THE UNIVERSITY OF SYDNEY

School of Information Technologies
A/Prof Uwe Roehm

COMP5349: Cloud Computing **1.Sem./2017**

Week 3: Introduction to Azure HDInsight and HDFS

23.03.2017

Before starting with the tutorial content, please take 5 minutes to complete the Programming Experience Survey under the "Tutorials" menu in eLearning. This will help the tutor gauge the levels of programming proficiency within the class and adjust their teaching appropriately.

Most of the lab exercies and the assignments require access to a Hadoop cluster. There are two cluster options: a **shared** cluster running locally on a few virtual machines or a **dedicated** cluster running on Microsoft Azure.

The **shared** cluster is installed and maintained by the School of IT. All students in the cloud computing course have access to it. This cluster is always on but may become slow when there are many users competing for the limited resources.

Alternatively, you may provision your own Hadoop cluster on Microsoft Azure using the credit awarded by Azure. This will be your own **dedicated** cluster and will be managed by you. It provides a good chance for you to learn cluster configuration and management skills. However, the running hours of the dedicated cluster is limited by the credit you have. You won't be able to keep a reasonably sized clustering running all the time. You will need to shut it down when you're not using it. Additionally, provisioning a cluster takes a very long time. We have timed the provision time of a cluster with 4 nodes to be around 25 minutes.

We encourage students to experience both clusters. This week's lab will introduce you to both clusters, and let you play with one of the fundamental components of any Hadoop cluster: the HDFS. Because an Azure cluster takes a long time to provision, we suggest that you start the lab by provisioning the Azure Hadoop cluster and practice HDFS on the local shared cluster while waiting for it to be completed.

Azure HDInsight Cluster: Start Initialising

This section regards using Azure to provision a dedicated Hadoop cluster for you to use.

Azure provides a few Hadoop cluster options. We will use HDInsight, which is released by Microsoft with good support and documentation. HDInsight is built on Hortonworks' HDP platform. An HDInsight Hadoop cluster requires at least three virtual machines: two head nodes and at least a worker node. The nodes will each be initialised with a range of Apache Hadoop related products, many of which are not covered in this course.

The provisioning of a cluster involves many activities such as creating virtual machines, configuring the various Azure and Hadoop related services, and so on. This is the reason it takes very long for a cluster to be provisioned and ready to use.

An HDInsight cluster stores data on separate Azure storage account, not on the file system attached to the virtual machine. This means you can shut down a cluster, kill all VMs with the data kept safe somewhere else. This feature is especially useful if you need to work on the same data set in multiple time periods. You may create a storage account during cluster provisioning. But it is good practice to have one created beforehand.

Azure storage provides a few storage services. Blob storage is used to store file data. HDInsight stores HDFS data on blob storage. Blob storage organizes data in a two level hierarchy: container and blob. All HDFS data should be stored in a container.

This document gives a good overview of Azure storage:
`https://azure.microsoft.com/en-us/documentation/articles/storage-introduction/`

You should now create an Azure Storage account. Browse the portal and try to find the link to create a new storage account. The default settings are fine for this class, as the cost of a storage account shouldn't amount to more than a dollar or two per month with the data sizes used here. As such, simply provide a name and location (which you'll also have to use for the HDInsight cluster) for the account and create it.

After you have created the storage account, find the link to create a new HDInsight cluster (hint: `Intelligence + analytics`) and proceed with creation, bearing in mind the following requirements and recommendations:

- In step 1 select Hadoop as the Cluster type.
- In step 2, select the storage account you just created and don't worry about any of the other settings.
- In step 3, modify the cluster size. Specify 2 worker nodes, and get both the worker and head nodes to use the A3 node size.
- Observe the displayed cost per hour of keeping your cluster up. Consider how much of your Azure credit will be used if you forgot that the cluster was active and left it on overnight.
- Submit!

There's a lot more information regarding cluster creation here, if you're interested:
`https://docs.microsoft.com/en-us/azure/hdinsight/hdinsight-hadoop-provision-linux-clusters`

While your cluster starts to provision, you can switch to work on the local cluster.

Local Hadoop Cluster

The local hadoop cluster currently consists of 30 Linux virtual machines. All machines have 4 core and 8G memory. Each has 250G storage. One machine is configured as the

master or "name" node while remaining are all slave or "data" nodes. These are the data nodes:

- soit-hdp-pro-2.ucc.usyd.edu.au
- soit-hdp-pro-3.ucc.usyd.edu.au
- soit-hdp-pro-4.ucc.usyd.edu.au
- ...
- soit-hdp-pro-30.ucc.usyd.edu.au

The name node is `soit-hdp-pro-1.ucc.usyd.edu.au`. You can access the name node Web UI from `http://soit-hdp-pro-1.ucc.usyd.edu.au:50070`.

You will see basic information such as living (and/or dead) data nodes and file system information there.

You can log in to any of the slave nodes with your `unikey` and `password`.

You need to set up the following environment variables to use Java, Hadoop and HDFS on the cluster:

```
# Environment variables required for Java, Hadoop and HDFS usage.
export JAVA_HOME=/usr/local/jdk1.8.0_40
export HADOOP_CONF_DIR=/usr/local/hadoop/etc/hadoop
export HADOOP_HOME=/usr/local/hadoop
export PATH=/usr/local/hadoop/bin:/usr/local/hbase/bin:$PATH
```

Instead of having to copy and execute these commands each time you login to the cluster, you can add the commands to your `~/.bashrc` file so that they're automatically executed on login. To do that, execute `nano ~/.bashrc` to edit the RC file, copy/paste the above commands to the bottom of the file, and then save and exit.

HDFS practice

The following exercises assume that you are using the local cluster and have logged in to one of the data nodes, with proper environment variables in place. You can do similar exercise in HDInsight with minor adjustments.

You can practice using HDFS shell commands to put this file on your cluster.

Question 1: Interacting with HDFS using shell commands

Basic HDFS shell command has the following format:

```
hdfs dfs -cmd [args]
```

HDFS shell commands can be issued from any node of the Hadoop cluster. Most of the shell commands are quite similar to Linux file system commands. Note that there is no `cd` command; you'll need to use the full path name in any command.

a) To list files in HDFS, use the `ls` command with a parameter indicating the directory name. For instance, the following command shows you what is under the root directory:

```
hdfs dfs -ls /
```

There are several folders under the root directory. A specific user's data is normally put under `/user/login`. This is the home directory of that user. Your home directory has been created for you on the university cluster, but not in HDInsight. The following command shows the current content of `/user`.

```
hdfs dfs -ls /user
```

b) To download a file from HDFS to your local file system, use the `get` command. The following command will download the file `place.txt` in `/share` folder on hdfs to the current directory in your local file system:

```
hdfs dfs -get /share/place.txt place.txt
```

c) To upload a file from your local file system to HDFS, use the put command. The following command will upload the `place.txt` to HDFS under the current user's home directory:

```
hdfs dfs -put place.txt place.txt
```

You can view the content of your home directory using the command

```
hdfs dfs -ls /user/<login>
```

d) To delete a file in HDFS, use the `rm` command. You can use `rm -r` to delete a directory. The following command will delete the file you just uploaded:

```
hdfs dfs -rm place.txt
```

For other shell commends that you can use, check `http://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/FileSystemShell.html`

Question 2: HDFS Storage Block and Replications

a) Open a web browser and point it to the HDFS WebUI. The url is `http://soit-hdp-pro-1.ucc.usyd.edu.au:50070`. From the HDFS home page, click the menu `Utilities` then the menu item `Browse the filesystem` and navigate to the folder `/share/photo`, click file `n07.txt` to inspect its block distribution. The default HDFS block size is 128MB. `n07.txt` has a size of 397.62MB and is stored in 4 blocks. Each block has three replicas. Try to find out the location of the replica for each block and answer the following questions:

- Are replicas distributed across all cluster nodes?
- Are all blocks have the same size?
- Is there any node which has all the blocks of this file?
- Which node has the meta data information such as the number of blocks and their replica locations?

b) When you use PUT command to upload a file to HDFS, a `HDFS-WRITE` request is sent to the cluster to write the new file. Now try to put back the `place.txt` file and describe the overall write flow, especially how does the client pushes the data to all nodes. You can check the namenode log to see what operations are taken and which nodes are involved in completing the `HDFS-WRITE` request. The logs is accessible from menu `Utilities` then menu items `logs`, however opening an 80Mb log file in the browser is not a good idea - instead right click the link and download the file, and open it in a program like Notepad++ (on Windows) to search through it.

Azure HDInsight Cluster: Play Around

After the cluster finishes provisioning, the status of your HDInsight cluster will be changed to "`running`". Click the cluster dashboard from the `Quick Links`, and from the menu of dashboards it then displays, click to view the HDInsight cluster dashboad. This will bring you to the management and monitoring site. You need to type in the cluster user name (`admin` if you didn't change it) and password you entered in step 6. Once authenticated, you will see the Ambari Web UI for cluster management and monitoring. If you do not see a list of all component installed on the left side bar, click the `Ambari` icon in the top left corner to return to the main dashboard. Click `HDFS` from the left menu to go to the HDFS management screen. It shows you what HDFS related services have been started and various performance metrics.

Your HDInsight also has a HDFS Web UI which allows you to see the data nodes and file system information. You can find the URL from the `Quick Links` drop down near the top of the HDFS page within Ambari. The URL points to domains which are not available to the public (internet), but only within the Azure cluster. This means you'll need to set up SSH Tunneling to access the HDFS Web UI, and any other internal web UI that Ambari may be linking to. If you are interested, you may follow this guide to set up SSH tunneling:

`https://azure.microsoft.com/en-us/documentation/articles/hdinsight-linux-ambari-ssh-tunnel/`

Question 3: Try HDFS on HDInsight.

You can run HDFS shell commands against your HDInsight cluster by logging in to one of the cluster's nodes. Azure only gives you SSH access to a particular node. To find out which node you can SSH to, click the `Secure Shell` button from the cluster's overview page in the Azure portal. This icon is in the row of icons directly above the `Essentials` blade. Clicking `Secure Shell` will bring up a blade showing the host name and a brief guide on how to connect to the node. Use the appropriate SSH account (`sshuser` if you didn't change it) to login to that host.

Once you SSH into the host, you can run HDFS commands such as `hdfs dfs -ls /users` to inspect the HDFS content. You may find that your own user `sshuser` does

not have a directory under `/user`. To create a home directory for `sshuser` user, run the following commands:

```
hdfs dfs -mkdir /user/sshuser
hdfs dfs -chown sshuser /user
```

A copy of `place.txt` has been put on a publicly visible Azure storage blob. You can download this using the wget command:
`wget https://2017sem1comp5349.blob.core.windows.net/photo-data/place.txt`

Once your home directory is ready, try repeating the questions from Question 1 above. You'll need to slightly adjust the commands to work with the different login and file locations.

Always remember to delete your cluster when not in use! You are responsible for your Azure credit; if it runs out you will have to wait until the next month to get more.