

Word Count on Shared-Memory/Hadoop/Swift

CS553 Fall 2014

Summary

This report represents a detailed analysis of word-count application implemented in three different languages **Hadoop, Shared-memory in Java and Swift**. The respective performance and speedup of each module across various scales ranging from 1 node to 16 nodes is determined and a performance evaluation chart was generated. The most optimal condition was obtained after running a series of experiments.

1. Introduction:

Cloud computing: An Internet cloud of resources can be either a centralized or a distributed computing system. The cloud applies parallel or distributed computing, or both. Clouds can be built with physical or virtualized resources over larger data centers that are centralized or distributed.

In this report, we will try to highlight the steps involved in setting up virtual cluster of 1 node up to 16 nodes using *Amazon Web Services (AWS) Elastic Cloud Computing (EC2)* and present our results by increasing the number of nodes when running *shared memory program, Hadoop framework and swift parallel programming language*.

2. Virtual Cluster(1-node):

Amazon EC2 is a web service that provides resizable compute capacity in the cloud. It is designed to make cloud computing easier for developers.

Amazon EC2 offers complete control of your computing resources, reduces the time required to obtain and boot new server instances in minutes and allows you to quickly scale capacity, both up and down, as your computing requirements change.

Running a virtual server in Amazon EC2 is as easy as setting up a Facebook account

for the first time, where an EC2 instance can be created by simple button clicks by selecting the configuration required for the servers. These virtual servers on AWS is called *EC2 Instances* and yours instances are initialized using machine images known as *Amazon Machine Images (AMIs)*.

A quick guide to launch to an Amazon EC2 *C3.large* instance is as follows:

- a. Open Amazon EC2 Console
- b. On the dashboard, click *Launch Instance*.
- c. Then choose an Amazon Machine Image (AMI) page displays a list of available AMIs. Select the *64-bit "Ubuntu Server 14.04 LTS (HVM), SSD Volume Type"* AMI. This AMI is marked Free tier eligible.
- d. On the *Choose an Instance Type* page, the "*c3.large*" instance type is selected.
- e. On the *Configure Instance Details* page, we can specify the number of instances that we would like to start. Default is one. Also, select Purchasing option "*Request Spot Instances*" where the pricing for running in the system is based on bidding which doesn't provide a guaranteed resource but provides a low cost compared to the On-Demand pricing.
- f. On the *Add Storage* page we can notice the amount of storage allocated for *c3.large* instance type having a default value of *2*16 SSD*.
- g. On the *Tag Instance* page, we would specify the name for our instance.
- h. On the *Configure Security Group* page we can set the firewall rules that control the traffic for our instances. By default *SSH Port 22* is open.
- i. *Review and Click Launch*.
- j. In the *Select an existing key pair or create a new key pair* dialog box, select

Word Count on Shared-Memory/Hadoop/Swift

CS553 Fall 2014

Create a new key pair, enter a name for the key pair, and then click Download Key Pair.

- k. When you are ready, select the acknowledgment check box, and then click Launch Instances.
- l. Now your EC2 instance appears on the Instances page under Spot Request.
- m. Once the resource is granted the state changes to running, and it receives a public DNS name.

3. Shared-Memory Word Count:

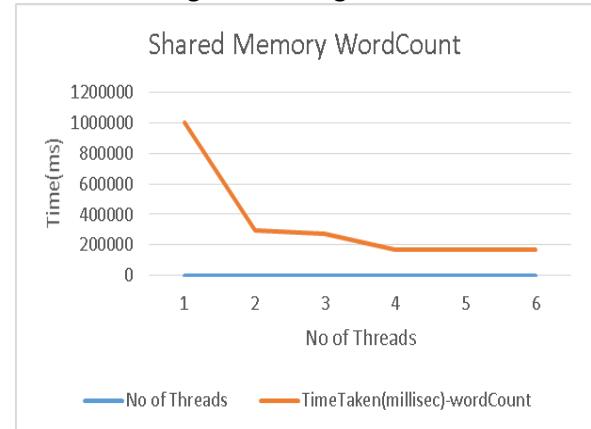
This module intends to calculate the performance of a Word Count Application when executed using single and multiple threads. The Word Count application is written using JAVA programming language and the experiment is done by reading 10 GB text file (wiki10gb.txt).

The application makes use of a **java.io.LineNumberReader** to read file line by line and a regular expression ("\\'? ([a-zA-z0-9'-]+)\\'?") to extract words from line using Regex pattern matcher. Each unique word is stored in a concurrent hash map static to the class WordProcessor.

Word Count Update: hm.put(hm.get(word), (hm.get(word)!=null?hm.get(word)+1:1)); here hm is the hash map storing the count of words as ConcurrentHashMap<String , Integer>(). This program uses a ConcurrentHashMap lineCheck to memoize the data read by threads, each thread checks the **memoized array** before reading the new line. If the line no. is found in the memorized array, the current line is skipped otherwise it is fed to tokenize String (String line) to count the words and update static hashmap hm.

Word count application is executed on an Amazon c3.large spot instance with 32GB local storage and the results are shown as on graph. Reading a 10gb file and performing the word count takes more than

999999 ms and throws an OutOfMemoryException, But on increasing the no of threads we can see a considerable decrease in the processing time for the application and it takes almost constant time for more than 6 threads. Efficiency of the program will differ on different hardware, though it can be further improved by tuning the cycle of garbage collection using runtime arguments.



4. Virtual Cluster (17-node):

Amazon EC2 offer scaling from 1 node virtual cluster to multi node in easy steps. We can proceed with setting up a virtual cluster with 16 nodes by creating an image of the system from step #1 and launch 16 instance from the AMI created. By doing this, all the settings will be replicated will be replicated across all the 16 nodes.

5. Hadoop:

The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application

Word Count on Shared-Memory/Hadoop/Swift

CS553 Fall 2014

layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures.

In this assignment we have used **Hadoop 2.4.1** version and the source code for **Hadoop has been re-compiled for 64 bit system for HadoopNativeLibrary support.** **Hadoop Distributed File System (HDFS™)** A distributed file system that provides high-throughput access to application data has been configured to make use of the 32 GB local disk offered by C3.Large instance for Hadoop file system storage. **Hadoop YARN:** A framework for job scheduling and cluster resource management was used for job tracking. **Hadoop MapReduce:** A YARN-based system for parallel processing of large data sets has been implemented for distributing the work across all the slave nodes.

Note: The jobs can be tracked and viewed using Hadoop Dashboard:

http://<public_ip>:8088/cluster

6. Hadoop WordCount:

Hadoop Word Count is an example to demonstrate the work flow distribution of large data set across cluster and process the data in parallel and distributed using MapReduce programming model.

A Hadoop cluster includes a single master and multiple worker nodes. Master node consists of a JobTracker, TaskTracker, NameNode and DataNode. A slave or worker node acts as both a DataNode and TaskTracker, though it is possible to have data-only worker nodes and compute-only worker nodes.

The **etc/hadoop/master** file contains the DNS mapping to the nodes which acts as a Master Node. The **etc/hadoop/slaves** file

contains the DNS mapping to the slave nodes which acts as the slave/workers nodes in the cluster. The **etc/Hadoop/core-site.xml** is a Site-specific configuration for a given hadoop installation which users can add additional resources. The **etc/Hadoop/hdfs-site.xml** contains the configuration parameters for the HDFS deamons NameNode and DataNode, which are specified by *dfs.namenode.name.dir* and *dfs.datanode.name.dir* name property. The **etc/Hadoop/mapred-site.xml** is a configuration option for Hadoop 2 which has the capability to specify a framework name for MapReduce, setting the *mapreduce.framework.name* property. In this install we will use the value of "yarn" to tell MapReduce that it will run as a YARN application.

A sanity check was performed on single node to ensure the Hadoop was configured and setup correctly before moving forward to the cluster setup. The results of the single node is provided and the value is compared with shared memory program in the performance section[9].

While moving forward from one node **Pseudo-Distributed Mode** to **Fully-Distributed Mode**, in Master node, the master file contains the master name and the slave file contains the list of all slaves while in slave nodes, the master file is empty and the slave file contains localhost. In the **hdfs-site.xml**, an extra property "**dfs.namenode.hosts**" was added which contains a list of permitted DataNodes and in **yarn-site.xml**, extra few properties were added pointing to the master node's resourcemanger's port numbers to ensure that all slaves and masters can communicate to the master over the single port to detect any failures.

The number of mappers and reduce task can be modified by defining the property in

Word Count on Shared-Memory/Hadoop/Swift

CS553 Fall 2014

etc/Hadoop/mapred-site.xml by providing the value for “**mapreduce.jobs.maps**” and “**mapreduce.job.reduces**” name tag. The default value for the number of map tasks per job is 2 and the default number of reduce task per job is 1 which is modified to 2 for full utilization of c3.large instance.

The cluster settings is replicated to all nodes in the cluster having 1 Master node and 16 slave nodes. The Hadoop word count application has been tested across the cluster and the comparison for scaling from 1 node to 16 nodes is analyzed and depicted in the performance section.

7. Swift:

Swift is a simple scripting language for executing many instances of ordinary application programs on distributed parallel resources. Swift scripts run many copies of ordinary programs concurrently. Swift acts like a structured "shell" language. It runs programs concurrently as soon as their inputs are available, reducing the need for complex parallel programming. Swift's **foreach {}** statement is the **main parallel workhorse** of the language.

The version used for the experiment is **swift-0.95-RC6**. Initially we tried with the stable version swift-0.94.1, but ended up facing problems during configuration in local-host. Since swift is a java based application it is important that we install a JDK and export the JAVA_HOME path from the configuration file of swift. The path of swift also has to be set up such that it points to the /bin folder. **source setup.sh** command is used to setup an environment where we can run swift programs.

In order to run swift on cloud, **an IAM account** is required. After the account is created, we should create a username and a user group in which the user should be

added. A credentials.csv file should be downloaded which will be used for configuration of head-node and workers. The most important part of creating a cluster is the **Launchpad**, from which all the worker nodes including the head-node are triggered. Before that, the **config** file must be setup in order to specify the path of **.pem** file, credentials and the number of workers. Once that is done execute “**source setup.sh**”. The head node along with the requested number of workers will get created in EC2. Connect to the head-node and upload the wiki dump and run the swift program.

The example programs were run to make sure all the workers are being assigned jobs.

8. Swift WordCount:

The word count program is implemented to run on both 1 node and 16 nodes respectively where the swift program assigns tasks for all the workers using the head-node and **swift.conf** file.

The word count analysis is performed using shell script program which is encapsulated by the swift code which performs the work distribution over the cluster to all the worker nodes. The intermediate output from each worker is stored in the centralized location (EBS) and the merge of the output is performed by another bash script which is invoked by swift for parallel processing.

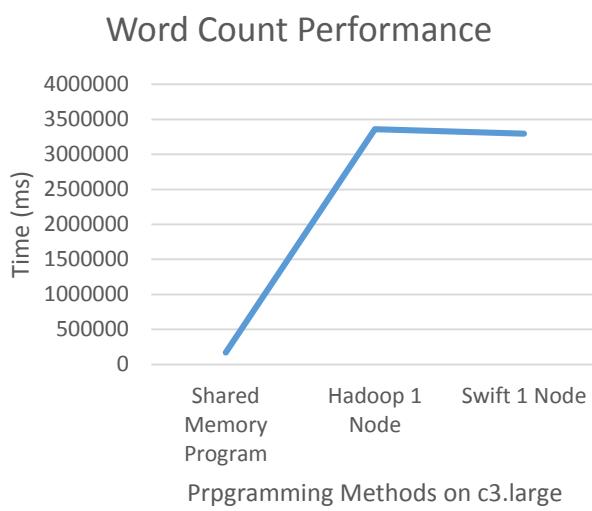
The Swift word count application has been tested across the cluster and the comparison for scaling from 1 node to 16 nodes is analyzed and depicted in the performance section.

Word Count on Shared-Memory/Hadoop/Swift

CS553 Fall 2014

9. Performance:

The three versions of Word Count - Shared-Memory, Hadoop and Swift on 1 node scale performance's is compared by the bases of reading a wiki dump file of size



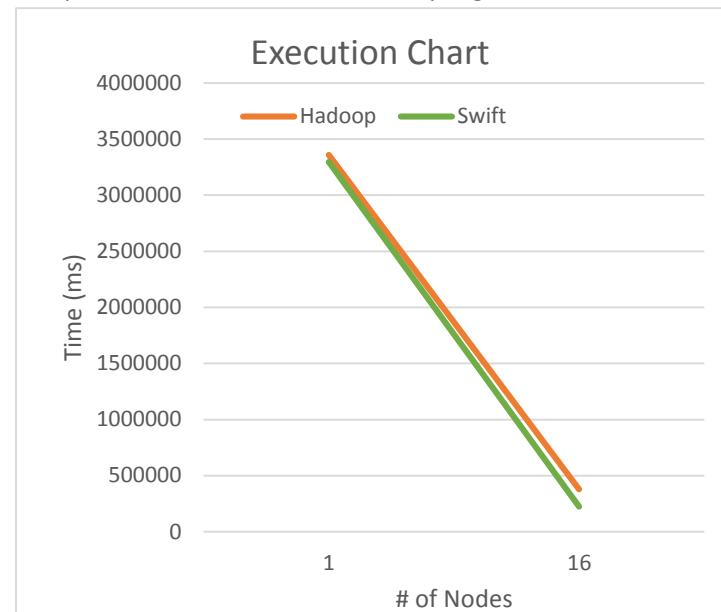
10 GB and the time taken to count the occurrence of the words has been plotted. The run time comparison is made with Java 8 threads, Hadoop run time is calculated based on the total time taken by the map tasks + total time taken by the reduce tasks and Swift run time is calculated based on the total time taken for the program execution.

From the graph above we can see that performance of the java shared-memory program running with 8 threads seems to be faster when compared to Hadoop and Swift. From the results we can depict that Shared memory program best utilizes the system resources.

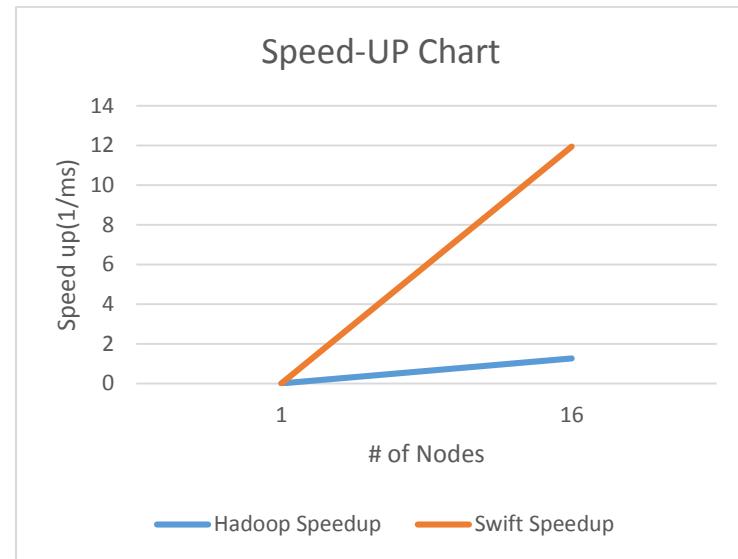
The Hadoop and Swift application are scaled from 1 node to 16 node cluster on c3.large instance in Amazon EC2. The results on the scaling is plotted as below.

From the graph given below we can infer that, As we scale to higher number of nodes

Swift gives a steep increase in its performance , whereas Hadoop lags behind.



$$\text{Speedup} = S = T / [\alpha T + (1-\alpha)T/n] = 1 / [\alpha + (1-\alpha)/n]$$



Scalability (1-N nodes): We can infer from the speed up chart and the execution chart that the performance increased when we scale from 1 to 16 nodes but this doesn't guarantee the increase in performance when we scale to 1000 nodes as the data size doesn't increase and the system infrastructure in cluster stays underutilized, With reference to **Amdahl's Law**.

Word Count on Shared-Memory/Hadoop/Swift

CS553 Fall 2014

10. Sort on Shared-Memory, Hadoop, Swift, and MPI:

The Key-Value pairs of the output is sorted using Shared-Memory, Hadoop and Swift programs. The Sort outputs are provided in the /output folder.

11. Constraints:

a. S3: We tried to configure mount **S3** on EC2 instance and encountered errors while performing the mount which is shown below.

In order to mount s3 as storage system on EC2 instance we are required to configure S3FS .

1.

```
->/ $ s3fs --version
```

```
Amazon Simple Storage Service File System  
1.74
```

```
Copyright (C) 2010 Randy Rizun  
<rrizun@gmail.com>
```

```
License GPLv2: GNU GPL version 2  
<http://gnu.org/licenses/gpl.html>
```

This is free software: you are free to change and redistribute it.

There is NO WARRANTY, to the extent permitted by law.

2.

```
->sudo s3fs -o
```

```
uid=1000,gid=1000,use_cache=/home/ubuntu/cache ssharm58 /mnt/s3
```

```
->mount
```

```
samuel@shashank-Lenovo-G410:/$ mount  
/dev/sda7 on / type ext4  
(rw,errors=remount-ro)
```

```
proc on /proc type proc  
(rw,noexec,nosuid,nodev)
```

```
sysfs on /sys type sysfs  
(rw,noexec,nosuid,nodev)
```

```
none on /sys/fs/cgroup type tmpfs (rw)
```

```
none on /sys/fs/fuse/connections type  
fusectl (rw)
```

```
none on /sys/kernel/debug type debugfs  
(rw)  
none on /sys/kernel/security type securityfs  
(rw)  
none on /sys/firmware/efi/efivars type  
efivars (rw)  
udev on /dev type devtmpfs  
(rw,mode=0755)  
devpts on /dev/pts type devpts  
(rw,noexec,nosuid,gid=5,mode=0620)  
tmpfs on /run type tmpfs  
(rw,noexec,nosuid,size=10%,mode=0755)  
none on /run/lock type tmpfs  
(rw,noexec,nosuid,nodev,size=5242880)  
none on /run/shm type tmpfs  
(rw,nosuid,nodev)  
none on /run/user type tmpfs  
(rw,noexec,nosuid,nodev,size=104857600,  
mode=0755)  
none on /sys/fs/pstore type pstore (rw)  
/dev/sda2 on /boot/efi type vfat (rw)  
binfmt_misc on /proc/sys/fs/binfmt_misc  
type binfmt_misc (rw,noexec,nosuid,nodev)  
systemd on /sys/fs/cgroup/systemd type  
cgroup  
(rw,noexec,nosuid,nodev,none,name=systemd)  
gvfsd-fuse on /run/user/1000/gvfs type  
fuse.gvfsd-fuse  
(rw,nosuid,nodev,user=samuel)  
gvfsd-fuse on /home/samuel/.gvfs type  
fuse.gvfsd-fuse (rw,nosuid,nodev)  
s3fs on /mnt/s3 type fuse.s3fs  
(rw,nosuid,nodev)  
3:->$:$/$ sudo ls /mnt/s3  
ls: cannot access /mnt/s3: Transport  
endpoint is not connected
```

b. MPI: Problem requires us to read 10GB text file and count unique words and perform sort using Message Passing Interface.

We found various implementation of open MPI in C and Java programming language.

Word Count on Shared-Memory/Hadoop/Swift

CS553 Fall 2014

C:<http://www.open-mpi.org/software/ompi/v1.4/>
Java: <http://mpj-express.org/>

We tried setting up the using MPJ Express (java implementation of openMPI) in local system and used the following test program available on MPJ express website.

```
import mpi.*;
public class HelloWorld {
    public static void main(String arg[]) throws
        Exception
    {
        MPI.Init(arg);
        int me = MPI.COMM_WORLD.Rank();
        int size = MPI.COMM_WORLD.Size();
        System.out.println("Running Job In
<"+me+">");
        MPI.Finalize();
    }
}
```

12. Conclusion:

From the following experiment we can observe that swift gives a better performance when scaled to N number of nodes with comparison to java and hadoop.

Word Count on Shared-Memory/Hadoop/Swift

CS553 Fall 2014

13. Screen Shots:

a. Virtual Cluster 1 Node:

The screenshot shows the AWS EC2 Management Console interface. On the left, there's a sidebar with navigation links like EC2 Dashboard, Events, Tags, Reports, Limits, Instances (selected), Spot Requests, Reserved Instances, Images, AMIs, Bundle Tasks, Elastic Block Store, Volumes, Snapshots, Network & Security, Security Groups, Elastic IPs, Placement Groups, Load Balancers, and Key Pairs. The main content area displays a table of instances. One instance is listed: i-fc05e8f0, which is a c3.large type running in us-west-2a. It has a Public DNS of ec2-54-69-98-192.us-west-2.compute.amazonaws.com and a Public IP of 54.69.98.192. The instance was launched on October 19, 2014, at 12:29:23 PM UTC-5.

The screenshot shows the AWS EC2 Management Console interface. On the left, there's a sidebar with navigation links like EC2 Dashboard, Events, Tags, Reports, Limits, Instances, Spot Requests (selected), Reserved Instances, Images, AMIs, Bundle Tasks, Elastic Block Store, Volumes, Snapshots, Network & Security, Security Groups, Elastic IPs, Placement Groups, and Key Pairs. The main content area displays a table of spot requests. A single spot request named "test" is listed, created on October 23, 2014, at 6:47:03 PM UTC-5. It's for a c3.large instance with an AMI ID of ami-5b85ca6b, with a maximum price of \$0.042. The state is open and pending-evaluation. The request is for Linux/UNIX and is associated with the launch-wizard-1 security group.

Word Count on Shared-Memory/Hadoop/Swift

CS553 Fall 2014

EC2 Management Console - Google Chrome

https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#SecurityGroups:sort=groupid

Services Edit

Create Security Group Actions

Filter by tags and attributes or search by keyword

1 to 3 of 3

Name	Group ID	Group Name	VPC ID	Description
sg-2787e142	launch-wizard-1	vpc-7b984d1e	launch-wizard-1 created 2014-10-21T17:27....	
sg-42ff9b27	MyNetworkGroup	vpc-7b984d1e	launch-wizard-1 created 2014-10-17T20:24....	

Security Group: sg-42ff9b27

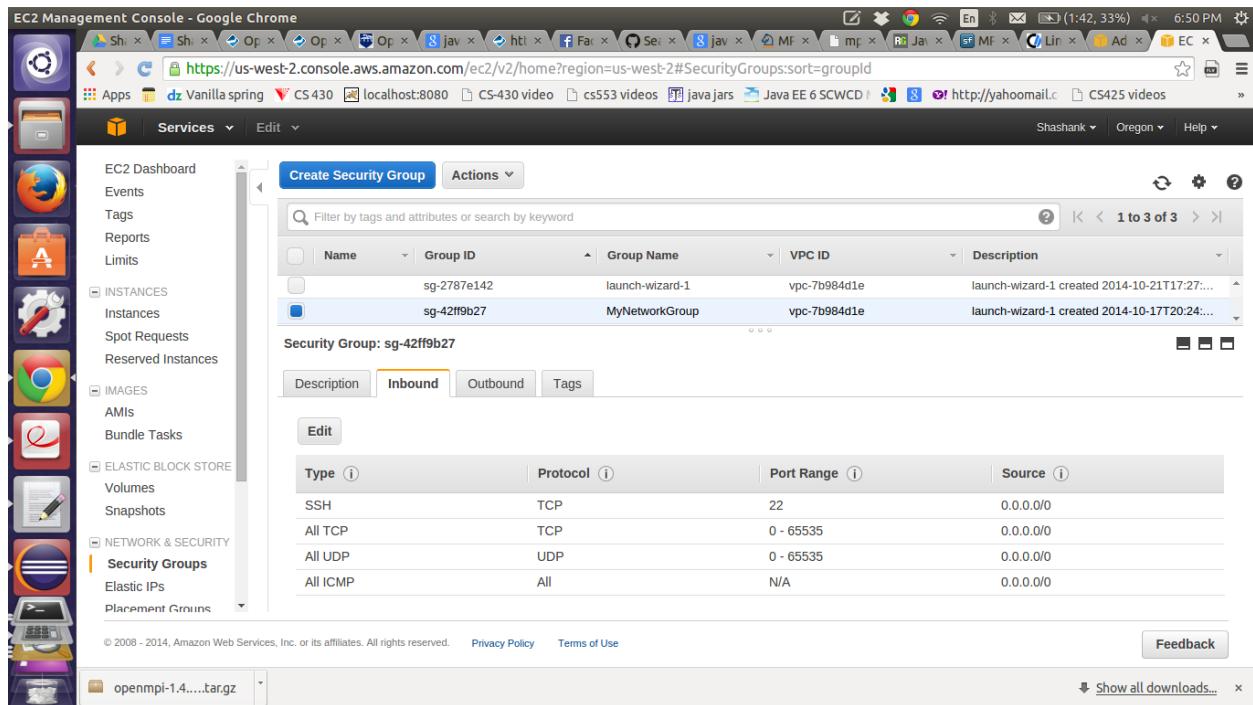
Description Inbound Outbound Tags

Edit

Type	Protocol	Port Range	Source
SSH	TCP	22	0.0.0.0/0
All TCP	TCP	0 - 65535	0.0.0.0/0
All UDP	UDP	0 - 65535	0.0.0.0/0
All ICMP	All	N/A	0.0.0.0/0

© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Feedback

openmpi-1.4....tar.gz Show all downloads...

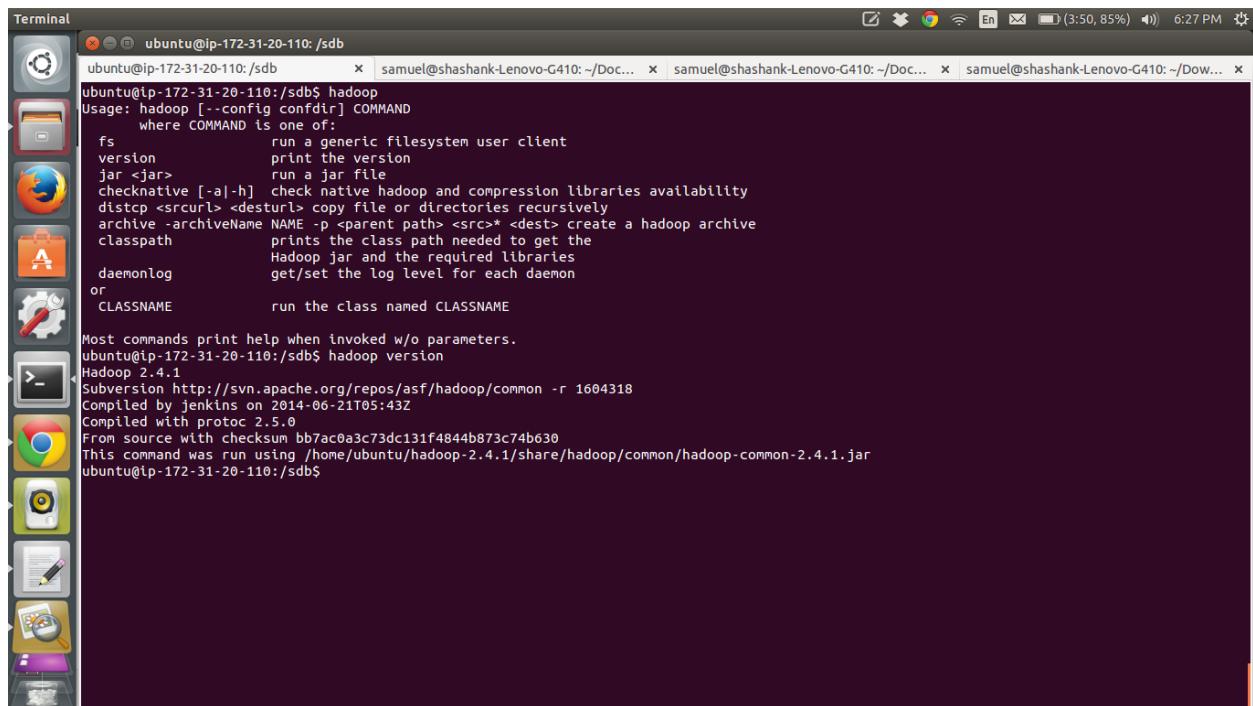


Terminal

ubuntu@ip-172-31-20-110: /sdb

```
ubuntu@ip-172-31-20-110:/sdb$ hadoop
Usage: hadoop [--config confdir] COMMAND
  where COMMAND is one of:
    fs          run a generic filesystem user client
    version     print the version
    jar <jar>    run a jar file
    checksum [-a|-h] check native hadoop and compression libraries availability
    distcp <srcurl> <desturl> copy file or directories recursively
    archive -archiveName NAME -p <parent path> <src>* <dest> create a hadoop archive
    classpath   prints the class path needed to get the
                Hadoop jar and the required libraries
    daemonlog   get/set the log level for each daemon
    or
    CLASSNAME   run the class named CLASSNAME

Most commands print help when invoked w/o parameters.
ubuntu@ip-172-31-20-110:/sdb$ hadoop version
Hadoop 2.4.1
Subversion http://svn.apache.org/repos/asf/hadoop/common - r 1604318
Compiled by jenkins on 2014-06-21T05:43Z
Compiled with protoc 2.5.0
From source with checksum bb7ac0a3c73dc131f4844b873c74b630
This command was run using /home/ubuntu/hadoop-2.4.1/share/hadoop/common/hadoop-common-2.4.1.jar
ubuntu@ip-172-31-20-110:/sdb$
```



Word Count on Shared-Memory/Hadoop/Swift

CS553 Fall 2014

The screenshot shows a Linux desktop environment with a dark theme. On the left is a vertical dock containing icons for various applications like a terminal, file manager, and system settings. The main window is a terminal window titled 'ubuntu@ip-172-31-32-64:~' showing Java and Swift version outputs:

```
ubuntu@ip-172-31-32-64:~$ java -version
java version "1.7.0_65"
OpenJDK Runtime Environment (IcedTea 2.5.2) (7u65-2.5.2-3-14.04)
OpenJDK 64-Bit Server VM (build 24.65-b04, mixed mode)

ubuntu@ip-172-31-32-64:~$ swift -version
Swift 0.95 RC6 swift-r7900 cog-r3908
ubuntu@ip-172-31-32-64:~$
```

Below the terminal is a Google Chrome window titled 'Namenode Information'. The address bar shows '54.187.113.51:50070/dfshealth.html#tab-datanode'. The page content is titled 'Datanode Information' and shows a table for 'In operation' nodes:

Node	Last contact	Admin State	Capacity	Used	Non DFS Used	Remaining	Blocks	Block pool used	Failed Volumes	Version
ip-172-31-25-19.us-west-2.compute.internal (127.0.0.1:50010)	1	In Service	15.62 GB	9.77 GB	880.34 MB	5 GB	83	9.77 GB (62.51%)	0	2.4.1

There is also a section titled 'Decommissioning' with a table:

Node	Last contact	Under replicated blocks	Blocks with no live replicas	Under Replicated Blocks In files under construction

At the bottom of the browser window, there are several tabs: 'hadoop_bash.sh', 'cluster_steps_from...', 'HadoopWordCo...jar', 'HadoopCluster.pem', and 'Show all downloads...'. A vertical toolbar on the left side of the desktop contains icons for file operations, system settings, and other applications.

Word Count on Shared-Memory/Hadoop/Swift

CS553 Fall 2014

All Applications - Google Chrome

54.187.113.51:8088/cluster

Logged in as: dr.wh

All Applications

hadoop

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
1	0	1	0	7	8 GB	8 GB	0 B	1	0	0	0	0

Show 20 entries Search:

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking UI
application_1413943201007_0001	ubuntu	Hadoop Word Count	MAPREDUCE	default	Wed, 22 Oct 2014 02:05:04 GMT	N/A	RUNNING	UNDEFINED		ApplicationMaster

Showing 1 to 1 of 1 entries First Previous 1 Next Last

About Apache Hadoop

hadoop_bash.sh cluster_steps_from... HadoopWordCo...jar HadoopCluster.pem Show all downloads...

```
ubuntu@ip-172-31-46-246:~$ Current workers : 0
Workers requested : 2
Starting worker(s) swift-worker-000 swift-worker-001
No worker_disk defined
[<Nodesize: id=c3.large, name=Compute Optimized Large Instance, ram=3750 disk=16
bandwidth=None price=0.105 driver=Amazon EC2 (us-west-2) ...>]
-----
NAME          | STATUS      | EXTERNAL_IP
headnode      | RUNNING     | 54.69.157.7
-----
Launchpad     | RUNNING     | 54.187.58.156
-----
ubuntu@ip-172-31-18-154:~/cloud-tutorials/ec2$ connect headnode^C
ubuntu@ip-172-31-18-154:~/cloud-tutorials/ec2$ connect headnode
Connecting to AWS node:headnode on 54.69.157.7 as ubuntu
Warning: Permanently added '54.69.157.7' (EDDSA) to the list of known hosts.
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-36-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
System information as of Wed Oct 22 19:45:03 UTC 2014
System load: 0.08      Memory usage: 1%    Processes:      55
Usage of /:  9.7% of 7.75GB   Swap usage:  0%   Users logged in: 0
Graph this data and manage this system at:
  https://landscape.canonical.com/
Get cloud support with Ubuntu Advantage Cloud Guest:
  http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

ubuntu@ip-172-31-46-246:~$
```

Word Count on Shared-Memory/Hadoop/Swift

CS553 Fall 2014

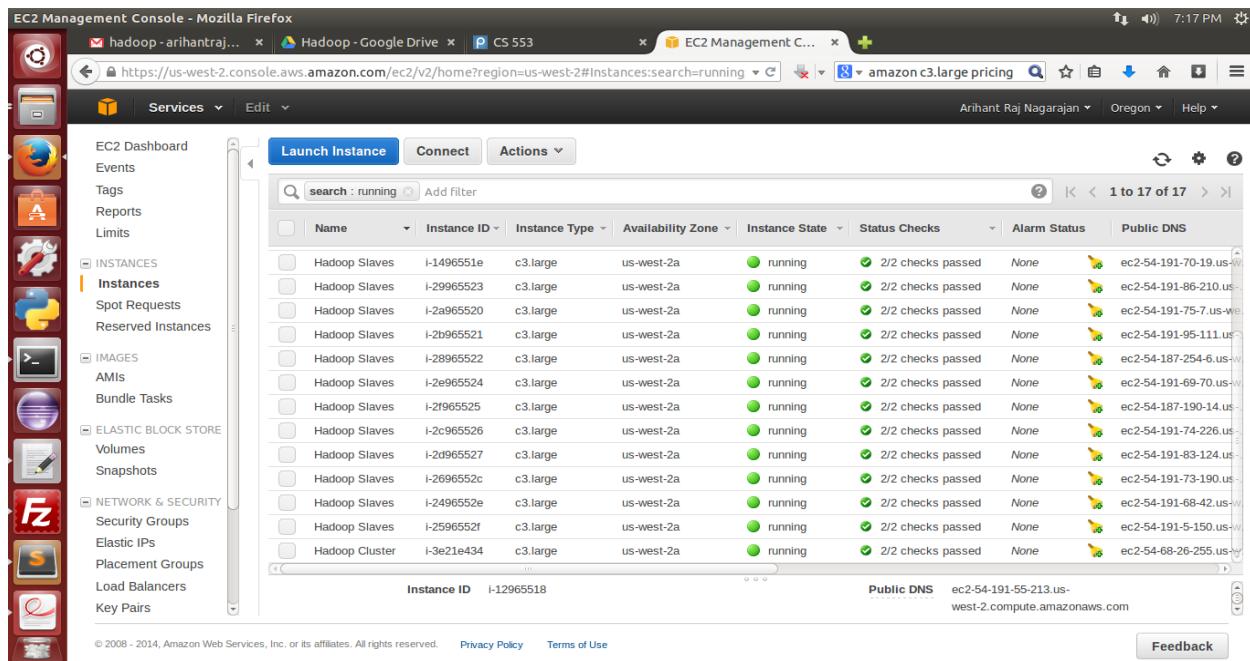
```
ubuntu@ip-172-31-18-154:/sdf/wordcount$ ./swift_bash.sh
Enter FQN file name
/sdf/wikitogb.txt
*****
Creating Input Directory
*****
Starting Swift Word Count
Swift 0.95 RC6 swift-r7900 cog-r3908
RunID: run001
Progress: Thu, 23 Oct 2014 09:23:46+0000
Progress: Thu, 23 Oct 2014 09:23:47+0000 Stage in:1
Progress: Thu, 23 Oct 2014 09:24:17+0000 Stage in:1
Progress: Thu, 23 Oct 2014 09:24:47+0000 Stage in:1
Progress: Thu, 23 Oct 2014 09:25:17+0000 Stage in:1
Progress: Thu, 23 Oct 2014 09:25:47+0000 Stage in:1
Progress: Thu, 23 Oct 2014 09:26:17+0000 Stage in:1
Progress: Thu, 23 Oct 2014 09:26:47+0000 Stage in:1
```

```
ubuntu@ip-172-31-24-234:~/wordcount$ ./wordcount
Progress: Thu, 23 Oct 2014 22:49:33+0000 Selecting site:9 Active:1 Finished successfully:23
Progress: Thu, 23 Oct 2014 22:49:44+0000 Selecting site:8 Stage in:1 Finished successfully:24
Progress: Thu, 23 Oct 2014 22:49:49+0000 Selecting site:8 Active:1 Finished successfully:24
Progress: Thu, 23 Oct 2014 22:50:19+0000 Selecting site:8 Active:1 Finished successfully:24
Progress: Thu, 23 Oct 2014 22:51:12+0000 Selecting site:7 Active:1 Finished successfully:25
Progress: Thu, 23 Oct 2014 22:51:14+0000 Selecting site:7 Active:1 Finished successfully:25
Progress: Thu, 23 Oct 2014 22:51:44+0000 Selecting site:7 Active:1 Finished successfully:25
Progress: Thu, 23 Oct 2014 22:52:12+0000 Selecting site:6 Stage in:1 Finished successfully:26
Progress: Thu, 23 Oct 2014 22:52:15+0000 Selecting site:6 Active:1 Finished successfully:26
Progress: Thu, 23 Oct 2014 22:52:45+0000 Selecting site:6 Active:1 Finished successfully:26
Progress: Thu, 23 Oct 2014 22:53:15+0000 Selecting site:6 Active:1 Finished successfully:26
Progress: Thu, 23 Oct 2014 22:53:45+0000 Selecting site:6 Active:1 Finished successfully:26
Progress: Thu, 23 Oct 2014 22:53:50+0000 Selecting site:5 Stage in:1 Finished successfully:27
Progress: Thu, 23 Oct 2014 22:53:56+0000 Selecting site:5 Active:1 Finished successfully:27
Progress: Thu, 23 Oct 2014 22:54:26+0000 Selecting site:5 Active:1 Finished successfully:27
Progress: Thu, 23 Oct 2014 22:54:56+0000 Selecting site:5 Active:1 Finished successfully:27
Progress: Thu, 23 Oct 2014 22:55:22+0000 Selecting site:3 Stage in:1 Finished successfully:29
Progress: Thu, 23 Oct 2014 22:55:28+0000 Selecting site:3 Active:1 Finished successfully:29
Progress: Thu, 23 Oct 2014 22:55:58+0000 Selecting site:3 Active:1 Finished successfully:29
Progress: Thu, 23 Oct 2014 22:56:28+0000 Selecting site:3 Active:1 Finished successfully:29
Progress: Thu, 23 Oct 2014 22:56:58+0000 Selecting site:3 Active:1 Finished successfully:29
Progress: Thu, 23 Oct 2014 22:57:19+0000 Selecting site:2 Stage in:1 Finished successfully:30
Progress: Thu, 23 Oct 2014 22:57:49+0000 Selecting site:2 Active:1 Finished successfully:30
Progress: Thu, 23 Oct 2014 22:58:19+0000 Selecting site:2 Active:1 Finished successfully:30
Progress: Thu, 23 Oct 2014 22:58:49+0000 Selecting site:2 Active:1 Finished successfully:30
Progress: Thu, 23 Oct 2014 22:59:19+0000 Selecting site:2 Active:1 Finished successfully:30
Progress: Thu, 23 Oct 2014 22:59:49+0000 Selecting site:2 Stage in:1 Finished successfully:31
Progress: Thu, 23 Oct 2014 23:00:03+0000 Selecting site:1 Active:1 Finished successfully:31
Progress: Thu, 23 Oct 2014 23:00:33+0000 Selecting site:1 Active:1 Finished successfully:31
Progress: Thu, 23 Oct 2014 23:01:03+0000 Selecting site:1 Active:1 Finished successfully:31
Progress: Thu, 23 Oct 2014 23:01:33+0000 Selecting site:1 Active:1 Finished successfully:31
Progress: Thu, 23 Oct 2014 23:01:52+0000 Stage in:1 Finished successfully:32
Progress: Thu, 23 Oct 2014 23:01:57+0000 Active:1 Finished successfully:32
Progress: Thu, 23 Oct 2014 23:02:27+0000 Active:1 Finished successfully:32
Progress: Thu, 23 Oct 2014 23:02:57+0000 Active:1 Finished successfully:32
Progress: Thu, 23 Oct 2014 23:03:17+0000 Active:1 Finished successfully:33
Final status:Thu, 23 Oct 2014 23:03:36+0000 Finished successfully:34
*****
Time taken to Hashmap and Sort: 54 min 55 sec 295 ms
*****
Thank you for using Swift Word Count
ubuntu@ip-172-31-24-234:~/wordcount$
```

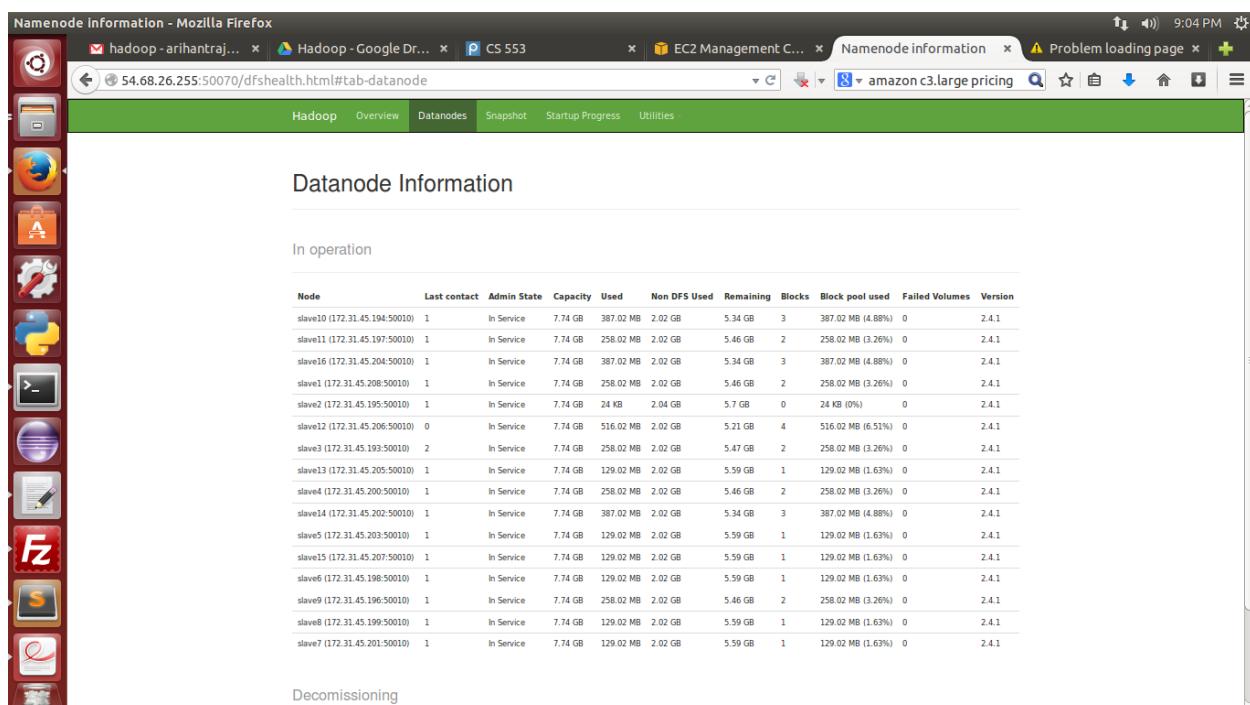
Word Count on Shared-Memory/Hadoop/Swift

CS553 Fall 2014

b. Virtual Cluster 17 Nodes (1 Master and 16 Workers) Hadoop & Swift:



The screenshot shows the AWS EC2 Management Console interface. On the left, there's a sidebar with various services like EC2 Dashboard, Events, Tags, Reports, Limits, Instances, AMIs, ELASTIC BLOCK STORE, NETWORK & SECURITY, and more. The main area displays a table of 17 running instances. The columns include Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, and Public DNS. Most instances are named 'Hadoop Slaves' and one is 'Hadoop Cluster'. All instances are in the 'running' state. The Public DNS column shows URLs starting with ec2-54-191-. For example, the first instance has a Public DNS of ec2-54-191-70-19.us-west-2.compute.amazonaws.com.



The screenshot shows the Hadoop Namenode Information page. At the top, there are tabs for Overview, Datanodes, Snapshot, Startup Progress, and Utilities. The main section is titled 'Datanode Information' and has a sub-section 'In operation'. It lists 17 datanodes with their details: Node, Last contact, Admin State, Capacity, Used, Non DFS Used, Remaining, Blocks, Block pool used, Failed Volumes, and Version. Slave10 is marked as 'In Service'. Other nodes like slave11, slave12, slave13, etc., are also listed with their respective details. Below this, there's a 'Decommissioning' section which is currently empty.

Word Count on Shared-Memory/Hadoop/Swift

CS553 Fall 2014

Namenode Information - Mozilla Firefox

54.68.26.255:50070/dfshealth.html#tab-overview

9:04 PM

Summary

Security is off.
Safemode is off.
3 files and directories, 48 blocks = 51 total filesystem object(s).
Heap Memory used 51.36 MB of 174.5 MB Heap Memory. Max Heap Memory is 889 MB.
Non Heap Memory used 29.13 MB of 30.44 MB Committed Non Heap Memory. Max Non Heap Memory is 214 MB.

Configured Capacity:	123.83 GB
DFS Used:	5.92 GB
Non DFS Used:	32.44 GB
DFS Remaining:	85.46 GB
DFS Used%:	4.78%
DFS Remaining%:	69.02%
Block Pool Used:	5.92 GB
Block Pool Used%:	4.78%
DataNodes usages% (Min/Median/Max/stdDev):	1.63% / 4.88% / 9.77% / 2.47%
Live Nodes	16 (Decommissioned: 0)
Dead Nodes	0 (Decommissioned: 0)
Decommissioning Nodes	0
Number of Under-Replicated Blocks	0
Number of Blocks Pending Deletion	0

NameNode Journal Status

Current transaction ID: 149

Journal Manager State

Nodes of the cluster - Mozilla Firefox

54.68.26.255:8088/cluster/nodes

9:09 PM

Logged in as: dr.who

hadoop

Nodes of the cluster

Cluster Metrics

Rack	Node State	Node Address	Node HTTP Address	Last health-update	Health-report	Containers	Mem Used	Mem Avail	Version
/default-rack	RUNNING	slave16:45411	slave16:8042	22-Oct-2014 02:09:06	6	6 GB	2 GB	2.4.1	
/default-rack	RUNNING	slave15:53065	slave15:8042	22-Oct-2014 02:09:07	0	0 B	8 GB	2.4.1	
/default-rack	RUNNING	slave2:35418	slave2:8042	22-Oct-2014 02:09:06	7	8 GB	0 B	2.4.1	
/default-rack	RUNNING	slave2:38668	slave2:8042	22-Oct-2014 02:09:07	0	0 B	8 GB	2.4.1	
/default-rack	RUNNING	slave2:36323	slave2:8042	22-Oct-2014 02:09:07	2	2 GB	6 GB	2.4.1	
/default-rack	RUNNING	slave10:46279	slave10:8042	22-Oct-2014 02:09:06	5	5 GB	3 GB	2.4.1	
/default-rack	RUNNING	slave12:34054	slave12:8042	22-Oct-2014 02:09:06	8	8 GB	0 B	2.4.1	
/default-rack	RUNNING	slave5:40556	slave5:8042	22-Oct-2014 02:09:06	6	6 GB	2 GB	2.4.1	
/default-rack	RUNNING	slave7:33583	slave7:8042	22-Oct-2014 02:09:06	0	0 B	8 GB	2.4.1	
/default-rack	RUNNING	slave11:37088	slave11:8042	22-Oct-2014 02:09:07	5	5 GB	3 GB	2.4.1	
/default-rack	RUNNING	slave14:56896	slave14:8042	22-Oct-2014 02:09:06	6	6 GB	2 GB	2.4.1	
/default-rack	RUNNING	slave13:56686	slave13:8042	22-Oct-2014 02:09:06	5	5 GB	3 GB	2.4.1	
/default-rack	RUNNING	slave1:55332	slave1:8042	22-Oct-2014 02:09:06	2	2 GB	6 GB	2.4.1	
/default-rack	RUNNING	slave6:49611	slave6:8042	22-Oct-2014 02:09:06	1	1 GB	7 GB	2.4.1	
/default-rack	RUNNING	slave3:47079	slave3:8042	22-Oct-2014 02:09:06	5	5 GB	3 GB	2.4.1	
/default-rack	RUNNING	slave4:53861	slave4:8042	22-Oct-2014 02:09:07	0	0 B	8 GB	2.4.1	

Showing 1 to 16 of 16 entries

About Apache Hadoop

Word Count on Shared-Memory/Hadoop/Swift

CS553 Fall 2014

NEW,NEW_SAVING,SUBMITTED,ACCEPTED,RUNNING Applications - Mozilla Firefox 10:02 PM

The screenshot shows the Hadoop Cluster Scheduler interface. The top navigation bar includes tabs for 'hadoop - arihantraj... x', 'Hadoop - Google Dr... x', 'CS 553 x', 'EC2 Management C... x', 'Namenode information x', 'NEW,NEW_SAVING,SU... x', and 'amazon c3.large pricing x'. A message 'Logged in as: dr who' is displayed.

Cluster Metrics

	Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
2	0	0	2	0	0	0 B	128 GB	0 B	16	0	0	0	0

Application Queues

Legend: Capacity (Grey), Used (Green), Used (over capacity) (Orange), Max Capacity (Grey)

- root
 - default

0.0% used
0.0% used

'default' Queue Status

Queue State: RUNNING

Used Capacity: 0.0%

Absolute Used Capacity: 0.0%

Absolute Capacity: 100.0%

Absolute Max Capacity: 100.0%

User Resources: <memory:0, vCores:0>

Num Scheduled Applications: 0

Num Non-Schedulable Applications: 0

Num Containers: 0

Max Applications: 10000

Max Applications Per User: 10000

Max Schedulable Applications: 13

Max Schedulable Applications Per User: 13

Configured Capacity: 100.0%

Configured Max Capacity: 100.0%

Configured Minimum User Limit Percent: 100%

Configured User Limit Factor: 1.0

Active users: 0

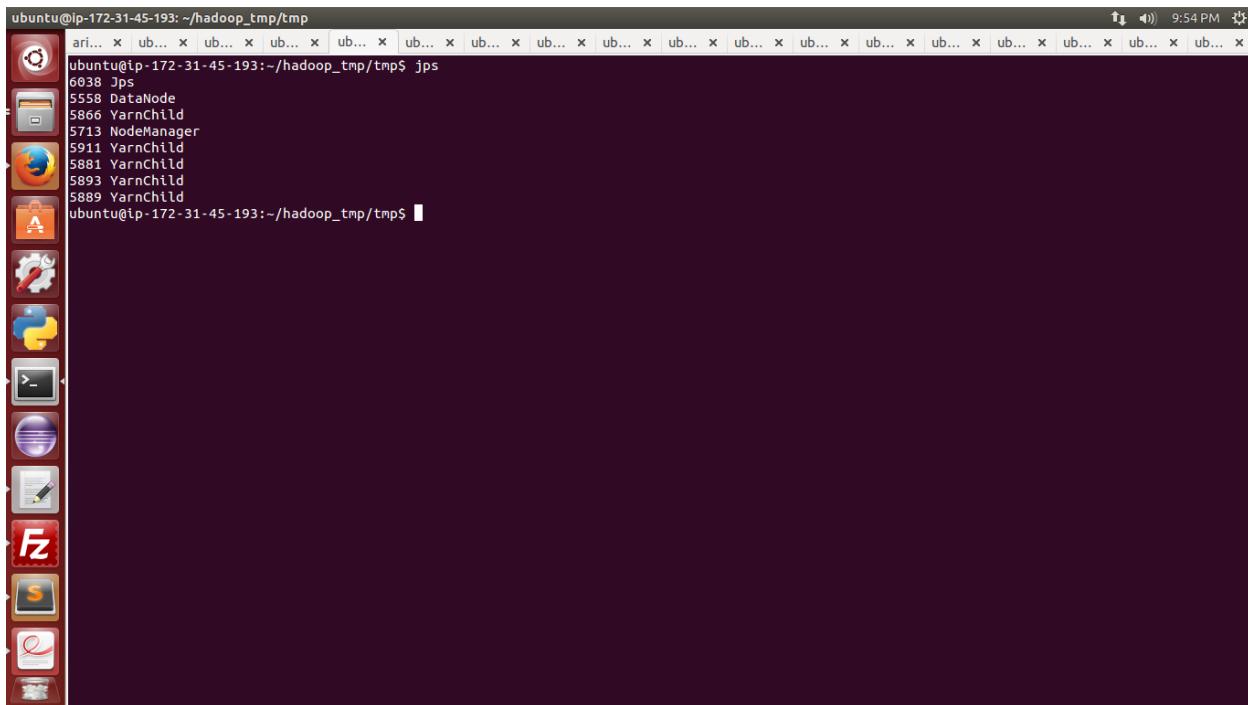
Show 20 entries Search: _____

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking UI
No data available in table										

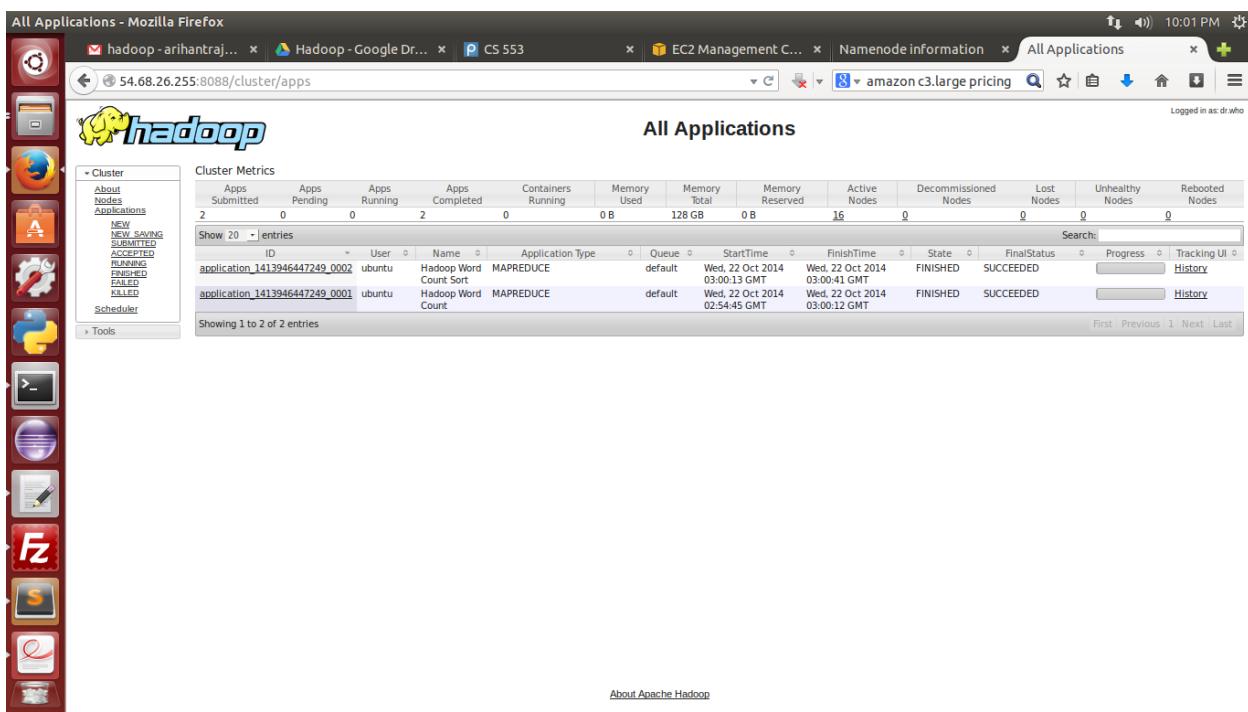
Showing 0 to 0 of 0 entries First Previous Next Last

Word Count on Shared-Memory/Hadoop/Swift

CS553 Fall 2014



ubuntu@ip-172-31-45-193: ~/hadoop_tmp/tmp\$ jps
6038 Jps
5558 DataNode
5866 YarnChild
5713 NodeManager
5911 YarnChild
5881 YarnChild
5893 YarnChild
5889 YarnChild
ubuntu@ip-172-31-45-193: ~/hadoop_tmp/tmp\$



All Applications - Mozilla Firefox

hadoop - arianhraj... | Hadoop - Google Dr... | CS 553 | EC2 Management C... | Namenode information | All Applications | 54.68.26.255:8088/cluster/apps | amazon c3.large pricing | Logged in as: dr.who

hadoop

All Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
2	0	0	2	0	0 B	128 GB	0 B	16	0	0	0	0

Show: 20 - 2 entries

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking UI
application_1413946447249_0002	ubuntu	Hadoop Word Count Sort	MAPREDUCE	default	Wed, 22 Oct 2014 03:00:13 GMT	Wed, 22 Oct 2014 03:00:41 GMT	FINISHED	SUCCEEDED		History
application_1413946447249_0001	ubuntu	Hadoop Word Count	MAPREDUCE	default	Wed, 22 Oct 2014 02:54:45 GMT	Wed, 22 Oct 2014 03:00:12 GMT	FINISHED	SUCCEEDED		History

Showing 1 to 2 of 2 entries

First Previous 1 Next Last

About Apache Hadoop

Word Count on Shared-Memory/Hadoop/Swift

CS553 Fall 2014

EC2 Management Console - Mozilla Firefox

Swift User Guide | Swift Tutorial for ... | EC2 Manage... | IAM Manage... | Setting Up with... | CS 553 (104 un... | cloud-tutorials/... | yadudoc/cloud-... | sudo apt-get install java

Oct 22 16:23:26 | Oct 22 | Rahul Krish | Oregon | Help

Services Edit

EC2 Dashboard Events Tags Reports Limits

Instances

- Spot Requests
- Reserved Instances

Images AMIs Bundle Tasks

Elastic Block Store Volumes Snapshots

Network & Security Security Groups Elastic IPs Placement Groups Load Balancers Key Pairs Network Interfaces

Auto Scaling Launch Configurations Auto Scaling Groups

Launch Instance Connect Actions

Filter by tags and attributes or search by keyword

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP	Key Name	Monitoring	Launch Time
swift-worker-...	i-02bchf0d	c3.large	us-west-2c	running	2/2 checks ...	None	ec2-54-69-248-31.us-w...	54.69.248.31	launchpad-key	disabled	October 22, 2014 4:16:45
swift-worker-...	i-0ebchf01	c3.large	us-west-2c	running	2/2 checks ...	None	ec2-54-69-209-229.us...	54.69.209.229	launchpad-key	disabled	October 22, 2014 4:16:44
swift-worker-...	i-0ebchf00	c3.large	us-west-2c	running	2/2 checks ...	None	ec2-54-187-246-98.us...	54.187.246.98	launchpad-key	disabled	October 22, 2014 4:16:43
headnode	i-1838fb12	c3.large	us-west-2b	running	2/2 checks ...	None	ec2-54-69-157-7.us-w...	54.69.157.7	launchpad-key	disabled	October 22, 2014 2:44:01
swift-worker-...	i-23fbff02	c3.large	us-west-2c	running	2/2 checks ...	None	ec2-54-187-108-169.us...	54.187.108.169	launchpad-key	disabled	October 22, 2014 4:16:46
swift-worker-...	i-3cbff033	c3.large	us-west-2c	running	2/2 checks ...	None	ec2-54-68-190-121.us...	54.68.190.121	launchpad-key	disabled	October 22, 2014 4:16:47
Launchpad	i-44826548	c3.large	us-west-2a	running	2/2 checks ...	None	ec2-54-187-58-156.us...	54.187.58.156	launchpad-key	disabled	October 22, 2014 1:50:20
swift-worker-...	i-98bdff97	c3.large	us-west-2c	running	2/2 checks ...	None	ec2-54-69-221-148.us...	54.69.221.148	launchpad-key	disabled	October 22, 2014 4:16:41
swift-worker-...	i-9ebdf992	c3.large	us-west-2c	running	2/2 checks ...	None	ec2-54-187-72-13.us-w...	54.187.72.13	launchpad-key	disabled	October 22, 2014 4:16:42
swift-worker-...	i-a0bcfeaf	c3.large	us-west-2c	running	2/2 checks ...	None	ec2-54-187-198-131.us...	54.187.198.131	launchpad-key	disabled	October 22, 2014 4:16:46
swift-worker-...	i-a2bcfead	c3.large	us-west-2c	running	2/2 checks ...	None	ec2-54-69-93-113.us-w...	54.69.32.31	launchpad-key	disabled	October 22, 2014 4:16:45
ec2-54-69-157-7.us-west-2b	i-1838fb12	c3.large	us-west-2b	running	2/2 checks ...	None	ec2-54-69-157-7.us-w...	54.69.157.7	Launchpad Key	disabled	October 22, 2014 4:16:45

Select an instance above

Feedback

EC2 Management Console - Mozilla Firefox

Swift User Guide | EC2 Manage... | Billing Manage... | CS 553 (122 un... | cloud-tutorials/... | yadudoc/cloud-... | sudo apt-get install java

Oct 23 22:34:06 | Oct 23 | Rahul Krish | Oregon | Help

Services Edit

EC2 Dashboard Events Tags Reports Limits

Instances

- Spot Requests
- Reserved Instances

Images AMIs Bundle Tasks

Elastic Block Store Volumes Snapshots

Network & Security Security Groups Elastic IPs Placement Groups Load Balancers Key Pairs Network Interfaces

Auto Scaling Launch Configurations Auto Scaling Groups

Launch Instance Connect Actions

Filter by tags and attributes or search by keyword

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP	Key Name	Monitoring	Launch Time
headnode	i-149e511e	c3.large	us-west-2b	running	2/2 checks ...	None	ec2-54-69-200-191.us...	54.69.200.191	launchpad-key	disabled	October 23, 2014 7:51:40
Launchpad	i-44826548	c3.large	us-west-2a	stopped	None	None	ec2-54-191-252-93.us...	54.191.252.93	launchpad-key	disabled	October 23, 2014 1:40:43
LaunchPad50	i-d161dd06	c3.large	us-west-2a	running	2/2 checks ...	None	ec2-54-191-252-93.us...	54.191.252.93	launchpad-key	disabled	October 23, 2014 2:55:56
swift-worker-...	i-44826548	c3.large	us-west-2c	running	2/2 checks ...	None	ec2-54-69-255-193.us...	54.69.255.193	launchpad-key	disabled	October 23, 2014 7:52:15
swift-worker-...	i-78804727	c3.large	us-west-2b	running	2/2 checks ...	None	ec2-54-186-89-43.us-w...	54.186.89.43	launchpad-key	disabled	October 23, 2014 7:52:13
swift-worker-...	i-4a834cf0	c3.large	us-west-2b	running	2/2 checks ...	None	ec2-54-187-250-147.us...	54.187.250.147	launchpad-key	disabled	October 23, 2014 7:52:14
swift-worker-...	i-209e512a	c3.large	us-west-2b	running	2/2 checks ...	None	ec2-54-191-44-78.us-w...	54.191.44.78	launchpad-key	disabled	October 23, 2014 7:52:15
swift-worker-...	i-1814e115	c3.large	us-west-2b	running	2/2 checks ...	None	ec2-54-191-56-171.us...	54.191.56.171	launchpad-key	disabled	October 23, 2014 7:52:16
swift-worker-...	i-fb834cf1	c3.large	us-west-2b	running	2/2 checks ...	None	ec2-54-68-44-50.us-w...	54.68.44.50	launchpad-key	disabled	October 23, 2014 7:52:17
swift-worker-...	i-239ef5129	c3.large	us-west-2b	running	2/2 checks ...	None	ec2-54-190-90-63.us-w...	54.190.90.63	launchpad-key	disabled	October 23, 2014 7:52:17
swift-worker-...	i-18814e12	c3.large	us-west-2b	running	2/2 checks ...	None	ec2-54-69-42-71.us-w...	54.69.42.71	launchpad-key	disabled	October 23, 2014 7:52:18

Instance: **i-d161dd06 (LaunchPad50)** Public DNS: **ec2-54-191-252-93.us-west-2.compute.amazonaws.com**

Description Status Checks Monitoring Tags

Instance ID	i-d161dd06
Instance state	running
Instance type	c3.large
Private DNS	ip-172-31-24-234.us-west-2.compute.internal
Private IPs	172.31.24.234
Secondary private IPs	
VPC ID	vpc-c7hd30a2
Subnet ID	subnet-2ca05e5b
Network interfaces	eth0
Source/dest. check	True
EBS-optimized	False
Root device type	efs

Public DNS: ec2-54-191-252-93.us-west-2.compute.amazonaws.com
 Public IP: 54.191.252.93
 Elastic IP: -
 Availability zone: us-west-2a
 Security groups: swift_security_group1, view_rules
 Scheduled events: No scheduled events
 AMI ID: Swift Launch Pad (ami-4b57197b)
 Platform: -
 IAM role: -
 Key pair name: launchpad-key
 Owner: 484859707345
 Launch time: October 23, 2014 2:55:58 PM UTC-5 (7 hours)
 Termination protection: False

Feedback

Word Count on Shared-Memory/Hadoop/Swift

CS553 Fall 2014

```
ubuntu@ip-172-31-32-64:~/wordcount
Progress: Fri, 24 Oct 2014 03:26:20+0000 Selecting site:94 Stage in:10 Submitted:3 Active:3 Stage out:2 Finished successfully:389
Progress: Fri, 24 Oct 2014 03:26:21+0000 Selecting site:86 Stage in:12 Submitting:1 Submitted:4 Active:3 Finished successfully:395
Progress: Fri, 24 Oct 2014 03:26:22+0000 Selecting site:83 Stage in:10 Submitting:1 Submitted:4 Active:4 Stage out:1 Finished successfully:398
Progress: Fri, 24 Oct 2014 03:26:23+0000 Selecting site:80 Stage in:12 Submitted:5 Active:3 Finished successfully:401
Progress: Fri, 24 Oct 2014 03:26:24+0000 Selecting site:78 Stage in:7 Submitting:5 Active:7 Stage out:1 Finished successfully:403
Progress: Fri, 24 Oct 2014 03:26:25+0000 Selecting site:72 Stage in:13 Submitting:1 Submitted:4 Active:2 Finished successfully:409
Progress: Fri, 24 Oct 2014 03:26:26+0000 Selecting site:71 Stage in:13 Submitted:5 Active:1 Stage out:1 Finished successfully:410
Progress: Fri, 24 Oct 2014 03:26:27+0000 Selecting site:68 Stage in:13 Submitting:1 Submitted:4 Active:2 Finished successfully:413
Progress: Fri, 24 Oct 2014 03:26:28+0000 Selecting site:65 Stage in:11 Submitted:4 Active:6 Finished successfully:414
Progress: Fri, 24 Oct 2014 03:26:29+0000 Selecting site:61 Stage in:9 Submitted:4 Active:7 Finished successfully:417
Progress: Fri, 24 Oct 2014 03:26:30+0000 Selecting site:57 Stage in:11 Submitting:2 Submitted:3 Active:4 Finished successfully:424
Progress: Fri, 24 Oct 2014 03:26:31+0000 Selecting site:54 Stage in:14 Submitted:2 Active:1 Finished successfully:428
Progress: Fri, 24 Oct 2014 03:26:32+0000 Selecting site:51 Stage in:8 Submitting:1 Submitted:4 Active:2 Finished successfully:413
Progress: Fri, 24 Oct 2014 03:26:33+0000 Selecting site:47 Stage in:10 Submitting:2 Active:3 Active:4 Stage out:1 Finished successfully:434
Progress: Fri, 24 Oct 2014 03:26:34+0000 Selecting site:44 Stage in:10 Submitted:5 Active:4 Stage out:1 Finished successfully:437
Progress: Fri, 24 Oct 2014 03:26:35+0000 Selecting site:40 Stage in:12 Submitting:1 Submitted:4 Active:2 Stage out:1 Finished successfully:441
Progress: Fri, 24 Oct 2014 03:26:36+0000 Selecting site:37 Stage in:15 Submitting:1 Submitted:4 Active:5 Finished successfully:444
Progress: Fri, 24 Oct 2014 03:26:37+0000 Selecting site:37 Stage in:9 Submitted:5 Active:5 Stage out:1 Finished successfully:444
Progress: Fri, 24 Oct 2014 03:26:38+0000 Selecting site:33 Stage in:8 Submitting:2 Submitted:2 Active:7 Finished successfully:449
Progress: Fri, 24 Oct 2014 03:26:39+0000 Selecting site:31 Stage in:9 Submitted:1 Submitted:4 Active:7 Stage out:2 Finished successfully:450
Progress: Fri, 24 Oct 2014 03:26:40+0000 Selecting site:27 Stage in:10 Submitted:4 Active:5 Finished successfully:455
Progress: Fri, 24 Oct 2014 03:26:41+0000 Selecting site:24 Stage in:11 Submitted:5 Active:3 Stage out:1 Finished successfully:457
Progress: Fri, 24 Oct 2014 03:26:42+0000 Selecting site:21 Stage in:10 Submitted:5 Active:5 Finished successfully:460
Progress: Fri, 24 Oct 2014 03:26:43+0000 Selecting site:20 Stage in:8 Submitted:5 Active:6 Stage out:1 Finished successfully:461
Progress: Fri, 24 Oct 2014 03:26:44+0000 Selecting site:15 Stage in:11 Submitted:5 Active:4 Finished successfully:466
Progress: Fri, 24 Oct 2014 03:26:45+0000 Selecting site:12 Stage in:13 Submitted:5 Active:1 Stage out:1 Finished successfully:469
Progress: Fri, 24 Oct 2014 03:26:46+0000 Selecting site:9 Stage in:13 Submitted:5 Active:2 Finished successfully:472
Progress: Fri, 24 Oct 2014 03:26:47+0000 Selecting site:7 Stage in:13 Submitted:5 Active:2 Finished successfully:474
Progress: Fri, 24 Oct 2014 03:26:48+0000 Selecting site:6 Stage in:11 Submitted:5 Active:4 Stage out:2 Finished successfully:475
Progress: Fri, 24 Oct 2014 03:26:49+0000 Selecting site:6 Stage in:11 Submitted:5 Active:4 Finished successfully:476
Progress: Fri, 24 Oct 2014 03:26:50+0000 Selecting site:5 Stage in:9 Submitted:5 Active:6 Finished successfully:476
Progress: Fri, 24 Oct 2014 03:26:51+0000 Selecting site:2 Stage in:5 Submitted:5 Active:9 Stage out:1 Finished successfully:479
Progress: Fri, 24 Oct 2014 03:26:52+0000 Stage in:6 Submitted:1 Active:8 Stage out:1 Finished successfully:485
Progress: Fri, 24 Oct 2014 03:26:53+0000 Stage in:5 Active:5 Finished successfully:491
Progress: Fri, 24 Oct 2014 03:26:54+0000 Active:5 Finished successfully:496
Progress: Fri, 24 Oct 2014 03:26:55+0000 Active:1 Finished successfully:501
Progress: Fri, 24 Oct 2014 03:27:26+0000 Stage in:1 Finished successfully:501
Progress: Fri, 24 Oct 2014 03:27:32+0000 Active:1 Finished successfully:501
Final status:Fri, 24 Oct 2014 03:27:43+0000 Finished successfully:502
*****
Time taken to Hashmap and Sort: 3 min 45 sec 225 ms
*****
Thank you for using Swift Word Count
ubuntu@ip-172-31-32-64:~/wordcount$
```

S3 Management Console - Mozilla Firefox

Services Edit

Upload Create Folder Actions

All Buckets / swift-bucket-rahal

Name	Storage Class	Size	Last Modified
Cache	--	--	--

Transfers

Automatically clear finished transfers

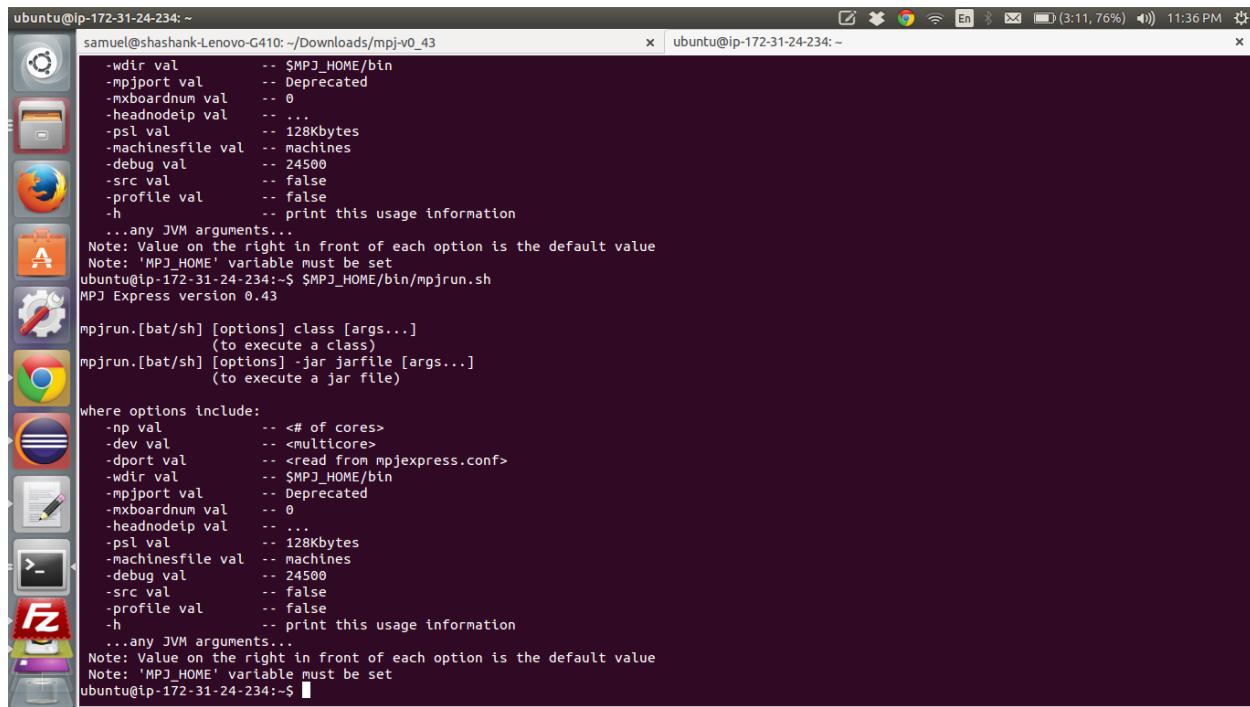
Done

Use Standard Storage and disable Server Side Encryption: Cache in swift-bucket-rahal

Feedback

Word Count on Shared-Memory/Hadoop/Swift

CS553 Fall 2014



ubuntu@ip-172-31-24-234: ~

```
samuel@shashank-Lenovo-G410: ~/Downloads/mpj-v0_43
samuel@shashank-Lenovo-G410: ~/Downloads/mpj-v0_43
-mdir val      -- $MPJ_HOME/bin
-mpjport val   -- Deprecated
-mxboardnum val -- 0
-headnodeip val -- ...
-psl val       -- 128Kbytes
-machinesfile val -- machines
-debug val     -- 24500
-src val       -- false
-profile val   -- false
-h             -- print this usage information
...any JVM arguments...
Note: Value on the right in front of each option is the default value
Note: 'MPJ_HOME' variable must be set
ubuntu@ip-172-31-24-234:~$ $MPJ_HOME/bin/mpjrun.sh
MPJ Express version 0.43

mpjrun.[bat/sh] [options] class [args...]
               (to execute a class)
mpjrun.[bat/sh] [options] -jar jarfile [args...]
               (to execute a jar file)

where options include:
-np val        -- <# of cores>
-dev val       -- <multicore>
-dport val     -- <read from mpjexpress.conf>
-wdir val      -- $MPJ_HOME/bin
-mpjport val   -- Deprecated
-mxboardnum val -- 0
-headnodeip val -- ...
-psl val       -- 128Kbytes
-machinesfile val -- machines
-debug val     -- 24500
-src val       -- false
-profile val   -- false
-h             -- print this usage information
...any JVM arguments...
Note: Value on the right in front of each option is the default value
Note: 'MPJ_HOME' variable must be set
ubuntu@ip-172-31-24-234:~$
```

Word Count on Shared-Memory/Hadoop/Swift

CS553 Fall 2014

Credits

1. Arihant Raj Nagarajan(A20334121)
2. Rahul Krishnamurthy(A20330185)
3. Shashank Sharma(A20330372)

12. References:

- a. <http://docs.oracle.com/javase/7/docs/api/>
- b. <http://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/SingleCluster.html>
- c. <https://aws.amazon.com/documentation/ec2/>
- d. <http://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/ClusterSetup.html>
- e. <http://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/ClusterSetup.html>
- f. https://github.com/hortonworks/hadoop-tutorials/blob/master/Community/T09_Write_And_Run_Your_Own_MapReduce_Java_Program_Poll_Result_Analysis.md
- g. <http://disi.unitn.it/~lissandrini/notes/installing-hadoop-on-ubuntu-14.html>
- h. <https://docs.google.com/document/d/1v-J19xwJnPw9F8OCgLn04dqKwYkGIOxyqRAIGpmHk0/edit?pli=1>
- i. <http://swift-lang.org/docs/index.php>
- j. <https://github.com/yadudoc/cloud-tutorials.git>
- k. <https://piazza.com/class/hza36qliup33pg?cid=232>
- l. <http://swift-lang.org/guides/trunk/userguide/userguide.html>
- m. <https://piazza.com/class/hza36qliup33pg?cid=181>