**LAB 2 : 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

## Thư mục build docker

.

├── config

│ ├── core-site.xml

│ ├── hdfs-site.xml

│ ├── mapred-site.xml

│ ├── ssh\_config

│ ├── start-hadoop.sh

│ └── yarn-site.xml

└── Dockerfile

## 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 path

# Ref: https://downloads.apache.org/hadoop/common/

RUN \

wget https://downloads.apache.org/hadoop/common/hadoop-3.3.6/hadoop-3.3.6.tar.gz && \

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" >> /opt/hadoop/etc/hadoop/hadoop-env.sh

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

A screenshot of a computer program

Description automatically generated

**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" >>/opt/hadoop/etc/hadoop/hadoop-env.sh

Nếu máy local là AMD64

RUN echo "export JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64">>/opt/hadoop/etc/hadoop/hadoop-env.sh

* **Kiểm tra máy local có CPU kiến trúc nào?**

**Trên Linux hoặc macOS**

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**

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

wmic os get osarchitecture

* + x86\_64: Kiến trúc AMD64
  + aarch64: Kiến trúc ARM64

A black background with white text

Description automatically generated

## Build docker image

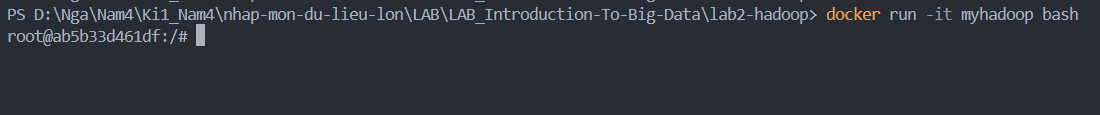
docker build -t myhadoop .

A screenshot of a computer

Description automatically generated

## Run docker image

docker run -it myhadoop bash



hadoop version (Kiểm tra version hadoop)

A screen shot of a computer code

Description automatically generated

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

bash start-hadoop.sh

A screenshot of a computer screen

Description automatically generated

Sau khi chạy xong, kiểm tra các file trên dfs

hdfs dfs -ls /

A screen shot of a computer

Description automatically generated

# LỆNH HADOOP

## Kiểm tra các file trên dfs

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

hdfs dfs -put -f <source\_file> <target\_file>

A screen shot of a computer

Description automatically generated

## Set replica

hdfs dfs -setrep <num\_replica> <filename>

hdfs dfs -stat %r <filename>

A screenshot of a computer program

Description automatically generated

# WORDCOUNT

## Tạo thư mục lab2

mkdir lab2 && cd lab2



## 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

A screen shot of a computer

Description automatically generated

## Tải file iuh.txt lên dfs

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

## Tạo file mapper – tách nội dung của file thành từng từ theo khoảng trắng và đếm 1 cho mỗi lẫn 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

cat iuh.txt | python3 mapper.py

A computer screen shot of a program code

Description automatically generated

## Cấp quyền thực thi cho file mapper

chmod a+x mapper.py

A screenshot of a computer

Description automatically generated

## Thực thi lại

cat iuh.txt | ./mapper.py

A screen shot of a computer

Description automatically generated

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

cat iuh.txt | ./mapper.py | sort -k1,1

A computer screen shot of a black screen

Description automatically generated

## 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

A screen shot of a computer

Description automatically generated

## Kiểm tra file reducer chạy được hay không

cat iuh.txt | ./mapper.py | sort -k1,1 | ./reducer.py

A screen shot of a computer

Description automatically generated

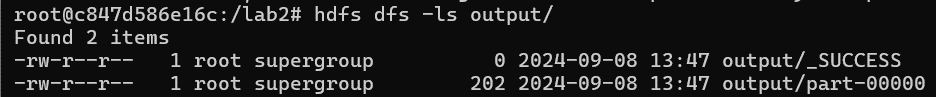
## Thực thi mapper vs reducer với hadoop

hadoop jar <jar\_file> -input <file\_in\_dfs> -output <dir\_output> -mapper <file\_mapper> -reducer <file\_reducer>

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

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

hdfs dfs -ls output/



Xem kết quả ở file output part-0000

hdfs dfs -cat output/part-00000

A screen shot of a computer

Description automatically generated

**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.jar

Hoặc thực hiện trực tiếp

hadoop jar /opt/hadoop/share/hadoop/tools/lib/hadoop-streaming-3.3.6.jar -input /iuh.txt -output output