

INFSCI 2750: Cloud Computing

Mini Project 1

Sai Charan Talipineni (sat122) || Sai Rakesh Ghanta (sag163)

Part 1: Setting up Hadoop

We built the Hadoop cluster following the sequence of preparatory steps to install and run Hadoop. We started the cluster and ran the default wordcount program that comes as part of the Hadoop package.

Master: 104.236.42.214

Slave: 104.236.73.214

Master

```
root@master: ~  
root@master:~# jps  
21760 ResourceManager  
21361 DataNode  
21890 NodeManager  
21576 SecondaryNameNode  
21147 NameNode  
22205 Jps
```

Slave

```
root@slave: ~  
root@slave:~# jps  
29874 NodeManager  
29730 DataNode  
30046 Jps  
root@slave:~#
```

Part 2: Building Hadoop Docker Image

We built a Ubuntu Docker image and based on the Ubuntu Docker image, you are required to build the Hadoop Docker image with the help of the steps in Part 1.

Then, we ran it locally and tested it with running a Wordcount job on it.

```
uncommented 1
under 75
unset 1
up 2
updating 1
usage 2
use 27
use. 2
used 4
used. 36
user 48
user. 2
user1,user2 2
user? 1
users 27
users,wheel". 18
uses 2
using 14
value 46
value. 1
value="20"/> 1
value="30"/> 1
values 4
variable 4
variables 4
version 1
version="1.0" 3
version="1.0"> 1
version="1.0"?> 6
via 3
view. 1
viewing 1
w/ 1
want 1
warnings. 1
when 9
where 4
which 7
while 1
who 6
will 24
window 1
window. 1
with 58
within 4
without 1
work 13
writing. 21
written 2
xmlns:xsl="http://www.w3.org/1999/XSL/Transform" 1
yarn.nodemanager.linux-container-executor.group 1
yarn.nodemanager.linux-container-executor.group=#configured 1
yarn.nodemanager.local-dirs 1
yarn.nodemanager.log-dirs 1
yarn.server.resourcemanager.appsummary.log.file 1
yarn.server.resourcemanager.appsummary.log.file=rm-appsummary.log 1
yarn.server.resourcemanager.appsummary.logger 2
yarn.server.resourcemanager.appsummary.logger=${hadoop.root.logger} 1
you 26
root@ee161bb516dd:/usr/local/hadoop#
```

Dockerfiles and supporting files for Ubuntu Docker and Hadoop Docker images can be found in Ubuntu Docker and Hadoop Docker folder respectively. README.md files are included in each folder to illustrate the procedure.

Part 3: Developing a Hadoop program (N-Gram)

We performed the digram as an example and shown below,

```
Bytes Written=45
root@master:/usr/local/hadoop# hdfs dfs -cat /user/root/test123output/*
17/02/25 21:37:00 WARN util.NativeCodeLoader: Unable to load native-hadoop
el      1
he      1
ld      1
ll      1
lo      1
or      1
ow      1
rl      1
wo      1
root@master:/usr/local/hadoop#
```

Part 4: Developing a Hadoop program to analyze real logs

1. How many hits were made to the website item “/assets/img/homelogo.png”?

Answer: 98744

2. How many hits were made from the IP: 10.153.239.5

Answer: 547

3. Which path in the website has been hit most? How many hits were made to the path?

Answer: ('/assets/css/combined.css', 117348)

4. Which IP accesses the website most? How many accesses were made by it?

Answer: ('10.216.113.172', 158614)

We performed the above programs and found the above output. For the 3,4 programs we changed the reducer part and found the maximum of the number.

>Note: Source codes are in the zip folder. Readme files are in the respective folders.