: 0.000130692301864	ID: 1332806 value:
558.654487513	
ID: 10399499 value: 0.00013020541465	ID: 10399499 valu
e: 556.572896654	
ID: 14963657 value: 0.000130036735847	ID: 14963657 valu

e: 555.852552059	
ID: 2578813 value: 0.000128410980262	ID: 2578813 value:
548.903982861	
ID: 2826544 value: 0.000128203747864	ID: 2826544 value:
548.016612766	
ID: 1575979 value: 0.000127322338903	ID: 1575979 value:
544.250217521	
ID: 1813634 value: 0.000127152522854	ID: 1813634 value:
543.523842162	
ID: 2778099 value: 0.000124107584482	ID: 2778099 value:
530.508306161	
ID: 13853369 value: 0.000120935161169	ID: 13853369 valu
e: 516.947866042	
ID: 9924814 value: 0.000120241485718	ID: 9924814 value:
513.981605526	
ID: 4568647 value: 0.000115778325695	ID: 4568647 value:
494.903642197	
ID: 12785678 value: 0.000114506611887	ID: 12785678 valu
e: 489.468364768	
ID: 7467127 value: 0.000114472347202	ID: 7467127 value:
489.322081091	
ID: 9742161 value: 0.000114300899384	ID: 9742161 value:
488.588372678	
ID: 3328327 value: 0.000113592887172	ID: 3328327 value:
485.561886269	
ID: 10246542 value: 0.000113264541758	ID: 10246542 valu
e: 484.158326084	
ID: 3591832 value: 0.000113234971214	ID: 3591832 value:
484.03201348	
ID: 5274313 value: 0.000113192001912	ID: 5274313 value:
483.848323732	
ID: 14727077 value: 0.000112910401522	ID: 14727077 valu
e: 482.643757536	
ID: 14709489 value: 0.000112415650593	ID: 14709489 valu
e: 480.529379289	
ID: 5908108 value: 0.000112186213975	ID: 5908108 value:
479.548861821	
ID: 3973000 value: 0.000112119387579	ID: 3973000 value:
479.262711842	

In examining this we find the Ids are exactly the same for both the graphX and homegrown pagerank. The values are obviously different but their ranking seems to be the same. So no, the ranked pages have not changed. We believe the GraphX values are not normalized, explaining the difference in their values whereas the homegrown spark version is normalized so all the total mass adds up to 1

## Questions 4 and 5

See the following link: <https://github.com/dunmireg/HW13/blob/master/MIDS-LSML-HW13.ipynb>  
(<https://github.com/dunmireg/HW13/blob/master/MIDS-LSML-HW13.ipynb>)

In [ ]: