



Министерство науки и высшего образования Российской Федерации
Калужский филиал федерального государственного автономного
образовательного учреждения высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУК Информатика и управление

КАФЕДРА ИУК4 Программное обеспечение ЭВМ, информационные технологии

ЛАБОРАТОРНАЯ РАБОТА

«ЦЕПОЧКИ MAPREDUCE ЗАДАЧ. СРАВНЕНИЕ ДОКУМЕНТОВ»

по дисциплине: «Технологии обработки больших данных»

Выполнил: студент группы ИУК4-72Б

(Подпись)

Губин Е.В.

(И.О. Фамилия)

Проверил:

(Подпись)

Голубева С.Е.

(И.О. Фамилия)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:

Калуга, 2025

Цель: формирование практических навыков использования цепочек MapReduce для решения сложных задач обработки больших данных.

Задачи:

1. Получить навыки реализации цепочки MapReduce задач.
2. Изучить интерфейс Hadoop MapReduce.
3. Изучить алгоритмы анализа, сравнение текстовых документов.
4. Получить практические навыки обработки и анализа текстовых данных.

Вариант 9

Задание:

Подсчитать среднюю стоимость показа рекламы по городам России и вывести максимальную стоимость.

Входной файл имеет формат:

userId, country, city, campaign_id, creative_id, payment.

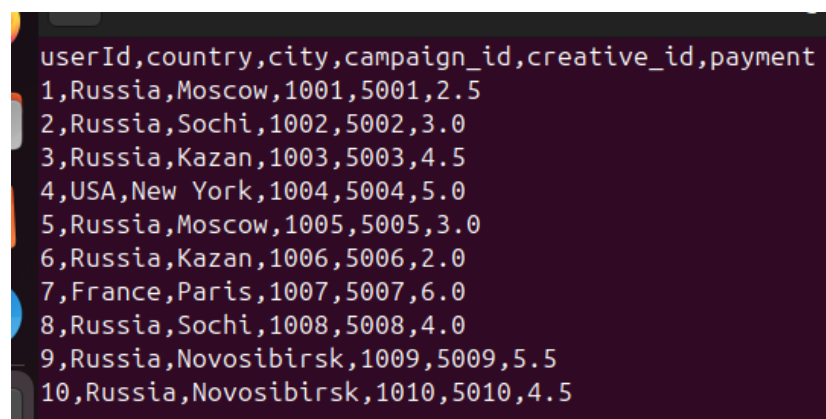
Результат должен быть сохранен в двух файлах:

- 1 файл: city, av_av_payment
- 2 файл (1 запись): max_payment

Ход выполнения лабораторной работы:

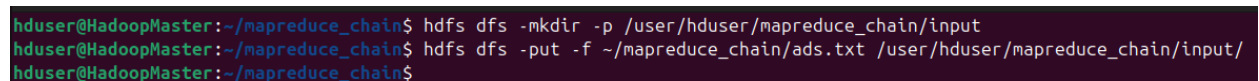
В ходе выполнения работы были реализованы две MapReduce задачи:

- MR1 выполняет map-reduce агрегацию — вычисляет среднее значение payment для каждого города, но только если country=Russia.
- MR2 берёт результаты первого MR и находит максимальное значение среди всех средних — это "максимальная стоимость показа рекламы".



```
userId, country, city, campaign_id, creative_id, payment
1, Russia, Moscow, 1001, 5001, 2.5
2, Russia, Sochi, 1002, 5002, 3.0
3, Russia, Kazan, 1003, 5003, 4.5
4, USA, New York, 1004, 5004, 5.0
5, Russia, Moscow, 1005, 5005, 3.0
6, Russia, Kazan, 1006, 5006, 2.0
7, France, Paris, 1007, 5007, 6.0
8, Russia, Sochi, 1008, 5008, 4.0
9, Russia, Novosibirsk, 1009, 5009, 5.5
10, Russia, Novosibirsk, 1010, 5010, 4.5
```

Рисунок 1 Входной файл



```
hduser@HadoopMaster:~/mapreduce_chain$ hdfs dfs -mkdir -p /user/hduser/mapreduce_chain/input
hduser@HadoopMaster:~/mapreduce_chain$ hdfs dfs -put -f ~/mapreduce_chain/ads.txt /user/hduser/mapreduce_chain/input/
hduser@HadoopMaster:~/mapreduce_chain$
```

Рисунок 2 Загружаем файл в файловую систему HDFS

```

hduser@HadoopMaster:~/mapreduce_chain$ mv ~/Downloads/mapper1.py ./
hduser@HadoopMaster:~/mapreduce_chain$ mv ~/Downloads/mapper2.py ./
hduser@HadoopMaster:~/mapreduce_chain$ mv ~/Downloads/reducer2.py ./
hduser@HadoopMaster:~/mapreduce_chain$ mv ~/Downloads/reducer1.py ./
hduser@HadoopMaster:~/mapreduce_chain$ mv ~/Downloads/run_chain.sh ./
hduser@HadoopMaster:~/mapreduce_chain$ ls'
> ^C
hduser@HadoopMaster:~/mapreduce_chain$ ls
ads.txt mapper1.py mapper2.py reducer1.py reducer2.py run_chain.sh
hduser@HadoopMaster:~/mapreduce_chain$ chmod +x mapper1.py mapper2.py reducer1.py reducer2.py run_chain.sh
hduser@HadoopMaster:~/mapreduce_chain$

```

Рисунок 3 Делаем все программные файлы исполняемыми

```

Kazan    3.25
Moscow   2.75
Novosibirsk    5.00
Sochi    3.50
~
~

```

Рисунок 4 Средняя стоимость рекламы по городам

```

5.00
~
~
~
~
~

```

Рисунок 5 Максимальная средняя стоимость рекламы среди средних значений стоимости по городам

Листинги программ:

mapper1.py:

```

#!/usr/bin/env python3
import sys
for line in sys.stdin:
    line = line.strip()
    if line.startswith("userId"):
        continue
    parts = line.split(',')
    if len(parts) < 6:
        continue
    userId, country, city, campaign_id, creative_id, payment = parts
    if country.strip().lower() == 'russia':
        try:
            print(f"{city.strip()}\t{float(payment)}")
        except:
            continue

```

mapper2.py:

```
#!/usr/bin/env python3
import sys
for line in sys.stdin:
    line = line.strip()
    if not line:
        continue
    city, avg = line.split('\t')
    try:
        print(f"max\t{float(avg)}")
    except:
        continue
```

reducer1.py:

```
#!/usr/bin/env python3
import sys

current_city = None
total = 0.0
count = 0

for line in sys.stdin:
    line = line.strip()
    if not line:
        continue
    city, payment = line.split('\t', 1)
    payment = float(payment)
    if current_city is None:
        current_city = city
    if city != current_city:
        print(f"{current_city}\t{total / count:.2f}")
        current_city = city
        total = payment
        count = 1
    else:
        total += payment
        count += 1

if current_city:
    print(f"{current_city}\t{total / count:.2f}")
```

reducer2.py:

```
#!/usr/bin/env python3
import sys

max_value = None
for line in sys.stdin:
    line = line.strip()
    if not line:
        continue
    key, val = line.split('\t', 1)
    try:
        val = float(val)
        if max_value is None or val > max_value:
            max_value = val
    except:
        continue
```

```
if max_value is not None:
    print(f"{max_value:.2f}")
```

run_chain.sh:

```
#!/bin/bash
set -e

hdfs dfs -rm -r -f /user/hduser/mapreduce_chain/output1
/user/hduser/mapreduce_chain/output2 || true

STREAMING_JAR=$(ls $HADOOP_HOME/share/hadoop/tools/lib/hadoop-streaming*.jar
| head -n1)

echo "=== MR1: Средняя стоимость по городам России ==="
hadoop jar "$STREAMING_JAR" \
  -files mapper1.py, reducer1.py \
  -input /user/hduser/mapreduce_chain/input \
  -output /user/hduser/mapreduce_chain/output1 \
  -mapper "python3 mapper1.py" \
  -reducer "python3 reducer1.py"

echo "=== MR2: Максимальная средняя стоимость ==="
hadoop jar "$STREAMING_JAR" \
  -files mapper2.py, reducer2.py \
  -input /user/hduser/mapreduce_chain/output1/part-00000 \
  -output /user/hduser/mapreduce_chain/output2 \
  -mapper "python3 mapper2.py" \
  -reducer "python3 reducer2.py"

echo "=== Выгрузка результатов ==="
mkdir -p ~/mapreduce_chain/first_task ~/mapreduce_chain/second_task
hdfs dfs -get -f /user/hduser/mapreduce_chain/output1/part-00000
~/mapreduce_chain/first_task/city_avg.txt
hdfs dfs -get -f /user/hduser/mapreduce_chain/output2/part-00000
~/mapreduce_chain/second_task/max_payment.txt
```

Вывод: в ходе лабораторной работы была реализована цепочка MapReduce задач.