UNIVERSITY OF SCIENCE - VNU, HCM CITY

INTRODUCTION TO BIGDATA

LAB1 - A GENTLE INTRODUCTION TO HADOOP

Students:
Vo Trung Hoang Hung
ID: 21120011

Tran Ky Thanh
ID: 21120556

Huynh Cong Triet
ID: 21120577

Hoang The Trung
ID: 21120583

Teachers:
Nguyen Ngoc Thao
Do Trong Le
Bui Huynh Trung Nam

Contents

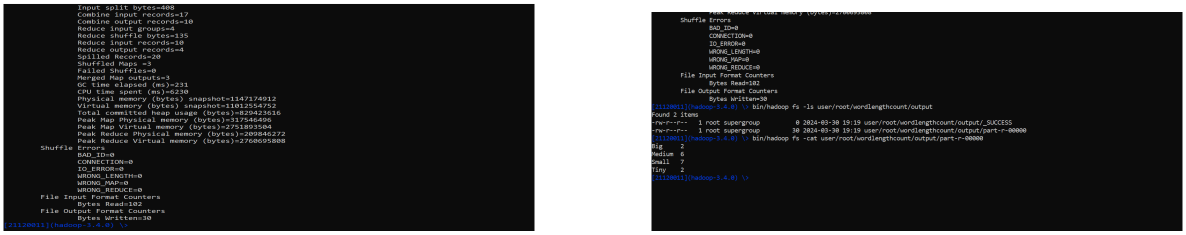
1 Team's Result

Section	**Percent**	**Notes**
1. Setting up SNC	100%	
2. Paper Reading	100%	
3. Running Word Count	100%	
4. Bonus	100%	

2 Answer

2.1 Setting up Single-node Hadoop Cluster

2.1.1 21120011:



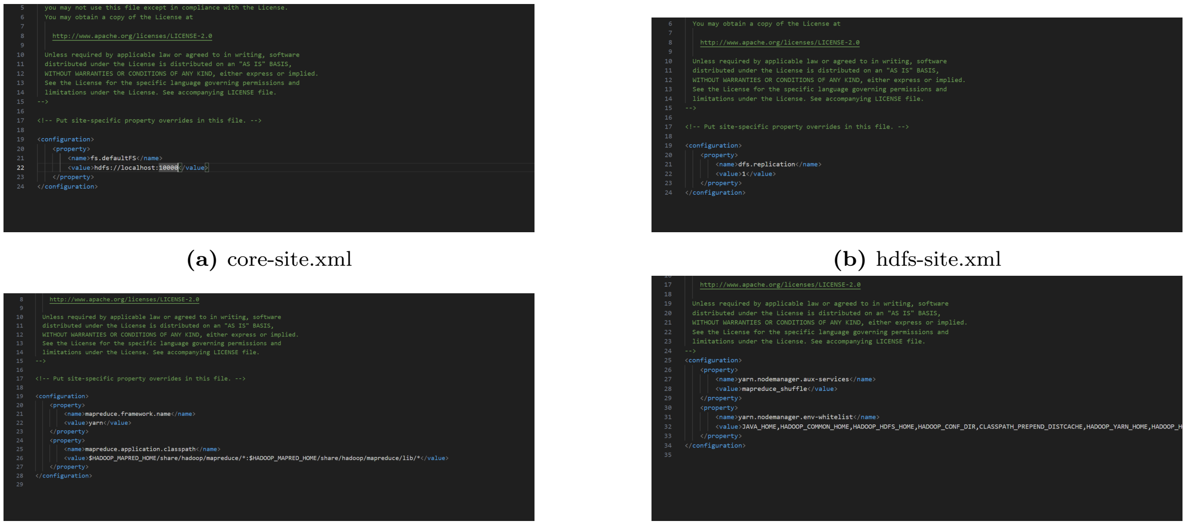
```
                Input split bytes=408
                Combine input records=17
                Combine output records=10
                Reduce input groups=4
                Reduce shuffle bytes=135
                Reduce input records=10
                Reduce output records=4
                Spilled Records=20
                Shuffled Maps =3
                Failed Shuffles=0
                Merged Map outputs=3
                GC time elapsed (ms)=231
                CPU time spent (ms)=6230
                Physical memory (bytes) snapshot=1147174912
                Virtual memory (bytes) snapshot=11012554752
                Total committed heap usage (bytes)=829423616
                Peak Map Physical memory (bytes)=317546496
                Peak Map Virtual memory (bytes)=2751893504
                Peak Reduce Physical memory (bytes)=209846272
                Peak Reduce Virtual memory (bytes)=2760695808
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=102
        File Output Format Counters
                Bytes Written=30
[21120011](hadoop-3.4.0) \>
```

(a) hadoop-mapreduce-examples-3.4.0.jar

```
                Peak Reduce Virtual memory (bytes)=2760695808
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=102
        File Output Format Counters
                Bytes Written=30
[21120011](hadoop-3.4.0) \> bin/hadoop fs -ls user/root/wordlengthcount/output
Found 2 items
-rw-r--r--   1 root supergroup          0 2024-03-30 19:19 user/root/wordlengthcount/output/_SUCCESS
-rw-r--r--   1 root supergroup         30 2024-03-30 19:19 user/root/wordlengthcount/output/part-r-00000
[21120011](hadoop-3.4.0) \> bin/hadoop fs -cat user/root/wordlengthcount/output/part-r-00000
Big     2
Medium  6
Small   7
Tiny    2
[21120011](hadoop-3.4.0) \>
```

(b) List all files in output folders

Figure 1: Standalone Operation



```
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
    <property>
        <name>fs.defaultFS</name>
        <value>hdfs://localhost:10000</value>
    </property>
</configuration>
```

(a) core-site.xml

```
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

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

(b) hdfs-site.xml

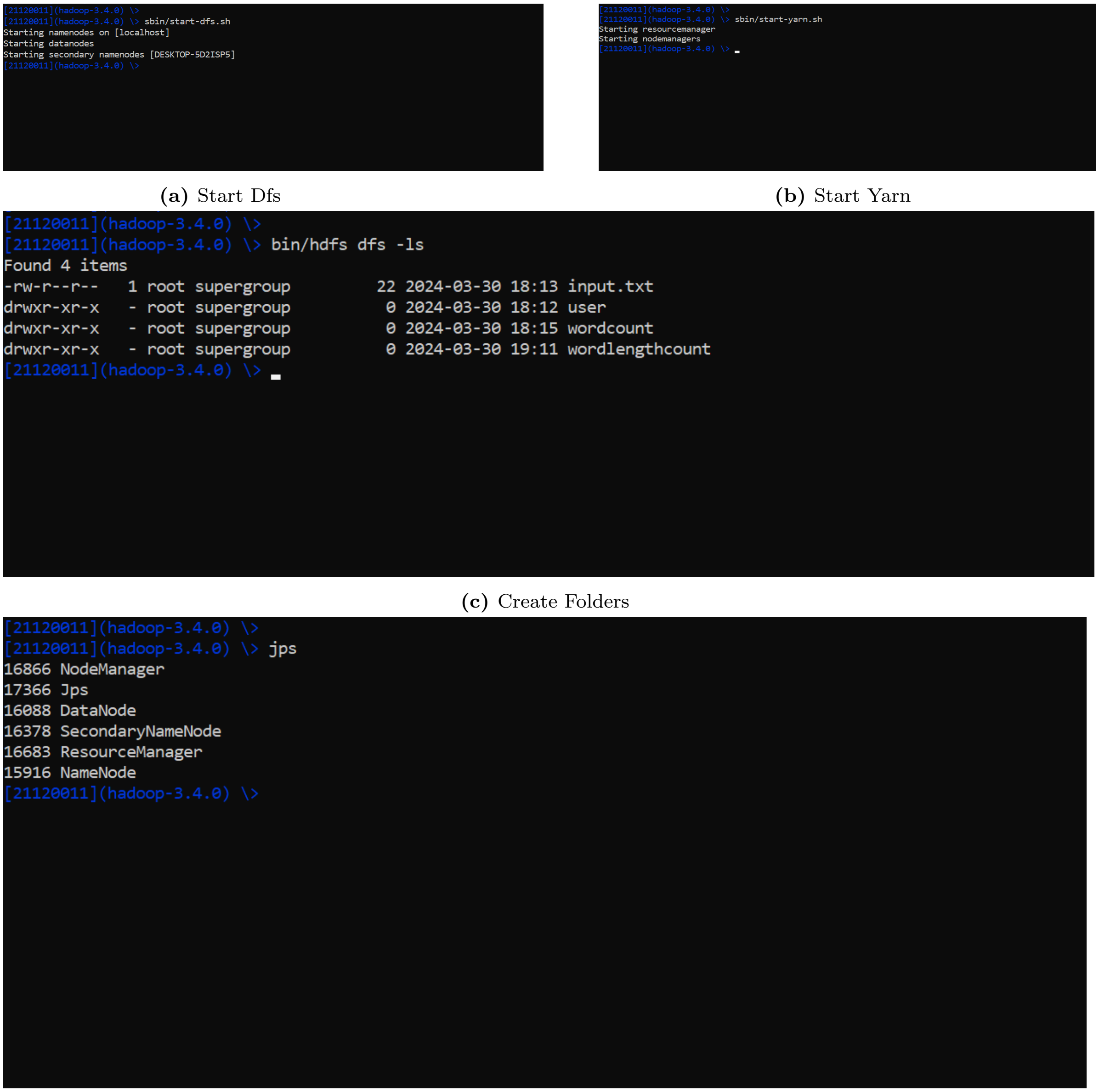
```
    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<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>

```

(c) mapred-site.xml

```
    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->
<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_PREPEND_DISTCACHE,HADOOP_YARN_HOME,HADOOP_H
    </property>
</configuration>

```

(d) yarn-site.xml

Figure 2: Hadoop configuration



```
[21120011](hadoop-3.4.0) \>
[21120011](hadoop-3.4.0) \> sbin/start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [DESKTOP-5D2ISP5]
[21120011](hadoop-3.4.0) \>
```

(a) Start Dfs

```
[21120011](hadoop-3.4.0) \>
[21120011](hadoop-3.4.0) \> sbin/start-yarn.sh
Starting resourcemanager
Starting nodemanagers
[21120011](hadoop-3.4.0) \>
```

(b) Start Yarn

```
[21120011](hadoop-3.4.0) \>
[21120011](hadoop-3.4.0) \> bin/hdfs dfs -ls
Found 4 items
-rw-r--r--   1 root supergroup         22 2024-03-30 18:13 input.txt
drwxr-xr-x   - root supergroup          0 2024-03-30 18:12 user
drwxr-xr-x   - root supergroup          0 2024-03-30 18:15 wordcount
drwxr-xr-x   - root supergroup          0 2024-03-30 19:11 wordlengthcount
[21120011](hadoop-3.4.0) \>
```

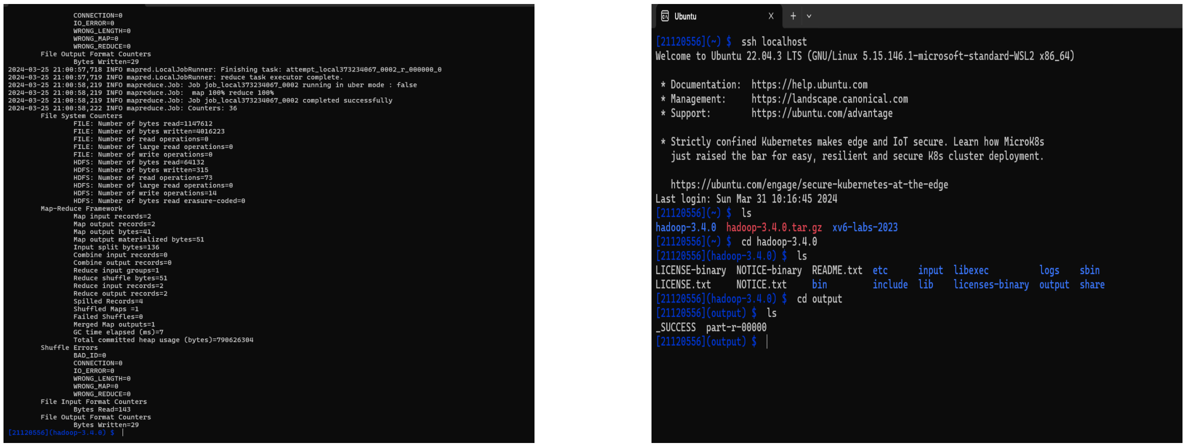
(c) Create Folders

```
[21120011](hadoop-3.4.0) \>
[21120011](hadoop-3.4.0) \> jps
16866 NodeManager
17366 Jps
16088 DataNode
16378 SecondaryNameNode
16683 ResourceManager
15916 NameNode
[21120011](hadoop-3.4.0) \>
```

(d) JPS

Figure 3: Pseudo-Distributed Operation

2.1.2 21120556:



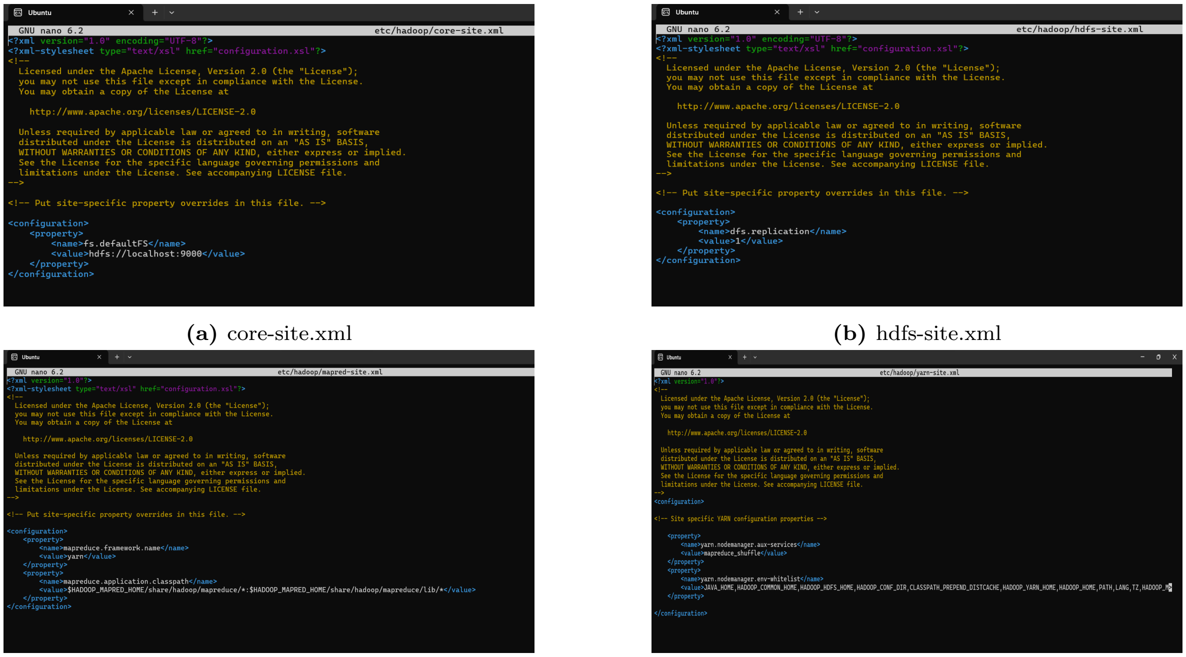
```
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Output Format Counters
                Bytes Written=29
2024-03-25 21:00:57,718 INFO mapred.LocalJobRunner: Finishing task: attempt_local373234067_0002_r_000000_0
2024-03-25 21:00:57,719 INFO mapred.LocalJobRunner: reduce task executor complete.
2024-03-25 21:00:58,219 INFO mapreduce.Job: Job job_local373234067_0002 running in uber mode : false
2024-03-25 21:00:58,219 INFO mapreduce.Job:  map 100% reduce 100%
2024-03-25 21:00:58,219 INFO mapreduce.Job: Job job_local373234067_0002 completed successfully
2024-03-25 21:00:58,222 INFO mapreduce.Job: Counters: 36
        File System Counters
                FILE: Number of bytes read=1147612
                FILE: Number of bytes written=4016223
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=64132
                HDFS: Number of bytes written=315
                HDFS: Number of read operations=73
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=14
                HDFS: Number of bytes read erasure-coded=0
        Map-Reduce Framework
                Map input records=2
                Map output records=2
                Map output bytes=41
                Map output materialized bytes=51
                Input split bytes=136
                Combine input records=0
                Combine output records=0
                Reduce input groups=1
                Reduce shuffle bytes=51
                Reduce input records=2
                Reduce output records=2
                Spilled Records=4
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=7
                Total committed heap usage (bytes)=790626304
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=143
        File Output Format Counters
                Bytes Written=29
[21120556](hadoop-3.4.0) $ |
```

(a) hadoop-mapreduce-examples-3.4.0.jar

```
[21120556](~) $ ssh localhost
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.146.1-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge
Last login: Sun Mar 31 10:16:45 2024
[21120556](~) $ ls
hadoop-3.4.0  hadoop-3.4.0.tar.gz  xv6-labs-2023
[21120556](~) $ cd hadoop-3.4.0
[21120556](hadoop-3.4.0) $ ls
LICENSE-binary  NOTICE-binary  README.txt  etc      input  libexec         logs    sbin
LICENSE.txt     NOTICE.txt     bin         include  lib    licenses-binary  output  share
[21120556](hadoop-3.4.0) $ cd output
[21120556](output) $ ls
_SUCCESS  part-r-00000
[21120556](output) $ |
```

(b) List all files in output folders

Figure 4: Standalone Operation



```
  GNU nano 6.2                                etc/hadoop/core-site.xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
    <property>
        <name>fs.defaultFS</name>
        <value>hdfs://localhost:9000</value>
    </property>
</configuration>
```

(a) core-site.xml

```
  GNU nano 6.2                                etc/hadoop/hdfs-site.xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

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

(b) hdfs-site.xml

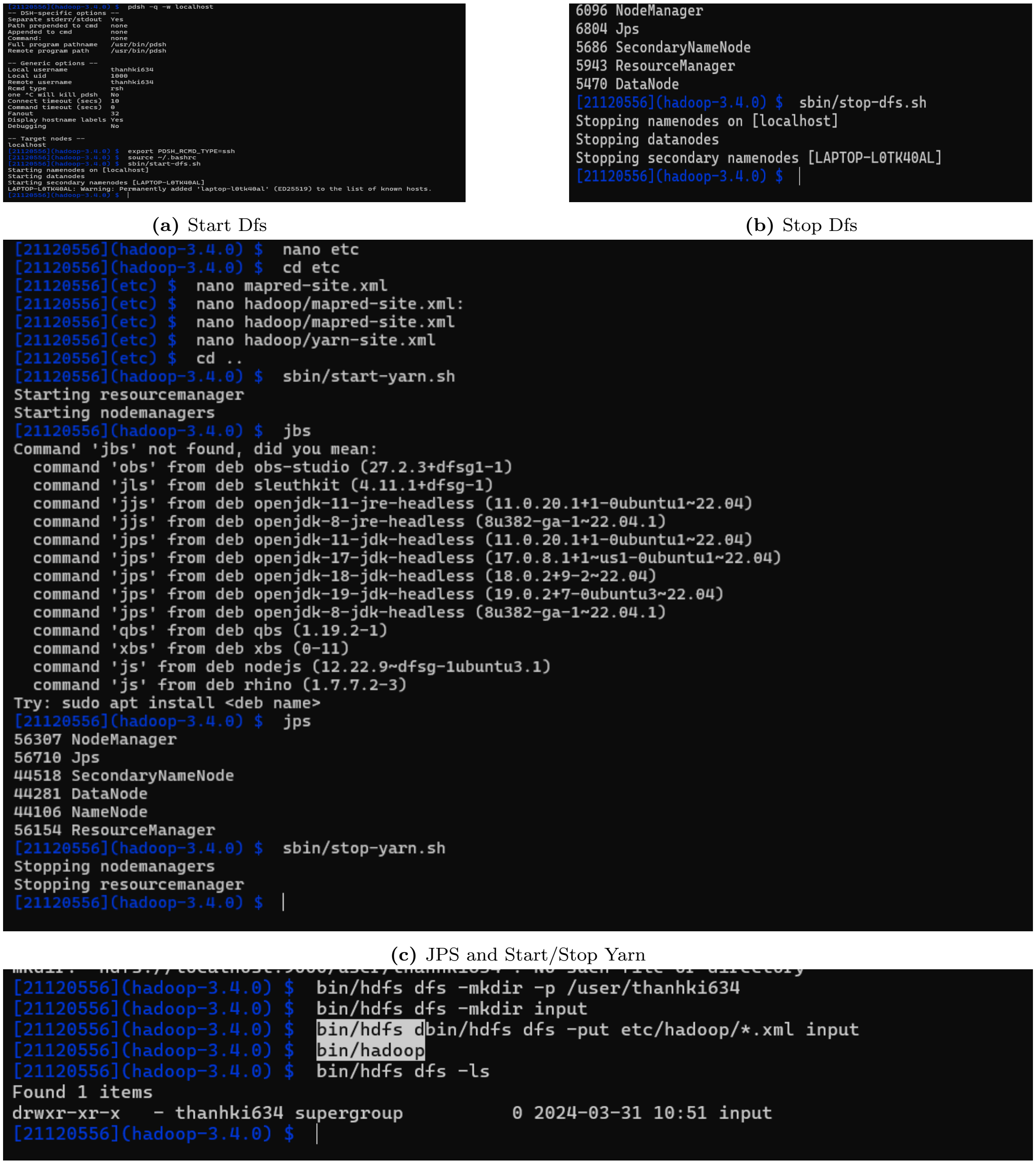
```
  GNU nano 6.2                                etc/hadoop/mapred-site.xml
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<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>
```

(c) mapred-site.xml

```
  GNU nano 6.2                                etc/hadoop/yarn-site.xml
<?xml version="1.0"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->
<configuration>

<!-- Site specific YARN configuration properties -->

    <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_PREPEND_DISTCACHE,HADOOP_YARN_HOME,HADOOP_HOME,PATH,LANG,TZ,HADOOP_M>
    </property>

</configuration>
```

(d) yarn-site.xml

Figure 5: Hadoop configuration



```
[21120556](hadoop-3.4.0) $  pdsh -q -w localhost
-- DSH-specific options --
Separate stderr/stdout  Yes
Path prepended to cmd   none
Appended to cmd         none
Command:                none
Full program pathname   /usr/bin/pdsh
Remote program path     /usr/bin/pdsh

-- Generic options --
Local username          thanhki634
Local uid               1000
Remote username         thanhki634
Rcmd type               rsh
one ^C will kill pdsh   No
Connect timeout (secs)  10
Command timeout (secs)  0
Fanout                  32
Display hostname labels Yes
Debugging               No

-- Target nodes --
localhost
[21120556](hadoop-3.4.0) $  export PDSH_RCMD_TYPE=ssh
[21120556](hadoop-3.4.0) $  source ~/.bashrc
[21120556](hadoop-3.4.0) $  sbin/start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [LAPTOP-L0TK40AL]
LAPTOP-L0TK40AL: Warning: Permanently added 'laptop-l0tk40al' (ED25519) to the list of known hosts.
[21120556](hadoop-3.4.0) $  |
```

(a) Start Dfs

```
6096 NodeManager
6804 Jps
5686 SecondaryNameNode
5943 ResourceManager
5470 DataNode
[21120556](hadoop-3.4.0) $  sbin/stop-dfs.sh
Stopping namenodes on [localhost]
Stopping datanodes
Stopping secondary namenodes [LAPTOP-L0TK40AL]
[21120556](hadoop-3.4.0) $  |
```

(b) Stop Dfs

```
[21120556](hadoop-3.4.0) $  nano etc
[21120556](hadoop-3.4.0) $  cd etc
[21120556](etc) $  nano mapred-site.xml
[21120556](etc) $  nano hadoop/mapred-site.xml:
[21120556](etc) $  nano hadoop/mapred-site.xml
[21120556](etc) $  nano hadoop/yarn-site.xml
[21120556](etc) $  cd ..
[21120556](hadoop-3.4.0) $  sbin/start-yarn.sh
Starting resourcemanager
Starting nodemanagers
[21120556](hadoop-3.4.0) $  jbs
Command 'jbs' not found, did you mean:
  command 'obs' from deb obs-studio (27.2.3+dfsg1-1)
  command 'jls' from deb sleuthkit (4.11.1+dfsg-1)
  command 'jjs' from deb openjdk-11-jre-headless (11.0.20.1+1-0ubuntu1~22.04)
  command 'jjs' from deb openjdk-8-jre-headless (8u382-ga-1~22.04.1)
  command 'jps' from deb openjdk-11-jdk-headless (11.0.20.1+1-0ubuntu1~22.04)
  command 'jps' from deb openjdk-17-jdk-headless (17.0.8.1+1~us1-0ubuntu1~22.04)
  command 'jps' from deb openjdk-18-jdk-headless (18.0.2+9-2~22.04)
  command 'jps' from deb openjdk-19-jdk-headless (19.0.2+7-0ubuntu3~22.04)
  command 'jps' from deb openjdk-8-jdk-headless (8u382-ga-1~22.04.1)
  command 'qbs' from deb qbs (1.19.2-1)
  command 'xbs' from deb xbs (0-11)
  command 'js' from deb nodejs (12.22.9~dfsg-1ubuntu3.1)
  command 'js' from deb rhino (1.7.7.2-3)
Try: sudo apt install <deb name>
[21120556](hadoop-3.4.0) $  jps
56307 NodeManager
56710 Jps
44518 SecondaryNameNode
44281 DataNode
44106 NameNode
56154 ResourceManager
[21120556](hadoop-3.4.0) $  sbin/stop-yarn.sh
Stopping nodemanagers
Stopping resourcemanager
[21120556](hadoop-3.4.0) $  |
```

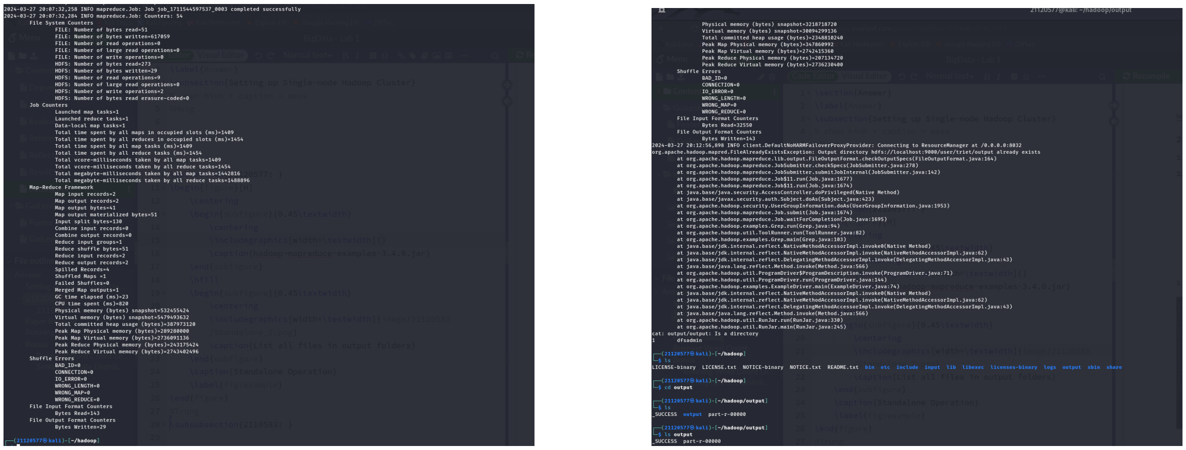
(c) JPS and Start/Stop Yarn

```
mkdir: `hdfs://localhost:9000/user/thanhki634': No such file or directory
[21120556](hadoop-3.4.0) $  bin/hdfs dfs -mkdir -p /user/thanhki634
[21120556](hadoop-3.4.0) $  bin/hdfs dfs -mkdir input
[21120556](hadoop-3.4.0) $  bin/hdfs dbin/hdfs dfs -put etc/hadoop/*.xml input
[21120556](hadoop-3.4.0) $  bin/hadoop
[21120556](hadoop-3.4.0) $  bin/hdfs dfs -ls
Found 1 items
drwxr-xr-x   - thanhki634 supergroup          0 2024-03-31 10:51 input
[21120556](hadoop-3.4.0) $  |
```

(d) Create Folder

Figure 6: Pseudo-Distributed Operation

2.1.3 21120577:



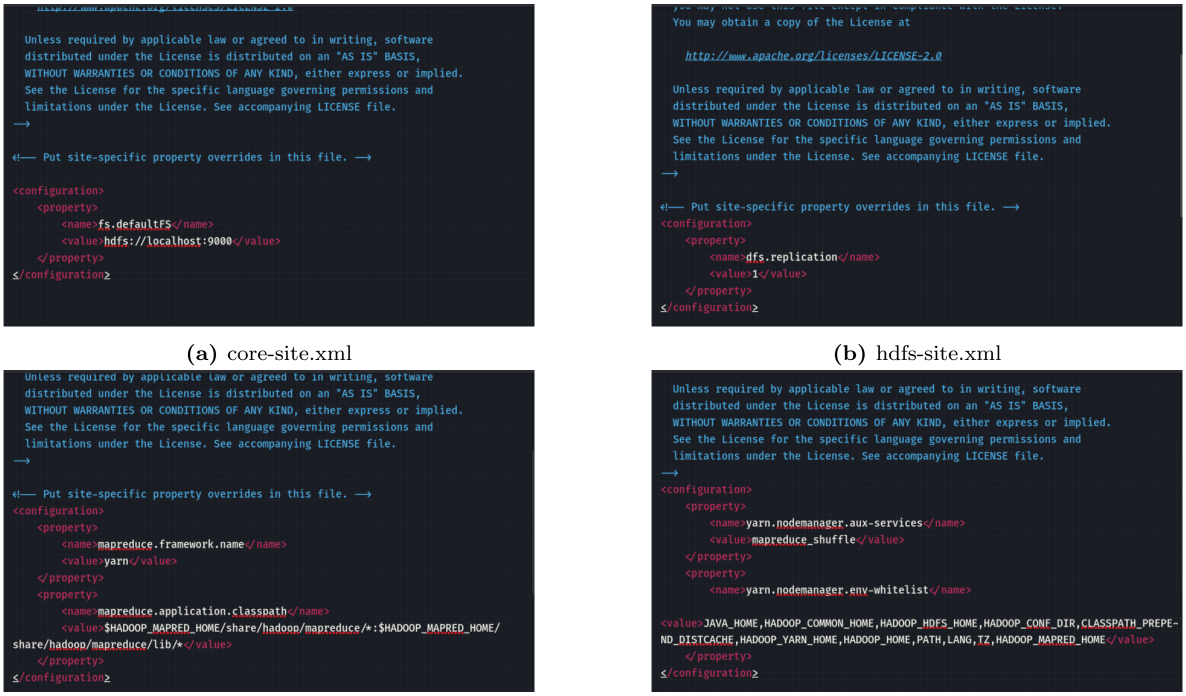
```
2024-03-27 20:07:32,258 INFO mapreduce.Job: Job job_1711544597537_0003 completed successfully
2024-03-27 20:07:32,284 INFO mapreduce.Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=51
                FILE: Number of bytes written=617059
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=273
                HDFS: Number of bytes written=29
                HDFS: Number of read operations=9
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
                HDFS: Number of bytes read erasure-coded=0
        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)=1409
                Total time spent by all reduces in occupied slots (ms)=1454
                Total time spent by all map tasks (ms)=1409
                Total time spent by all reduce tasks (ms)=1454
                Total vcore-milliseconds taken by all map tasks=1409
                Total vcore-milliseconds taken by all reduce tasks=1454
                Total megabyte-milliseconds taken by all map tasks=1442816
                Total megabyte-milliseconds taken by all reduce tasks=1488896
        Map-Reduce Framework
                Map input records=2
                Map output records=2
                Map output bytes=41
                Map output materialized bytes=51
                Input split bytes=130
                Combine input records=0
                Combine output records=0
                Reduce input groups=1
                Reduce shuffle bytes=51
                Reduce input records=2
                Reduce output records=2
                Spilled Records=4
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=23
                CPU time spent (ms)=820
                Physical memory (bytes) snapshot=532455424
                Virtual memory (bytes) snapshot=5479493632
                Total committed heap usage (bytes)=387973120
                Peak Map Physical memory (bytes)=289280000
                Peak Map Virtual memory (bytes)=2736091136
                Peak Reduce Physical memory (bytes)=243175424
                Peak Reduce Virtual memory (bytes)=2743402496
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=143
        File Output Format Counters
                Bytes Written=29

┌──(21120577㉿kali)-[~/hadoop]
```

(a) hadoop-mapreduce-examples-3.4.0.jar

```
21120577@kali: ~/hadoop/output
                Physical memory (bytes) snapshot=3218718720
                Virtual memory (bytes) snapshot=30094299136
                Total committed heap usage (bytes)=2348810240
                Peak Map Physical memory (bytes)=347860992
                Peak Map Virtual memory (bytes)=2742415360
                Peak Reduce Physical memory (bytes)=207134720
                Peak Reduce Virtual memory (bytes)=2736230400
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=32550
        File Output Format Counters
                Bytes Written=143
2024-03-27 20:12:56,898 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
org.apache.hadoop.mapred.FileAlreadyExistsException: Output directory hdfs://localhost:9000/user/triet/output already exists
        at org.apache.hadoop.mapreduce.lib.output.FileOutputFormat.checkOutputSpecs(FileOutputFormat.java:164)
        at org.apache.hadoop.mapreduce.JobSubmitter.checkSpecs(JobSubmitter.java:278)
        at org.apache.hadoop.mapreduce.JobSubmitter.submitJobInternal(JobSubmitter.java:142)
        at org.apache.hadoop.mapreduce.Job$11.run(Job.java:1677)
        at org.apache.hadoop.mapreduce.Job$11.run(Job.java:1674)
        at java.base/java.security.AccessController.doPrivileged(Native Method)
        at java.base/javax.security.auth.Subject.doAs(Subject.java:423)
        at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1953)
        at org.apache.hadoop.mapreduce.Job.submit(Job.java:1674)
        at org.apache.hadoop.mapreduce.Job.waitForCompletion(Job.java:1695)
        at org.apache.hadoop.examples.Grep.run(Grep.java:94)
        at org.apache.hadoop.util.ToolRunner.run(ToolRunner.java:82)
        at org.apache.hadoop.examples.Grep.main(Grep.java:103)
        at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.base/java.lang.reflect.Method.invoke(Method.java:566)
        at org.apache.hadoop.util.ProgramDriver$ProgramDescription.invoke(ProgramDriver.java:71)
        at org.apache.hadoop.util.ProgramDriver.run(ProgramDriver.java:144)
        at org.apache.hadoop.examples.ExampleDriver.main(ExampleDriver.java:74)
        at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.base/java.lang.reflect.Method.invoke(Method.java:566)
        at org.apache.hadoop.util.RunJar.run(RunJar.java:330)
        at org.apache.hadoop.util.RunJar.main(RunJar.java:245)
cat: output/output: Is a directory
1       dfsadmin

┌──(21120577㉿kali)-[~/hadoop]
└─$ ls
LICENSE-binary  LICENSE.txt  NOTICE-binary  NOTICE.txt  README.txt  bin  etc  include  input  lib  libexec  licenses-binary  logs  output  sbin  share

┌──(21120577㉿kali)-[~/hadoop]
└─$ cd output

┌──(21120577㉿kali)-[~/hadoop/output]
└─$ ls
_SUCCESS  output  part-r-00000

┌──(21120577㉿kali)-[~/hadoop/output]
└─$ ls output
_SUCCESS  part-r-00000
```

(b) List all files in output folders

Figure 7: Standalone Operation



```
    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
    <property>
        <name>fs.defaultFS</name>
        <value>hdfs://localhost:9000</value>
    </property>
</configuration>
```

(a) core-site.xml

```
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->
<configuration>
    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>
</configuration>
```

(b) hdfs-site.xml

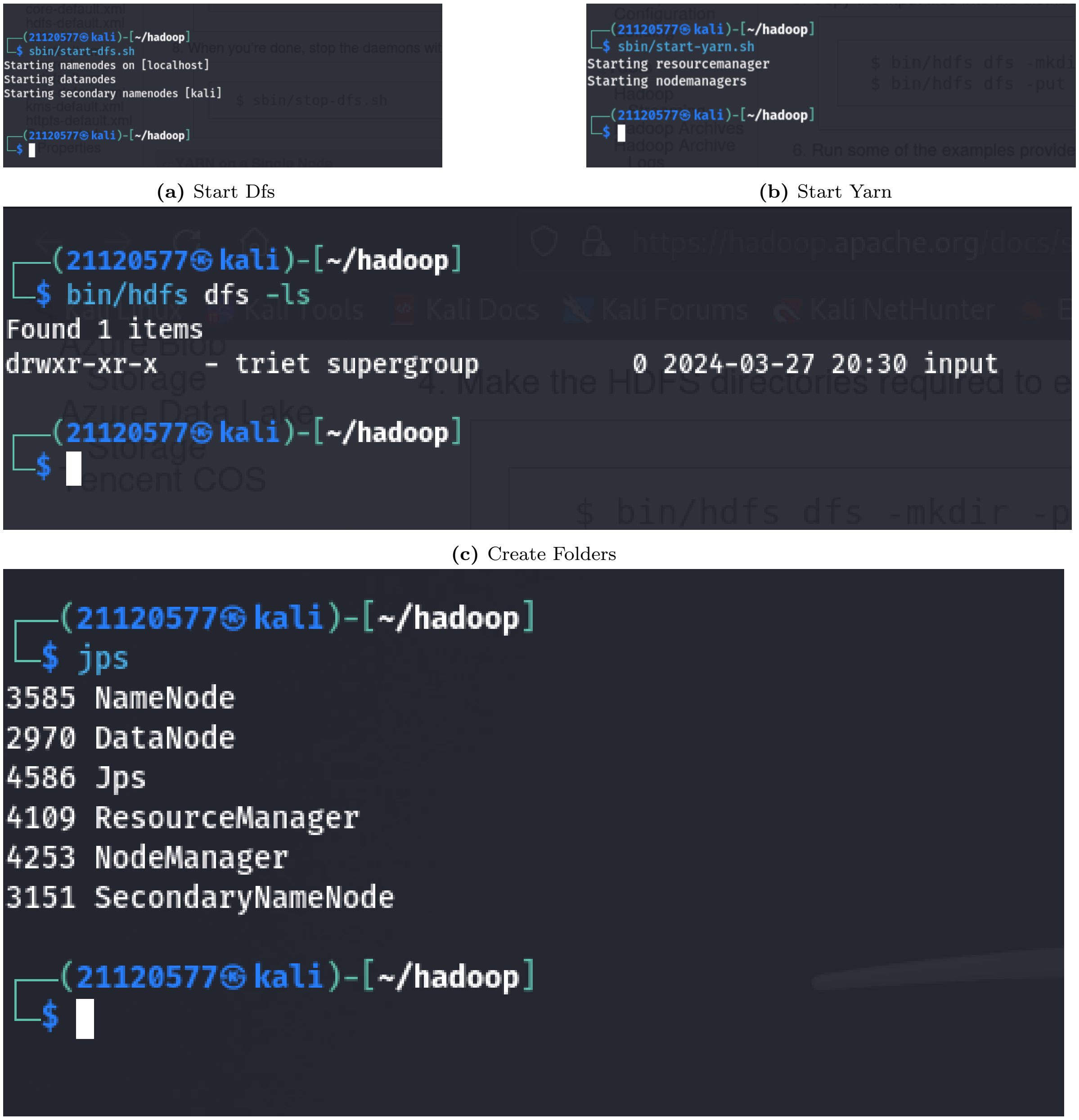
```
  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->
<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>
```

(c) mapred-site.xml

```
  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->
<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_PREPE-
ND_DISTCACHE,HADOOP_YARN_HOME,HADOOP_HOME,PATH,LANG,TZ,HADOOP_MAPRED_HOME</value>
    </property>
</configuration>
```

(d) yarn-site.xml

Figure 8: Hadoop configuration



```
┌──(21120577㉿kali)-[~/hadoop]
└─$ sbin/start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [kali]

┌──(21120577㉿kali)-[~/hadoop]
└─$ 
```

(a) Start Dfs

```
┌──(21120577㉿kali)-[~/hadoop]
└─$ sbin/start-yarn.sh
Starting resourcemanager
Starting nodemanagers

┌──(21120577㉿kali)-[~/hadoop]
└─$ 
```

(b) Start Yarn

```
┌──(21120577㉿kali)-[~/hadoop]
└─$ bin/hdfs dfs -ls
Found 1 items
drwxr-xr-x   - triet supergroup          0 2024-03-27 20:30 input

┌──(21120577㉿kali)-[~/hadoop]
└─$ 
```

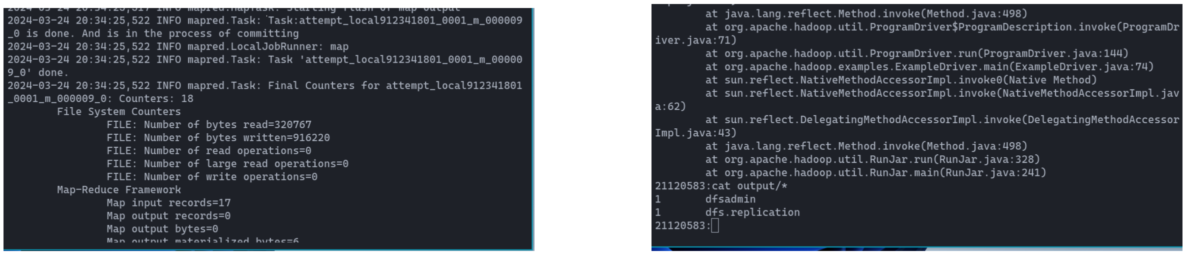
(c) Create Folders

```
┌──(21120577㉿kali)-[~/hadoop]
└─$ jps
3585 NameNode
2970 DataNode
4586 Jps
4109 ResourceManager
4253 NodeManager
3151 SecondaryNameNode

┌──(21120577㉿kali)-[~/hadoop]
└─$ 
```

(d) JPS

Figure 9: Pseudo-Distributed Operation

2.1.4 21120583:



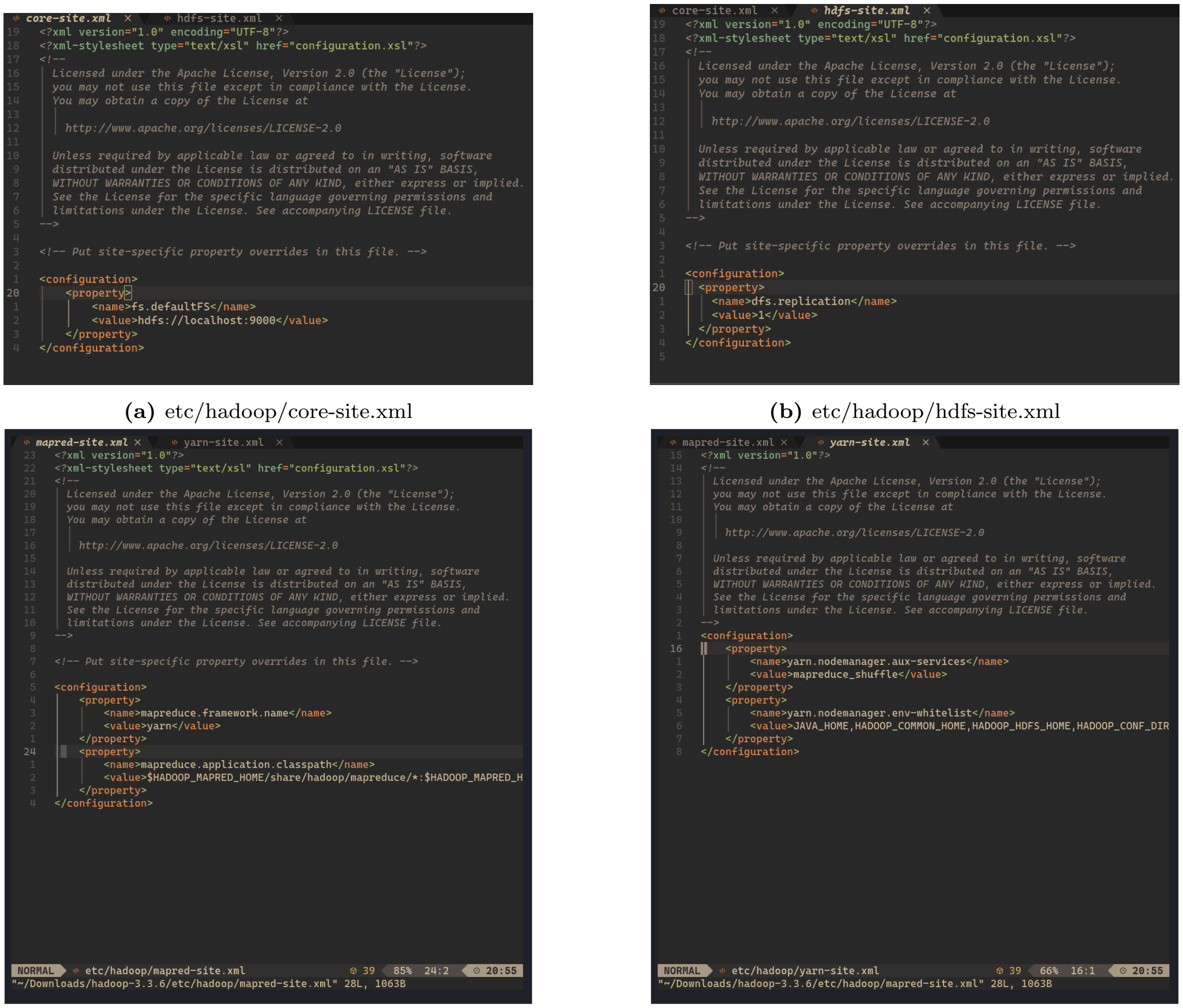
```
2024-03-24 20:34:25,517 INFO mapred.MapTask: Starting flush of map output
2024-03-24 20:34:25,522 INFO mapred.Task: Task:attempt_local912341801_0001_m_000009
_0 is done. And is in the process of committing
2024-03-24 20:34:25,522 INFO mapred.LocalJobRunner: map
2024-03-24 20:34:25,522 INFO mapred.Task: Task 'attempt_local912341801_0001_m_00000
9_0' done.
2024-03-24 20:34:25,522 INFO mapred.Task: Final Counters for attempt_local912341801
_0001_m_000009_0: Counters: 18
        File System Counters
                FILE: Number of bytes read=320767
                FILE: Number of bytes written=916220
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
        Map-Reduce Framework
                Map input records=17
                Map output records=0
                Map output bytes=0
                Map output materialized bytes=6
```

(a) hadoop-mapreduce-examples-3.3.6.jar

```
        at java.lang.reflect.Method.invoke(Method.java:498)
        at org.apache.hadoop.util.ProgramDriver$ProgramDescription.invoke(ProgramDr
iver.java:71)
        at org.apache.hadoop.util.ProgramDriver.run(ProgramDriver.java:144)
        at org.apache.hadoop.examples.ExampleDriver.main(ExampleDriver.java:74)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.jav
a:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessor
Impl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:498)
        at org.apache.hadoop.util.RunJar.run(RunJar.java:328)
        at org.apache.hadoop.util.RunJar.main(RunJar.java:241)
21120583:cat output/*
1       dfsadmin
1       dfs.replication
21120583:
```

(b) List all files in output folders

Figure 10: Standalone Operation



```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
    <property>
        <name>fs.defaultFS</name>
        <value>hdfs://localhost:9000</value>
    </property>
</configuration>
```

(a) etc/hadoop/core-site.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

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

(b) etc/hadoop/hdfs-site.xml

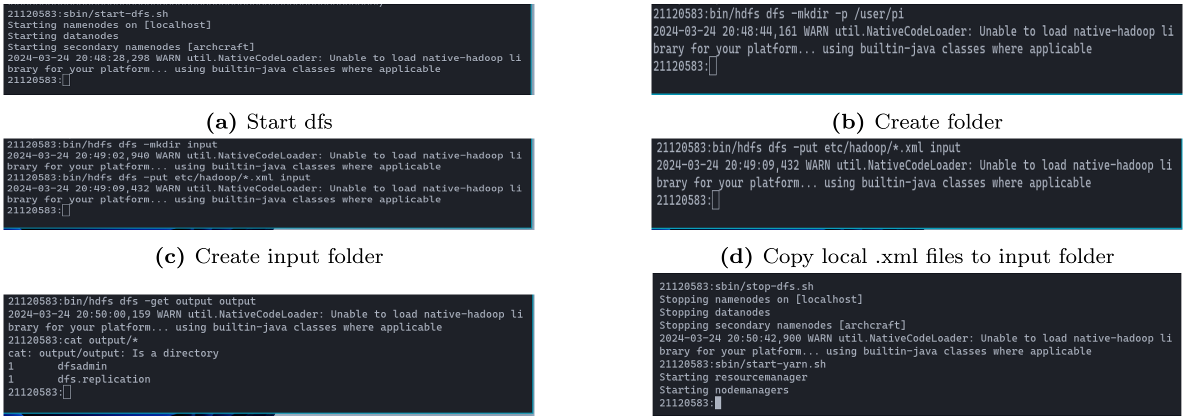
```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<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_H
    </property>
</configuration>

NORMAL  etc/hadoop/mapred-site.xml                39  85%  24:2   20:55
"~/Downloads/hadoop-3.3.6/etc/hadoop/mapred-site.xml" 28L, 1063B
```

(c) etc/hadoop/mapred-site.xml

```
<?xml version="1.0"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->
<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
    </property>
</configuration>

NORMAL  etc/hadoop/yarn-site.xml                  39  66%  16:1   20:55
"~/Downloads/hadoop-3.3.6/etc/hadoop/mapred-site.xml" 28L, 1063B
```

(d) etc/hadoop/yarn-site.xml

Figure 11: Hadoop configuration



```
21120583:sbin/start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [archcraft]
2024-03-24 20:48:28,298 WARN util.NativeCodeLoader: Unable to load native-hadoop li
brary for your platform... using builtin-java classes where applicable
21120583:
```

(a) Start dfs

```
21120583:bin/hdfs dfs -mkdir -p /user/pi
2024-03-24 20:48:44,161 WARN util.NativeCodeLoader: Unable to load native-hadoop li
brary for your platform... using builtin-java classes where applicable
21120583:
```

(b) Create folder

```
21120583:bin/hdfs dfs -mkdir input
2024-03-24 20:49:02,940 WARN util.NativeCodeLoader: Unable to load native-hadoop li
brary for your platform... using builtin-java classes where applicable
21120583:bin/hdfs dfs -put etc/hadoop/*.xml input
2024-03-24 20:49:09,432 WARN util.NativeCodeLoader: Unable to load native-hadoop li
brary for your platform... using builtin-java classes where applicable
21120583:
```

(c) Create input folder

```
21120583:bin/hdfs dfs -put etc/hadoop/*.xml input
2024-03-24 20:49:09,432 WARN util.NativeCodeLoader: Unable to load native-hadoop li
brary for your platform... using builtin-java classes where applicable
21120583:
```

(d) Copy local .xml files to input folder

```
21120583:bin/hdfs dfs -get output output
2024-03-24 20:50:00,159 WARN util.NativeCodeLoader: Unable to load native-hadoop li
brary for your platform... using builtin-java classes where applicable
21120583:cat output/*
cat: output/output: Is a directory
1       dfsadmin
1       dfs.replication
21120583:
```

(e) Show results

```
21120583:sbin/stop-dfs.sh
Stopping namenodes on [localhost]
Stopping datanodes
Stopping secondary namenodes [archcraft]
2024-03-24 20:50:42,900 WARN util.NativeCodeLoader: Unable to load native-hadoop li
brary for your platform... using builtin-java classes where applicable
21120583:sbin/start-yarn.sh
Starting resourcemanager
Starting nodemanagers
21120583:
```

(f) Start YARN

Figure 12: Pseudo-Distributed Operation

2.2 Paper Reading

1. How do the input keys-values, the intermediate keys-values, and the output keys-values relate?

Answer:

MapReduce model includes 2 main operations: Map and Reduce

- ***The Map function***: written by the user, takes an input pair and produces a set of intermediate key/value pairs.
- ***The Reduce function***, also written by the user, accepts an intermediate key I and a set of values for that key. It merges together these values to form a possibly smaller set of values.

The related of input keys-values, the intermediate keys-values, and the output keys-values:

- Input keys-values represent the initial data
- Intermediate keys-values represent the intermediate results during computation, they are essentially the output of the mapping phase and the input to the reducing phase in MapReduce
- Output keys-values represent the final results after processing

2. How does MapReduce deal with node failures?

Answer:

MapReduce deal with node failures by re-executing tasks and ensuring deterministic output.There 2 main kind of failure:

- ***Master Failure***: If the master task dies, a new copy can be started from the last checkpointed state. However, given that there is only a single master, its failure is unlikely; therefore our current implementation aborts the MapReduce computation if the master fails.
- ***Worker Failure***: The master pings every worker periodically. If no response is received from a worker in a certacertain amount of time, the master marks the worker as failed. Any map tasks completed by the worker are reset back to their initial idle state, and therefore become eligible for scheduling on other workers. Similarly, any map task or reduce task in progress on a failed worker is also reset to idle and becomes eligible for rescheduling. Completed map tasks are re-executed on a failure because their output is stored on the local disk(s) of the failed machine and is therefore inaccessible. Completed reduce tasks do not need to be re-executed since their output is stored in a global file system.

3. What is the meaning and implication of locality? What does it use?

Answer:

Meaning of Locality: Conserving network bandwidth by taking advantage of the fact that the input data is stored on the local disks of the machines that make up our cluster. When running large MapReduce operations on a significant fraction of the workers in a cluster, most input data is read locally and consumes no network bandwidth.

System tries to do the work on the computer where the data is stored. This way, it does not have to move the data over long distances, which can be slow. It's like having the toolbox (the data) right where you're working (the computer doing the task). This makes everything faster and less crowded on the network, like having a clear path in your house to walk without tripping over things. MapReduce uses this idea to make sure it can do a lot of work quickly and without causing traffic jams in the data network.

Usage of locality in HDFS:
Data placement policy: HDFS follows a data placement policy that tries to store replica blocks on different racks and nodes to improve locality and fault tolerance.
Task scheduling: When scheduling tasks, HDFS tries to assign tasks to the nodes where the required data blocks are already present, taking advantage of data locality.
Rack-aware replica placement: HDFS tries to place one replica on the same node, another on a different node in the same rack, and a third replica on a different node in a different rack, ensuring locality and fault tolerance.
Speculative execution: If a task is running slowly due to network or hardware issues, HDFS may speculatively execute another copy of the task on a different node with better data locality, to improve overall performance.

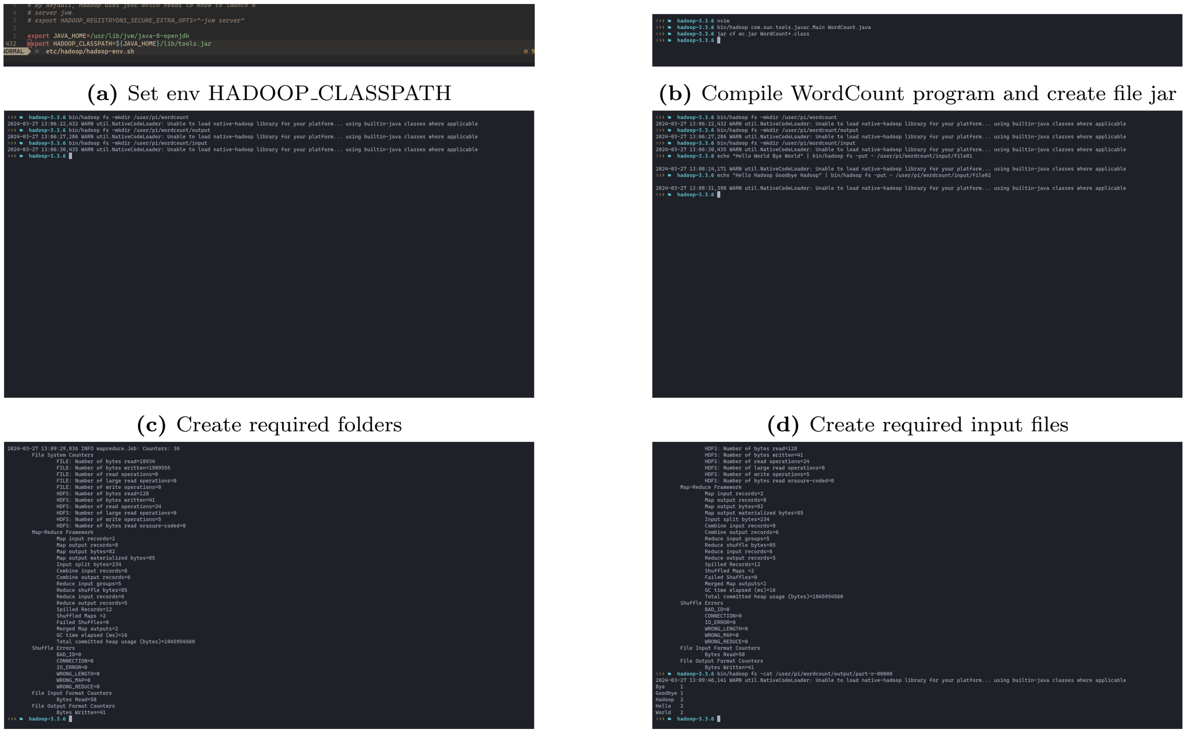
4. Which problem is addressed by introducing a combiner function to the MapReduce model?

   Answer:

   When there is significant repetition in the intermediate keys produced by each map task, and the user-specified Reduce function is commutative and associative, it costs a significant amount of time to send all intermediate keys to the Reduce function. So, a combiner function comes in handy to address this issue by partial merging of this data before it is sent over the network.

   The combiner function works like a mini-reducer for each map task. It takes the output of the map tasks and combines or summarizes it before it's sent over the network to the reduce tasks

2.3 Running Word Count



```
# By default, Hadoop uses jsvc which needs to know to launch a
# server jvm.
# export HADOOP_REGISTRYDNS_SECURE_EXTRA_OPTS="-jvm server"

export JAVA_HOME=/usr/lib/jvm/java-8-openjdk
export HADOOP_CLASSPATH=${JAVA_HOME}/lib/tools.jar
ORMAL   etc/hadoop/hadoop-env.sh
```

(a) Set env HADOOP_CLASSPATH

```
>>>   hadoop-3.3.6 nvim
>>>   hadoop-3.3.6 bin/hadoop com.sun.tools.javac.Main WordCount.java
>>>   hadoop-3.3.6 jar cf wc.jar WordCount*.class
>>>   hadoop-3.3.6
```

(b) Compile WordCount program and create file jar

```
>>>   hadoop-3.3.6 bin/hadoop fs -mkdir /user/pi/wordcount
2024-03-27 13:06:22,432 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
>>>   hadoop-3.3.6 bin/hadoop fs -mkdir /user/pi/wordcount/output
2024-03-27 13:06:27,286 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
>>>   hadoop-3.3.6 bin/hadoop fs -mkdir /user/pi/wordcount/input
2024-03-27 13:06:30,435 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
>>>   hadoop-3.3.6
```

(c) Create required folders

```
>>>   hadoop-3.3.6 bin/hadoop fs -mkdir /user/pi/wordcount
2024-03-27 13:06:22,432 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
>>>   hadoop-3.3.6 bin/hadoop fs -mkdir /user/pi/wordcount/output
2024-03-27 13:06:27,286 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
>>>   hadoop-3.3.6 bin/hadoop fs -mkdir /user/pi/wordcount/input
2024-03-27 13:06:30,435 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
>>>   hadoop-3.3.6 echo "Hello World Bye World" | bin/hadoop fs -put - /user/pi/wordcount/input/file01

2024-03-27 13:08:14,171 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
>>>   hadoop-3.3.6 echo "Hello Hadoop Goodbye Hadoop" | bin/hadoop fs -put - /user/pi/wordcount/input/file02

2024-03-27 13:08:31,398 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
>>>   hadoop-3.3.6
```

(d) Create required input files

```
2024-03-27 13:09:29,836 INFO mapreduce.Job: Counters: 36
        File System Counters
                FILE: Number of bytes read=10934
                FILE: Number of bytes written=1909555
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=128
                HDFS: Number of bytes written=41
                HDFS: Number of read operations=24
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=5
                HDFS: Number of bytes read erasure-coded=0
        Map-Reduce Framework
                Map input records=2
                Map output records=8
                Map output bytes=82
                Map output materialized bytes=85
                Input split bytes=234
                Combine input records=8
                Combine output records=6
                Reduce input groups=5
                Reduce shuffle bytes=85
                Reduce input records=6
                Reduce output records=5
                Spilled Records=12
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=16
                Total committed heap usage (bytes)=1045954560
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=50
        File Output Format Counters
                Bytes Written=41
>>>   hadoop-3.3.6
```

(e) Run the application

```
                HDFS: Number of bytes read=128
                HDFS: Number of bytes written=41
                HDFS: Number of read operations=24
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=5
                HDFS: Number of bytes read erasure-coded=0
        Map-Reduce Framework
                Map input records=2
                Map output records=8
                Map output bytes=82
                Map output materialized bytes=85
                Input split bytes=234
                Combine input records=8
                Combine output records=6
                Reduce input groups=5
                Reduce shuffle bytes=85
                Reduce input records=6
                Reduce output records=5
                Spilled Records=12
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=16
                Total committed heap usage (bytes)=1045954560
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=50
        File Output Format Counters
                Bytes Written=41
>>>   hadoop-3.3.6 bin/hadoop fs -cat /user/pi/wordcount/output/part-r-00000
2024-03-27 13:09:46,141 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Bye     1
Goodbye 1
Hadoop  2
Hello   2
World   2
>>>   hadoop-3.3.6
```

(f) output

Figure 13: Word Count

Walk through:

```
public void map(Object key, Text value, Context context)
throws IOException, InterruptedException {
  StringTokenizer itr = new StringTokenizer(value.toString());
  while (itr.hasMoreTokens()) {
    word.set(itr.nextToken());
    context.write(word, one);
  }
}
```

The Mapper operates by processing each line individually through its map method, sourced from the chosen TextInputFormat. It then breaks down each line into separate words using StringTokenizer, finally emitting key-value pairs in the format $< word, 1 >$

```
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);
}
```

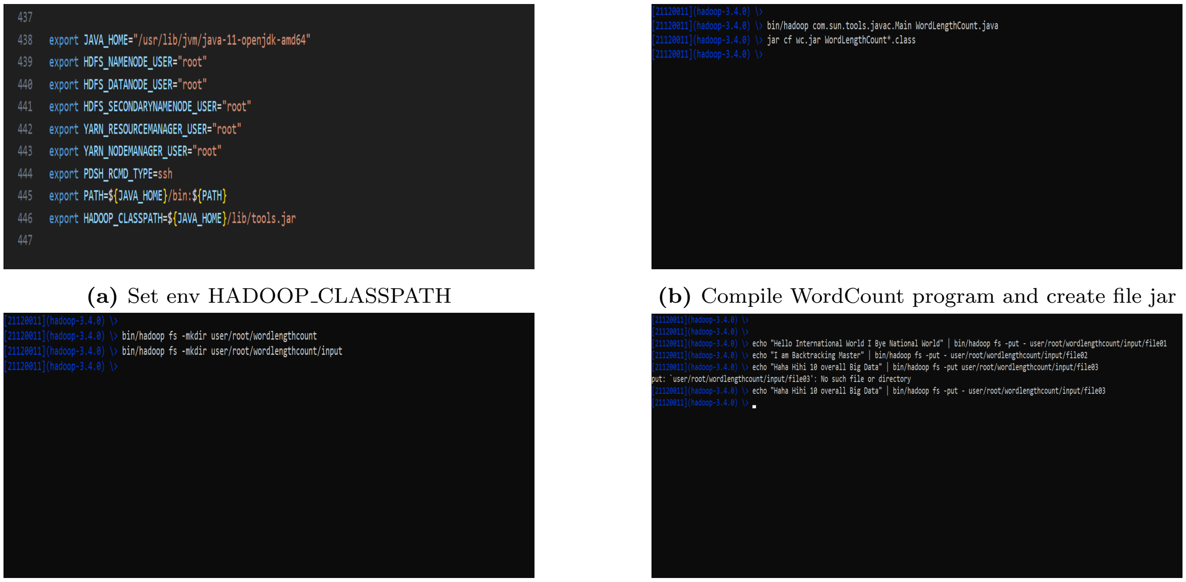
The Reduce function computes the sum of integer values associated with a key and emits the key along with the computed sum. Keep in mind that, on this program we use Comnbiner

```
    job.setCombinerClass(IntSumReducer.class);
```

The output of each map is passed through the local combiner the function. So that, the value returned by `val.get()` may vary depending on whether it’s called within the Combiner or the Reducer.

2.4 Bonus

2.4.1 Running Word Length Count



```
export JAVA_HOME="/usr/lib/jvm/java-11-openjdk-amd64"
export HDFS_NAMENODE_USER="root"
export HDFS_DATANODE_USER="root"
export HDFS_SECONDARYNAMENODE_USER="root"
export YARN_RESOURCEMANAGER_USER="root"
export YARN_NODEMANAGER_USER="root"
export PDSH_RCMD_TYPE=ssh
export PATH=${JAVA_HOME}/bin:${PATH}
export HADOOP_CLASSPATH=${JAVA_HOME}/lib/tools.jar
```

(a) Set env HADOOP_CLASSPATH

```
[21120011](hadoop-3.4.0) \>
[21120011](hadoop-3.4.0) \> bin/hadoop com.sun.tools.javac.Main WordLengthCount.java
[21120011](hadoop-3.4.0) \> jar cf wc.jar WordLengthCount*.class
[21120011](hadoop-3.4.0) \>
```

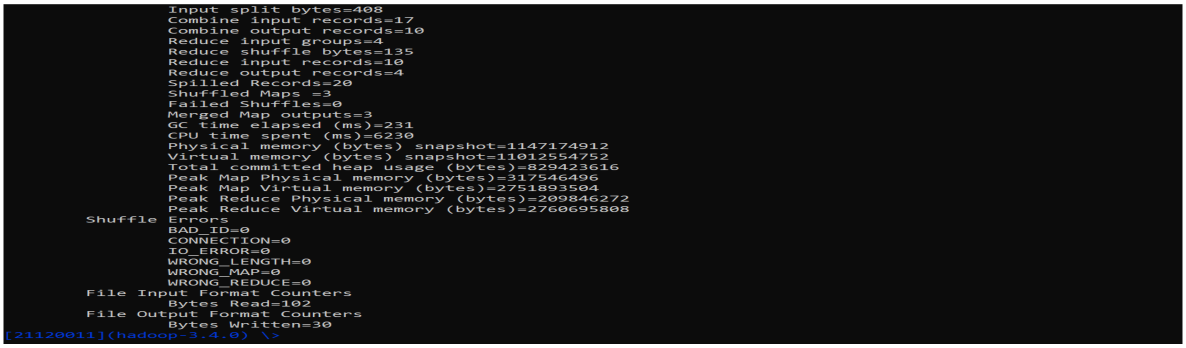
(b) Compile WordCount program and create file jar

```
[21120011](hadoop-3.4.0) \>
[21120011](hadoop-3.4.0) \> bin/hadoop fs -mkdir user/root/wordlengthcount
[21120011](hadoop-3.4.0) \> bin/hadoop fs -mkdir user/root/wordlengthcount/input
[21120011](hadoop-3.4.0) \>
```

(c) Create required folders

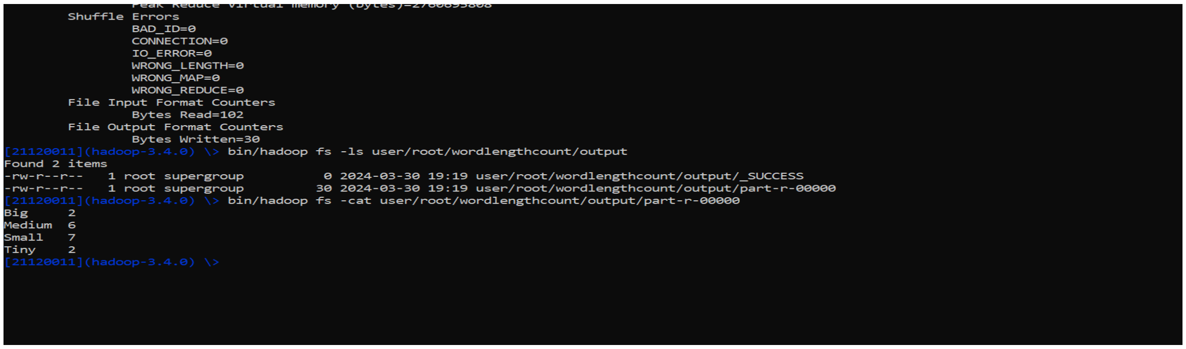
```
[21120011](hadoop-3.4.0) \>
[21120011](hadoop-3.4.0) \>
[21120011](hadoop-3.4.0) \> echo "Hello International World I Bye National World" | bin/hadoop fs -put - user/root/wordlengthcount/input/file01
[21120011](hadoop-3.4.0) \> echo "I am Backtracking Master" | bin/hadoop fs -put - user/root/wordlengthcount/input/file02
[21120011](hadoop-3.4.0) \> echo "Haha Hihi 10 overall Big Data" | bin/hadoop fs -put user/root/wordlengthcount/input/file03
put: `user/root/wordlengthcount/input/file03': No such file or directory
[21120011](hadoop-3.4.0) \> echo "Haha Hihi 10 overall Big Data" | bin/hadoop fs -put - user/root/wordlengthcount/input/file03
[21120011](hadoop-3.4.0) \> 
```

(d) Create required input files



```
                Input split bytes=408
                Combine input records=17
                Combine output records=10
                Reduce input groups=4
                Reduce shuffle bytes=135
                Reduce input records=10
                Reduce output records=4
                Spilled Records=20
                Shuffled Maps =3
                Failed Shuffles=0
                Merged Map outputs=3
                GC time elapsed (ms)=231
                CPU time spent (ms)=6230
                Physical memory (bytes) snapshot=1147174912
                Virtual memory (bytes) snapshot=11012554752
                Total committed heap usage (bytes)=829423616
                Peak Map Physical memory (bytes)=317546496
                Peak Map Virtual memory (bytes)=2751893504
                Peak Reduce Physical memory (bytes)=209846272
                Peak Reduce Virtual memory (bytes)=2760695808
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=102
        File Output Format Counters
                Bytes Written=30
[21120011](hadoop-3.4.0) \>
```

Figure 15: Run the application



```
                Peak Reduce Virtual memory (bytes)=2760695808
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=102
        File Output Format Counters
                Bytes Written=30
[21120011](hadoop-3.4.0) \> bin/hadoop fs -ls user/root/wordlengthcount/output
Found 2 items
-rw-r--r--   1 root supergroup          0 2024-03-30 19:19 user/root/wordlengthcount/output/_SUCCESS
-rw-r--r--   1 root supergroup         30 2024-03-30 19:19 user/root/wordlengthcount/output/part-r-00000
[21120011](hadoop-3.4.0) \> bin/hadoop fs -cat user/root/wordlengthcount/output/part-r-00000
Big     2
Medium  6
Small   7
Tiny    2
[21120011](hadoop-3.4.0) \>
```

Figure 16: output

Walk through:

```
public void map(Object key, Text value, Context context
) throws IOException, InterruptedException {
    StringTokenizer itr = new StringTokenizer(value.toString());
    while (itr.hasMoreTokens()) {
        String token = itr.nextToken();
        String type = classifyWordLength(token);
        word.set(type);
        context.write(word, one);
    }
}

private String classifyWordLength(String token) {
    if(token.length() == 1) return "Tiny";
    else if (token.length() >= 2 && token.length() <= 4) {
        return "Small";
    } else if (token.length() >= 5 && token.length() <= 9) {
        return "Medium";
    }
    return "Big";
}
```

We employs a custom Mapper class named **TokenizerMapper**, which extends Hadoop's **Mapper** class. This class is responsible for reading input text data, tokenizing the text into individual words, and categorizing each word based on its length. For each word, the **map** method emits a key-value pair, with the key being one of the predefined categories ("Tiny", "Small", "Medium", "Big") and the value being an **'IntWritable'** of 1, representing a single occurrence of a word in that category.

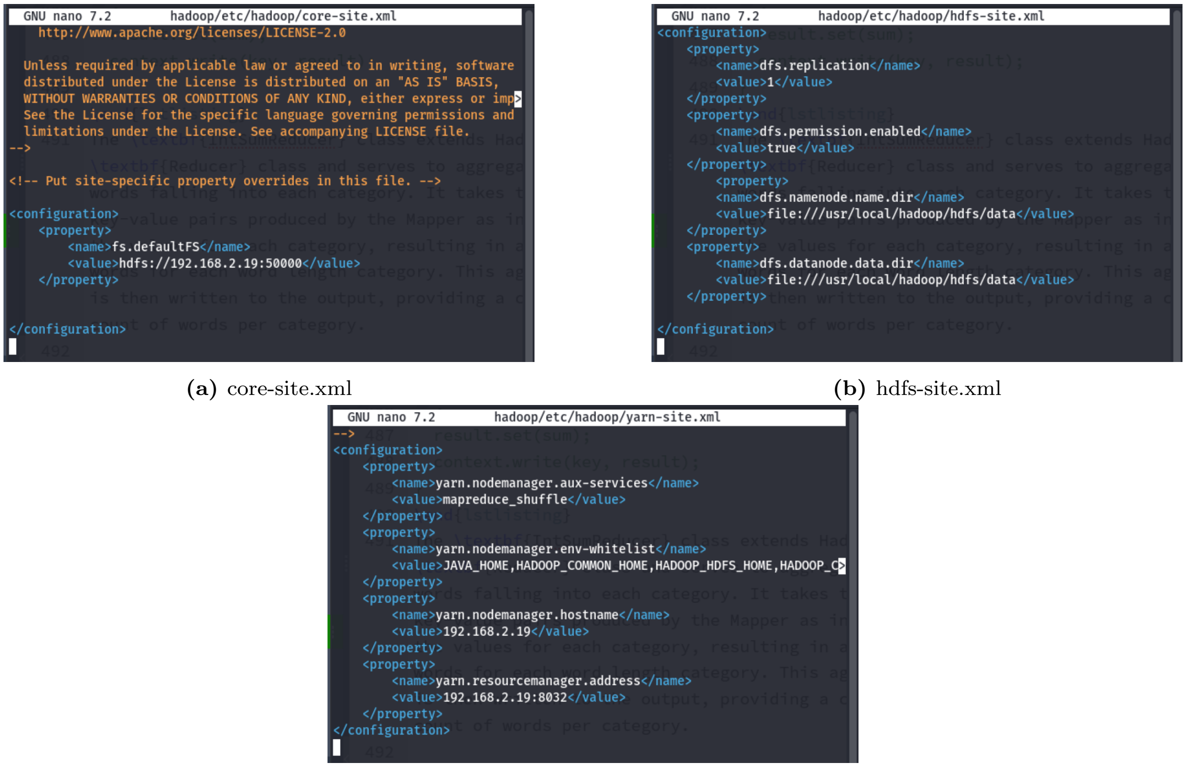
```
public void reduce(Text key, Iterable<IntWritable> values, Context context) throws
     IOException, InterruptedException {
  int sum = 0;
  for (IntWritable val : values) {
    sum += val.get();
  }
  result.set(sum);
  context.write(key, result);
}
```

The **IntSumReducer** class extends Hadoop's **Reducer** class and serves to aggregate the counts of words falling into each category. It takes the intermediate key-value pairs produced by the Mapper as input and sums up the values for each category, resulting in a total count of words for each word length category. This aggregated result is then written to the output, providing a comprehensive count of words per category.

2.4.2 Setting up Fully Distributed Mode

a) Hadoop Cluster Setup in Non-Secure Mode:

1. Create user named ”hadoop” on all machines
2. Make sure our machines can run ssh localhost.
3. Connect ssh to workers without password.
4. Edit configuration files on Master and Slave node:
   (a) Master:



```
  GNU nano 7.2         hadoop/etc/hadoop/core-site.xml
    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or imp>
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
    <property>
        <name>fs.defaultFS</name>
        <value>hdfs://192.168.2.19:50000</value>
    </property>


</configuration>
```

(a) core-site.xml

```
  GNU nano 7.2         hadoop/etc/hadoop/hdfs-site.xml
<configuration>
    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>
    <property>
        <name>dfs.permission.enabled</name>
        <value>true</value>
    </property>
        <property>
        <name>dfs.namenode.name.dir</name>
        <value>file:///usr/local/hadoop/hdfs/data</value>
    </property>
    <property>
        <name>dfs.datanode.data.dir</name>
        <value>file:///usr/local/hadoop/hdfs/data</value>
    </property>

</configuration>
```

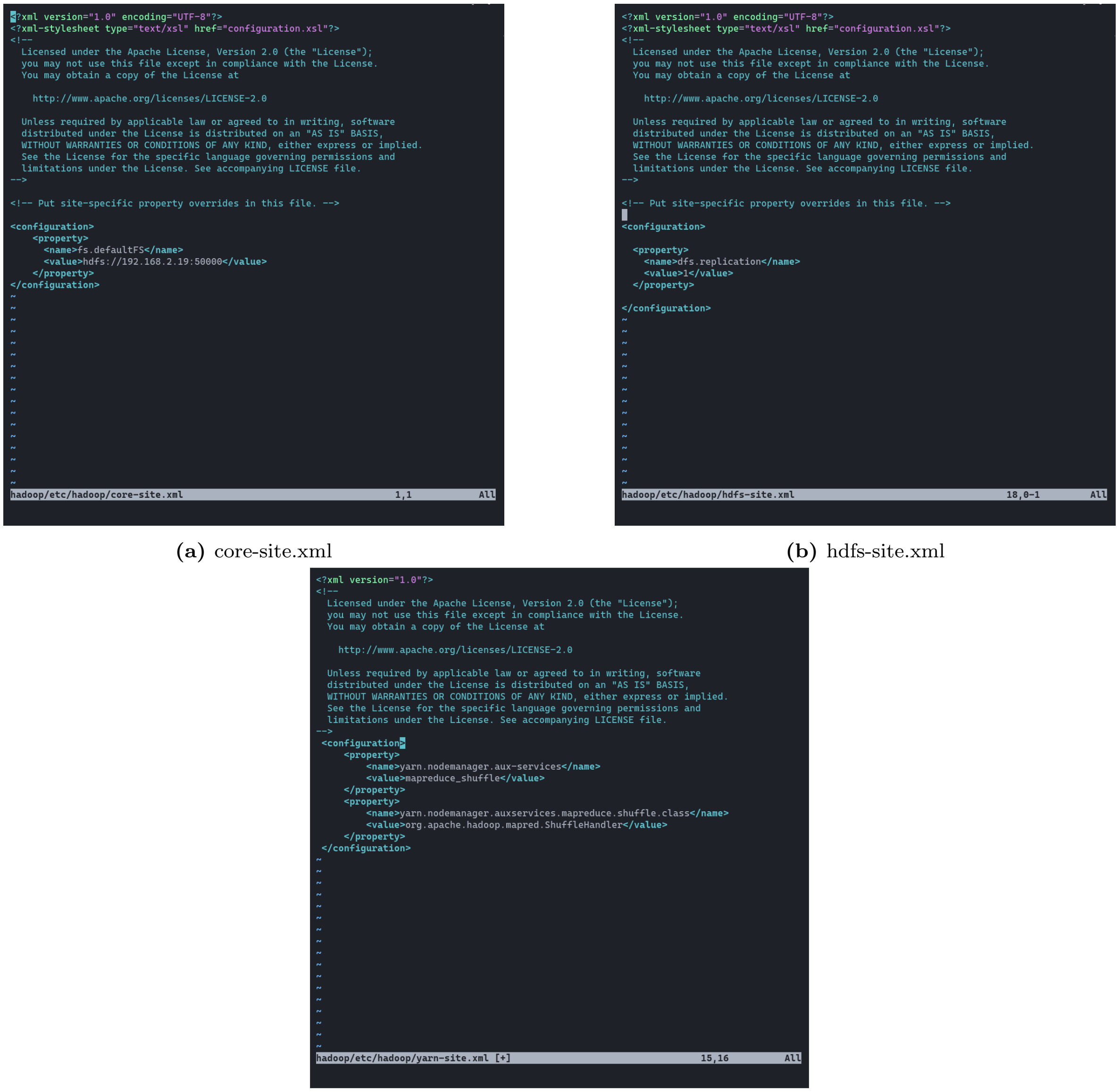
(b) hdfs-site.xml

```
  GNU nano 7.2         hadoop/etc/hadoop/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_C>
    </property>
    <property>
        <name>yarn.nodemanager.hostname</name>
        <value>192.168.2.19</value>
    </property>
    <property>
        <name>yarn.resourcemanager.address</name>
        <value>192.168.2.19:8032</value>
    </property>
</configuration>
```

(c) yarn-site.xml

Figure 17: Namenode configuration

(b) Slave:



```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
    <property>
      <name>fs.defaultFS</name>
      <value>hdfs://192.168.2.19:50000</value>
    </property>
</configuration>
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
hadoop/etc/hadoop/core-site.xml                                    1,1           All
```

(a) core-site.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>

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

</configuration>
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
hadoop/etc/hadoop/hdfs-site.xml                                    18,0-1        All
```

(b) hdfs-site.xml

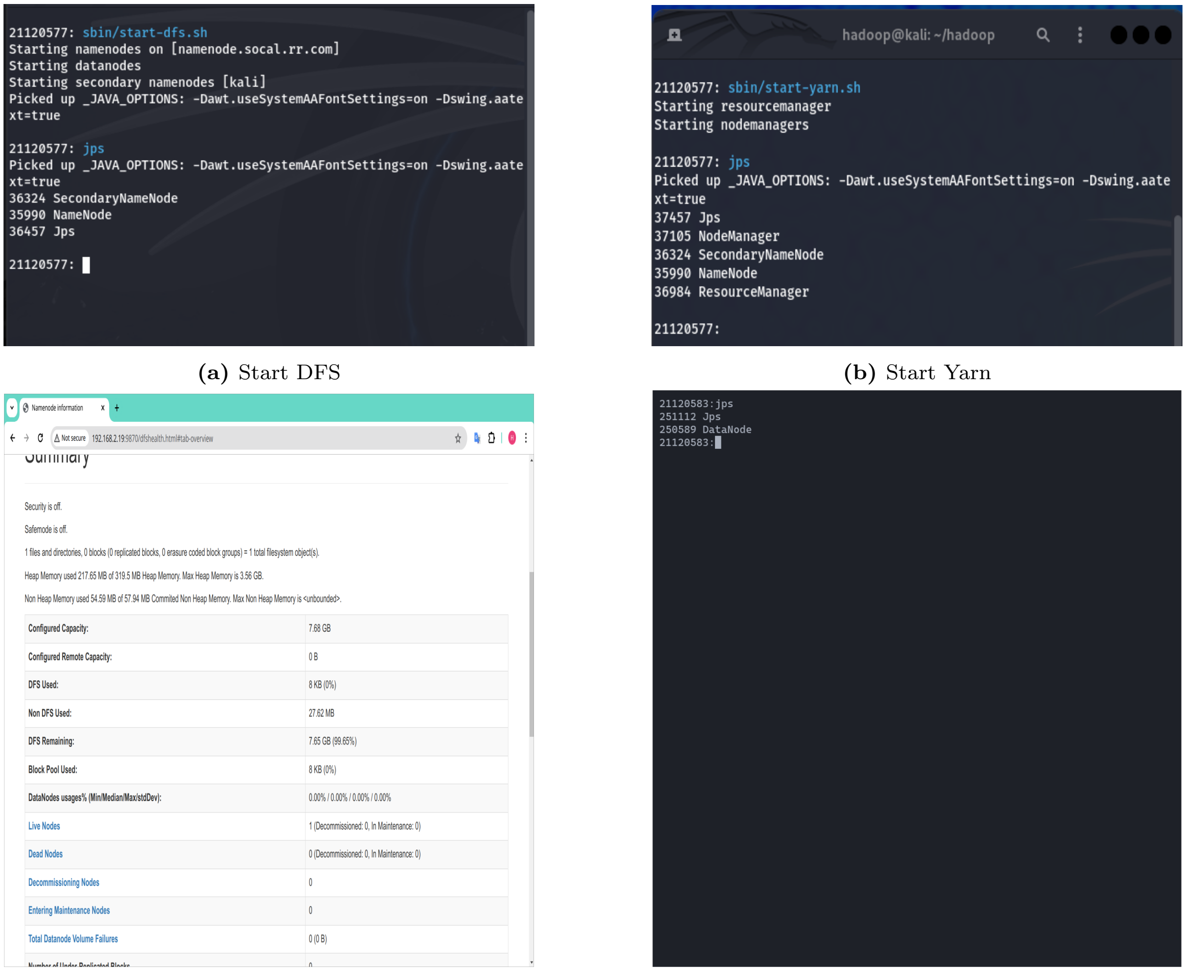
```
<?xml version="1.0"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->
 <configuration>
     <property>
         <name>yarn.nodemanager.aux-services</name>
         <value>mapreduce_shuffle</value>
     </property>
     <property>
         <name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>
         <value>org.apache.hadoop.mapred.ShuffleHandler</value>
     </property>
 </configuration>
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
hadoop/etc/hadoop/yarn-site.xml [+]                                15,16         All
```

(c) yarn-site.xml

Figure 18: Worker-node configuration

5. Format namenode again

6. Start hadoop on Master node



```
21120577: sbin/start-dfs.sh
Starting namenodes on [namenode.socal.rr.com]
Starting datanodes
Starting secondary namenodes [kali]
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aate
xt=true

21120577: jps
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aate
xt=true
36324 SecondaryNameNode
35990 NameNode
36457 Jps

21120577: 
```

(a) Start DFS

```
21120577: sbin/start-yarn.sh
Starting resourcemanager
Starting nodemanagers

21120577: jps
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aate
xt=true
37457 Jps
37105 NodeManager
36324 SecondaryNameNode
35990 NameNode
36984 ResourceManager

21120577:
```

(b) Start Yarn

Summary

Security is off.

Safemode is off.

1 files and directories, 0 blocks (0 replicated blocks, 0 erasure coded block groups) = 1 total filesystem object(s).

Heap Memory used 217.65 MB of 319.5 MB Heap Memory. Max Heap Memory is 3.56 GB.

Non Heap Memory used 54.59 MB of 57.94 MB Commited Non Heap Memory. Max Non Heap Memory is <unbounded>.

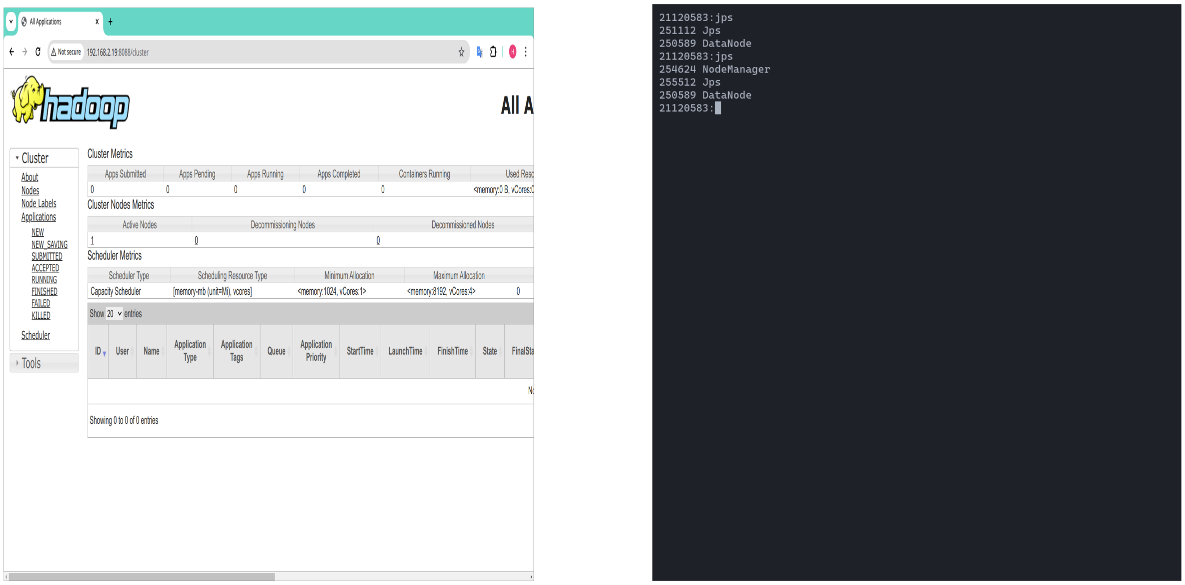
Configured Capacity:	7.68 GB
Configured Remote Capacity:	0 B
DFS Used:	8 KB (0%)
Non DFS Used:	27.62 MB
DFS Remaining:	7.65 GB (99.65%)
Block Pool Used:	8 KB (0%)
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%
Live Nodes	1 (Decommissioned: 0, In Maintenance: 0)
Dead Nodes	0 (Decommissioned: 0, In Maintenance: 0)
Decommissioning Nodes	0
Entering Maintenance Nodes	0
Total Datanode Volume Failures	0 (0 B)
Number of Under-Replicated Blocks	0

(c) Check connection in UI

```
21120583:jps
251112 Jps
250589 DataNode
21120583:
```

(d) Check datanode start at worker machine

Figure 19: Verify namenode start on slave successfully



Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Used Resc
0	0	0	0	0	<memory:0 B, vCores:(

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes
1	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:1024, vCores:1>	<memory:8192, vCores:4>	0

ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalSta
											N

Showing 0 to 0 of 0 entries

```
21120583:jps
251112 Jps
250589 DataNode
21120583:jps
254624 NodeManager
255512 Jps
250589 DataNode
21120583:
```

Figure 20: Verify YARN start on slave successfully

Note:

- To avoid unexpected behaviours, we should create the same user name for all machines.
- All machine must connect the same local network.
- Installing the same Hadoop version on each machine is required

b) Research about Security in Hadoop Set-up

1. Is your Hadoop secure? Give a short explanation if your answer is yes. Otherwise, give some examples of risks to your system.
   Answer:

The Hadoop system would not be considered secure. Here's some example of risks to my system:

User to Service Authentication:
+ No User Authentication on NameNode or JobTracker: If there is no mechanism to authenticate users, then any client can potentially claim to be any user, which is a significant security risk.
+ No User Authorization on DataNode: Prior to version 0.21, users could read/write any block, leading to the possibility of unauthorized data access and potential data leakage.
+ No User Authorization on JobTracker: Without proper authorization, users could modify or kill jobs that do not belong to them, affecting the data processing integrity and potentially causing a denial of service.

Service to Service Authentication:
+ No Authentication of DataNodes and TaskTrackers: Without authentication, there is a risk of unauthorized nodes joining the cluster and interfering with its operation or stealing data.

No Encryption on Wire or Disk:
+ Data transmitted across the network (on the wire) or stored on disk without encryption can be intercepted or accessed by unauthorized entities, leading to data breaches.

2. From your perspective, which method is better when securing your HDFS: authentication, authorization, or encryption? Give an explanation about your choices.
   Answer:

Authentication ensures that only identified and verified users can access the system. It's the first line of defense, preventing unauthorized access.

Authorization goes one step further by defining access rights and permissions for authenticated users. This ensures that users can only perform actions permitted by their role, which is essential for enforcing the principle of least privilege.

Encryption protects data at rest and in transit, ensuring confidentiality and integrity. Even if an attacker circumvents authentication and authorization mechanisms, encryption can prevent them from reading or tampering with the data.

Each of these methods serves a different purpose:
- Authentication protects against unauthorized entry into the system.
- Authorization ensures that users can't perform actions they're not permitted to do.
- Encryption keeps the data secure even if the system is breached.

In the context of a student project, authentication would be the most suitable method to prioritize when securing Hadoop Distributed File System (HDFS).
Authentication ensures that only authorized individuals can access the HDFS by verifying their identity. By implementing strong authentication mechanisms, such as username/-password authentication or integration with external authentication systems like LDAP, the project can enforce access controls and prevent unauthorized users from gaining access to the HDFS.

3 Reflection

3.1 Journey to the Deadline:

Our work is divided into 4 main phase:

1. *Each member implements Standalone Operation and Pseudo-Distributed Operation of Hadoop.*
2. *Reading Paper and Answer question about MapReduce model*
3. *Running example of Word Count program and implementing MapReduce programs: Word Length Count*
4. *Implementing Fully-connected Distributed Operation on 2 physical devices and researching about Security in Hadoop Set-up*

Phase	Time-consumption	Bugs
1	~6 days	+ ssh: connection denied port 22 + java version is not compatible with hadoop version + cannot create folder in hdfs
2	~2 days	
3	~2 days	
4	~3 days	+ cannot start datanode on worker device + cannot set up the datanode workspace

3.2 Overcoming Challenges

1. SSH problem: we have researched the problem and found that it can be solve by restarting ssh service.

```
sudo service ssh restart
```

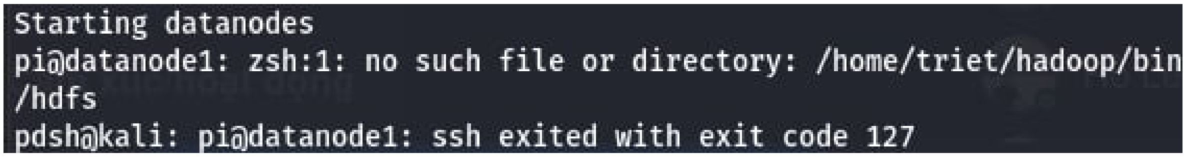
```
ssh localhost
```

2. java version is not compatible with hadoop version (3.4.0). We found that cannot run hdfs with jdk-21 so we solved by installing jdk-11.
3. Cannot create folder on hdfs. The problem is default root of hdfs which start `/user/username` so we need create that folder first of add / before any command.
4. Cannot start datanode on worker device. After configurating, we did not connect ssh between master device and worker device that why master node cannot connect to worker node. Another problem is the permission of hadoop folder is root. We solved by run commands change owner of hadoop folder

```
sudo chown hadoop:hadoop -R /home/hadoop
```

```
chmod 700 /home/hadoop
```

5. Cannot set up the datanode workspace. The default workspace of datanode is `/home/username/hadoop/bin/hdfs`, so when we start datanode from master we cannot find `/home/master-username/hadoop/bin/hdfs` on worker device and we try to configure `hdfs-site.xml` but it's still not working. We have to create the user named hadoop on both master and worker device to solved this problem.



```
Starting datanodes
pi@datanode1: zsh:1: no such file or directory: /home/triet/hadoop/bin
/hdfs
pdsh@kali: pi@datanode1: ssh exited with exit code 127
```

Figure 21: No such file error

3.3 Lesson Learned

After setting up a Hadoop cluster and running a MapReduce program, there are several valuable lessons that can be learned. Firstly, the process of setting up the cluster itself provides insights into distributed systems and infrastructure management. Understanding the configuration and interplay of various components such as NameNode, DataNode, ResourceManager, and NodeManager enhances knowledge of cluster architecture.

Running a MapReduce program offers practical experience in processing large-scale data in a distributed environment. It highlights the importance of data partitioning, task scheduling, and fault tolerance mechanisms. Dealing with input/output formats, understanding the Map and Reduce functions, and optimizing data locality contribute to a deeper comprehension of parallel processing and data manipulation.

Moreover, resolving the problems encountered during the setup of the Hadoop cluster can significantly improve our team's Linux command skills. The process of configuring and troubleshooting the cluster often involves working with various Linux commands and utilities. Setting up a Hadoop cluster typically requires tasks such as installing and configuring the operating system, managing user permissions, modifying network settings, and installing and configuring software packages. These tasks often involve executing Linux commands to navigate the file system, modify configuration files, manage services, and troubleshoot issues.

4 Reference

- Hadoop - Multi-Node Cluster
- Hadoop Multi-node Cluster Installation
- Viblo - Set-up multi node cluster hadoop
- Safe Mode in Hadoop
- Apache Hadoop 3.3.6 – Hadoop Cluster Setup
- Apache Hadoop 3.3.6 – Hadoop: Setting up a Single Node Cluster
- Apache Hadoop 3.3.6 – Hadoop in Secure Mode
- Apache Hadoop 3.3.6 – MapReduce Tutorial