Università degli Studi di Messina

Data Science

Big Data Acquisition MapReduce

Prof. Daniele Ravi

Honorary Associate Professor
University College London (UCL) – UK
Department of Computer Science

dravi@unime.it

Hadoop Streaming

- **Hadoop Streaming** is a powerful utility within the Hadoop ecosystem that allows you to process large datasets in a distributed manner using any programming language, including Python.

- It works by enabling the user to define MapReduce jobs with custom mappers and reducers in a non-Java language like Python.

How Hadoop Streaming Works

- **Mapper**: The mapper reads data line by line from standard input (stdin) and outputs key-value pairs to standard output (stdout).
- In Python, you can write a mapper function as a simple script that processes each line of input.

- **Reducer**: The reducer reads the key-value pairs output by the mapper and aggregates or processes them as needed, then outputs the final result to stdout.

Example of Hadoop Streaming with Python:

Mapper (mapper.py):

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

# input comes from STDIN (standard input)
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()
    # split the line into words
    words = line.split()
    # output each word with a count of 1
    for word in words:
        print(f'{word}\t1')
```

Example of Hadoop Streaming with Python:

Reducer (reducer.py):

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

current_word = None
current_count = 0

# input comes from STDIN
for line in sys.stdin:
    line = line.strip()
    word, count = line.split('\t', 1)
    count = int(count)

    if current_word == word:
        current_count += count
    else:
        if current_word:
            print(f'{current_word}\t{current_count}')
        current_word = word
        current_count = count

if current_word == word:
    print(f'{current_word}\t{current_count}')
```

Running Hadoop Streaming with Python:

- To run these scripts on Hadoop, you would use a command like this:

- hadoop jar /path/to/hadoop-streaming.jar \ -input /input/path \ -output /output/path \ -mapper mapper.py \ -reducer reducer.py

Key Features

1.Language Flexibility: Python (or any language) can be used for map and reduce operations, giving flexibility beyond Java.

2.Data Processing: It processes large amounts of data efficiently by distributing the computation across a cluster of machines.

3.Text-based Communication: Input and output are passed via standard I/O, making it easier to integrate with external systems.

Hadoop Streaming with Python is widely used for prototyping and simpler tasks

Exercise 1: **invert an index** for documents

- This is useful in search engines where you want to create a list of documents in which each word appears.

- **Problem Description:**
- We have a set of documents, and we want to build an inverted index, i.e., for each word, we want to output the list of document IDs where that word appears.

- **Input Format:**
  - Each line of input contains a document ID followed by the content of the document, e.g.,:
    1. *Doc1 -> Hadoop Streaming is powerful*
    2. *Doc2 -> Hadoop allows you to process big data*
    3. *Doc3 -> Streaming data with Hadoop*

- **Output Format:**
  - For each word, list the document IDs where it appears, e.g.,
    1. *Hadoop -> doc1,doc2,doc3*
    2. *Streaming -> doc1,doc3 is doc1*
    3. *Powerful -> doc1*
    4. *Allows doc2*

Exercise 2: Calculating the average temperature

Calculating the average temperature per year from a large dataset of weather station readings.

Problem Description:

- We have a large dataset where each line contains weather data, including the year and temperature for a specific day. We want to calculate the **average temperature for each year**.

Input Format:

- Each line contains weather data in the following format: year temperature
  - For example:
    - *2020 35*
    - *2020 28*
    - *2021 30*
    - *2021 32*

Goal:

- For each year, compute the average temperature.