Семинар №2 SQL и BigData

1. Инструментарий:

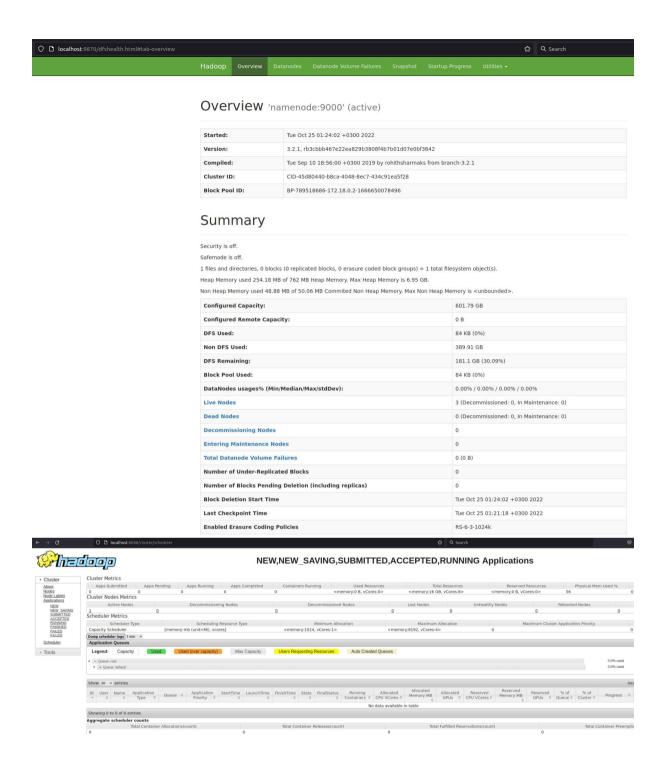
<u>Лекция</u> <u>Презентация</u>

Задание:

- 1) Для работы нужно установить дистрибутив <u>Docker</u>:
 - https://docs.docker.com/get-docker/
 - O https://docs.docker.com/desktop/install/windows-install/
 - O https://docs.docker.com/desktop/install/mac-install/
 - O https://docs.docker.com/desktop/install/linux-install/
- 2) Выполняем команду docker compose up:

```
| COMMANID | COMMANID
```

3) Наблюдаем и изучаем настройки и состояние NM и RM в вебинтерфейсе



HDFS

Intro

Oсновные команды hdfs: https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsUserGuide

Команды для работы с файлами:

https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/FileSystemShell.html

Задание:

- 1) Загрузите датасет, доступный на kaggle.com: https://www.kaggle.com/pieca111/music-artists-popularity
- 2) Положите его в hdfs
- 3) Выяснить сколько места занимают файл (hdfs dfs -du -h)
- 4) Изменить фактор репликации (hdfs dfs -setrep <path>)
- 5) Удалить файл (hdfs dfs -rm <file>). При удалении файл перемещается в корзину (~/.Trash). При этом в корзине воссоздаётся абсолютный путь к файлу (.../.Trash/user/your_user/your_dir/your_file). Для удаления в обход корзины можно использовать флаг --skipTrash

Пример решения:

```
sudo docker exec -it namenode bash
hdfs dfs -mkdir /my_dir
hdfs dfs -touchz /my_dir/another_dir/emptyfile
hdfs dfs -rm -r /my_dir
dfs dfs -copyFromLocal example.txt /
hdfs dfs -head /error.txt
hdfs dfs -tail /error.txt
hdfs dfs -cat /error.txt
```

Hive

Задание:

- 1) Сделать таблицу artists в Hive и вставить туда значения, используя датасет https://www.kaggle.com/pieca111/music-artists-popularity
- 2) Используя Hive найти
 - а) Исполнителя с максимальным числом скробблов
 - b) Самый популярный тэг на ластфм
 - c) Самые популярные исполнители 10 самых популярных тегов ластфм

d)

Пример решения:

MapReduce

Задание:

- 1) Загрузите датасет по ценам на жилье Airbnb, доступный на kaggle.com: https://www.kaggle.com/dgomonov/new-york-city-airbnb-open-data
- 2) Используя Python, реализуйте скрипт mapper.py и reducer.py для расчета. В итоге у вас должно получиться 2 скрипта: mapper и reducer для каждой величины.

Пример решения:

```
def mapper():
    for line in csv.reader(sys.stdin):
        if len(line[9]) > 0 and line[9] != 'price':
            sys.stdout.write(f'{line[9]}\t1\n')

def reduce():
    s = 0
    counter = 0
    for line in sys.stdin:
        vals = line.strip().split("\t")
        s += float(vals[0])
        counter += 1

sys.stdout.write("{}\n".format(s / counter))

root@fb82701789af:/# hdfs dfs -cat /block3/output/part-00000
152.7206871868289
```

2. Домашнее задание

Условие:

- 1) Загрузите датасет по ценам на жилье Airbnb, доступный на kaggle.com: https://www.kaggle.com/dgomonov/new-york-city-airbnb-open-data
- 2) Подсчитайте среднее значение и дисперсию по признаку "price" в hive
- 3) Используя Python, реализуйте скрипт mapper.py и reducer.py для расчета
- 4) Проверьте правильность подсчета статистики методом mapreduce в сравнении со hive.