**CS5543 Real-Time Big Data Analytics**

**MapReduce & Spark Programing**

ICE-7

10/18/2016

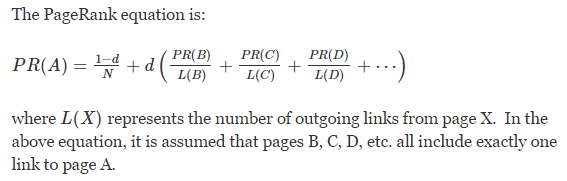
Name: Bill Capps

Class ID:

**MapReduce & Spark Programming - PageRanking**

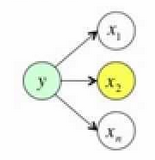
Create a PageRanking for Wikipedia with the use of Hadoop and Spark and Storm. The Wikipedia (en) has 3.7M articles at the moment and is still growing. Each article has many links to other articles. With those incomming and outgoing links we can determine which page is more important than others, which basically is what PageRanking does.

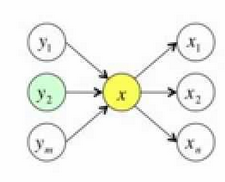
**Hint:**



A damping factor 0<d<1 is a sort of weighted average between the two extremes (typically, a value between 0.5 and 0.85).

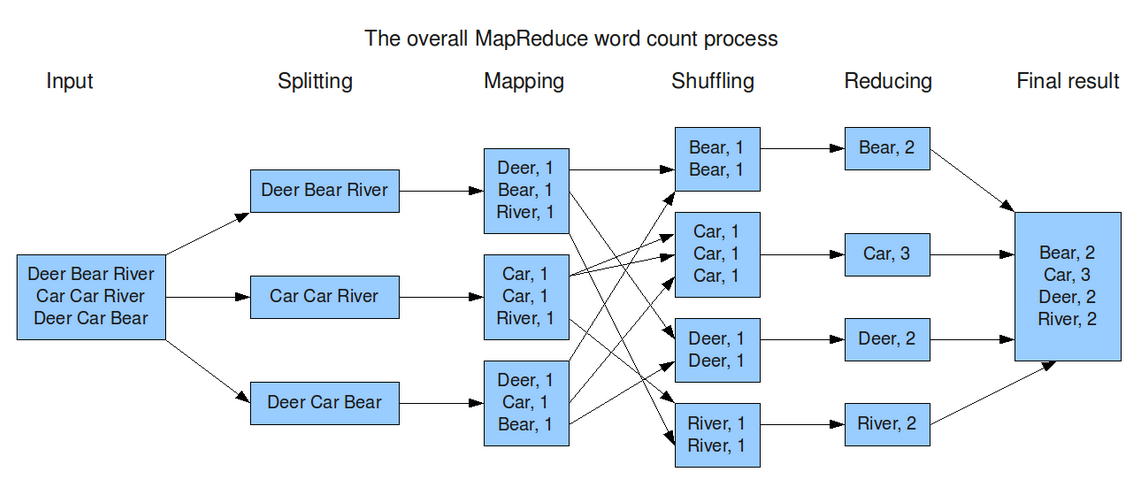
In the Hadoop mapping phase, get the article's name and its outgoing links.  
In the Hadoop reduce phase, get for each wikipage the links to other pages.

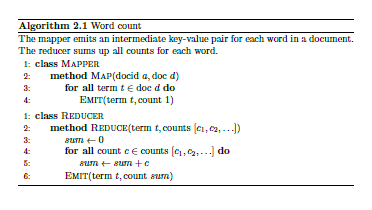




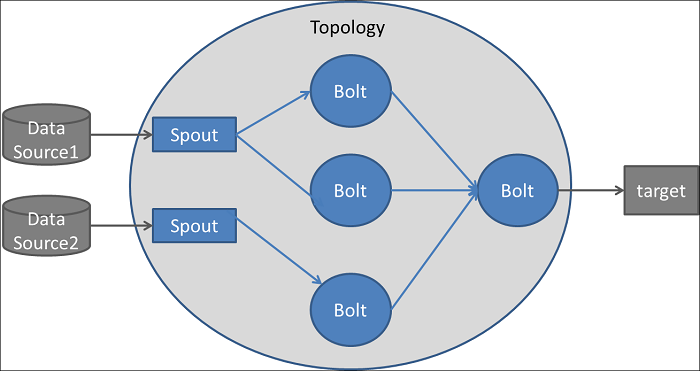
Given the problem description,

1. Draw the MapReduce Diagram (show the steps)
2. Sketch the MapReduce Algorithm.
3. Sketch the Spark Scala Algorithm
4. Draw the Storm Topology

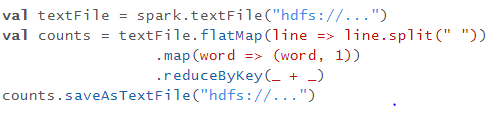




Storm Topology Example



**Spark Scala Code for WordCount**



|  |  |
| --- | --- |
| **flatMap(func)** | Similar to map, but each input item can be mapped to 0 or more output items (so func should return a Seq rather than a single item). |
| **reduceByKey**(*func*, [*numTasks*]) | When called on a dataset of (K, V) pairs, returns a dataset of (K, V) pairs where the values for each key are aggregated using the given reduce function *func*, which must be of type (V,V) => V. Like in groupByKey, the number of reduce tasks is configurable through an optional second argument. |