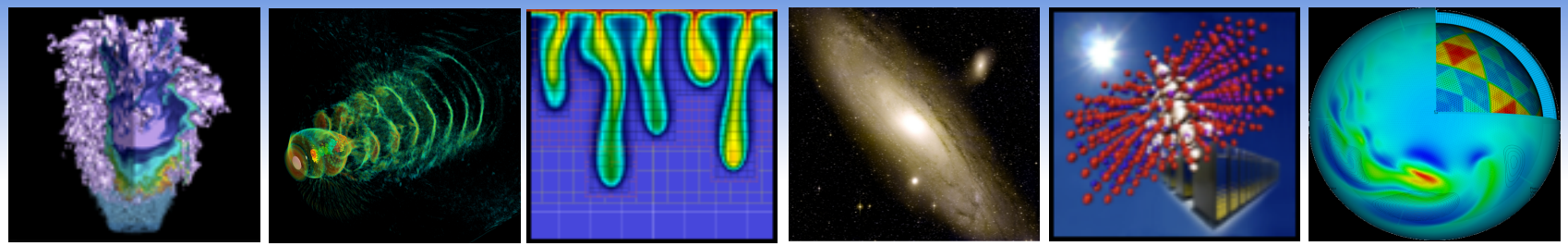

Hadoop Hands-On Exercises

Lawrence Berkeley National Lab

Oct 2011

U.S. DEPARTMENT OF ENERGY | Office of Science

NERSC

We will …

- Training accounts/User Agreement forms
- Test access to carver
- HDFS commands
- Monitoring
- Run the word count example
- Simple streaming with Unix commands
- Streaming with simple scripts
- Streaming “Census” example
- Pig Examples
- Additional Exercises

Instructions

http://tinyurl.com/nerschadoopoct

Login and Environment

ssh [username]@carver.nersc.gov

echo $SHELL

- **should be bash**

Remote Participants

Visit: http://maghdp01.nersc.gov:50030/

http://magellan.nersc.gov

(Go to Using Magellan -> Creating a SOCKS proxy)

Environment Setup

$ ssh [username]@carver.nersc.gov
$ echo $SHELL

If your shell doesn't show /bin/bash please change your shell
$ bash

Setup your environment to use Hadoop on Magellan system
$ module load tig hadoop

Hadoop Command

hadoop command [genericOptions] [commandOptions]

Examples:-

command – fs, jar, job

[genericOptions] - -conf, -D, -files, -libjars, -archives

[commandOptions] - -ls, -submit

HDFS Commands [1]

$ hadoop fs –ls
If you see an error do the following where [username] is your training account username

$ hadoop fs -mkdir /user/[username]

$ vi testfile1 [Repeat for testfile2]
This is file 1
This is to test HDFS

$ hadoop fs -mkdir input

$ hadoop fs -put testfile* input

You can get help on commands -
$ hadoop fs -help

HDFS Commands [2]

$ hadoop fs -cat input/testfile1

$ hadoop fs -cat input/testfile*

Download the files from HDFS into a directory called input and check there is a input directory.

$ hadoop fs -get input input

$ ls input/

Monitoring

http://maghdp01.nersc.gov:50030/

http://maghdp01.nersc.gov:50070/

```
$   hadoop job -list
```

Wordcount Example

Input in HDFS

$ hadoop fs -mkdir wordcount-in

$ hadoop fs -put /global/scratch/sd/lavanya/ hadooptutorial/wordcount/* wordcount-in/

Run example

$ hadoop jar /usr/common/tig/hadoop/ hadoop-0.20.2+228/hadoop-0.20.2+228-examples.jar wordcount wordcount-in wordcount-op

View output

$ hadoop fs -ls wordcount-op

$ hadoop fs -cat wordcount-op/part-r-00000

$ hadoop fs -cat wordcount-op/p* | grep Darcy

Wordcount: Number of reduces

```
$  hadoop dfs -rmr wordcount-op

$  hadoop jar /usr/common/tig/hadoop/
   hadoop-0.20.2+228/hadoop-0.20.2+228-examples.jar
   wordcount -Dmapred.reduce.tasks=4 wordcount-in
   wordcount-op
```

http://maghdp01.nersc.gov:50030/

Wordcount: GPFS

Setup permissions for Hadoop user [ONE-TIME]

$ mkdir /global/scratch/sd/[username]/hadoop

$ chmod -R 755 /global/scratch/sd/[username]

$ chmod -R 777 /global/scratch/sd/[username]/hadoop/

Run Job

$ hadoop jar /usr/common/tig/hadoop

/hadoop-0.20.2+228/hadoop-0.20.2+228-examples.jar wordcount -Dfs.default.name=file://// /global/scratch/sd/lavanya/hadooptutorial/wordcount/ /global/scratch/sd/[username]hadoop/wordcount-gpfs/

Set perms for yourself

$ fixperms.sh /global/scratch/sd/[username]/hadoop/wordcount-gpfs/

Streaming with Unix Commands

```
$ hadoop  jar $HADOOP_HOME/contrib/streaming/
  hadoop*-streaming.jar -input wordcount-in -output
  wordcount-streaming-op -mapper /bin/cat -reducer /
  usr/bin/wc
$ hadoop fs -cat wordcount-streaming-op/p*
```

Streaming with Unix Commands/ GPFS

```
$ hadoop  jar $HADOOP_HOME/contrib/streaming/
  hadoop*-streaming.jar -Dfs.default.name=file:/// -
  input /global/scratch/sd/lavanya/hadooptutorial/
  wordcount/ -output /global/scratch/sd/[username]/
  hadoop/wordcount-streaming-op -mapper /bin/cat -
  reducer /usr/bin/wc
$  fixperms.sh  /global/scratch/sd/[username]/hadoop/
  wordcount-streaming-op
```

Streaming with Scripts

$ mkdir simple-streaming-example

$ cd simple-streaming-example

$ vi cat.sh

cat

Now let us test this

$ hadoop fs -mkdir cat-in

$ hadoop fs -put /global/scratch/sd/lavanya/hadooptutorial/cat/in/* cat-in/

$ hadoop jar /usr/common/tig/hadoop/hadoop-0.20.2+228/contrib/streaming/hadoop*streaming*.jar -mapper cat.sh -input cat-in -output cat-op -file cat.sh

Streaming with scripts – Number of reducers and mappers

```
$ hadoop jar /usr/common/tig/hadoop/
  hadoop-0.20.2+228/contrib/streaming/
  hadoop*streaming*.jar -Dmapred.reduce.tasks=0 -
  mapper cat.sh -input cat-in -output cat-op -file cat.sh
```

```
$  hadoop jar /usr/common/tig/hadoop/
  hadoop-0.20.2+228/contrib/streaming/
  hadoop*streaming*.jar -
  Dmapred.min.split.size=91212121212  -mapper cat.sh
  -input cat-in -output cat-op -file cat.sh
```

Census sample

```
$  mkdir census
$  cd census
$  cp /global/scratch/sd/lavanya/hadooptutorial/
   census/censusdata.sample .
$ mkdir census
$  cd census
$  cp /global/scratch/sd/lavanya/hadooptutorial/
   census/censusdata.sample .
```

Mapper

```
#The code is available in
    $   vi mapper.sh
    while read line; do
    if [[ "$line" == *Alabama* ]]; then
        echo "Alabama 1"
    fi
    if [[ "$line" == *Alaska* ]]; then
        echo -e "Alaska\t1"
    fi
    done
$   chmod 755 mapper.sh
$   cat censusdata.sample | ./mapper.sh
```

Census Run

```
$  hadoop fs -mkdir census
$  hadoop fs -put  /global/scratch/sd/lavanya/
   hadooptutorial/census/censusdata.sample census/
$  hadoop jar /usr/common/tig/hadoop/
   hadoop-0.20.2+228/contrib/streaming/
   hadoop*streaming*.jar -mapper mapper.sh -input
   census -output census-op -file mapper.sh –reducer /
   usr/bin/wc
$  hadoop fs -cat census-op/p*
```

Census Run: Mappers and Reducers

```
$ hadoop fs -rmr census-op

$  hadoop jar /usr/common/tig/hadoop/
   hadoop-0.20.2+228/contrib/streaming/
   hadoop*streaming*.jar -Dmapred.map.tasks=10 -
   Dmapred.reduce.tasks=2 -mapper  mapper.sh -input
   census -output census-op/ -file mapper.sh -reducer /
   usr/bin/wc
```

Census: Custom Reducer

```
$ vi reducer.sh
   last_key="Alabama"
   while read line; do
      key=`echo $line | cut -f1 -d' '`
            val=`echo $line | cut -f2 -d' '`
      if [[ "$last_key" = "$key" ]];then
               let "count=count+1";
      else
               echo "**" $last_key $count
              last_key=${key};
              count=1;
      fi
   done
   echo "**" $last_key $count
```

Census Run with custom reducer

```
$  hadoop fs -rmr census-op

$  hadoop jar /usr/common/tig/hadoop/
   hadoop-0.20.2+228/contrib/streaming/
   hadoop*streaming*.jar -Dmapred.map.tasks=10 -
   Dmapred.reduce.tasks=2 -mapper mapper.sh -input
   census -output census-op -file mapper.sh -reducer
   reducer.sh -file reducer.sh
```