Cloud Computing

Hadoop Lab

Lab1: HDFS operations

Part I HDFS Shell basic operation (POSIX-like)

Lab Objective

This part make you be familiar with the operations of HDFS via its' shell commands.

Lab Steps

Step 1:

Make sure that your Hadoop cluster has already set up, and the HDFS is running.

```
$ bin/start-dfs.sh
```

Step 2:

In the Hadoop home directory. Enter the following command:

```
$ bin/hadoop fs
```

And you can see the following message:

```
hadoop@[illegible]:/opt/hadoop$ bin/hadoop fs
Usage: java FsShell
           [-ls <path>]
           [-lsr <path>]
           [-du <path>]
           [-dus <path>]
           [-count[-q] <path>]
           [-mv <src> <dst>]
           [-cp <src> <dst>]
           [-rm [-skipTrash] <path>]
           [-rmr [-skipTrash] <path>]
           [-expunge]
           [-put <localsrc> ... <dst>]
           [-copyFromLocal <localsrc> ... <dst>]
           [-moveFromLocal <localsrc> ... <dst>]
           [-get [-ignoreCrc] [-crc] <src> <localdst>]
           [-getmerge <src> <localdst> [addnl]]
           [-cat <src>]
           [-text <src>]
           [-copyToLocal [-ignoreCrc] [-crc] <src> <localdst>]
           [-moveToLocal [-crc] <src> <localdst>]
           [-mkdir <path>]
           [-setrep [-R] [-w] <rep> <path/file>]
           [-touchz <path>]
           [-test -[ezd] <path>]
           [-stat [format] <path>]
           [-tail [-f] <file>]
           [-chmod [-R] <MODE[,MODE]... | OCTALMODE> PATH...]
           [-chown [-R] [OWNER][:[GROUP]] PATH...]
           [-chgrp [-R] GROUP PATH...]
           [-help [cmd]]
```

There are HDFS shell commands (POSIX-like) that you can use to operate the

HDFS.

Step 3:

Upload the data to the HDFS.

```
$ bin/hadoop fs -mkdir input
$ bin/hadoop fs -put conf/ input/
```

To check out if the data is put in the input directory already.

```
$ bin/hadoop fs -ls input/
```

```
hadoop@ csmal :/opt/hadoop$ bin/hadoop fs -ls input
Found 13 items
-rw-r--r--   1 hadoop supergroup       3936 2010-10-05 14:25 /user/hadoop/i
nput/capacity-scheduler.xml
-rw-r--r--   1 hadoop supergroup        535 2010-10-05 14:25 /user/hadoop/i
nput/configuration.xsl
-rw-r--r--   1 hadoop supergroup        464 2010-10-05 14:25 /user/hadoop/i
nput/core-site.xml
-rw-r--r--   1 hadoop supergroup       2237 2010-10-05 14:25 /user/hadoop/i
nput/hadoop-env.sh
-rw-r--r--   1 hadoop supergroup       1245 2010-10-05 14:25 /user/hadoop/i
nput/hadoop-metrics.properties
-rw-r--r--   1 hadoop supergroup       4190 2010-10-05 14:25 /user/hadoop/i
nput/hadoop-policy.xml
-rw-r--r--   1 hadoop supergroup        259 2010-10-05 14:25 /user/hadoop/i
nput/hdfs-site.xml
-rw-r--r--   1 hadoop supergroup       2815 2010-10-05 14:25 /user/hadoop/i
nput/log4j.properties
-rw-r--r--   1 hadoop supergroup        277 2010-10-05 14:25 /user/hadoop/i
nput/mapred-site.xml
-rw-r--r--   1 hadoop supergroup         10 2010-10-05 14:25 /user/hadoop/i
nput/masters
-rw-r--r--   1 hadoop supergroup         10 2010-10-05 14:25 /user/hadoop/i
nput/slaves
-rw-r--r--   1 hadoop supergroup       1243 2010-10-05 14:25 /user/hadoop/i
nput/ssl-client.xml.example
-rw-r--r--   1 hadoop supergroup       1195 2010-10-05 14:25 /user/hadoop/i
nput/ssl-server.xml.example
```

Run the mapreduce job:

```
$ bin/hadoop jar hadoop-0.20.2-examples.jar wordcount input output
```

Then retrieve the result from the HDFS, you have two choices:

(1) `$ bin/hadoop fs -cat output/part-r-00000`

(2) `$ bin/hadoop fs -copyToLocal output/ output/`

And then, you can see the output directory in the path /opt/hadoop/

Step4:

And you can also delete the file/directory by HDFS shell command. For example, to delete the file output/part-r-00000:

```
$ bin/hadoop fs -rm output/part-r-00000
$ bin/hadoop fs -ls output/
```

And then delete the directory output:

```
$ bin/hadoop fs -rmr output
```

```
$ bin/hadoop fs -ls
```

Part II Java Program (using APIs)

Lab Objective

In this part, you must learn how to write simple programs using Hadoop APIs to interact with HDFS.

Lab Steps

Exersice1: Create a file on HDFS.

Step1: Modify the .profile in the user: hadoop home directory.

In order to link to the hadoop library, you need to add the CLASSPATH used by java compiler.

```
$ vim ~/.profile
```

Add the following lines at the end of the file.

```
...
CLASSPATH=/opt/hadoop/hadoop-0.20.2-core.jar
export CLASSPATH
```

And then do relogin.

```
$ exit
$ su - hadoop
```

Step2: Open a java file.

Create a file named “ex01.java”.

```
$ vim ex01.java
```

Paste the following program to it.

```
import org.apache.hadoop.conf.*;
import org.apache.hadoop.fs.*;

public class ex01 {
        public static void main(String[] args){
                if(args.length < 2){
                        System.out.print("Need two arguments\n");
                        System.out.flush();
                        return;
                }
```

```
                System.out.print("Creat File: " + args[0] + "\n");
                System.out.print("Content: " + args[1] + "\n");
                System.out.flush();

                try {
                        Configuration conf = new Configuration();
                        FileSystem hdfs = FileSystem.get( conf );

                        Path filepath = new Path( args[0] );
                        FSDataOutputStream output = hdfs.create( filepath );

                        byte[] buff = args[1].getBytes();
                        output.write(buff, 0, buff.length);
                } catch (Exception e) {
                        e.printStackTrace();
                }
        }
}
```

Step3: Compile ex01.java

```
$ javac ex01.java
```

Note: If the project is too large, you may need to pack you program into jar file. You can do this by the following command.

```
$ mkdir ex01
$ mv ex01.class ex01/
$ jar cvf ex01.jar -C ex01/ .
```

Step4: Run the program.

Run the program on hadoop.

```
$ bin/hadoop ex01 test “Hello World”
```

Then you can check the content of test on the HDFS.

```
$ bin/hadoop fs -cat test
```

Note: If you pack your program in a jar file. You can run it by this command:

```
$ bin/hadoop jar ex01.jar ex01 test “Hello World”
```

Exercise2: Upload a local file/directory to HDFS

Step1: Open a java file.

Create a file named “ex02.java”.

$ vim ex02.java

Paste the following program to it.

```
import org.apache.hadoop.conf.*;
import org.apache.hadoop.fs.*;

public class ex02 {
    public static void main( String[] args){
        if ( args.length < 2){
            System.out.print("Need two arguments\n");
            System.out.flush();
            return;
        }
        System.out.print("Source File: " +args[0]+ "\n");
        System.out.print("Destination File: " +args[1]+ "\n");
        System.out.flush();

        try {
            Configuration conf = new Configuration();
            FileSystem hdfs = FileSystem.get(conf);

            Path srcPath = new Path(args[0]);
            Path dstPath = new Path(args[1]);

            hdfs.copyFromLocalFile(srcPath, dstPath);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Step2: Compile ex02.java

$ javac ex02.java

Step3: Run the program.

Run the program on hadoop.

$ bin/hadoop ex02 conf/ input2/

```
hadoop@                    :/opt/hadoop$ bin/hadoop jar ex02.jar ex02 conf/ in
put2/
Source File: conf/
Destination File: input2/
```

Then you can check the content of test on the HDFS.

$ bin/hadoop fs -ls input2

```
hadoop@                :/opt/hadoop$ bin/hadoop fs -ls input2
Found 13 items
-rw-r--r--   1 hadoop supergroup       3936 2010-10-19 22:22 /user/hadoop/i
nput2/capacity-scheduler.xml
-rw-r--r--   1 hadoop supergroup        535 2010-10-19 22:22 /user/hadoop/i
nput2/configuration.xsl
-rw-r--r--   1 hadoop supergroup        464 2010-10-19 22:22 /user/hadoop/i
nput2/core-site.xml
-rw-r--r--   1 hadoop supergroup       2237 2010-10-19 22:22 /user/hadoop/i
nput2/hadoop-env.sh
-rw-r--r--   1 hadoop supergroup       1245 2010-10-19 22:22 /user/hadoop/i
nput2/hadoop-metrics.properties
-rw-r--r--   1 hadoop supergroup       4190 2010-10-19 22:22 /user/hadoop/i
nput2/hadoop-policy.xml
-rw-r--r--   1 hadoop supergroup        259 2010-10-19 22:22 /user/hadoop/i
nput2/hdfs-site.xml
-rw-r--r--   1 hadoop supergroup       2815 2010-10-19 22:22 /user/hadoop/i
nput2/log4j.properties
-rw-r--r--   1 hadoop supergroup        277 2010-10-19 22:22 /user/hadoop/i
nput2/mapred-site.xml
-rw-r--r--   1 hadoop supergroup         10 2010-10-19 22:22 /user/hadoop/i
nput2/masters
-rw-r--r--   1 hadoop supergroup         10 2010-10-19 22:22 /user/hadoop/i
nput2/slaves
-rw-r--r--   1 hadoop supergroup       1243 2010-10-19 22:22 /user/hadoop/i
nput2/ssl-client.xml.example
-rw-r--r--   1 hadoop supergroup       1195 2010-10-19 22:22 /user/hadoop/i
nput2/ssl-server.xml.example
```