과목 명: 시스템프로그래밍

담당 교수 명: 김 지 환

<<Assignment 6>>

**서강대학교 컴퓨터학과**

**[학번] 20140134**

**[이름] 조 예린**

**목 차**

1. 프로그램 개요 3

2. 프로그램 설명 3

2.1 프로그램 흐름도 3

3. 모듈 정의 3

4. 코드

# **프로그램 개요**

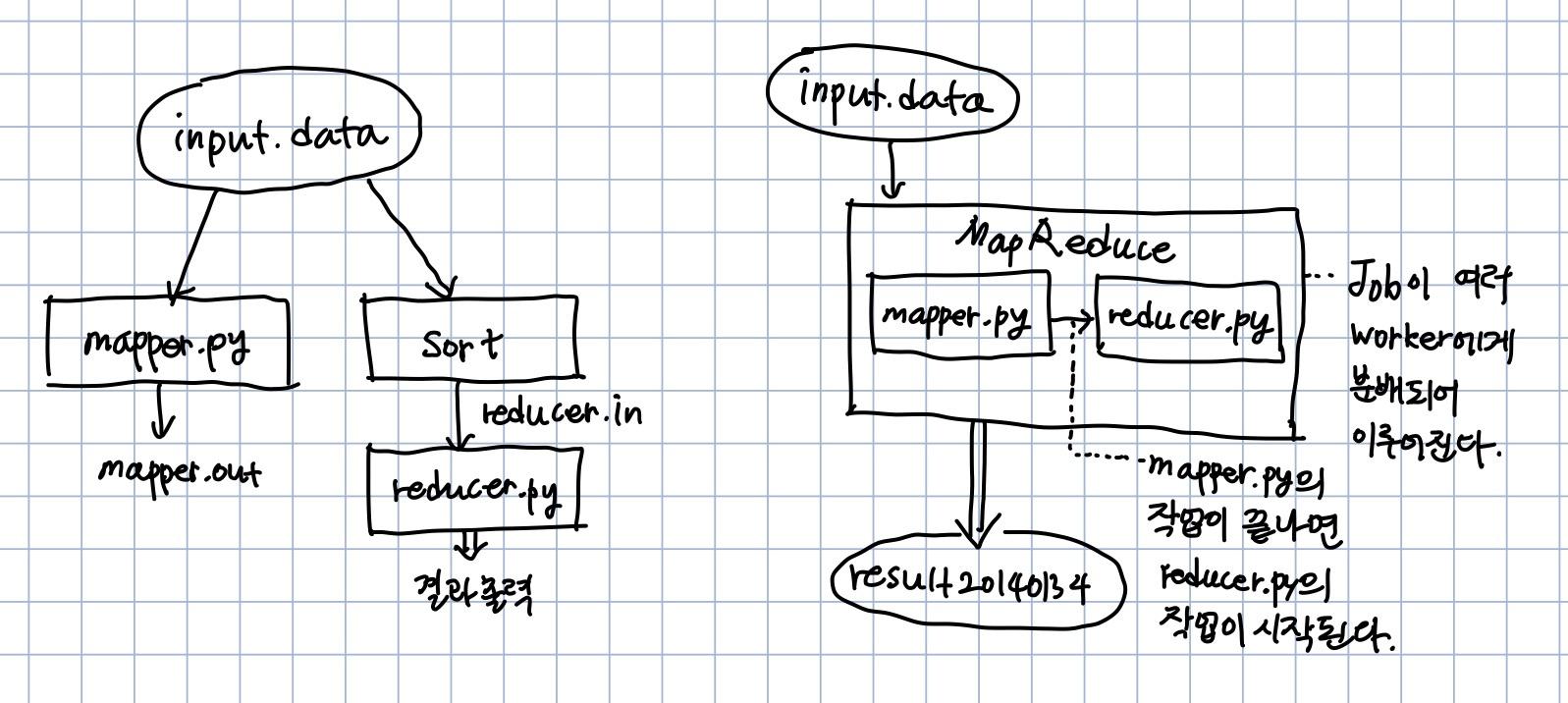
해당 프로젝트는 하둡과 맵리듀스의 기초 지식을 증진하기 위한 것이다. 프로젝트 내의 코드는 Python3을 사용하여 작성하며, Azure HDInsight service 환경에서 진행된다. 데이터 처리를 위해 MapReduce 테크닉을 사용하여 Group by Max problem을 1억개의 데이터에 대하여 진행한다.

GroupByMax problem은 각 그룹에서 가장 큰 값을 찾는 것이다. 입력은 한 개의 integer와 한 개의 real number로 각 라인이 이루어져 있으며, 각 숫자는 comma로 구분되어 있다. integer 숫자는 그룹 숫자를 의미하며, real 숫자는 값이다. 주어진 데이터에서 각 그룹의 최댓값을 찾는 것이 목표이다.

# **프로그램 설명**

## **프로그램 흐름도**

## 



(위의 데이터는 PPT에서 예시로 사용된 것을 차용하여 데이터 흐름도를 작성한 것으로, 실제 프로그램에서 사용되는 것은 generate.py를 통해 만들어진 1억개의 데이터 셋이다.)

cat input.data | python3 mapper.py > mapper.out

위의 명령어를 통하여 mapper.py에 input.data가 stdin으로 들어가고, 결과값이 mapper.out으로 만들어진다.

cat input.data | sort > reducer.in

위의 명령어를 통하여 input.data를 sort한 결과물이 reducer.in으로 들어가게 된다.

cat reducer.in | python reducer.py

위의 명령어를 통하여 reducer.in 파일이 reducer.py를 돌아, 그룹마다의 max값을 취하여 결과가 출력된다.

# **모듈 정의**

## **모듈 이름 : Mapper.py**

## *3.1.1 기능*

if \_\_name\_\_==”\_\_main\_\_” 문을 이용해, mapper.py가 직접 실행되었을 때에 main() 함수가 실행되게끔 되어있다. 메인 펑션이 호출되면, read\_input(sys.stdin)으로 cat을 통해 입력된 파일을 넘겨준다. 그러면 파일을 한 줄씩 순회하며 comma를 기준으로 split하여 key와 value로 리턴하고, 이를 다시 group이라는 이름의 딕셔너리에 키와 밸류 값으로 넘겨주는 형식이다. 이때 group의 key가 없을 경우를 대비하여 for문을 돌며 초기화를 진행한다. 그렇게 만들어진 group 딕셔너리는 return 값이 되어 다시 메인 함수로 돌아간다.

메인 함수는 받아온 group의 키 값을 for문 내에서 순회하며, key값의 value를 하나씩 출력한다. 이를 통해 sort가 되는 것이다.

## **모듈 이름: Reducer.py**

### *기능*

if \_\_name\_\_==”\_\_main\_\_” 문을 이용해, reducer.py가 직접 실행되었을 때에 main() 함수가 실행되게끔 되어있다. data라는 이름의 딕셔너리를 만들고, sys.stdin을 enumerate하며 index와 line으로 반환한다. 이때 받아온 line은 comma를 기준으로 split하여 group과 number로 반환해준다. 또한 data의 key값을 통해 data내의 그룹에 접근하여, 받아온 number와 data[group]의 number를 비교하여 큰 쪽을 취한다. 이후 data.items()를 통해 for문을 돌며 group과 number를 print해준다.

# **코드**

[Mapper.py]

import sys

def read\_input(file):

group = {}

for index in range(0,16) :

group[index]=[]

for line in file :

key, value = line.split(',')

group[int(key)].append(float(value))

return group

def main(seperator=','):

group = read\_input(sys.stdin)

for key in group.keys() :

for value in group[key] :

print('%s%s%s'%(key,seperator,float(value)))

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

main()

[Reducer.py]

import sys

def main() :

data = {}

for index, line in enumerate(sys.stdin):

group, number = line.split(',')

number = float(number)

if group in data.keys() :

data[group] = max(data[group], number)

else :

data[group] = number

for group, number in data.items() :

print('{}\t{}'.format(group, number))

if \_\_name\_\_ == '\_\_main\_\_' :

main()