빅데이터 분석 및 응용

L03.1: MapReduce Practice on a Cluster

Summer 2020

Kookmin University

Goals

- The goal of this practice is to get you familiar with the basics of following:
    - Hadoop and HDFS commands
    - Hadoop MapReduce API
    - Write programs with MapReduce

Contents

- **Hadoop and HDFS Commands**
- WordCount on a Cluster
- More practices

Hadoop and HDFS Commands

- Upload file into HDFS
  - Type ‘hdfs dfs –put <LOCAL PATH> <HDFS PATH>‘ in terminal
- Download file from HDFS
  - Type ‘hdfs dfs –get <HDFS PATH> <LOCAL PATH>‘ in terminal
  - Type ‘hdfs dfs –getmerge <HDFS PATH> <LOCAL PATH>‘ in terminal
  - You can download the files via HDFS web interface.

Hadoop and HDFS Commands

- List files in HDFS
  - Type ‘hdfs dfs –ls <HDFS PATH>‘ in terminal
  - You can see the files via HDFS web interface.
- Show file contents in HDFS
  - Type ‘hdfs dfs –cat <HDFS PATH>‘ in terminal
- Delete a file in HDFS
  - Type ‘hdfs dfs –rm <HDFS PATH>‘ in terminal
- Delete a directory in HDFS
  - Type ‘hdfs dfs –rm –r <HDFS PATH>‘ in terminal

Contents

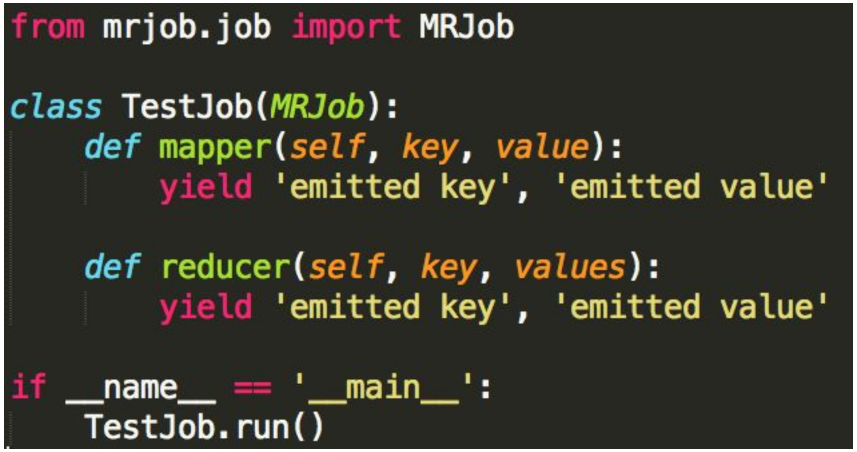
- **Hadoop and HDFS Commands**
- WordCount on a Cluster
- More practices

MapReduce in Python (mrjob) (1)

- A library to write MapReduce job in Python
- With mrjob, you can:
  - Write MapReduce job in pure Python
  - Test on your local machine
  - Run on a Hadoop cluster
  - Run in a cloud using Amazon Elastic MapReduce
  - Run in a cloud using Google Cloud Dataproc

MapReduce in Python (mrjob) (2)

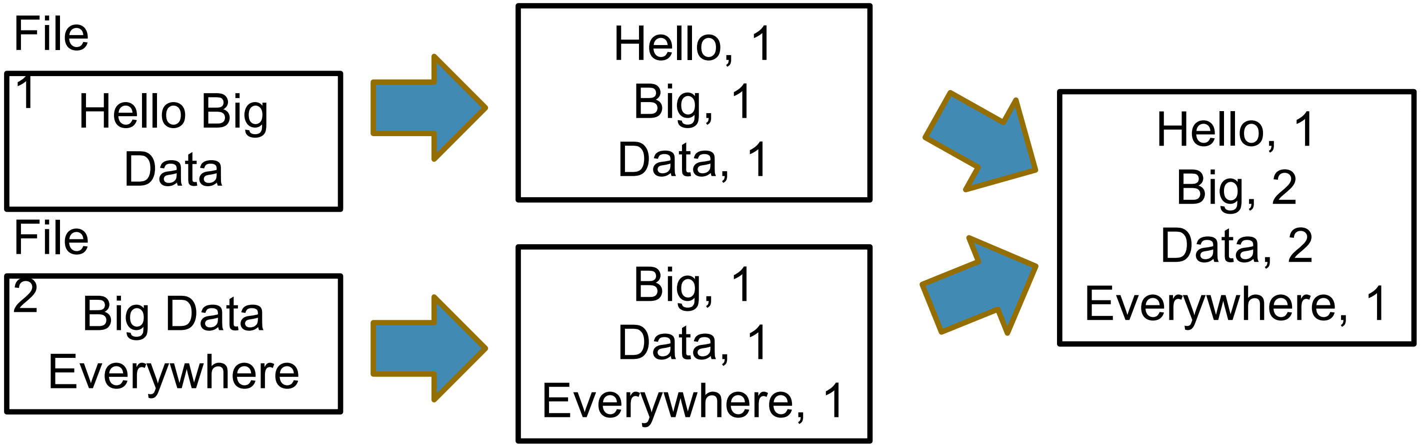
- Defining a MapReduce job
  1. Create a class derived from `mrjob.job.MRJob`
  2. Override `mapper` and `reducer` methods
  3. Add a main block to run the MapReduce job



```python
from mrjob.job import MRJob

class TestJob(MRJob):
    def mapper(self, key, value):
        yield 'emitted key', 'emitted value'

    def reducer(self, key, values):
        yield 'emitted key', 'emitted value'

if __name__ == '__main__':
    TestJob.run()
```

WordCount (MapReduce) (1)

- Counts the number of time each distinct word appears in the file



File 1

Hello Big Data

File 2

Big Data Everywhere

Hello, 1
Big, 1
Data, 1

Big, 1
Data, 1
Everywhere, 1

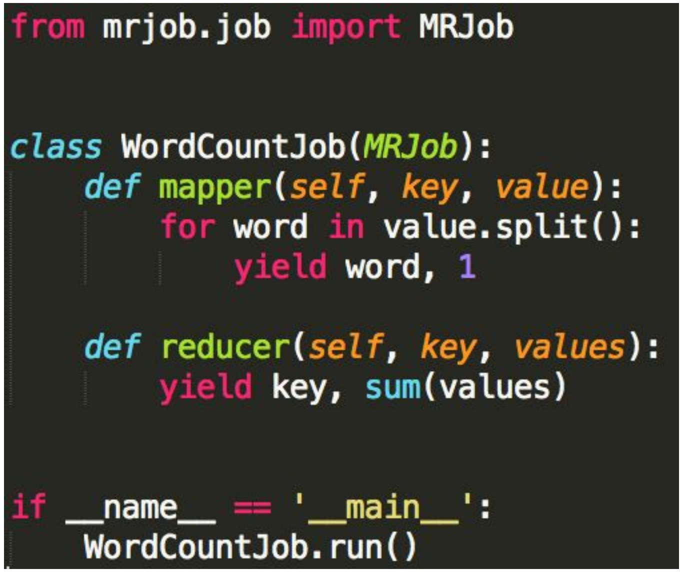
Hello, 1
Big, 2
Data, 2
Everywhere, 1

- Input: <line number, text> pair
- Output: <word, word frequency> pair

WordCount (MapReduce) (2)

- Map
  - Read each line in the input file and emit partial word–count pairs
- Reduce
  - Sum all partial word–count pairs with the same key

WordCount (MapReduce) (3)



```python
from mrjob.job import MRJob


class WordCountJob(MRJob):
    def mapper(self, key, value):
        for word in value.split():
            yield word, 1

    def reducer(self, key, values):
        yield key, sum(values)


if __name__ == '__main__':
    WordCountJob.run()
```

WordCount (MapReduce) (4)

- Run our WordCount MapReduce job
  1. Open terminal on a cluster
  2. Type `python wordcount.py -r hadoop hdfs:///wc-input --output-dir hdfs:///wc-output`
  3. Check the output via HDFS cat command

Contents

- **Hadoop and HDFS Commands**
- WordCount on a Cluster
- More practices

서울 공기질 데이터 다루기

- https://www.kaggle.com/bappekim/air-pollution-in-seoul
- 연습문제
  - 문제 1) 각 지역별로 평균, 최대, 최소 PM10 측정치 구하기
  - 문제 2) PM10, PM2.5 기준으로 공기의 질이 ‘좋음’ 수준이 가장 많이 측정된 지역은 어디인지 찾기
  - 문제 3) 데이터 변환하기 → 각 시간, 지역별로 모든 종류의 측정치 모아서 저장하기
  - 문제 4) 시간대를 기준으로 평균 공기질 구하기 (SO2, NO2, CO, O3, PM10, PM2.5 한꺼번에 구하기)

Questions?