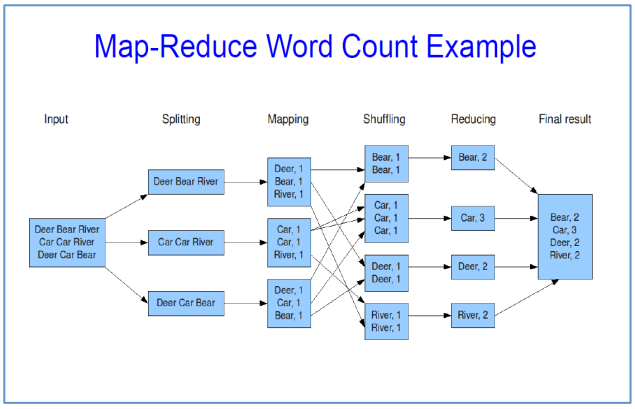
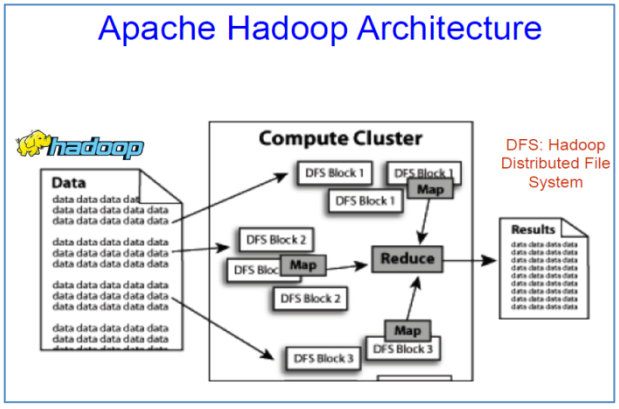
Homework 4 Problem
CS 4001/7001 Cloud Computing - Spring 2015

Dr. Prasad Calyam & Ronny Bazan Antequera (Contact: calyamp@missouri.edu)

This assignment is the implementation part of 'MapReduce' using AWS EMR (Elastic Map Reduce). EMR service provides several options to run your application depending on the type of program that you are developing; *for this assignment you will focus on 'streaming' type*. EMR uses other services such as EC2 to deploy the required instances and S3 to store your logs, source or input files, programs and output, as well as CloudWatch for collecting metrics related to your application.





For this assignment, you will need to create a bucket 'mapreduce-yourpawprint' and 3 folders inside your bucket ('logs', 'input' and 'programs'). Word Count Example will be applied that consists of following steps: opening a file, splitting, mapping, shuffling, reducing and finally showing the number of occurrences of each word.

1. Download the `input.txt` file from Blackboard, and store it to your S3 'input' folder.

2. Copy two files (`mapper.py` and `reducer.py`) into your S3 'programs' folder.
NOTE: Streaming Option in EMR *has support for following languages: Ruby, Perl, Python, PHP, R, Bash, or C++. To use this option, we are going to use the source code for mapper and reducer written in Python.*

mapper.py

```
#!/usr/bin/env python
import sys
import re
import string
pattern = re.compile("^[a-z][a-z0-9]*$")
for line in sys.stdin:
   line = line.strip()
   words = line.split()
   for word in words:
      lword = word.lower()
      if pattern.match(lword):
         print '%s%s%d' % (lword, "\t", 1)
```

reducer.py

```
#!/usr/bin/env python

from itertools import groupby
from operator import itemgetter
import sys
def read_mapper_output(file, separator='\t'):
   for line in file:
      yield line.rstrip().split(separator, 1)
def main(separator='\t'):
   data = read_mapper_output(sys.stdin, separator=separator)
   for current_word, group in groupby(data, itemgetter(0)):
      try:
         total_count = sum(int(count) for current_word, count in group)
         print "%s%s%d" % (current_word, separator, total_count)
      except ValueError:
         pass
if __name__ == "__main__":
   main()
```

3. Access your AWS EMR environment, create a new cluster and configure based on the following: Cluster name, Termination protection=No, Log folder S3 location, disable 'debugging' iption, remove Hive, Pig and Hue applications and in 'Steps' section configure auto-terminate=yes, and select 'streaming program' and click on 'Configure and add'. In there you will need to define a name for your cluster, enter your S3 path for mapper.py, reducer.py, Input and Output (note that output should be unique), and select 'Terminate Cluster' for 'Action on Failure' option.

4. After above configuration, Click on 'Create Cluster' and navigate through the options to see details of your cluster. Pay special attention to 'Steps' section in AWS EMR since once the status change to 'Completed' you will be able to see your output files in your S3 bucket.

QUESTIONS:

Q1. Mention the number of EC2 instances created by default, type of the instances and clearly explain the purpose of each of them.

Q2. The output will consist in many 'part files', list the first and last word in each file and the associated number of occurrences.

Q3. Run the Word Count example (input.txt) multiple times by increasing the Slave cores in your cluster, and complete the below chart. Explain relationship between the number of cores and the output parts.

Master	*Cores*	*Output parts*	*Cluster ID*
	2		
	3		
	4		
	5		
	6		

Q4. Provide a screenshot of the clusters created with EMR. Note that your name and several clusters need to be clearly visible.