Hadoop MapReduce with Docker

Giới thiệu

Hadoop – Map Reduce là một khung nền (software framework) mã nguồn mở, hỗ trợ người lập trình viết các ứng dụng theo mô hình Map/Reduce. Để hiện thực một ứng dụng theo mô hình Map/Reduce, sinh viên cần sử dụng các interface lập trình do Hadoop cung cấp như: Mapper, Reducer, JobConf, JobClient, Partitioner, OutputCollector, Reporter, InputFormat, OutputFormat, v.v..

Yêu cầu sinh viên tìm hiểu và chạy ứng dụng WordCount để hiểu rõ hoạt động của mô hình Map/Reduce và kiến trúc HDFS (Hadoop Distributed FileSystem). MapReduce Tutorial (apache.org)

Cài đặt

1. Thư mục build docker

```
.
├── config
│   ├── core-site.xml
│   ├── hdfs-site.xml
│   ├── mapred-site.xml
│   ├── ssh_config
│   ├── start-hadoop.sh
│   └── yarn-site.xml
└── Dockerfile
```

2. Chuẩn bị dockerfile

```
FROM ubuntu:latest

# set environment vars
ENV HADOOP_HOME /opt/hadoop
ENV JARs opt/hadoop/share/hadoop/tools/lib/hadoop-streaming-3.3.6.jar
ENV HDFS_NAMENODE_USER="root"
ENV HDFS_DATANODE_USER="root"
ENV HDFS_SECONDARYNAMENODE_USER="root"
ENV YARN_RESOURCEMANAGER_USER="root"
ENV YARN_NODEMANAGER_USER="root"

# install packages
RUN \
  apt-get update && apt-get install -y \
  ssh \
  rsync \
  vim \
  openjdk-8-jdk


# download and extract hadoop, set JAVA_HOME in hadoop-env.sh, update
# Ref: https://downloads.apache.org/hadoop/common/
RUN \
  wget https://downloads.apache.org/hadoop/common/hadoop-3.3.6/hadoop
  tar -xzvf hadoop-3.3.6.tar.gz

# Move dir hadoop to /opt/hadoop
RUN  mv hadoop-3.3.6 /opt/hadoop

# set env for hadoop
RUN  echo "export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-arm64" >>
RUN  echo "PATH=$PATH:/opt/hadoop/bin" >> ~/.bashrc

# create ssh keys
RUN \
  ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa && \
  cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys && \
  chmod 0600 ~/.ssh/authorized_keys

# copy hadoop configs
ADD config/*xml /opt/hadoop/etc/hadoop/

# copy ssh config
ADD config/ssh_config /root/.ssh/config

# copy script to start hadoop
ADD config/start-hadoop.sh start-hadoop.sh
```

Lưu ý: lệnh bên dưới là chọn đường dẫn của jdk java mà được cài đặt cho hadoop, dựa trên loại kiến trúc CPU của máy local (ARM64 hoặc AMD64).

Nếu máy local là ARM64

```
RUN  echo "export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-arm64" >>
```

Nếu máy local là AMD64

```
RUN  echo "export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64" >>
```

Để kiểm tra máy local có CPU kiến trúc nào:

Trên Linux hoặc macOS

1. **Sử dụng `uname`:**

   ```
   uname -m
   ```

   - Nếu kết quả là `x86_64`, đó là CPU AMD64 (64-bit).
   - Nếu kết quả là `aarch64`, đó là CPU ARM64 (64-bit).

Trên Windows

1. **Sử dụng `wmic`:**
   Mở Command Prompt và chạy lệnh sau:

   ```
   wmic os get osarchitecture
   ```

   - `x86_64`: Kiến trúc AMD64
   - `aarch64`: Kiến trúc ARM64

3. Build docker image

```
docker build -t myhadoop .
```

4. Run docker image

```
docker run -it myhadoop bash
```

Sau khi thực thi thành công, kiểm tra đã cài đặt hadoop:

```
hadoop version
```

Output:

```
root@196ba02a51de:/# hadoop version
Hadoop 3.3.6
Source code repository https://github.com/apache/hadoop.git -r 1be782
Compiled by ubuntu on 2023-06-18T08:22Z
Compiled on platform linux-x86_64
Compiled with protoc 3.7.1
From source with checksum 5652179ad55f76cb287d9c633bb53bbd
This command was run using /opt/hadoop/share/hadoop/common/hadoop-com
```

Sau khi đã cài đặt thành công, thực hiện lệnh sau để khởi động:

```
bash start-hadoop.sh
```

Sau khi chạy xong, kiểm tra các file trên dfs

```
hdfs dfs -ls /
```

Output:

```
root@196ba02a51de:/# hdfs dfs -ls /
Found 1 items
drwxrwx---   - root supergroup          0 2024-08-25 04:09 /tmp
```

Lệnh Hadoop

- Kiểm tra các file trên dfs

  Command

  ```
  hdfs dfs -ls <dir>
  ```

  Với <dir> là thư mục cần kiểm tra. ví dụ kiểm tra file tại thư mục “/”

  ```
  hdfs dfs -ls /
  ```

- Tải file lên dfs

  Command

  ```
  hdfs dfs -put -f <source_file> <target_file>
  ```

- Set replica

  Command

  ```
   hdfs dfs -setrep <num_replica> <filename>
  ```

WordCount

Cho 1 file txt bao gồm nội dung bất kỳ, hãy dùng mapper và reducer để đếm các từ xuất hiện trong file (các từ cách nhau bởi khoảng trắng).

Hướng dẫn

Sau khi khởi động hadoop

- Tạo thư mục lab1

```
mkdir lab1 && cd lab1
```

- Tạo file input như sau và lưu với tên iuh.txt

```
echo 'Industrial University of HoChiMinh City

Industrial University of HoChiMinh City

Industrial University of HoChiMinh City

Faculty of Information Technology

Department of Data Science ' > iuh.txt
```

- Tải file iuh.txt lên dfs

```
hdfs dfs -put -f ./iuh.txt /
```

- Tạo file mapper - tách nội dung của file thành từng từ theo khỏang trắng và đếm 1 cho mỗi lần xuất hiện

```
echo '#!/usr/bin/python3
import sys
def main(argv):
    for line in sys.stdin:
        wordlist = line.strip().split()
        for word in wordlist:
            print(word+"\t"+"1")

if __name__ == "__main__":
    main(sys.argv)' > mapper.py
```

- Kiểm tra file mapper chạy được hay không?

```
root@ebcd075e748e:/lab1# cat iuh.txt | python3 mapper.py
Industrial      1
University      1
of      1
HoChiMinh       1
City    1
Industrial      1
University      1
of      1
HoChiMinh       1
City    1
Industrial      1
University      1
of      1
HoChiMinh       1
City    1
Faculty 1
of      1
Information     1
Technology      1
Department      1
of      1
Data    1
Science 1
```

- Cấp quyền thực thi cho file mapper

```
chmod a+x mapper.py
```

```
root@ebcd075e748e:/lab1# ll
total 16
drwxr-xr-x 2 root root 4096 Aug 25 08:01 ./
drwxr-xr-x 1 root root 4096 Aug 25 08:01 ../
-rw-r--r-- 1 root root  190 Aug 25 08:01 iuh.txt
-rwxr-xr-x 1 root root  223 Aug 25 08:01 mapper.py*
```

- Thực thi lại

```
root@ebcd075e748e:/lab1# cat iuh.txt | ./mapper.py
Industrial      1
University      1
of      1
HoChiMinh       1
City    1
Industrial      1
University      1
of      1
HoChiMinh       1
City    1
Industrial      1
University      1
of      1
HoChiMinh       1
City    1
Faculty 1
of      1
Information     1
Technology      1
Department      1
of      1
```

```
Data    1
Science 1
```

- Sắp xếp lại output để các từ giống nhau sẽ đứng cạnh nhau

```
root@ebcd075e748e:/lab1# cat iuh.txt | ./mapper.py | sort -k1,1
City    1
City    1
City    1
Data    1
Department      1
Faculty 1
HoChiMinh       1
HoChiMinh       1
HoChiMinh       1
Industrial      1
Industrial      1
Industrial      1
Information     1
Science 1
Technology      1
University      1
University      1
University      1
of      1
of      1
of      1
of      1
of      1
```

- Tạo file reducer - Từ output của mapper, tạo danh sách các từ và đếm số lần xuất hiện

```
echo '#!/usr/bin/python3
import sys
```

```
def main(argv):
    current_word = None
    count = 0
    for line in sys.stdin:
        word, n = line.strip().split("\t",1)
        n = int(n)
        if current_word == word:
            count += n
        else:
            if current_word:
                print(current_word+"\t"+str(count))
            count = n
            current_word = word
    if current_word == word:
        print(current_word+"\t"+str(count))

if __name__ == "__main__":
    main(sys.argv)
' > reducer.py
```

- Cấp quyền thực thi cho reducer

```
chmod a+x reducer.py
```

- Kiểm tra file reducer chạy được hay không?

```
root@ebcd075e748e:/lab1# cat iuh.txt | ./mapper.py | sort -k1,1 | ./r
City    3
Data    1
Department      1
Faculty 1
HoChiMinh       3
Industrial      3
Information     1
Science 1
```

```
Technology      1
University      3
of      5
```

- Thực thi với mapper và reducer với hadoop

```
hadoop jar <jar_file> -input <file_in_dfs> -output <dir_output> -mappe
```

ví dụ:

```
 hadoop jar $JARs -input /iuh.txt -output output -mapper /lab1/mapper
```

Output

```
root@ebcd075e748e:/lab1# hadoop jar $JARs -input /iuh.txt -output out
2024-08-25 13:14:46,501 WARN util.NativeCodeLoader: Unable to load na
packageJobJar: [/tmp/hadoop-unjar8505073691030412411/] [] /tmp/stream
2024-08-25 13:14:46,925 INFO client.DefaultNoHARMFailoverProxyProvide
2024-08-25 13:14:47,081 INFO client.DefaultNoHARMFailoverProxyProvide
2024-08-25 13:14:47,236 INFO mapreduce.JobResourceUploader: Disabling
2024-08-25 13:14:47,453 INFO mapred.FileInputFormat: Total input file
2024-08-25 13:14:47,510 INFO mapreduce.JobSubmitter: number of splits
2024-08-25 13:14:48,024 INFO mapreduce.JobSubmitter: Submitting token
2024-08-25 13:14:48,024 INFO mapreduce.JobSubmitter: Executing with t
2024-08-25 13:14:48,164 INFO conf.Configuration: resource-types.xml n
2024-08-25 13:14:48,164 INFO resource.ResourceUtils: Unable to find '
2024-08-25 13:14:48,668 INFO impl.YarnClientImpl: Submitted applicati
2024-08-25 13:14:48,710 INFO mapreduce.Job: The url to track the job:
2024-08-25 13:14:48,712 INFO mapreduce.Job: Running job: job_17245728
2024-08-25 13:14:55,832 INFO mapreduce.Job: Job job_1724572860133_000
2024-08-25 13:14:55,835 INFO mapreduce.Job:  map 0% reduce 0%
2024-08-25 13:14:58,972 INFO mapreduce.Job:  map 100% reduce 0%
2024-08-25 13:15:03,008 INFO mapreduce.Job:  map 100% reduce 100%
2024-08-25 13:15:03,016 INFO mapreduce.Job: Job job_1724572860133_000
```

```
2024-08-25 13:15:03,121 INFO mapreduce.Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=279
                FILE: Number of bytes written=835565
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=447
                HDFS: Number of bytes written=117
                HDFS: Number of read operations=11
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
                HDFS: Number of bytes read erasure-coded=0
        Job Counters
                Launched map tasks=2
                Launched reduce tasks=1
                Data-local map tasks=2
                Total time spent by all maps in occupied slots (ms)=34
                Total time spent by all reduces in occupied slots (ms
                Total time spent by all map tasks (ms)=3404
                Total time spent by all reduce tasks (ms)=1447
                Total vcore-milliseconds taken by all map tasks=3404
                Total vcore-milliseconds taken by all reduce tasks=144
                Total megabyte-milliseconds taken by all map tasks=348
                Total megabyte-milliseconds taken by all reduce tasks
        Map-Reduce Framework
                Map input records=9
                Map output records=23
                Map output bytes=227
                Map output materialized bytes=285
                Input split bytes=162
                Combine input records=0
                Combine output records=0
                Reduce input groups=11
                Reduce shuffle bytes=285
                Reduce input records=23
                Reduce output records=11
                Spilled Records=46
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=196
                CPU time spent (ms)=1190
                Physical memory (bytes) snapshot=789528576
                Virtual memory (bytes) snapshot=7449767936
                Total committed heap usage (bytes)=569901056
                Peak Map Physical memory (bytes)=267350016
                Peak Map Virtual memory (bytes)=2480377856
                Peak Reduce Physical memory (bytes)=267165696
                Peak Reduce Virtual memory (bytes)=2489942016
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=285
        File Output Format Counters
                Bytes Written=117
2024-08-25 13:15:03,121 INFO streaming.StreamJob: Output directory: o
```

Sau khi thực hiện, kiểm tra output được lưu trên dfs

```
root@ebcd075e748e:/lab1# hdfs dfs -ls output/
2024-08-25 13:16:34,704 WARN util.NativeCodeLoader: Unable to load na
Found 2 items
```

```
-rw-r--r--   1 root supergroup          0 2024-08-25 13:15 output/_SU
-rw-r--r--   1 root supergroup        117 2024-08-25 13:15 output/par
```

Xem kết quả ở file output part-0000

```
root@ebcd075e748e:/lab1# hdfs dfs -cat output/part-00000
2024-08-25 13:17:43,424 WARN util.NativeCodeLoader: Unable to load na
City    3
Data    1
Department      1
Faculty 1
HoChiMinh       3
Industrial      3
Information     1
Science 1
Technology      1
University      3
of      5
```

Note: Nếu xảy ra lỗi với $JARs, kiểm tra lại biến môi trường $JARs. JARs là đường dẫn đến file “hadoop-streaming-3.3.6.jar”. Kiểm tra tại thư mục: “/opt/hadoop/share/hadoop/tools/lib/hadoop-streaming-3.3.6.jar”. Thực hiện gán cho biến môi trường

```
export JARs=/opt/hadoop/share/hadoop/tools/lib/hadoop-streaming-3.3.6
```

Hoặc thực hiện trực tiếp

```
hadoop jar /opt/hadoop/share/hadoop/tools/lib/hadoop-streaming-3.3.6.
```