Big Data
- Coding Setup

Dr. Qing “Matt” Zhang
ITU

Outline

- WordCount
- Setup env
- Input/output
- Code and compile
- Create jar and run

WordCount

- “Hello World” of MapReduce programming
- Count the number of occurrences of each word in a file
- Single file contains Mapper, Reducer, and driver program
- Takes input and writes output to/from hdfs

Environment Setup

- Confirm that Hadoop is setup and running

```
[hduser@localhost ~]$ jps
3840 DataNode
4104 SecondaryNameNode
3690 NameNode
22922 ResourceManager
26636 Jps
23054 NodeManager
```

Prepare input/output

- Create the code directory
  - *cd*
  - *mkdir develop*
- Create the local input/output directory
  - *mkdir input*
  - *mkdir output*
  - Copy the WCTestInput file into the input directory

Load input files to hdfs

- *hadoop fs -mkdir /wcinput*
- *hadoop fs –put input/WCTestInput /wcinput/*

Edit the code and compile

- Edit the WordCount.java file
  - *vi WordCount.java*
  - Copy the content from my uploaded *WordCount.java* file
- Compile the code
  - *javac -classpath `hadoop classpath` WordCount.java*
    - Use back quotes
    - This runs the'hadoop classpath' command and uses its' output as part of the javac command

Create jar file and run

- Create jar file
  - *jar cvf wc.jar *.class*
- Run the binary
  - *hadoop jar wc.jar WordCount /wcinput /wcoutput*
- Check output

hadoop fs -ls /wcoutput

```
Found 2 items
-rw-r--r--   1 hduser supergroup          0 2015-08-13 02:28 /wcoutput/_SUCCESS
-rw-r--r--   1 hduser supergroup        129 2015-08-13 02:28 /wcoutput/part-r-00000
```

hadoop fs -cat /wcoutput/part-r-00000

```
Count    1
Hadoop   2
Let      1
…
```