**Hadoop MapReduce Streaming Using Bash Script.**

MapReduce has a feature known as **Hadoop Streaming** that gives the flexibility to write code in your favorite language other than Java. You can use Ruby, Perl, Python or even quickly write a MapReduce job using shell script.

By definition, any language that can read from standard input and write to standard output can be used with Hadoop streaming. Remember, appropriate packages needs to be installed on each node in the cluster. For example, if we are using Perl, make sure the Perl is installed on all nodes in the cluster. Here is example code to demonstrate how to use Hadoop streaming with shell scripting.

The input should be a text file. This program will output the average length of a word that starts with a particular character. You may add checks to ignore characters like quotation, astrick, slash, etc and improve the performance by using lesser pipes and easier commands.

**Step 1: Create a mapper script(word\_length.sh) on your local file system**

#!/bin/bash

#This mapper script will read one line at a time and then break it into words

#For each word starting LETTER and LENGTH of the word are emitted

while read line

do

for word in $line do

if [ -n $word ] then

wcount=`echo $word | wc -m`;

wlength=`expr $wcount - 1`;

letter=`echo $word | head -c1`;

echo -e "$lettert$wlength";

fi

done

done

#The output of the mapper would be “starting letter of each word” and “its length”, separated by a tab space.

**Step 2:  Create a reducer script(average\_word\_length.sh)**

#!/bin/bash

#This reducer script will take-in output from the mapper and emit starting letter of each word and average length

#Remember that the framework will sort the output from the mappers based on the Key

#Note that the input to a reducer will be of a form(Key,Value)and not (Key,

#This is unlike the input i.e.; usually passed to a reducer written in Java.

lastkey="";

count=0;

total=0;

iteration=1

while read line

do

newkey=`echo $line | awk '{print $1}'`

value=`echo $line | awk '{print $2}'`

if [ "$iteration" == "1" ] then

lastkey=$newkey;c

iteration=`expr $iteration + 1`;

fi

if [[ "$lastkey" != "$newkey" ]] then

average=`echo "scale=5;$total / $count" | bc`;

echo -e "$lastkeyt$average"

count=0;

lastkey=$newkey;

total=0;

average=0;

fi

total=`expr $total + $value`;

lastkey=$newkey;

count=`expr $count + 1`;

done

#The output would be Key,Value pairs(letter,average length of the words starting with this letter)

**Step 3: Run the job from the terminal using Hadoop streaming command**

hadoop jar /usr/lib/hadoop-0.20-mapreduce/contrib/streaming/hadoop-streaming\*.jar -input /input -output /avgwl -mapper mapper.sh -reducer reducer.sh -file /home/user/mr\_streaming\_bash/mapper.sh -file /home/user/mr\_streaming\_bash/reducer.sh

Use “-file” parameter to copy scripts to each of the nodes that will run parts of this file. The job will not run if this is not specified.

You can do some performance comparison between running the same job using different languages like Perl, Ruby, Bash, Python,etc.