Class Number: SEC 01(BOS-2-TR) (CRN: 13036)

HW Number: 4 **Name: Anoop Pai**

Design Discussion:

Initially, the Scala Spark program creates a pair RDD which contains a page name and its outlinks as a key-value pair. This is accomplished by making use of the Java program - WikiBz2Parser from Assignment3 and calling it on every line of the input data set using a map() function. The next map() method removes the '~' separating the page name and its outlinks that was returned by the Java Parser and constructs a pair RDD with page name as its key and outlinks as its value. Once the whole data set has been parsed, a count() action is performed on this pair RDD to obtain the total number of nodes in the graph. This RDD is cached as it is used in each iteration of PageRank so as to get around lazy evaluation.

Next, the Spark version creates a new pair RDD where it initializes every node with a page rank which is equal to 1/totalNumNodes using a map function. We join the initial pagerank RDD with the adjacency graph RDD so that we can access both the page rank and the outlinks of a particular node by using join(). We perform a flatMap on its values in order to send the contribution of each node to each of its outlinks, thereby resulting in a new pair RDD which contains a node name and its respective contribution. Also, for all those nodes which have no outlinks i.e dangling nodes, their pageranks are accumulated to maintain a delta value.

Next, the RDD containing the contribution for each node is reduced by key to aggregate all the values of pagerank for the same key via reduceByKey(). A new value of PageRank is computed using the initial page rank value or its value from the previous iteration and the accumulated value of delta for that particular iteration using map(). The structure of 'pageranks' RDD is maintained by performing subtractByKey() and union() so that no nodes were lost during the process. This updated RDD is then reused for the next iteration to compute a PageRank value for that particular iteration. (Only the node name and its pagerank are passed to the next iteration)

Lastly, the topK results are computed upon the completion of 10^{th} iteration of PageRank to retrieve the top 100 pagerank values and its corresponding page name in decreasing order by making use of the sortBy() function in Spark.

Comparison between Hadoop MapReduce and Spark Implementations of PageRank:

(Pre-Processing phase)

In the case of Hadoop MapReduce, the pre-processing phase was separately executed in a different job to construct an adjacency list for each node in the graph. A global counter was maintained to keep track of the total number of nodes in the graph. These steps occur in the

Mapper phase of the Preprocessing job in MapReduce version of the program, where the map function is called on each line, parsing it and writing to HDFS which is read by the next job. It is clear that the Spark version doesn't require any data to be written to the HDFS nor does it need to maintain counters to keep track of the total nodes which is needed for the next job. The Spark version doesn't require a separate job to construct the adjacency list, instead this step is accomplished by one single line clearly demonstrating how Spark leads to less verbose code.

(PageRank initialization and contributions distributed among a nodes outlinks)

The initialization step of page rank values to every node in the graph takes place in the Mapper phase of the new job (PageRank computation job) if the job was running for the first iteration of PageRank. Otherwise, the value of PageRank from the previous iteration is taken into account for calculation of PageRank for that particular iteration. This PageRank job runs continuously for ten iterations. Each incoming line from the constructed adjacency list is read from the HDFS and the outlinks encountered are added to a data structure. If the outlinks are empty i.e it is a dangling node, the page rank values for the corresponding nodes are accumulated to form a delta value. This delta value is once again stored in a global counter in order to make use of it in the Reducer across different iterations as they keep getting updated. If they are not empty, contributions are sent along for each of the outlinks corresponding to a particular node which is given by its pagerank divided by the size of its outlinks.

(Pagerank value aggregation and new page rank value computation)

In the reducer phase, the page rank values are aggregated for the same key and then the page rank formula is applied to compute a new page rank value for the given value of delta and the value of pagerank from the previous iteration. The subtractbyKey() and union performed in Spark is done so that the initial structure of the graph can be recovered and kept intact so that no nodes are lost, both of which are performed in the Reducer phase of the MapReduce version of PageRank.

(TopK Results)

A separate job is used in the Hadoop version to obtain the topK Results in decreasing order. In the mapper phase of this job, we emit those locally topK results from each mapper which is then read by the reducer to output the globally topK results.

Finally, it is easy to conclude based on the above analysis of the MapReduce version of the program that iterative problems lead to increase in verbosity of code, and therefore make Spark more elegant when it comes to handling such problems. In additional, the flexibility and powerfulness of the APIs in Spark results in making them ideal to perform complex operations in a simple way, which would require large number of operations in the case of MapReduce. Also, in the case of MapReduce, it is necessary to have multiple jobs for the pre-processing phase, page rank computation phase and the TopK phase whereas the Spark approach goes about it in a very gracefully manner. Lastly, the fact that a MapReduce job would have to write to HDFS so that the next job can read from it makes them inefficient in comparison to the Spark version as there is a large amount of data that is being transferred from HDFS to Mappers and Mappers to Reducers.

The advantage of the MapReduce approach is that, even though they increase code verbosity, they are easy to visualize and implement in comparison to functional programming in Spark. This can be summed up by mentioning that MapReduce performs a large number of simple operations whereas the Spark version performs fewer operations but most of which are complex.

In MapReduce, the data is distributed over the cluster and processed.

The difference in Spark is that it performs in-memory processing of data. This in-memory processing is a faster process as there is no time spent in moving the data/processes in and out of the disk, whereas MapReduce requires a lot of time to perform these input/output operations thereby increasing latency.

Real-time data can be processed on MapReduce but its speed is nowhere close to that of Spark.

Spark claims to process data 100x faster than MapReduce, while 10x faster with the disks.

While most of these covered the short comings of MapReduce, here are a few shortcomings of Spark.
Its “in-memory” capability can become a bottleneck when it comes to cost-efficient processing of big data.

Does not have its file management system, so you need to integrate with Hadoop, or other cloud based data platform.

Performance Comparison:

	6 m4.large machines	**11 m4.large machines**
Spark Execution Time	1 hr 50 mins	1 hr 05 mins
Hadoop Execution Time	42 mins 12 secs	23 mins 02 secs

As evident from the above table, although ideally Spark should run faster Hadoop, Hadoop ends up having faster execution time for both 6 m4.large and 11 m4.large. This can be explained by the fact that Spark consumes a lot of memory, and issues around memory consumption and garbage collection are not handled in a user friendly manner. It has to be noted that for the simple data set, Hadoop finishes executing in 15 mins while Spark takes only 10 mins. Therefore, it can be argued that with heavy data sets, Spark takes too long to take the data set into memory (in our example, loading the adjacency graph into memory as we cache it) and as it performs lazy evaluation most of the time i.e when the data is not cached, it ends up taking longer to execute.

Top-100 Wikipedia Pages:

Spark Execution:
Simple data set:

(United_States_09d4,0.0039169513674001235)
(Wikimedia_Commons_7b57,0.0029717784069910533)
(Country,0.0024656204995873055)
(England,0.0016460575221108559)
(Europe,0.001641968619116063)
(United_Kingdom_5ad7,0.0016305609828408197)
(Water,0.0016211884046934538)
(Germany,0.0015989831279761388)
(France,0.0015697609115602037)
(Earth,0.0015345884235528494)
(Animal,0.0015246077160271673)
(City,0.0014912812051641163)
(Week,0.0012984316385861651)
(Asia,0.0012158256243519814)
(Sunday,0.0011925884968242513)
(Monday,0.0011747206098511114)
(Wednesday,0.0011628472831638102)
(Wiktionary,0.0011513576873514188)
(Money,0.0011499520204621622)
(Friday,0.0011344423058911)
(Plant,0.0011252560212964568)
(Saturday,0.0011215810116942818)
(Thursday,0.0011070203866967827)
(Tuesday,0.0010991587943733703)
(English_language,0.0010922730150181262)
(Computer,0.001088110486203107)
(Government,0.001074662911425146)
(Italy,0.0010707791152227988)
(India,0.001065319484168118)
(Number,0.001000536136461699)
(Spain,9.74970395980969E-4)
(Day,9.498592621522064E-4)
(Japan,9.393511847940663E-4)
(Canada,9.37315247322181E-4)
(People,9.052845954507006E-4)
(Human,8.847296824980124E-4)

(Wikimedia_Foundation_83d9,8.65886282307737E-4)
(China,8.525218694912798E-4)
(Australia,8.495992625268686E-4)
(Energy,8.437820769452533E-4)
(index,8.437305439039336E-4)
(Sun,8.222562712880822E-4)
(Food,8.165448446976429E-4)
(Science,8.096769114012377E-4)
(Mathematics,7.943697910586104E-4)
(Television,7.534533329730566E-4)
(Capital_(city),7.499854615816616E-4)
(Russia,7.399926695376285E-4)
(State,7.363050107985816E-4)
(Year,7.141528072430708E-4)
(Music,7.096119207165143E-4)
(Language,6.938095711434104E-4)
(Greece,6.928162987403319E-4)
(Scotland,6.861846107356928E-4)
(Wikipedia,6.746258186908348E-4)
(Metal,6.736750727801323E-4)
(Greek_language,6.674789264579284E-4)
(2004,6.62498633807186E-4)
(Planet,6.593872446163026E-4)
(Sound,6.405654183532833E-4)
(Religion,6.372514552971203E-4)
(Africa,6.335000227508006E-4)
(London,6.27503869954384E-4)
(Geography,5.990721083616879E-4)
(Poland,5.982726425593083E-4)
(20th_century,5.969608376054568E-4)
(Law,5.9415680763417E-4)
(Liquid,5.885986783613313E-4)
(19th_century,5.832250595991726E-4)
(World,5.823363489037499E-4)
(Society,5.742543521078996E-4)
(Scientist,5.708110783080367E-4)
(Atom,5.549418130838433E-4)
(History,5.529876457842867E-4)
(Latin,5.508016234981965E-4)
(Sweden,5.455232925664394E-4)

(War,5.437043788290274E-4)
(Light,5.431488136447683E-4)
(Culture,5.366407656580828E-4)
(Netherlands,5.340783456082946E-4)
(Building,5.24666844764102E-4)
(Turkey,5.19826621707203E-4)
(God,5.150355120389613E-4)
(Plural,5.14463662060596E-4)
(Information,5.112972168903008E-4)
(Portugal,4.988315435744951E-4)
(Centuries,4.988026253214741E-4)
(Chemical_element,4.949518423703227E-4)
(Denmark,4.908386832419067E-4)
(Capital_city,4.8819573828310845E-4)
(Inhabitant,4.8673331492248104E-4)
(Austria,4.8497712587650875E-4)
(Cyprus,4.8339983536058894E-4)
(Ocean,4.766700176856042E-4)
(North_America_e7c4,4.711831895001456E-4)
(Moon,4.681832751190429E-4)
(University,4.673879797436611E-4)
(Species,4.659740714451532E-4)
(Disease,4.6549375629881837E-4)
(Book,4.6523261165892706E-4)

Full data set:

(United_States_09d4,0.001277811408428384)
(2006,0.001177256048465967)
(United_Kingdom_5ad7,6.190141335584234E-4)
(2005,5.411645427380356E-4)
(Canada,4.060945922441726E-4)
(France,4.050957734927372E-4)
(Biography,3.972303323291158E-4)
(England,3.9208039965098944E-4)
(2004,3.7163366745698845E-4)
(Germany,3.50438525743235E-4)
(Unicode,3.2858635085038094E-4)
(Australia,3.2547138976955515E-4)
(Latin_alphabet,3.1064635367530544E-4)

(Geographic_coordinate_system,3.0738903196367827E-4)
(English_language,3.002489459684073E-4)
(2003,2.984009890848165E-4)
(International_Phonetic_Alphabet_96f8,2.9283773304448845E-4)
(India,2.898045909054034E-4)
(Japan,2.8975441536526126E-4)
(Italy,2.50106407550077E-4)
(Wiktionary,2.479910155077009E-4)
(2001,2.37751485049847E-4)
(2002,2.3709738598807302E-4)
(Europe,2.3253870223256E-4)
(Internet_Movie_Database_7ea7,2.2536992162624768E-4)
(2000,2.2314092184861798E-4)
(World_War_II_d045,2.2246098900855496E-4)
(London,2.0983668387523138E-4)
(Spain,2.0184968882066187E-4)
(Russia,1.975075044027153E-4)
(1999,1.9695518443286635E-4)
(Wikimedia_Commons_7b57,1.95970457869001E-4)
(Population_density,1.942765137122077E-4)
(Record_label,1.8781575762310236E-4)
(Race_(United_States_Census)_a07d,1.7851413873240735E-4)
(1998,1.704113245550251E-4)
(1945,1.6761315190949122E-4)
(1997,1.635730494390909E-4)
(Z,1.6341938480535093E-4)
(T,1.6025427861637514E-4)
(New_York_City_1428,1.599288702012009E-4)
(Scotland,1.5972650311624843E-4)
(Music_genre,1.58174392581174E-4)
(1984,1.5702631242895483E-4)
(1983,1.568392349013231E-4)
(1981,1.5603999986631842E-4)
(1982,1.5575910916149383E-4)
(Diacritic,1.5413145630794277E-4)
(1975,1.539741860596087E-4)
(1976,1.5316059382421062E-4)
(Paris,1.5285553464754396E-4)
(1985,1.5281783180319224E-4)
(Sweden,1.5271133464049287E-4)

(1996,1.5236087338466077E-4)
(Television,1.5213519602186838E-4)
(1987,1.5195956160061626E-4)
(Portugal,1.5159921610428785E-4)
(Austria,1.5138328681129877E-4)
(Brazil,1.510571744218113E-4)
(Album,1.5048904828769588E-4)
(Actor,1.4988598256180955E-4)
(Studio_album,1.4984068904757113E-4)
(Roman_numerals,1.4971621591747928E-4)
(Football_(soccer),1.4926066559513176E-4)
(Politician,1.4925340484952995E-4)
(Animal,1.4885169679103357E-4)
(Soviet_Union_a1df,1.4815950020440253E-4)
(South_Africa_1287,1.4809239831293695E-4)
(Personal_name,1.4674327882781182E-4)
(China,1.457834616118855E-4)
(1973,1.457457150887025E-4)
(1972,1.4564789950955557E-4)
(Latin,1.4514495916191625E-4)
(French_language,1.4499868804308393E-4)
(1971,1.446096578700351E-4)
(1995,1.4452174479928157E-4)
(California,1.4170787935419257E-4)
(Netherlands,1.4128099036265926E-4)
(Square_mile,1.405285928370884E-4)
(New_Zealand_2311,1.395292028531858E-4)
(Census,1.387006934410429E-4)
(1994,1.372678356540517E-4)
(1991,1.3321290282763094E-4)
(Cyrillic_alphabet,1.3160873436127802E-4)
(Punctuation,1.3015171259377778E-4)
(1993,1.3006233857534242E-4)
(1990,1.2907553346756923E-4)
(Film,1.2868719810322257E-4)
(Public_domain,1.286704392609703E-4)
(Minuscule,1.275959677652872E-4)
(New_York_3da4,1.262148296922881E-4)
(Palaeography,1.2573476198863981E-4)
(Poland,1.2494451066543693E-4)

(1992,1.2476577583736748E-4)
(Norway,1.2424394485138218E-4)
(Alphabets_derived_from_the_Latin_c33b,1.2412313128202753E-4)
(History_of_the_Latin_alphabet_ec67,1.2346886704142662E-4)
(Mathematics,1.2235744717126155E-4)
(Iran,1.2217569490769373E-4)
(Denmark,1.2082980823911644E-4)

MapReduce Execution:
Simple data set:

```
United_States_09d4:  0.005458351352856293
Wikimedia_Commons_7b57:  0.004365927492811352
Country:  0.003543398968855606
England:  0.0024407253491515795
Germany:  0.002380965125324824
Europe:  0.00237663724855411
United_Kingdom_5ad7:  0.0023679249857932274
Water:  0.0023371693279392106
France:  0.0023083966644495187
Animal:  0.002247760926708638
Earth:  0.0022097515921970915
City:  0.0020789290382824607
Week:  0.0018251695235291133
Asia:  0.0017575203048763915
Wiktionary:  0.0017081886805031
Sunday:  0.0016989365027587884
Monday:  0.0016729927772312484
Money:  0.0016679759859938876
Plant:  0.0016583072118842553
Wednesday:  0.001657531881654813
Friday:  0.0016177020974040028
Computer:  0.001610524372712971
Saturday:  0.0015994526544411134
English_language:  0.0015952802116620318
Thursday:  0.0015788453603585554
Italy:  0.0015752137766022205
Tuesday:  0.0015675384087696304
India:  0.001546042031531843
Government:  0.0015114025519524339
Number:  0.0014541077076003323
Spain:  0.0014199002592347367
Japan:  0.0013816755155036744
Day:  0.001336594584541612
People:  0.001314821905928574
Canada:  0.0013052590569882653
Human:  0.0013016762493325501
Wikimedia_Foundation_83d9:  0.001258468032999788
China:  0.0012449984539525997
```

Energy: 0.001219837304181776
Australia: 0.0012182500526570304
Food: 0.0012037604875280911
Science: 0.001179393463618545
Sun: 0.0011767638589870167
Mathematics: 0.0011709791915695347
index: 0.0011491499722276948 Television:
0.001117151880936274
Russia: 0.0010797140725622724
Music: 0.0010545668804916735
Year: 0.0010333828771668966
State: 0.0010300302581225974
Greece: 0.0010204693318833001
Language: 0.0010189503562045607
Capital_(city): 0.0010158517836664663
Scotland: 0.0010095119197330198
Metal: 9.901845957727999E-4
Wikipedia: 9.820742423295E-4
2004: 9.69800974033915E-4
Greek_language: 9.691041095275052E-4
London: 9.416007566927337E-4
Planet: 9.414039576193629E-4
Sound: 9.408390439254908E-4
Religion: 9.362983323975534E-4
Africa: 9.19530032174076E-4
Poland: 8.78267425375345E-4
20th_century: 8.691324817957274E-4
Geography: 8.626849781417545E-4
Law: 8.606866706063362E-4
Liquid: 8.564718543578831E-4
19th_century: 8.48754270783759E-4
World: 8.443068589640385E-4
Scientist: 8.337257210208316E-4
Society: 8.296480233459732E-4
Atom: 8.030029246576257E-4
History: 8.011747465506352E-4
Latin: 8.0029123897156E-4
Sweden: 7.981434947455812E-4
Light: 7.912789822080848E-4
War: 7.876680635409976E-4
Netherlands: 7.861933365863827E-4
Culture: 7.755963053692902E-4
God: 7.590794898430814E-4
Building: 7.554068306302891E-4
Turkey: 7.536018919083421E-4
Inhabitant: 7.464825122008327E-4
Plural: 7.44981035186368E-4
Information: 7.426612302161724E-4
Centuries: 7.327148962651551E-4
Portugal: 7.261256490071358E-4
Chemical_element: 7.257394886785997E-4
Denmark: 7.115370055190203E-4
Austria: 7.083242607692868E-4

```
Cyprus:  6.961411976910087E-4
Species:  6.950367104677497E-4
Book:  6.930899429413559E-4
Disease:  6.909513443532878E-4
University:  6.90131919762503E-4
Ocean:  6.892400615551847E-4
Biology:  6.863661383322514E-4
Capital_city:  6.862677397560723E-4
North_America_e7c4:  6.70863994035089E-4
```

Full data set:

```
United_States_09d4:  0.0010536895282195387
2006:  9.747476951720468E-4
United_Kingdom_5ad7:  5.386421744725589E-4
2005:  4.53936872728234E-4
Biography:  3.8307073123498213E-4
France:  3.4731055861474624E-4
England:  3.4522020339047225E-4
Canada:  3.300336014644149E-4
2004:  3.200369813314975E-4
Germany:  3.016317482282253E-4
Australia:  2.75352661479081E-4
India:  2.582408106633031E-4
2003:  2.5336661323952544E-4
Japan:  2.4251834984754107E-4
Geographic_coordinate_system:  2.1971322631715056E-4 Italy:
2.1501253865354768E-4
Internet_Movie_Database_7ea7:  2.1009493901645758E-4
2002:  2.0698284352624108E-4
2001:  2.0396955863792498E-4
Europe:  2.0163366733447037E-4
London:  1.9073469428723837E-4
World_War_II_d045:  1.8631261302942543E-4
2000:  1.840206098889639E-4
Record_label:  1.804332244787803E-4
English_language:  1.7653755067307316E-4
1999:  1.7315274624915368E-4
Spain:  1.7223300633700673E-4
Wiktionary:  1.699566547162186E-4
Russia:  1.636285383898102E-4
Music_genre:  1.518268303417644E-4
Wikimedia_Commons_7b57:  1.5127854775750732E-4
1998:  1.4994607752877065E-4
Football_(soccer):  1.433393973245959E-4
1997:  1.425673015010558E-4
Scotland:  1.3844263705762715E-4
Television:  1.346843600907152E-4
Sweden:  1.3439114215389354E-4
1996:  1.326985860307239E-4
New_York_City_1428:  1.3043538291118635E-4
1995:  1.2626375279739394E-4
China:  1.2480920383123938E-4
```

Netherlands: 1.2185520783497464E-4
1994: 1.2048442316747488E-4
New_Zealand_2311: 1.1863534773419481E-4
1991: 1.1494277239527325E-4
Public_domain: 1.1458177359541244E-4
Scientific_classification: 1.1404566478786986E-4
1993: 1.1403425802299908E-4
1990: 1.1186712284315937E-4
California: 1.1166500002639193E-4
Film: 1.1141904475000256E-4
Actor: 1.102721460813951E-4
1992: 1.0930719865493796E-4
Poland: 1.0763173489896846E-4
Norway: 1.063714689074772E-4
Population_density: 1.0602261687629597E-4
Ireland: 1.043890690441676E-4
1989: 1.0244401080987395E-4
Latin: 1.0235478846150315E-4
Brazil: 1.0058998935144309E-4
1980: 9.896944731565784E-5
January_1: 9.869113045162882E-5
Album: 9.730053882874919E-5
1986: 9.724069943605147E-5
New_York_3da4: 9.666738370030173E-5
Politician: 9.608201706592814E-5
Mexico: 9.594582657497062E-5
French_language: 9.576727283366687E-5
Record_producer: 9.549546679944901E-5
1985: 9.501850408449496E-5
1982: 9.459418586651086E-5
1979: 9.438177023233195E-5
1981: 9.411168592931413E-5
Paris: 9.408584670626314E-5
1984: 9.281392776893489E-5
1983: 9.246873067424363E-5
1987: 9.246021500839021E-5
1974: 9.237626654883038E-5
South_Africa_1287: 9.178604180887902E-5
Switzerland: 9.079577361505942E-5
Personal_name: 9.042285852931988E-5
1988: 9.040235316441634E-5
1970: 8.957904778988453E-5
1976: 8.950652498674797E-5
1975: 8.873279803260922E-5
Animal: 8.832745342105989E-5
Soviet_Union_ad1f: 8.781725161912768E-5
Greece: 8.770261741221979E-5
1945: 8.717669622760933E-5
1969: 8.691958893279049E-5
1972: 8.66156544038222E-5
1977: 8.630722603877652E-5
1978: 8.600470338267009E-5
Portugal: 8.517152849655552E-5

Austria: 8.439110840993373E-5
1973: 8.435518811491949E-5
Studio_album: 8.421852228247513E-5
Iran: 8.357296772018084E-5
Denmark: 8.347051581970005E-5
1971: 8.278672109717845E-5

Although the value of the PageRank differs slightly, the order in which they appear in the TopK output i.e the pagenames are roughly in the same order. This can be explained that the logic/algorithm followed in Spark version slightly differs with the way MapReduce was implemented, particularly when it comes to handling dangling nodes, secondary dangling nodes and ghost nodes.