Week-3

Java Primitive	Writeable Implementation
Boolean	BooleanWritable
Byte	ByteWritable
Int	IntWritable
Short	ShortWritable
Long	LongWritable
Double	DoubleWritable

Java Class	Writeable
String	Text

MapReduce

Input → Mapper() → Shuffle() & Sort() → Reducer() → Output

Sudoku Problem

- vi solve_this.txt
- hdfs dfs -put /home/hdoop/solve_this.txt /Sudoku/
- /home/hdoop/hadoop-3.3.4/bin/hadoop jar '/home/hdoop/hadoop-3.3.4/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.4.jar' sudoku Solve_this.txt
- /home/hdoop/hadoop-3.3.4/bin/hadoop jar '/home/hdoop/hadoop-3.3.4/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.4.jar' /Sudoku/solve_this.txt /Sudoku/Wcount_result1
- hdfs dfs -cat /Sudoku/wcount_result1/*

Step 1: Create an input file on the local system. (Sudoku & word count)
Step 2: Copy the file from local system to hadoop cluster
Step 3: Access the jar file present on the local system
Step 4: Give the file on hadoop cluster as the input file
Step 5: Create an output directory on Hadoop Cluster
Step 6: Cat the output directory to see the output

MapReduce Architecture & Process

Local System — Hadoop Cluster

① Create Input File — Copy → Paste here

② Locate & Access Jar File — Input → Above copied file

or

③ Create a Jar File
↳ Java Mapper & Reducer Programs (File name.java)
[Compile to create jar ←
↳ $ javac cp 'hadoop classpath' filename.java

④ Run MapReduce Program → Create output directory

⑤ Cat the output directory [Hdfs dfs - cat 'path/*']

Word Count Problem (Using own jar file)
Create Input File
↳ Move file to Hadoop
↳ Write Java Mapper & Reducer Program
↓
Compile the code to create a jar file
↓
Put the jar file in a directory
↓
Run MapReduce
↓
Cat output files.

- vi wc.txt
  Hello World
  Hello Class
- hdfs dfs -put /home/hdoop/wc.txt /batch3/20030848
- wordcount_classes - WordCount.java
- javac cp 'hadoop classpath' wordcount_classes WordCount.java
- jar -cvf wordcount.jar -c wordcount_class/*
- hadoop jar wordcount.jar . /wc.txt wc_output
- hdfs dfs -cat wc_output/*

  Hello 2
  Class 1
  World 1