Hands on Hadoop

Daya Adianto

(credit : Samuel Louvan, Remmy Augusta Menzata Zen)

Tujuan:

1. Mampu melakukan perintah-perintah dasar operasi file di HDFS
2. Mampu mengeksekusi beberapa program contoh MapReduce dan menginspeksi hasilnya

Dalam kegiatan praktek ini, Anda akan mencoba untuk mengeksekusi perintah-perintah dasar di *command line* untuk berinteraksi dengan HDFS (Hadoop Distributed File System). Selain itu, Anda akan mencoba untuk menjalankan contoh program yang sudah ada dalam instalasi *default* Hadoop.

Dalam praktek ini, Anda harus *login* terlebih dahulu karena Anda akan mengeksekusi program/perintah di *remote server*.

Operasi Dasar Operasi *File* pada HDFS

Apabila Anda familiar dengan sistem seperti Linux, pada dasarnya perintah yang digunakan oleh HDFS tidak jauh berbeda dengan perintah pada Linux. Namun diawali dengan **hadoop fs -[perintah]**. Untuk melihat referensi yang lebih lengkap silakan cek https://hadoop.apache.org/docs/r2.7.3/hadoop-project-dist/hadoopcommon/FileSystemShell.html

1. Untuk menampilkan berkas-berkas pada HDFS Anda gunakan perintah
   hadoop fs -ls
   Contoh tampilan :
   

2. Buatlah sebuah file baru **test.txt** di komputer Anda. Isi **test.txt** bisa acak, atau berupa teks yang Anda ambil dari sebuah artikel/berita. Kemudian unggah file **test.txt** ke server menggunakan aplikasi FTP. Catat lokasi dimana Anda mengunggah file tersebut di server.

3. Untuk mengkopi berkas test.txt dari lokal (server) ke HDFS gunakan perintah

hadoop fs -copyFromLocal test.txt
Atau bisa juga menggunakan perintah
hadoop fs -put test.txt

Contoh tampilan ls:

```
bd@master:~$ hadoop fs -ls
Found 1 items
-rw-r--r--   1 bd supergroup         38 2017-07-20 12:18 test.txt
bd@master:~$
```

4. Untuk mengkopi berkas test.txt dari HDFS ke HDFS gunakan perintah
   hadoop fs -cp test.txt test2.txt
   Contoh tampilan:

```
bd@master:~$ hadoop fs -cp test.txt test2.txt
bd@master:~$ hadoop fs -ls
Found 2 items
-rw-r--r--   1 bd supergroup         38 2017-07-20 12:18 test.txt
-rw-r--r--   1 bd supergroup         38 2017-07-20 12:19 test2.txt
bd@master:~$
```

5. Untuk mengkopi berkas **test.txt** dari HDFS ke lokal gunakan perintah
   hadoop fs -copyToLocal test2.txt test2.txt

   Bisa juga menggunakan perintah
   hadoop fs -get test2.txt

   Contoh tampilan:

```
bd@master:~$ hadoop fs -copyToLocal test2.txt test2.txt
bd@master:~$ ls
alfan  datanode  derby.log  hadoiq  metastore_db  namenode  samuel  test2.txt  test.txt
```

6. Untuk menghapus berkas test2.txt di HDFS gunakan perintah
   hadoop fs -rm test2.txt

   Contoh tampilan:

```
bd@master:~$ hadoop fs -rm test2.txt
17/07/20 12:22:05 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 0
minutes, Emptier interval = 0 minutes.
Deleted test2.txt
bd@master:~$ ls
alfan  datanode  derby.log  hadaiq  metastore_db  namenode  samuel  test2.txt  test.txt
bd@master:~$
```

LATIHAN

Operasi-operasi dasar lain juga berlaku untuk HDFS

a. Cobalah untuk membuat sebuah folder bernama **test**

b. Pindahkan **test.txt** ke dalam folder tersebut

c. Tampilkan isi **test.txt**

Menjalankan Aplikasi di Hadoop

1. Untuk melihat jenis aplikasi contoh yang ada di Hadoop gunakan perintah **hadoop jar /usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar**
   Contoh tampilan:

```
bd@master:~$ hadoop jar /opt/hadoop-2.7.3/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.3.jar
An example program must be given as the first argument.
Valid program names are:
  aggregatewordcount: An Aggregate based map/reduce program that counts the words in the input files.
  aggregatewordhist: An Aggregate based map/reduce program that computes the histogram of the words in
the input files.
  bbp: A map/reduce program that uses Bailey-Borwein-Plouffe to compute exact digits of Pi.
  dbcount: An example job that count the pageview counts from a database.
  distbbp: A map/reduce program that uses a BBP-type formula to compute exact bits of Pi.
  grep: A map/reduce program that counts the matches of a regex in the input.
  join: A job that effects a join over sorted, equally partitioned datasets
  multifilewc: A job that counts words from several files.
  pentomino: A map/reduce tile laying program to find solutions to pentomino problems.
  pi: A map/reduce program that estimates Pi using a quasi-Monte Carlo method.
  randomtextwriter: A map/reduce program that writes 10GB of random textual data per node.
  randomwriter: A map/reduce program that writes 10GB of random data per node.
  secondarysort: An example defining a secondary sort to the reduce.
  sort: A map/reduce program that sorts the data written by the random writer.
  sudoku: A sudoku solver.
  teragen: Generate data for the terasort
  terasort: Run the terasort
  teravalidate: Checking results of terasort
  wordcount: A map/reduce program that counts the words in the input files.
  wordmean: A map/reduce program that counts the average length of the words in the input files.
  wordmedian: A map/reduce program that counts the median length of the words in the input files.
  wordstandarddeviation: A map/reduce program that counts the standard deviation of the length of the w
ords in the input files.
bd@master:~$
```

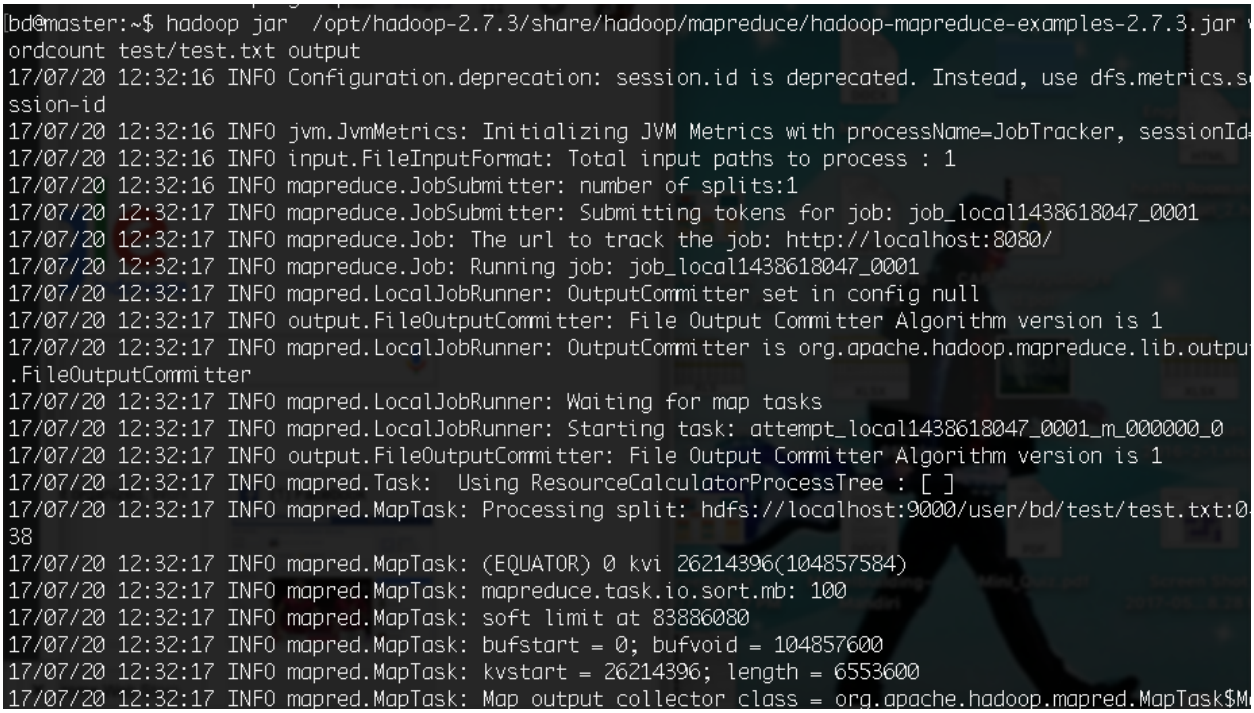
2. Kita akan mencoba menjalankan aplikasi WordCount terlebih dahulu. Untuk melihat cara penggunaan gunakan perintah
   hadoop jar /opt/hadoop-2.7.3/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.3.jar

wordcount

3. Diketahui bahwa aplikasi WordCount membutuhkan 1 (atau lebih) berkas masukan dan direktori keluaran. Perhatikan bahwa berkas di sini adalah berkas yang ada di HDFS **bukan pada direktori lokal**. Oleh karena itu kita akan menggunakan test.txt di awal sebagai masukan. Untuk menjalankan gunakan perintah

   hadoop jar /opt/hadoop-2.7.3/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.3.jar wordcount test/test.txt output

   Contoh potongan tampilan:



4. Dengan menggunakan perintah yang sudah dipelajari sebelumnya, kopi isi direktori tersebut ke filesystem lokal Anda.
5. Lihat isi dari berkas-berkas tersebut, apakah isinya?
6. **Selamat Anda telah menjalankan aplikasi pertama Anda di Hadoop.**

LATIHAN

1. Coba gunakan contoh lain seperti **pi** dan lihat hasilnya. Contoh ini melakukan estimasi dari nilai pi (π) dengan margin error tertentu. Error ini bergantung pada banyaknya sample yang kita miliki. Semakin banyak sample, estimasi semakin akurat. Jalankan contoh tersebut dengan format perintah berikut:

```
hadoop jar
/opt/hadoop/hadoop-2.7.3/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.3.jar
pi <jumlah_map> <jumlah_sample>
```

<jumlah_map> adalah jumlah mapper jobs dan <num_samples> adalah jumlah sample, misal jika kita kita menggunakan 10 mapper dan 5 sample maka command yang digunakan adalah:

```
hadoop jar
/opt/hadoop/hadoop-2.7.3/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.3.jar
pi 10 5
```

Cobalah kombinasi berikut untuk jumlah mapper dan jumlah sample dan observasi *running time* dan keakuratan estimasi.

Jumlah Map	**Jumlah Sample**	**Waktu (s)**	**Nilai π**
2	10	1.394	3.800000000
5	10	1.301	3.28000000
10	10	1.376	3.20000000
2	100	1.287	3.12000000
10	100	1.317	3.140000

2. Coba gunakan wordcount untuk masukan yang berupa folder
3. Coba gunakan wordcount dengan file dengan ukuran yang lebih besar:
   Anda bisa menggunakan file yang terdapat di direktori :
   /home/bd/samuel/gutenberg
   Copy direktori tersebut ke HDFS Anda lalu eksekusi WordCount seperti biasa. Kemudian lihat hasilnya di direktori output Anda.