The Maui High Performance Computing Center Department of Defense Supercomputing Resource Center (MHPCC DSRC)

Hadoop Implementation on Riptide

Table of Contents

Executive Summary

The Maui High Performance Computing Center is a Department of Defense (DoD) Supercomputing Resource Centers (DSRC) and is one of the five DSRCs in the DoD's High Performance Computing Modernization Program (HPCMP). It is a US Air Force Research Laboratory (AFRL) Center and part of the US Air Force Research Laboratory's Directed Energy Directorate at Kirtland Air Force Base, New Mexico.

The Center has an infrastructure that is connected to the Defense Research and Engineering Network (DREN) and other proprietary networks. It has a large-scale parallel computing environment with terabytes of disk and online tape storage and high-speed communications.

One of the HPC systems at MHPCC is called Riptide. Riptide is an IBM iDataPlex. The login and compute nodes are populated with two Intel Sandy Bridge 8-core processors. Riptide uses the FDR 10 Infiniband interconnect and uses IBM's General Parallel File System (GPFS). Riptide has 756 compute nodes with Direct Water Cooling; memory is not shared across the nodes. Each diskless compute node has two 8-core processors (16 cores) with its own Red Hat Enterprise Linux OS, sharing 32 GBytes of memory. Riptide is rated at 251.6 peak TFLOPS.

The use of Hadoop on a large-scale computing platform came about as part of the research and development trust that the Center provides. What is Hadoop? Hadoop is an open-source software for distributed computing on large clusters. What separates this software from other distributed computing software is that it allows for the distributed processing of large data sets across clusters of commodity computers using simple programming models. It can scale up from single server to thousands of machines offering local computation and storage.

Since Hadoop is designed to work on local storage, it has been the objective of this paper to prove that even with an infrastructure like Riptide with diskless compute nodes, Hadoop can still be installed and perused as it was originally designed for. This is where the myHadoop software comes in. myHadoop is a set of scripts that allows a Hadoop installation to work with a high-performance computing environment like Riptide. The MyHadoop is a study conducted at San Diego Supercomputer Center at the University of California at San Diego by Dr. Sriram Krishan. Further details on myHadoop can be found at http://myhadoop.sourceforge.net/.

Getting Started

Hadoop

Introduction

Since Hadoop is a framework that runs on large clusters, it needs a mechanism to distribute data across the compute nodes in the cluster and to be able to store files within its framework. This is resolved by its two main components: Map/Reduce and Hadoop Distributed File System (HDFS). Map/Reduce is Hadoop's implementation of dividing the application into multiple small pieces of work that can then be executed on any of the cluster's compute nodes. HDFS, on the other hand, is where data is stored. Hadoop stores data in files and does not index them. This is why Hadoop cannot be really a direct substitute for a database. Finding something in Hadoop usually means running a Map/Reduce job. The upside to this argument is that Hadoop excels in areas where data/information retrieval/search becomes too overwhelming for a regular database

query. Furthermore, Hadoop, with its Map/Reduce component, can process very large unstructured and non-related data sets.

Installation

A valid Hadoop installation is the first prerequisite to getting the myHadoop setup on Riptide work. Hadoop version 0.20.2 is the recommended version for myHadoop to work. However, a later stable version can be used as well provided that the environment variables, corresponding file system paths, and use of necessary jars within the scripts are pointing to the latest installed version.

Hadoop installers can be found at http://hadoop.apache.org/releases.html#Download. At the time of this writing, the current stable release is the 1.2 version. Extract the downloaded compressed file into the folder of your choice. A typical Hadoop installation usually has the following files/folders:

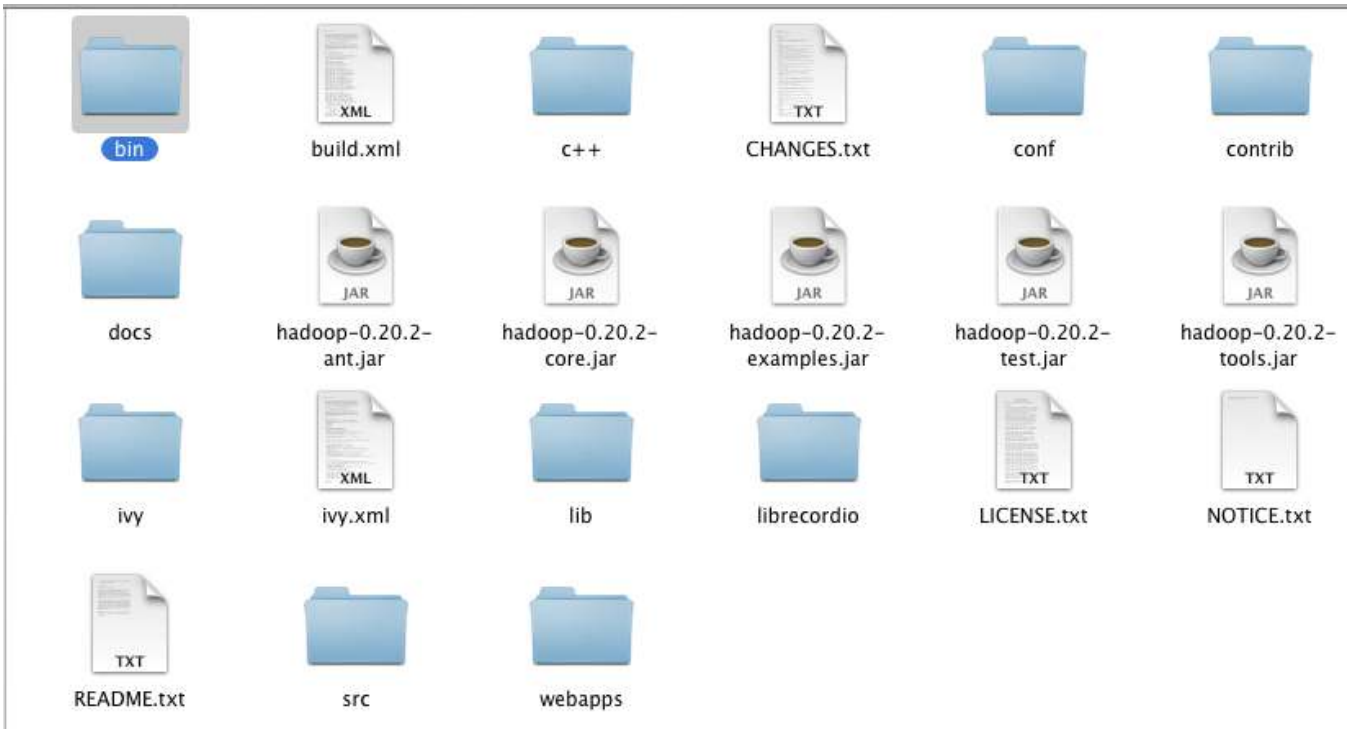

Figure 1 Typical Hadoop Installation

On Riptide, this folder is located at /gpfs/pkgs/mhpcc/hadoop-<version>. In the scripts, this will be referred to as HADOOP_HOME.

myHadoop

Introduction

As has been aforementioned, myHadoop is a set of scripts that enable a user to submit Hadoop jobs on an HPC system like Riptide via PBS. In typical HPC clusters, jobs are usually submitted in batch using some type of resource management system such as PBS. While Hadoop provide its own scheduling and manages it own job

submissions, it may seem redundant for a Hadoop implementation to run on an HPC environment and sometimes considered a challenge. It is for this purpose that myHadoop is born. Performance-wise, Hadoop is known to scale on clusters of up to 4000 nodes. Sort performance is good on 900 nodes. Sorting 9TB of data on 900 nodes takes around 1.8 hours according to wiki.apache.org.

Installation

The installer can be downloaded from http://myhadoop.sourceforge.net. Unzip the downloaded compressed file into the folder of your choice. Once the files have been extracted, check out the preliminary details on how to get started modifying pertinent scripts within myHadoop under the "docs" folder. A typical myHadoop installation would have the following file structure:

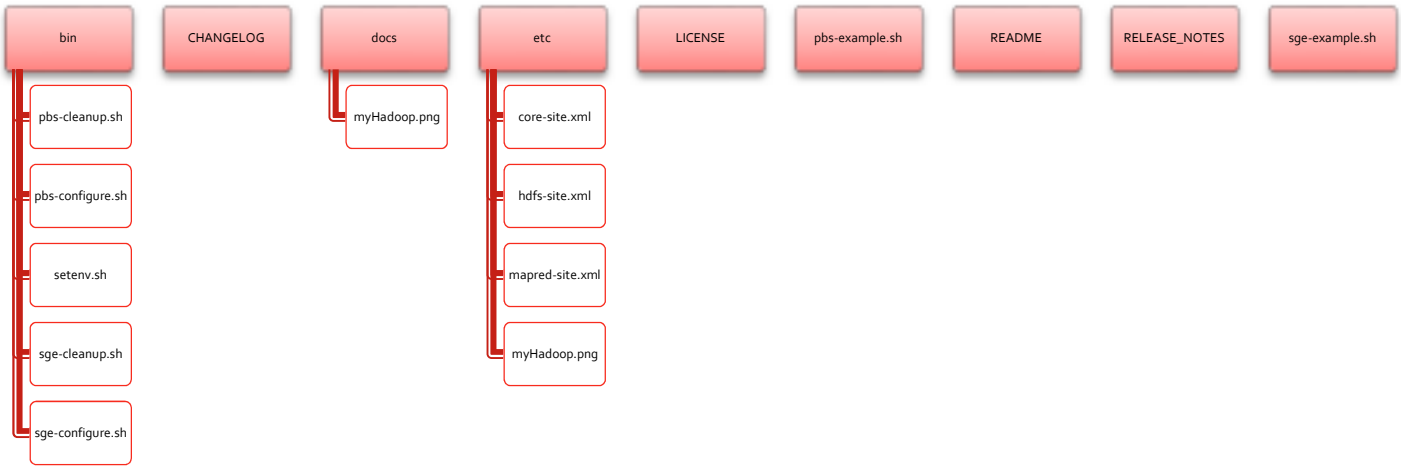

Figure 2 Typical myHadoop installation

Job submissions in myHadoop will not be successful without a valid Hadoop installation. Hadoop 0.20.2 is the recommended version but any later, stable version will also be applicable provided that appropriate scripts within myHadoop are updated accordingly. Once a valid Hadoop installation is completed, ensure that the following scripts within the myHadoop folder are modified to fit the target system/cluster.

On Riptide, this folder is installed on /gpfs/pkgs/mhpcc/myHadoop-<version>. This will be referred to as MY_HADOOP_HOME in the scripts.

Configuration

1. setenv.sh
   Please refer to Figure 2 Typical myHadoop installation. This file is located under the bin folder and is called from *pbs-hadoop.sh*. This contains the relevant environment variables for setting/running Hadoop. This is file is under /gpfs/pkgs/mhpcc/hadoop/myHadoop-<version>/bin folder and should be updated with the correct values,

otherwise, Hadoop will not run or error out. The following are the current settings that may/may not need tweaking:

a. export MY_HADOOP_HOME="/gpfs/pkgs/mhpcc/hadoop/myHadoop-0.2a"
b. export HADOOP_HOME="/gpfs/pkgs/mhpcc/hadoop/hadoop-0.20.2"
c. export HADOOP_DATA_DIR=${WORKDIR}/hadoop/data
d. export HADOOP_LOG_DIR=${WORKDIR}/hadoop/log

2. pbs-hadoop.sh

This script can probably be considered the main script in executing Hadoop jobs via PBS. This is an entirely new file that was created based on the *pbs-example.sh* provided by myHadoop. This file is under /gpfs/pkgs/mhpcc/hadoop/myHadoop-<version> folder. This is the main script that is submitted via PBS. Please make sure to edit this file for Hadoop testing. This script contains sample Hadoop jobs with text files as sample source data. Results of the Hadoop word count are being copied over to $WORKDIR/results.$PBS_JOBID. Therefore, if the Hadoop job is successful, a file "part-r-00000" will be written under this folder. This file should contain the word count results. Just be aware that anything that is placed under $WORKDIR will be purged at a later point. Therefore, make sure copies of the results are copied over to a more persistent location to avoid losing any data.

Furthermore, this script contains the pertinent PBS directives that will configure the PBS job when this is invoked. In order to run this via PBS, simply issue the following command: ***qsub pbs_hadoop.sh.*** This assumes that the script invocation is started from the MY_HADOOP_HOME root folder. Otherwise, please make the necessary path correction in the command line.

SCRIPT CONTENTS/DETAILS

a. PBS Directives

The following are the PBS directives that are needed to configure a PBS job:

```
#PBS -q standard
#PBS -N hadoop_job
#PBS -l walltime=00:03:00
#PBS -l select=2:ncpus=1
#PBS -l place=scatter
#PBS -o <path>/hadoop_run.out
#PBS -j oe
#PBS -A <account>
#PBS -V
```

Please refer to the PBS documentation at http://www.pbsworks.com/SupportDocuments.aspx for a detailed explanation of each directive but here are the basics:

-q is the queue where the PBS job is supposed to be
-N is the name of the PBS job
-l walltime is number of hours/minutes/seconds for the entire job to complete its run
-l select is the number nodes while ncpus is the number of cpus per node
-o the path and name of the log file
-j oe simply tells PBS to combine the output and error logs into; in this instance, combine them into the output log
-A the account where the jobs is supposed to run under
-V to export all PBS variables mentioned in the above.

b. Export HADOOP_CONF_DIR variable

This is the base path for all the necessary HADOOP variables. The files under this directory are usually the same files found on $*HADOOP_HOME/conf* folder with a slight difference. The files: *mapred-site.xml*, *core-site.xml*, *hdfs-site.xml*, and *hadoop-env.sh* are files taken from the $MY_HADOOP_HOME under *etc*. Modify these files under *$MY_HADOOP_HOME/etc* only when really necessary otherwise, the settings should suffice for a simple Hadoop job.

c. Set the environment variables

At this point, the script calls *setenv.sh*, which, in turn, sets the pertinent environment variables listed in 1 above1 above

d. Set up the cluster configuration

At this stage, the script is ready to call *pbs-configure.sh*. This script expects the following parameters: total number of nodes needed and the $HADOOP_CONF_DIR.

e. Format HDFS

Once all the necessary configurations are set up, it is now time format the HDFS by issuing the following:

```
$HADOOP_HOME/bin/hadoop --config $HADOOP_CONF_DIR namenode -format
```

Since a persistent Hadoop cluster is not the objective of this study, it is deemed necessary that whenever a PBS-Hadoop job is submitted, the HDFS needs to be initialized/formatted.

f. Run Hadoop jobs

Running Hadoop would mean starting all the necessary daemons. This is accomplished by executing the following:

```
$HADOOP_HOME/bin/start-all.sh
```

Once all the necessary daemons are started, it is time to run some Hadoop like the following:

```
echo "Run some test Hadoop jobs"
$HADOOP_HOME/bin/hadoop version
echo "creating runData directory..."
$HADOOP_HOME/bin/hadoop dfs -mkdir runData
echo "Copying file from local to HDFS..."
$HADOOP_HOME/bin/hadoop dfs -copyFromLocal $HOME/davinci.txt runData
echo "Checking file copied (HDFS)..."
$HADOOP_HOME/bin/hadoop dfs -ls runData/davinci.txt
echo "Running wordcount on the file inside HDFS..."
$HADOOP_HOME/bin/hadoop jar $HADOOP_HOME/hadoop-examples-1.2.1.jar wordcount runData/davinci.txt
runData/Outputs
echo "Checking the wordcount output under Outputs..."
$HADOOP_HOME/bin/hadoop dfs -ls runData/Outputs
echo "Copying results from HDFS to local FS..."
$HADOOP_HOME/bin/hadoop dfs -copyToLocal runData/Outputs $WORKDIR/results.$PBS_JOBID
echo "Sample Hadoop job finished!"
```

g. Clean up

This part just basically removes the HADOOP_LOG_DIR and HADOOP_DATA_DIR used in the run. If results of the Hadoop job run are necessary for later analysis, please make sure to copy the files out of the HDFS prior to finishing the Hadoop run.

3. pbs-configure.sh

This file resides under myHadoop-<version>/bin. This script, as its name implies, sets up the necessary environment variables and file configuration for Hadoop to work.

4. hadoop-env.sh

This is the only file that needs editing in the Hadoop installation folder. This file resides under <hadoop installation>/conf folder. The only line that is required here is the JAVA_HOME. Set it to the correct JAVA_HOME and it's good to go.

Testing

Once all the necessary scripts have been edited and updated with the correct values. The framework is now ready for testing. Run some Hadoop jobs and make sure to Check the log location under HADOOP_LOG_DIR and the "hadoop_run.out" file under $WORKDIR.