



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.04.01 Информатика и вычислительная техника

МАГИСТЕРСКАЯ ПРОГРАММА 09.04.01/07 Интеллектуальные системы анализа,
обработки и интерпретации больших данных

О Т Ч Е Т

по лабораторным работам № 7-8

Название: Работа с Hadoop и Spark

Дисциплина: Технология параллельных систем баз данных

Студент

ИУ6-12М

(Группа)

(Подпись, дата)

С.В. Астахов

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

(И.О. Фамилия)

Москва, 2023

Введение

1. Цель работы: приобретение навыков инсталляции и работы с продуктами Apache Hadoop и Apache Spark, поддерживающих технологию MapReduce. Эти системы используются для обработки больших данных (Big Data).

Ход выполнения

2. Установка Hadoop и Spark.

Для упрощения процесса установки воспользуемся docker-контейнерами с hadoop и spark. Склонируем репозиторий с описанием конфигурации docker-compose “git clone git@github.com:Marcel-Jan/docker-hadoop-spark.git”. Конфигурация системы контейнеров приведена в листинге 1 (некоторые контейнеры, такие как hive server, опущены для краткости, т.к. не используются в данной работе).

Листинг 1 — файл docker-compose.yml

```
version: "3"

services:
  namenode:
    image: bde2020/hadoop-namenode:2.0.0-hadoop3.2.1-java8
    container_name: namenode
    restart: always
    ports:
      - 9870:9870
      - 9010:9000
    volumes:
      - hadoop_namenode:/hadoop/dfs/name
    environment:
      - CLUSTER_NAME=test
      - CORE_CONF_fs_defaultFS=hdfs://namenode:9000
    env_file:
      - ./hadoop.env

  datanode:
    image: bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8
    container_name: datanode
    restart: always
    volumes:
      - hadoop_datanode:/hadoop/dfs/data
    environment:
      SERVICE_PRECONDITION: "namenode:9870"
      CORE_CONF_fs_defaultFS: hdfs://namenode:9000
    ports:
```

```

- "9864:9864"
env_file:
- ./hadoop.env

resourcemanager:
image: bde2020/hadoop-resourcemanager:2.0.0-hadoop3.2.1-java8
container_name: resourcemanager
restart: always
environment:
  SERVICE_PRECONDITION: "namenode:9000 namenode:9870 datanode:9864"
env_file:
- ./hadoop.env

nodemanager1:
image: bde2020/hadoop-nodemanager:2.0.0-hadoop3.2.1-java8
container_name: nodemanager
restart: always
environment:
  SERVICE_PRECONDITION: "namenode:9000          namenode:9870          datanode:9864
resourcemanager:8088"
env_file:
- ./hadoop.env

spark-master:
image: bde2020/spark-master:3.0.0-hadoop3.2
container_name: spark-master
depends_on:
- namenode
- datanode
ports:
- "8080:8080"
- "7077:7077"
environment:
- INIT_DAEMON_STEP=setup_spark
- CORE_CONF_fs_defaultFS=hdfs://namenode:9000

spark-worker-1:
image: bde2020/spark-worker:3.0.0-hadoop3.2
container_name: spark-worker-1
depends_on:
- spark-master
ports:
- "8081:8081"
environment:
- "SPARK_MASTER=spark://spark-master:7077"
- CORE_CONF_fs_defaultFS=hdfs://namenode:9000

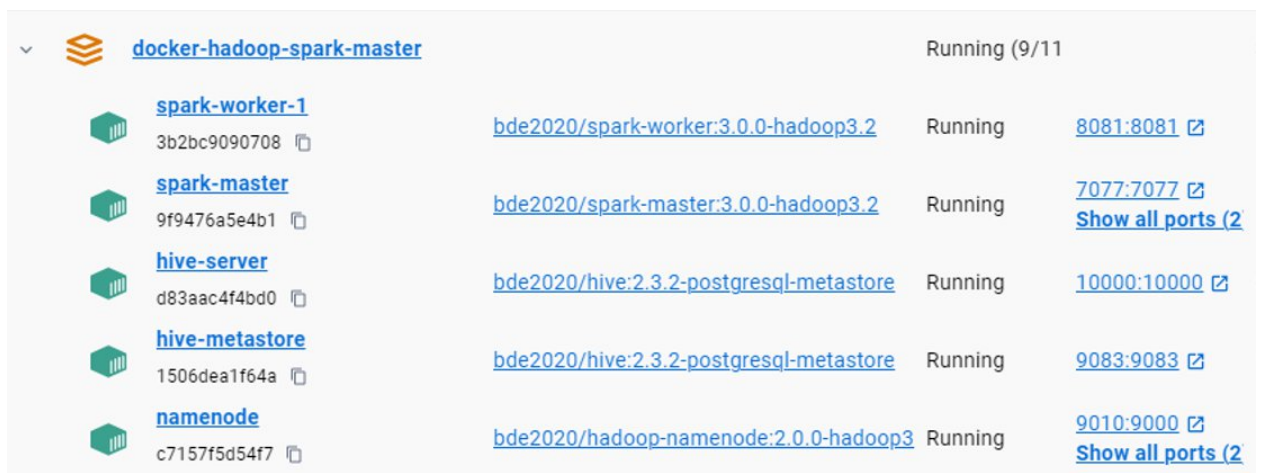
volumes:
hadoop_namenode:
hadoop_datanode:
hadoop_historyserver:

```

Данная конфигурация описывает соответствие портов на хост-машине и внутри виртуальной сети docker, используемые носители, а так же дополнительные параметры для контейнеров. Назначение основных контейнеров в конфигурации:

- namenode — хранит имена файлов и расположение их сегментов;
- datanode — хранит сегменты файлов;
- resource manager и node manager — управляют ресурсами всего кластера и отдельных узлов соответственно;
- spark-master — управляет задачами spark;
- spark-worker — выполняет задачи spark.

Перейдя в папку с файлом конфигурации, запустим контейнеры командой “docker-compose up” (рисунок 1). Примечание: некоторые контейнеры не уместились на одном скриншоте.



The screenshot shows the Docker Desktop interface with a list of running containers under the 'docker-hadoop-spark-master' project. The containers are: spark-worker-1, spark-master, hive-server, hive-metastore, and namenode. Each container is in a 'Running' state. The table below summarizes the information visible in the screenshot.

Container Name	Image	Status	Ports
spark-worker-1	bde2020/spark-worker:3.0.0-hadoop3.2	Running	8081:8081
spark-master	bde2020/spark-master:3.0.0-hadoop3.2	Running	7077:7077
hive-server	bde2020/hive:2.3.2-postgresql-metastore	Running	10000:10000
hive-metastore	bde2020/hive:2.3.2-postgresql-metastore	Running	9083:9083
namenode	bde2020/hadoop-namenode:2.0.0-hadoop3	Running	9010:9000, 9870:9870

Рисунок 1 — запущенные контейнеры

Проверим работу hadoop с помощью команды jps (рисунок 2). Hadoop работает корректно.

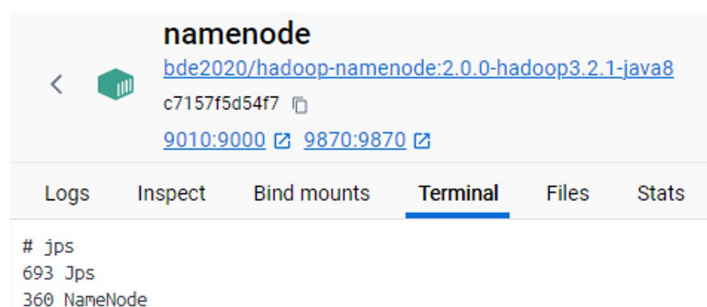


Рисунок 2 — проверка работы hadoop

Проверим работу веб-интерфейсов hadoop и spark (рисунки 3, 4).

Hadoop	Overview	Datanodes	Datanode Volume Failures	Snapshot	Startup Progress
Overview 'namenode:9000' (active)					
Started:	Wed Nov 29 12:40:14 +0300 2023				
Version:	3.2.1, rb3cbbb467e22ea829b3808f4b7b01d07e0bf3842				
Compiled:	Tue Sep 10 18:56:00 +0300 2019 by rohithsharmaks from branch-3.2.1				
Cluster ID:	CID-d76bb543-b5be-4fe9-8f6c-241ebc74effb				
Block Pool ID:	BP-1755031343-172.20.0.9-1701244642935				

Рисунок 3 — веб-интерфейс hadoop

← → ↺ ⓘ localhost:8080

bMarks root Bookmarks 2023 Курс: Теоретическ... (PDF) Attention is A... IET A Sur

Spark 3.0.0 **Spark Master at spark://9f9476a5e4b1:7077**

URL: spark://9f9476a5e4b1:7077
Alive Workers: 1
Cores in use: 8 Total, 0 Used
Memory in use: 2.7 GiB Total, 0.0 B Used
Resources in use:
Applications: 0 Running, 0 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers (1)

Worker Id
worker-20231129094033-172.20.0.7-32927

Рисунок 4 — веб-интерфейс spark

3. Работа с hadoop и spark

В консоли контейнера spark-master запустим ruyspark и создадим датафрейм из пар чисел с их названиями (рисунок 5).


```
spark-master
bde2020/spark-master:3.0.0-hadoop3.2
9f9476a5e4b1
7077:7077 8080:8080

Logs Inspect Bind mounts Terminal Files Stats

>>>
>>>
>>>
>>> quit()
/ # spark/bin/spark-shell --master spark://spark-master:7077
23/11/29 09:51:40 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://9f9476a5e4b1:4040
Spark context available as 'sc' (master = spark://spark-master:7077, app id = app-20231129095149-0001).
Spark session available as 'spark'.
Welcome to

      _/ _/ _/ _/ _/ _/ _/ _/
     /_/_/_/_/_/_/_/_/_/_/_/_/
    /_/_/_/_/_/_/_/_/_/_/_/_/
   /_/_/_/_/_/_/_/_/_/_/_/_/
  /_/_/_/_/_/_/_/_/_/_/_/_/
 /_/_/_/_/_/_/_/_/_/_/_/_/
/_/_/_/_/_/_/_/_/_/_/_/_/

version 3.0.0

Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 1.8.0_252)
Type in expressions to have them evaluated.
Type :help for more information.

scala> :q
/ #
```

Рисунок 7 — проверка консоли scala

Скопируем файл с пьесой «Гамлет» с хост-машины в контейнер (рисунок 8).

```
C:\Users\sergey.astakhov\Desktop\Bmstu-M1\db\lab78>docker cp gamlet_en.txt namenode:/lab78x/gamlet_en.txt
Successfully copied 190kB to namenode:/lab78x/gamlet_en.txt

C:\Users\sergey.astakhov\Desktop\Bmstu-M1\db\lab78>
```

Рисунок 8 — копирование файла в контейнер

Внутри контейнера скопируем файл в hdfs и проверим его наличие в hdfs (рисунок 9).

```
# hdfs dfs -mkdir /lab78
# mkdir /lab78x
# hdfs dfs -put /lab78x/gamlet_en.txt /chapter5
2023-11-29 09:58:16,265 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false
# hdfs dfs -ls /chapter5
Found 2 items
drwxr-xr-x - root supergroup 0 2023-11-29 09:48 /chapter5/example.csv
-rw-r--r-- 3 root supergroup 188041 2023-11-29 09:58 /chapter5/gamlet_en.txt
#
```

Рисунок 9 — копирование файла в hdfs

Войдем в консоль `ruspark` и выполним скрипт подсчета слов (рисунок 10). Программа выполняет следующие шаги:

- импорт библиотек, чтение файла из hdfs;
- разбиение файла на слова (метод flatMap);
- каждому слову ставится в соответствие значение 1 (метод map);
- для каждого уникального слова-ключа значения складываются между собой (reduceByKey);
- сохранение расчетов в файл;
- сумма повторений всех слов рассчитывается и выводится на экран.

Welcome to

[illegible]

```
Using Python version 2.7.18 (default, May 3 2020 22:58:12)
SparkSession available as 'spark'.
>>> from pyspark import SparkContext
>>> from datetime import datetime
>>> f = sc.textFile("hdfs://namenode:9000/chapter5/gamlet_en.txt")
>>> counts = f.flatMap(lambda line:line.split(" ")) \
...   .map(lambda word: (word , 1)) \
...   .reduceByKey(lambda a, b: a + b) \
...   .sortBy(lambda a: a[1], ascending=False)
>>> counts.saveAsTextFile("/home/hduser/res/")
>>> sm = counts.map(lambda x: x[1]).sum()
>>> print ("Summa %d" % (sm))
Summa 33140
>>> quit()
```

Рисунок 10 — подсчет слов

На рисунке 11 показано содержимое RDD counts.


```
spark-master
bde2020/spark-master:3.0.0-hadoop3.2
9f9476a5e4b1
7077:7077 8080:8080

Logs Inspect Bind mounts Terminal Files Stats
Open in external terminal

>>> print(counts.collect())
[(u'the', 995), (u'and', 701), (u'of', 641), (u'to', 606), (u'I', 511), (u'a', 449), (u'my', 444), (u'in', 385), (u'you', 363), (u'Ham', 358), (u'is', 297), (u'his', 281), (u'it', 268), (u'not', 255), (u'And', 250), (u'J', 244), (u'that', 225), (u'your', 224), (u'with', 222), (u'this', 203), (u'Hamlet', 201), (u'be', 186), (u'William', 177), (u'Shakespeare', 176), (u'The', 174), (u'OriginalBook.Ru', 172), (u'for', 168), (u'he', 162), (u'have', 159), (u'as', 144), (u'but', 144), (u'me', 142), (u'will', 132), (u'by', 125), (u'hin', 122), (u'are', 117), (u'To', 116), (u'That', 112), (u'King', 111), (u'Hor.', 110), (u'what', 108), (u'our', 107), (u'do', 106), (u'so', 105), (u'we', 102), (u'on', 102), (u'shall', 101), (u'no', 90), (u'all', 86), (u'Pol.', 86), (u'A', 84), (u'But', 84), (u'from', 84), (u'thou', 82), (u'thy', 81), (u'lord', 79), (u'or', 78), (u'Queen.', 77), (u'they', 76), (u'at', 76), (u'what', 75), (u'As', 72), (u'her', 70), (u'more', 70), (u'most', 70), (u'My', 68), (u'if', 68), (u'like', 68), (u'was', 67), (u'good', 67), (u'you', 64), (u'would', 63), (u'For', 63), (u'Laer.', 62), (u'[Enter', 61), (u'may', 58), (u'let', 58), (u'Oph.', 58), (u'lord.', 57), (u'very', 57), (u'O', 57), (u'I'll', 56), (u'How', 56), (u'know', 56), (u'It', 55), (u'hath', 54), (u'must', 54), (u'their', 54), (u'some', 52), (u'Ros.', 51), (u'should', 51), (u'am', 50), (u'an', 50), (u'You', 49), (u'come', 48), (u'This', 47), (u'did', 46), (u'O', 46), (u'us', 46), (u'such', 46), (u'Clown.', 46), (u'tis', 45), (u'then', 44), (u'nake', 44), (u'much', 43), (u'With', 42), (u'Of', 42), (u'love', 41), (u'If', 41), (u'He', 40), (u'there', 39), (u'out', 39), (u'had', 38), (u'So', 38), (u'upon', 38), (u'give', 38), (u'think', 38), (u'than', 38), (u'1', 37), (u'which', 36), (u'these', 36), (u'tell', 36), (u'speak', 36), (u'In', 36), (u'own', 36), (u'Mar.', 35), (u'go', 35), (u'now', 35), (u'thee', 34), (u'Let', 34), (u'it', 34), (u'Guil.', 34), (u'she', 34), (u'ay', 34), (u'me', 33), (u'too', 32), (u'Or', 32), (u'man', 32), (u'Hamlet', 30), (u'say', 30), (u'we', 30), (u'when', 29), (u'see', 29), (u'time', 29), (u'can', 29), (u'one', 29), (u'how', 28), (u'father', 27), (u'into', 27), (u'sir', 27), (u'might', 27), (u'lord?', 27), (u'take', 27), (u'here', 26), (u'dear', 26), (u'why', 26), (u'where', 26), (u'well', 26), (u'king', 26), (u'up', 26), (u'made', 26), (u'been', 25), (u'when', 25), (u'is', 25), (u'osr.', 25), (u'hin', 25), (u'could', 25), (u'Tis', 25), (u'yet', 25), (u'it', 24), (u'great', 24), (u'No', 24), (u'were', 24), (u'cannot', 24), (u'where', 23), (u'hin', 23), (u'mine', 23), (u'doth', 23), (u'now', 22), (u'Nay.', 22), (u'hear', 22), (u'put', 22), (u'i", 22), (u'whose', 22), (u'Do', 22), (u'[Exeunt', 22), (u'you', 22), (u'But', 22), (u'does', 22), (u'Hamlet.', 21), (u'Polonius.', 21), (u'nor', 21), (u'call', 21), (u'thus', 21), (u'hold', 21), (u'Ber.', 21), (u'me', 21), (u'Good', 21), (u'those', 20), (u'poor', 20), (u'itself', 20), (u'Scene', 20), (u'Ghost.', 20), (u'Give', 20), (u'set', 20), (u'look', 20), (u'pray', 19), (u'[Exit', 19), (u'which', 19), (u'[Exit.', 19), (u'who', 19), (u'show', 19), (u'against', 18), (u'old', 18), (u'leave', 18), (u'father's', 18), (u'comes', 18), (u'Laertes', 18), (u'then', 18), (u'Then', 18), (u'why', 18), (u'not', 18), (u'part', 18), (u'both', 18), (u'sweet', 18), (u'Will', 17), (u'They', 17), (u'dead', 17), (u'head', 17), (u'that', 17), (u'play', 17), (u'nothing', 17), (u'Hls', 17), (u'Come', 17), (u'seen', 17), (u'soul', 17), (u'many', 17), (u'I', 17), (u'noble', 17), (u'world', 17), (u'No', 17), (u'Your', 17), (u'find', 17), (u'By', 17), (u'Horatio', 16), (u'Nor', 16), (u'though', 16), (u'never', 16), (u'heaven', 16), (u'two', 16), (u'King', 16), (u'A', 16), (u'keep', 16), (u'this', 16), (u'then', 15), (u'fair', 15), (u'little', 15), (u'Rosencrantz', 15), (u'Have', 15), (u'Not', 15), (u'Upon', 15), (u'Guild
```

Рисунок 11 — содержимое counts

Вывод: в ходе работы были приобретены навыки инсталляции и работы с продуктами Apache Hadoop и Apache Spark, поддерживающих технологию MapReduce. Эти системы используются для обработки больших данных (Big Data).