**Hadoop vs Java Wordcount**

Repositori ini berisi panduan instalasi Hadoop di sistem operasi Linux, terutama Ubuntu 22.04. Hadoop digunakan untuk menjalankan program Wordcount yang berfungsi untuk menghitung jumlah kata dalam sebuah file teks. Nantinya, program Wordcount Hadoop akan dibandingkan dengan program Wordcount menggunakan Java tanpa Hadoop.

**Penginstalan Hadoop pada Ubuntu 22.04**

1. Install java

sudo apt install openjdk-8-jdk

1. Pindah direktori ke java

Cd /usr/lib/jvm

1. Konfigurasi open.bashrc

export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64   
export PATH=$PATH:/usr/lib/jvm/java-8-openjdk-amd64/bin   
export HADOOP\_HOME=~/hadoop-3.2.3/   
export PATH=$PATH:$HADOOP\_HOME/bin   
export PATH=$PATH:$HADOOP\_HOME/sbin   
export HADOOP\_MAPRED\_HOME=$HADOOP\_HOME   
export YARN\_HOME=$HADOOP\_HOME   
export HADOOP\_CONF\_DIR=$HADOOP\_HOME/etc/hadoop   
export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_HOME/lib/native   
export HADOOP\_OPTS="-Djava.library.path=$HADOOP\_HOME/lib/native"   
export HADOOP\_STREAMING=$HADOOP\_HOME/share/hadoop/tools/lib/hadoop-streaming-3.2.3.jar  
export HADOOP\_LOG\_DIR=$HADOOP\_HOME/logs   
export PDSH\_RCMD\_TYPE=ssh

1. Install SSH

sudo apt-get install ssh

1. Download Hadoop 3.2.2

https://hadoop.apache.org/release/3.2.2.html

1. Extract file

tar -zxvf ~/Downloads/hadoop-3.2.3.tar.gz

1. Pindah ke folder Hadoop

cd hadoop-3.2.3/etc/hadoop

1. Melakukan konfigurasi hadoop-env.h

now open hadoop-env.h

sudo nano hadoop-env.h

1. Cari JAVA\_HOME dan lakukan perubahan

JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64

1. Melakukan konfigurasi pada file

cd etc/hadoop

core-site.xml

<configuration>   
<property>   
<name>fs.defaultFS</name>   
<value>hdfs://localhost:9000</value> </property>   
<property>   
<name>hadoop.proxyuser.dataflair.groups</name> <value>\*</value>   
</property>   
<property>   
<name>hadoop.proxyuser.dataflair.hosts</name> <value>\*</value>   
</property>   
<property>   
<name>hadoop.proxyuser.server.hosts</name> <value>\*</value>   
</property>   
<property>   
<name>hadoop.proxyuser.server.groups</name> <value>\*</value>   
</property>   
</configuration>

hdfs-site.xml

<configuration>   
<property>   
<name>dfs.replication</name>   
<value>1</value>   
</property>   
</configuration>

mapred-site.xml

<configuration>   
<property>   
<name>mapreduce.framework.name</name> <value>yarn</value>   
</property>   
<property>  
<name>mapreduce.application.classpath</name>   
  
<value>$HADOOP\_MAPRED\_HOME/share/hadoop/mapreduce/\*:$HADOOP\_MAPRED\_HOME/share/hadoop/mapreduce/lib/\*</value>   
</property>   
</configuration>

yarn-site.xml

<configuration>   
<property>   
<name>yarn.nodemanager.aux-services</name>   
<value>mapreduce\_shuffle</value>   
</property>   
<property>   
<name>yarn.nodemanager.env-whitelist</name>   
  
<value>JAVA\_HOME,HADOOP\_COMMON\_HOME,HADOOP\_HDFS\_HOME,HADOOP\_CONF\_DIR,CLASSPATH\_PREP END\_DISTCACHE,HADOOP\_YARN\_HOME,HADOOP\_MAPRED\_HOME</value>   
</property>   
</configuration>

1. Jalankan ssh

ssh localhost   
ssh-keygen -t rsa -P '' -f ~/.ssh/id\_rsa   
cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys   
chmod 0600 ~/.ssh/authorized\_keys   
hadoop-3.2.3/bin/hdfs namenode -format

1. Format file sistem

export PDSH\_RCMD\_TYPE=ssh

1. Menjalankan Hadoop

start-all.sh

**Menjalakan wordcount pada Hadoop**

1. Format

hadoop-3.2.3/bin/hdfs namenode -format

1. Jalankan hadoop

start-all.sh

1. Buat direktori input

hadoop fs -mkdir /input

1. Siapkan file yang ingin dihitung, kemudian pindah file tersebut ke direktori input

hadoop fs -put text.txt /input

1. Buat program java untuk wordcount dan simpan dengan nama WordCount.java

sudo nano WordCount.java

1. Contoh program java untuk wordcount

**import** **java.io.IOException**;

**import** **org.apache.hadoop.conf.Configuration**;

**import** **org.apache.hadoop.fs.Path**;

**import** **org.apache.hadoop.io.IntWritable**;

**import** **org.apache.hadoop.io.LongWritable**;

**import** **org.apache.hadoop.io.Text**;

**import** **org.apache.hadoop.mapreduce.Job**;

**import** **org.apache.hadoop.mapreduce.Mapper**;

**import** **org.apache.hadoop.mapreduce.Reducer**;

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

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

**public** **class** **WordCount** {

// Map function

**public** **static** **class** **MyMapper** **extends** Mapper<LongWritable, Text, Text, IntWritable>{

**private** Text word = **new** Text();

**public** **void** **map**(LongWritable key, Text value, Context context)

**throws** IOException, InterruptedException {

// Splitting the line on spaces

String[] stringArr = value.toString().split("\\s+");

**for** (String str : stringArr) {

word.set(str);

context.write(word, **new** IntWritable(**1**));

}

}

}

// Reduce function

**public** **static** **class** **MyReducer** **extends** Reducer<Text, IntWritable, Text, IntWritable>{

**private** IntWritable result = **new** IntWritable();

**public** **void** **reduce**(Text key, Iterable<IntWritable> values, Context context)

**throws** IOException, InterruptedException {

**int** sum = **0**;

**for** (IntWritable val : values) {

sum += val.get();

}

result.set(sum);

context.write(key, result);

}

}

**public** **static** **void** **main**(String[] args) **throws** Exception{

Configuration conf = **new** Configuration();

Job job = Job.getInstance(conf, "WC");

job.setJarByClass(WordCount.class);

job.setMapperClass(MyMapper.class);

job.setReducerClass(MyReducer.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

FileInputFormat.addInputPath(job, **new** Path(args[**0**]));

FileOutputFormat.setOutputPath(job, **new** Path(args[**1**]));

System.exit(job.waitForCompletion(**true**) ? **0** : **1**);

}

}

1. Export classpath

export HADOOP\_CLASSPATH=$($HADOOP\_HOME/bin/hadoop classpath)

1. Buat folder untuk menyimpan hasil WordCount.java

sudo mkdir WordCountCompiled

1. Ubah permission pada folder tersebut

sudo chmod -R 777 WordCountCompiled

1. Compile WordCount.java

javac -classpath $HADOOP\_CLASSPATH -d WordCountCompiled/WordCount.java

1. Mengubah file executable .jar

jar -cvf WordCount.jar -C WordCountCompiled/.

1. Menjalankan jar tersebut untuk menghitung jumlah kata pada file text.txt

hadoop jar WordCount.jar WordCount /input/text.txt /WordCount-Result

1. Melihat hasil perhitungan wordcount

hadoop fs -cat /WordCount-Result/part-r-00000

**Menjalankan program wordcount java tanpa hadoop**

1. Buka file WordCount.java di atas

A screenshot of a computer program

Description automatically generated with medium confidence

1. Ubah filePath sesuai letak file yang ingin dihitung

A screenshot of a computer program

Description automatically generated with medium confidence

1. Klik tombol panah hijau pada public class Main

A screenshot of a computer program

Description automatically generated with medium confidence

1. Ubah konfigurasi saat menjalankan program

A screenshot of a computer program

Description automatically generated with medium confidence

1. Tambahkan perintah -Xms16g

A screenshot of a computer

Description automatically generated

-Xms16g agar program java tersebut dapat menjalankan program yang dapat memakai hingga 16 GB RAM. Digunakan untuk menghindari error outofmemory.

1. Jalankan program

**Referensi**

<https://codewitharjun.medium.com/install-hadoop-on-ubuntu-operating-system-6e0ca4ef9689>

<https://www.youtube.com/watch?v=Slbi-uzPtnw>

<http://malifauzi.lecture.ub.ac.id/2019/04/tutorial-pembuatan-program-wordcount-pada-hadoop-menggunakan-java/>

<https://www.i3s.unice.fr/~jplozi/hadooplab_lsds_2015/datasets/>