|  |  |
| --- | --- |
|  | **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 | |  |  | Губин Е.В. | |
|  | | (Подпись) |  | (И.О. Фамилия) | |
| Проверил: | |  |  | Голубева С.Е. | |
|  | | (Подпись) |  | (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 MapReuce 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".

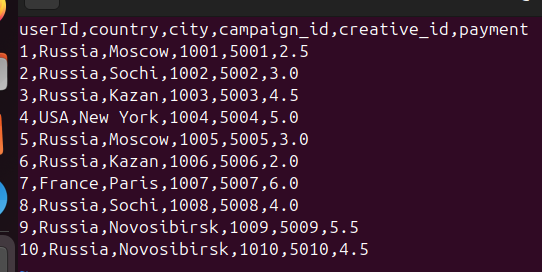


Рисунок 1 Входной файл

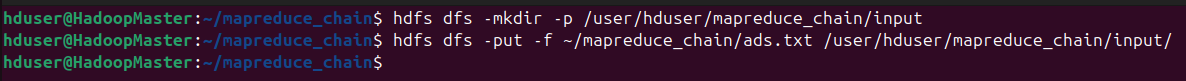


Рисунок 2 Загружаем файл в файловую систему HDFS

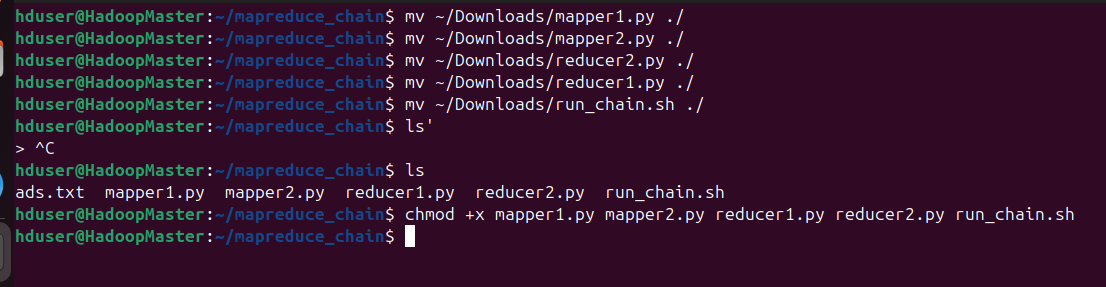


Рисунок 3 Делаем все программыне файлы исполняемыми

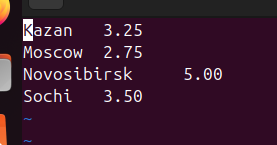


Рисунок 4 Средняя стоимость рекламы по городам

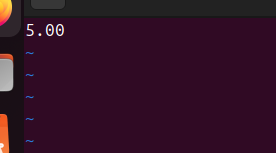


Рисунок 5 Максимальная средняя стоимость рекламы среди средних значений стоимости по городам

**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.