**Big Data Computing** Lab - III

# **August 17, 2021**horizontal line

1. **Wikipedia data analysis**

Wikipedia contains a vast amount of data. It is possible to make use of this data in computer programs for a variety of purposes. However, the sheer size of Wikipedia makes this difficult. Moreover, accessing data from wikipedia programmatically may generate large volumes of additional traffic which will most likely result in banning of your IP address by Wikipedia. There are a variety of Wikipedia dump files available. We will utilize **map-reduce** programs on an XML file: ‘**input.txt**’, containing information about different Wikipedia articles to calculate the importance of each page through the **PageRank** algorithm. An overview about the associated tags for each record are explained below:

| **Tag name** | **Description** |
| --- | --- |
| title | The title of the page |
| id | The internal Wikipedia ID of the page |
| redirect | What this page redirects to. |
| **Revision** | |
| id | Unique identifier of a revision |
| timestamp | Timestamp for recording the revision (Date and Time value in YYYY-MM-DD HH:MM:SS) |
| **Revision -> Contributor** | |
| id | Unique identifier of registered user who performed this revision |
| comment | Comment inserted by the user who performed this revision |

1. You are provided with the instructions & source code of several map-reduce programs which performs the following task:
   1. **DriverClass.java** : Runs map-reduce jobs to generate the LinkGraph, process PageRank and Cleanup & Sorting. This file then launches the LinkGraph map-reduce jobs for computing the PageRank followed by invoking PageRank.java and finally the PageRankSorting.java map-reduce jobs.
   2. **LinkGraph.java :** Used to extract the outgoing links from each page and calculation of initial pagerank for each node.
   3. **PageRank.java :** Used to compute pagerank and is run for 10 iterations.
   4. **PageRankSorting.java** : Used in sorting the pagerank based upon values.
2. **Temperature analysis with multiple input files**

Given a set of input files, recording temperatures measured by multiple sensors for different time-stamps, the task is to output the maximum temperature corresponding to each date using **map-reduce** programs. An overview of the columns involved include:

| **Column** | **Description** |
| --- | --- |
| SensorId | ID of the temperature sensor |
| Date | Date for recording the measurement (YYYY-MM-DD) |
| Hour | Hour of recording the measurement (HH:MM) |
| Temperature | Corresponding temperature value recorded in appropriate units |

If the first input file contains columns in order <SensorID, Date, Hour,Temperature> while the second in the order <Date, Hour, Temperature, SensorID>, you are provided with 4 source code file(s) performing the following tasks:

* 1. **TotalDriver.java** : Driver class invoking multiple mapper and reducer classes
  2. **TotalMapper1.java & TotalMapper2.java**: Fetches the date and temperature from first & second input files.
  3. **TotalReducer.java :** Extracts the maximum temperature for the corresponding date.