과목 명: 시스템프로그래밍

담당 교수 명: 소 정 민

<<Assignment 6>>

**서강대학교 컴퓨터공학과**

**[20151561]**

**[신용하]**

목 차

1. 프로그램 개요 3

2. 프로그램 설명 3

2.1 프로그램 흐름도 3

3. 모듈 정의 3

3.1 모듈 이름 : 전체 네트워크의 흐름 3

4. 코드 4

# 프로그램 개요

Hadoop과 MapReduce에 대한 기본 지식을 향상시킨다.

# 프로그램 설명

16개의 그룹, 1억개의 숫자들을 각 그룹 별 최댓값을 찾는 방식으로 MapReduce를 진행한다.

## 프로그램 흐름도

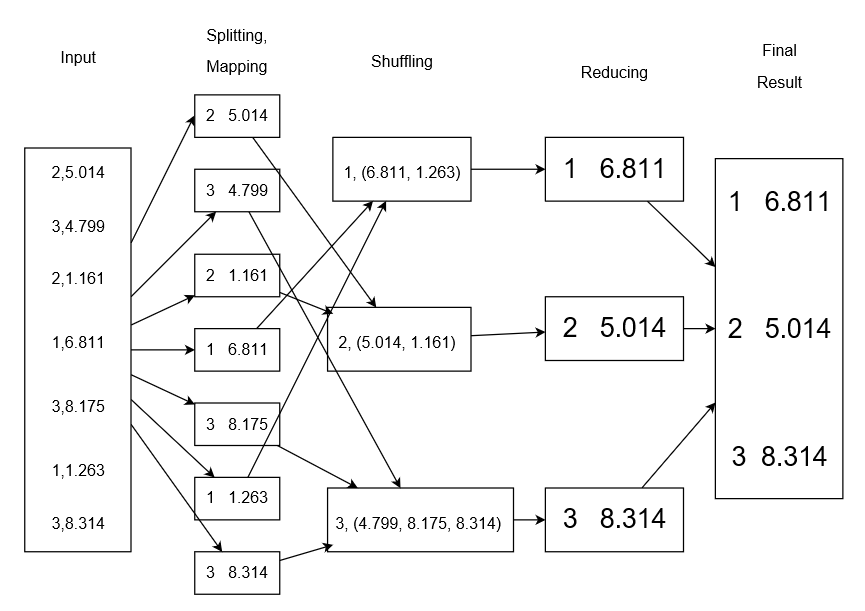


그림 1> 다이어그램

# 모듈 정의

## 전체 네트워크의 흐름

1) 하둡 클러스터를 만든다.(리소스 그룹)

2) 맵리듀스 코드와 input.data를 클라우드 쉘에 올린다.

3) 파일을 클러스터로 옮긴다.

4) 맵리듀스를 진행한다.

5) 완료 후 리소스 그룹을 삭제한다.

# 코드

mapper.py

#!/usr/bin/env python

# Use the sys module

import sys

# 'file' in this case is STDIN

def read\_input(file):

# Split each line into words

for line in file:

yield line.split(",")

def main(separator='\t'):

# Read the data using read\_input

data = read\_input(sys.stdin)

# Process each word returned from read\_input

for words in data:

# Process each word

# Write to STDOUT

print('%s%s%s' % (words[0], separator, words[1]))

if \_\_name\_\_ == "\_\_main\_\_":

main()

reducer.py

#!/usr/bin/env python

# import modules

from itertools import groupby

from operator import itemgetter

import sys

# 'file' in this case is STDIN

def read\_mapper\_output(file, separator='\t'):

# Go through each line

for line in file:

# Strip out the separator character

yield line.rstrip().split(separator)

def main(separator='\t'):

# Read the data using read\_mapper\_output

data = read\_mapper\_output(sys.stdin, separator=separator)

# Group words and counts into 'group'

# Since MapReduce is a distributed process, each word

# may have multiple counts. 'group' will have all counts

# which can be retrieved using the word as the key.

for current\_word, group in groupby(data, itemgetter(0)):

try:

# For each word, pull the count(s) for the word

# from 'group' and create a total count

total\_count = max(float(count) for current\_word, count in group)

# Write to stdout

print("%s%s%f" % (current\_word, separator, total\_count))

except ValueError:

# Count was not a number, so do nothing

pass

if \_\_name\_\_ == "\_\_main\_\_":

main()