215229140

Exercise 02a: Map Reduce applications for Word Counting

Previous exercise described how to save input file in to HDFS. This exercise train students to do MapReduce process using word counting application.

Prerequisites

Ensure that Hadoop is installed, configured and is running. More details: Single

Node Setup for first-time users.

Cluster Setup for large, distributed clusters.

MapReduce Overview

Hadoop MapReduce is a software framework for easily writing applications which process vast amounts of data (multi-terabyte data-sets) in-parallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner.

A MapReduce *job* usually splits the input data-set into independent chunks which are processed by the *map tasks* in a completely parallel manner. The framework sorts the outputs of the maps, which are then input to the *reduce tasks*. Typically both the input and the output of the job are stored in a file-system. The framework takes care of scheduling tasks, monitoring them and re-executes the failed tasks.

Typically the compute nodes and the storage nodes are the same, that is, the MapReduce framework and the Hadoop Distributed File System are running on the same set of nodes. This configuratio n allows the framework to effectively schedule tasks on the nodes where data is already present, resulting in very high aggregate bandwidth across the cluster.

The MapReduce framework consists of a single master `ResourceManager`, one worker `NodeManager` per cluster-node, and `MRAppMaster` per application.

Minimally, applications specify the input/output locations and supply *map* and *reduce* functions via implementations of appropriate interfaces and/or abstract-classes. These, and other job parameters, comprise the *job configuration*.

The Hadoop *job client* then submits the job (jar/executable etc.) and configuration to the `ResourceManager` which then assumes the responsibility of distributing the software/configuration to the workers, scheduling tasks and monitoring them, providing status and diagnostic information to the job-client.

Inputs and Outputs

The MapReduce framework operates exclusively on `<key, value>` pairs, that is, the framework views the input to the job as a set of `<key, value>` pairs and produces a set of `<key, value>` pairs as the output of the job, conceivably of different types.

The `key` and `value` classes have to be serializable by the framework and hence need to implement the Writable interface. Additionally, the key classes have to implement the WritableComparable interface to facilitate sorting by the framework.

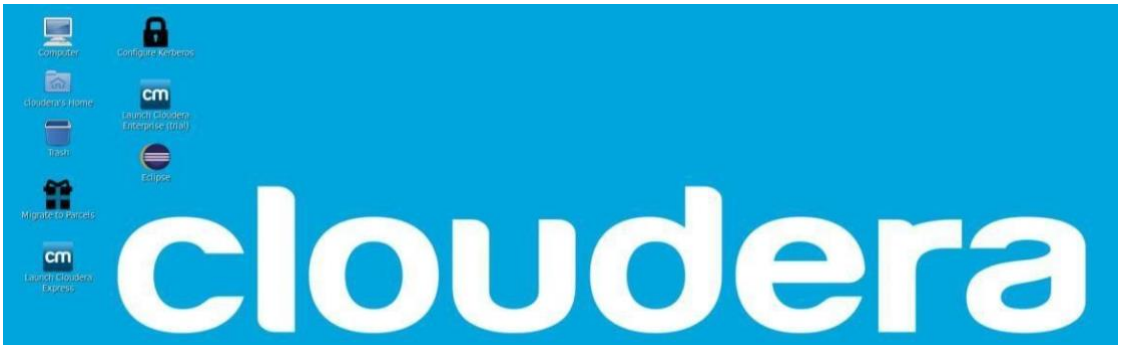
Input and Output types of a MapReduce job:

(input) `<k1, v1>` -> **map** -> `<k2, v2>` -> **combine** -> `<k2, v2>` -> **reduce** -> `<k3, v3>` (output)

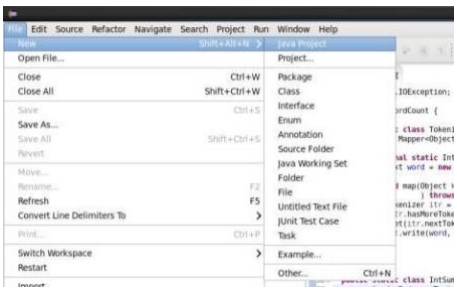
Step 1

Compile `WordCount.java` and create a jar:

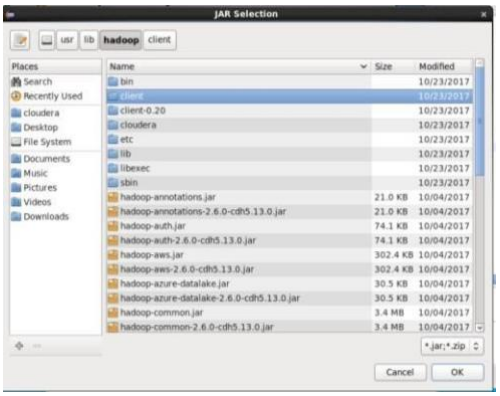
(i) Open Eclipse in Clouderea

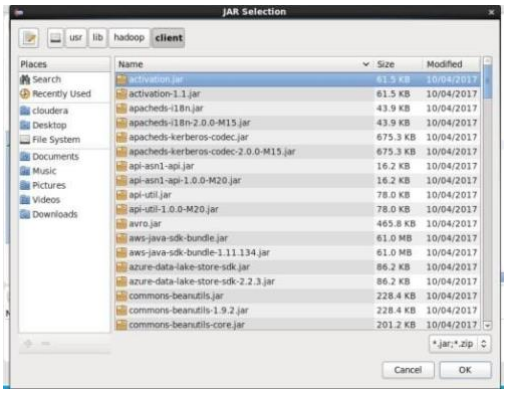


(ii) Create 'WordCount' java project

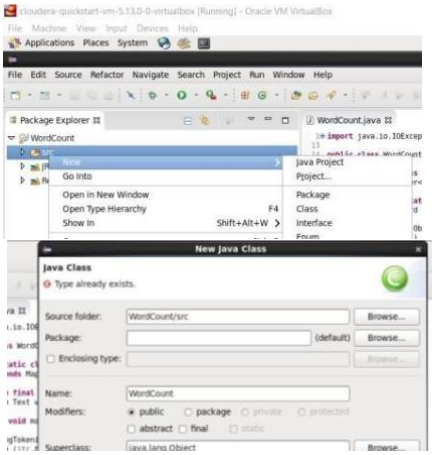


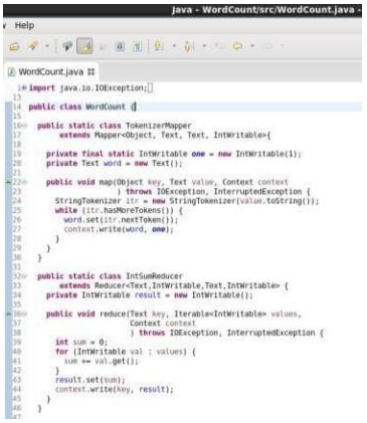
Import following Jar files



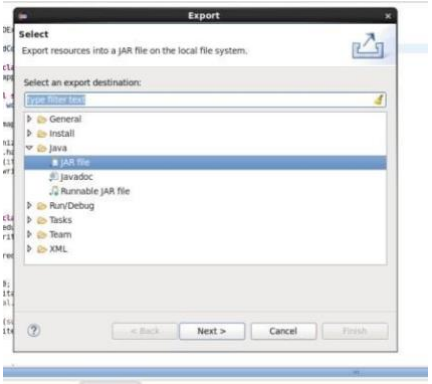


(iii) Create ‘WordCount.java’ in src folder





(iv) Create WordCount.jar file



Step 2

Create following folders in HDFS:

- `/input` - input directory in HDFS
- `/output` - output directory in HDFS

```
File  Edit  View  Search  Terminal  Help
[cloudera@quickstart ~]$ ls
cloudera-manager  enterprise-deployment.json  Pictures   Videos
cm_api.py         express-deployment.json     Public     WordCount.jar
Desktop           kerberos                    te         workspace
Documents         lib                         temp
Downloads         Music                       Templates
eclipse           parcels                     to
[cloudera@quickstart ~]$ ls
cloudera-manager  Desktop    Downloads  enterprise-deployment.json  kerberos  Music    Pictures  te    Templates  Videos        workspace
cm_api.py         Documents  eclipse    express-deployment.json     lib       parcels  Public    temp  to         WordCount.jar
[cloudera@quickstart ~]$ hdfs dfs -mkdir /in00
[cloudera@quickstart ~]$ hdfs dfs ls/
ls/: Unknown command
[cloudera@quickstart ~]$ hdfs dfs ls /
ls: Unknown command
Did you mean -ls?  This command begins with a dash.
[cloudera@quickstart ~]$ hdfs dfs -ls /
Found 8 items
drwxrwxrwx   - hdfs     supergroup          0 2017-10-23 09:15 /benchmarks
drwxr-xr-x   - hbase    supergroup          0 2022-07-27 23:46 /hbase
drwxr-xr-x   - cloudera supergroup          0 2022-08-03 00:09 /in00
drwxr-xr-x   - solr     solr                0 2017-10-23 09:18 /solr
drwxr-xr-x   - cloudera supergroup          0 2022-07-31 10:07 /temp
drwxrwxrwt   - hdfs     supergroup          0 2022-07-22 22:58 /tmp
drwxr-xr-x   - hdfs     supergroup          0 2017-10-23 09:17 /user
drwxr-xr-x   - hdfs     supergroup          0 2017-10-23 09:17 /var
```

Step 3

Create and copy sample text-files into input folder:

```
[cloudera@quickstart ~]$ cat>WCFile.txt
Hadoop
Big Data
Hadoop Distributed File System
Map Reduce
^C
[cloudera@quickstart ~]$ hdfs dfs -ls/in00/
-ls/in00/: Unknown command
[cloudera@quickstart ~]$ hdfs dfs -ls /in00/
[cloudera@quickstart ~]$ hdfs dfs -ls /in00
[cloudera@quickstart ~]$ hdfs dfs -copyFromLocal WCFile.txt /in00
[cloudera@quickstart ~]$ hdfs dfs -ls /in00
Found 1 items
-rw-r--r--   1 cloudera supergroup         58 2022-08-03 00:17 /in00/WCFile.txt
[cloudera@quickstart ~]$
[cloudera@quickstart ~]$ hdfs dfs -mkdir /in00/WCFile.txt/out00
mkdir: Parent path is not a directory: /in00/WCFile.txt WCFile.txt
[cloudera@quickstart ~]$ ls
cloudera-manager  Desktop    Downloads  enterprise-deployment.json  kerberos  Music    Pictures  te    Templates  Videos      WordCount.jar
cm_api.py         Documents  eclipse    express-deployment.json     lib       parcels  Public    temp  to         WCFile.txt  workspace
[cloudera@quickstart ~]$ hdfs dfs -mkdir /out00
[cloudera@quickstart ~]$ hadoop jar /home/cloudera/WordCount.jar WordCount /in00/WCFile.txt /out00
```

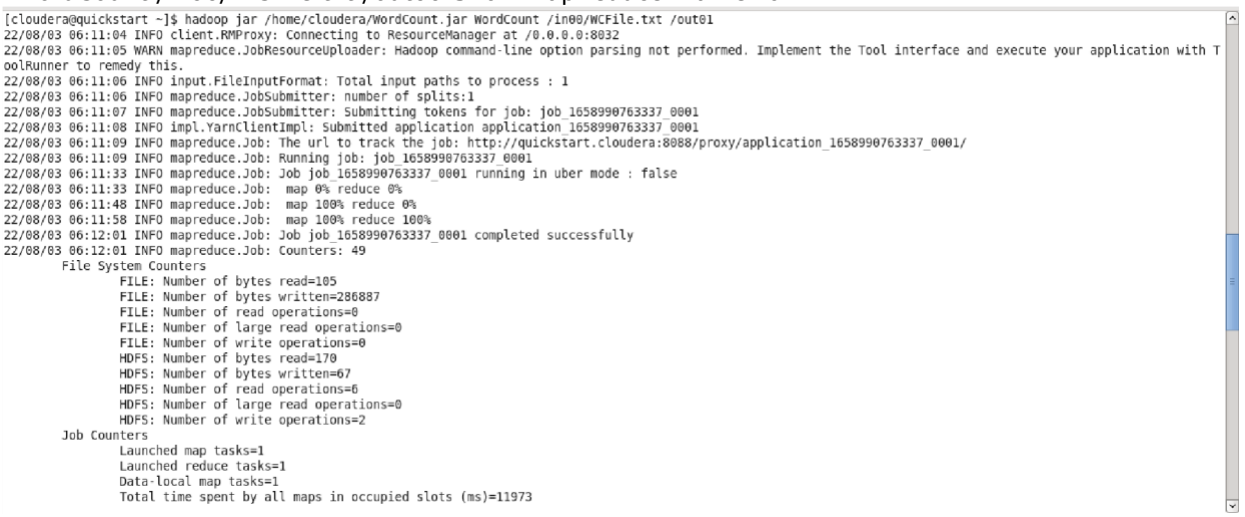
[cloudera@quickstart ~]$ hdfs dfs -ls /in00/

Found 1 items

-rw-r--r-- 1 cloudera supergroup 158 2021-08-15 04:32 /in00/WCFile.txt

Step 4

Run the MapReduce application: hadoop jar /home/cloudera/WordCount.jar WordCount /in00/WCFile.txt /out00 Show MapReduce Framework



```
[cloudera@quickstart ~]$ hadoop jar /home/cloudera/WordCount.jar WordCount /in00/WCFile.txt /out01
22/08/03 06:11:04 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
22/08/03 06:11:05 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with T
oolRunner to remedy this.
22/08/03 06:11:06 INFO input.FileInputFormat: Total input paths to process : 1
22/08/03 06:11:06 INFO mapreduce.JobSubmitter: number of splits:1
22/08/03 06:11:07 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1658990763337_0001
22/08/03 06:11:08 INFO impl.YarnClientImpl: Submitted application application_1658990763337_0001
22/08/03 06:11:09 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:8088/proxy/application_1658990763337_0001/
22/08/03 06:11:09 INFO mapreduce.Job: Running job: job_1658990763337_0001
22/08/03 06:11:33 INFO mapreduce.Job: Job job_1658990763337_0001 running in uber mode : false
22/08/03 06:11:33 INFO mapreduce.Job:  map 0% reduce 0%
22/08/03 06:11:48 INFO mapreduce.Job:  map 100% reduce 0%
22/08/03 06:11:58 INFO mapreduce.Job:  map 100% reduce 100%
22/08/03 06:12:01 INFO mapreduce.Job: Job job_1658990763337_0001 completed successfully
22/08/03 06:12:01 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=105
                FILE: Number of bytes written=286887
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=170
                HDFS: Number of bytes written=67
                HDFS: Number of read operations=6
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=11973
```

Step 5

Output:

[cloudera@quickstart ~]$ hdfs dfs -ls /out00/

Found 2 items

-rw-r--r-- 1 cloudera supergroup 0 2021-08-15 04:41 /out00/_SUCCESS

-rw-r--r-- 1 cloudera supergroup 113 2021-08-15 04:41 /out00/part-r-00000

[cloudera@quickstart ~]$ hdfs dfs -cat /out00/part-r-00000