# Homework 5: MapReduce

Introduction to Big Data Systems course

**Due: October 30, 2022** 23:59 China time. Late submission results in lower (or even no) scores.

For questions or concerns, contact TA (Huanqi Cao, Mingzhe Zhang) by WeChat. Or send an email to caohq18@mails.tsinghua.edu.cn or zmz21@mails.tsinghua.edu.cn if you could not use WeChat.

## Overview

In this assignment, you will read the MapReduce paper and implement a MapReduce program counting the out-degree of each vertex in a graph.

## Tasks

### Task 1 (40%)

Read the MapReduce paper attached, answer the following question:

- **Q1**. The result of mapping are shuffled and sent to reducers, which could cost a lot of network traffic. Propose an approach to address this issue.
- **Q2**. If a mapper or reducer task is too slow, which would make the whole map-reduce task very slow. Propose an approach to address this issue.

### Task 2 (60%)

(correctness 20%; report 40%; performance does not matter as long as it finishes in a reasonable time)

Implement a program that counts the out-degree of all vertices in a graph using MapReduce.

There are 2 graphs to be count. (`case1`, `case2`)

### Task 3 (Optional, bonus up to +10%)

Implement a program that find the top-20 biggest out-degree vertices also using MapReduce.

Note: For `case1`, just find top-2 biggest out-degree vertices.

## Environment

After login to the server, the executable related to this homework such as `hdfs`, `yarn` are located in `/hadoop/bin/` directory. You can use a command like `/hadoop/bin/hdfs [OPTIONS] SUBCOMMAND [SUBCOMMAND OPTIONS]`.

You may optionally add `/hadoop/bin/` to your `PATH` environment variable on the server, so that you can type `hdfs` directly.

```
export PATH=/hadoop/bin/:$PATH
```

# Graph

Out-degree is very easy to understand. It is the number of edges that start from a certain vertex.

For example, there are 4 edges in a graph: `(1, 2) (1, 3) (2, 3) (1, 3)`. So, out-degree(1)=**3**, out-degree(2)=1, out-degree(3)=0.

Note: Treat the duplicated edge as another edge.

# Code

## Java

You can use Java for this homework, for which we supplied the scripts to compile and run the program. If you wish, you may use your own scripts, as long as you use MapReduce and run with Hadoop, HDFS, and Yarn.

We have placed the starter code directory at `/data/hw5_src` on the server. You should copy it to your home (by `cp -r /data/hw5_src ~`).

- `WordCount.java` is a completed program that you can compile and run directly, this program counts the words in the input file. You can use it to learn about MapReduce. The output file's format will be `key value` (such as `aaa 2`) per line. We will show more details about the output in section Sample.
- `OutDegree.java` is an uncompleted program that counts the out-degree of vertices in a graph. You will need to fill in 2 functions: `map()` and `reduce()`.
- `run_wc.sh` is a script to run wordcount example.
- `run_od.sh` is a script to run your out-degree program.

## Other languages

You may use another programming language, as long as you follow the MapReduce programming, and you must be able to compile and run it on the server, with Hadoop, HDFS and Yarn environments.

You should supply your own running script if you use another language. Contact TA if you need something installed on the server.

# Data

The MapReduce program needs to read the input file from HDFS and write the output file to HDFS.

The graph data is located at the HDFS directory `/hw5_data/`

```
/hadoop/bin/hdfs dfs -ls /hw5_data
```

There are 4 files in it:

- `temp.txt`. Used as the sample to test `WordCount.java`.
- `edges.txt`. Used as the sample to test `OutDegree.java`.
- `case1`: |V|=9 (1~9), |E|=100
- `case2`: |V|=999986 (1~999986), |E|=10000000

`case1` & `case2` are the 2 directed graphs for this assignment. The format of these 2 files is `a u v w`. It means there is an edge from `u` to `v`, and its weight is `w`.

The MapReduce program need to read the input file from HDFS and write the output file to HDFS.

You can get these files from HDFS to your local directory:

```
/hadoop/bin/hdfs dfs -get <hdfs_file_path> <local_path>
```

For example,

```
/hadoop/bin/hdfs dfs -get /hw5_data/temp.txt .
```

Also, you can put the data to your own HDFS directory:

```
/hadoop/bin/hdfs dfs -put <local_file> <hdfs_directory>
```

For example,

```
/hadoop/bin/hdfs dfs -put temp.txt .
```

You may need to learn some other commands of `hdfs` by yourself.

Note: Pay attention to the 1 GB file size limit. The total size of your home directory plus your directory in HDFS should not exceed 1 GB.

# Sample

## Wordcount

Input file `temp.txt`, a file consist of 4 lines of words:

```
aaa
bbb
ccc
aaa
```

Output file `{output_path_you_defined}/part-r-00000`

```
aaa 2
bbb 1
ccc 1
```

## Graph

Input file `edges.txt`

```
a 1 2 0
a 3 4 0
a 5 1 0
a 2 4 0
a 4 5 0
a 2 5 0
a 2 3 0
a 3 2 0
```

Output after part 1 (Run `OutDegree`)

```
2 3
3 2
5 1
4 1
1 1
```

# Hand-in

Please submit your assignment containing your PDF report and code. Pack everything in a ZIP file. There is no strict format restrictions for this homework.

Please describe your solution in detail in your report. Besides, please tell us how to run your program successfully (e.g. run the provided `run_od.sh` scripts, or supply your own script and describe the usage).