**Finding the co-occurrence of author and their co-authors using Hadoop Map-reduce**

* We have taken the dataset from the open-source computer science bibliography database dblp.org, where we have taken the dataset of publications from springer in 10 different areas.
* We have created a directory in Hadoop file system with the input, where we have stored all the XML files that we have downloaded from dblp.org.

hdfs dfs -mkdir input.

* We have created a java file with mapper class, reducer class and scheduler class which we use to find the co-occurrences of authors.
* We have compiled the auth.java file with the command:

hadoop com.sun.tools.javac.Main Auth.java

* We have created a jar file which contains all the java class files, which we use to run the Hadoop MR program.

hadoop jar '/home/hadoop/Desktop/occurance/occurance.jar' Auth/Occurance/input /Occurance/Output

* In the mapper class we only consider the author names and remove all the unnecessary data.
* And we generate key-value pairs with an author as key and the co-author as value. We pass this key-value pairs to reducer, which aggregates the key-value pairs and generates output as author- co-author, co-author.
* We execute the MR program with the command:

Hadoop jar occurance.jar Auth input output

* To generate output from reducer:

hdfs dfs -cat /Occurance/Output/\*

* We have saved the output that is generated into local directory using the command:

hdfs -get output cat part-r-00000

We used following packages to implement our MR :

import java.io.IOException;

import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IntWritable;

import java.io.ByteArrayInputStream;

import java.io.IOException;

import javax.xml.stream.XMLInputFactory;

import javax.xml.stream.XMLStreamConstants;

import javax.xml.stream.XMLStreamReader;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.FSDataInputStream;

import org.apache.hadoop.fs.FileSystem;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.DataOutputBuffer;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.InputSplit;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.Mapper;

import org.apache.hadoop.mapreduce.RecordReader;

import org.apache.hadoop.mapreduce.Reducer;

import org.apache.hadoop.mapreduce.TaskAttemptContext;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.input.FileSplit;

import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

import java.util.ArrayList;