



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

«MAPREDUCE»

по дисциплине: «Технологии обработки больших данных»

Выполнил: студент группы ИУК4-72Б

(Подпись)

Моряков В.Ю.

(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 парадигмы MapReduce для обработки больших данных.

Задачи:

1. Изучить подход MapReduce.
2. Изучить принципы работы Hadoop MapReduce.
3. Получить практические навыки реализации MapReduce задач.
4. Уметь обрабатывать большие текстовые файлы с помощью MapReduce.

Формулировка задания (17 вариант):

Подсчитать среднюю длину слов в нескольких файлах. Результат должен содержать средний размер слов в файле и соответствующее название файла. Сохранить результат в файл в виде:

(7@file1 6@file1 13@file2 22@file2 ...)

Ход выполнения:

```
1 <?xml version="1.0"?>
2 <configuration>
3   <property>
4     <name>mapreduce.framework.name</name>
5     <value>yarn</value>
6   </property>
7
8   <property>
9     <name>yarn.app.mapreduce.am.env</name>
10    <value>HADOOP_MAPRED_HOME=/opt/hadoop</value>
11  </property>
12
13  <property>
14    <name>mapreduce.map.env</name>
15    <value>HADOOP_MAPRED_HOME=/opt/hadoop</value>
16  </property>
17
18  <property>
19    <name>mapreduce.reduce.env</name>
20    <value>HADOOP_MAPRED_HOME=/opt/hadoop</value>
21  </property>
22 </configuration>
23
```

Рисунок 1 Изменение конфигурации MAPReduce hadoop

```

        Reduce output records=2
        Spilled Records=6158
        Shuffled Maps =3
        Failed Shuffles=0
        Merged Map outputs=3
        GC time elapsed (ms)=274
        CPU time spent (ms)=3040
        Physical memory (bytes) snapshot=1140568064
        Virtual memory (bytes) snapshot=10858143744
        Total committed heap usage (bytes)=1226833920
        Peak Map Physical memory (bytes)=286351360
        Peak Map Virtual memory (bytes)=2714066944
        Peak Reduce Physical memory (bytes)=287854592
        Peak Reduce Virtual memory (bytes)=2717687808
    Shuffle Errors
        BAD_ID=0
        CONNECTION=0
        IO_ERROR=0
        WRONG_LENGTH=0
        WRONG_MAP=0
        WRONG_REDUCE=0
    File Input Format Counters
        Bytes Read=22990
    File Output Format Counters
        Bytes Written=70
2025-10-08 05:48:30 INFO  StreamJob:1029 - Output directory: /output
✓ Результат:
5@hdfs://namenode/input/file1.txt
5@hdfs://namenode/input/file2.txt

```

Рисунок 2 Результаты работы программы

```

<> mapred-site.xml X  run_it_docker.sh X

1 bash /init-hdfs.sh
2 bash /create_files.sh
3
4 chmod +x /mapper_avg_wordlen.py /reducer_avg_wordlen.py
5
6 hadoop jar /opt/hadoop/share/hadoop/tools/lib/hadoop-streaming-*.jar \
7 -files /mapper_avg_wordlen.py,/reducer_avg_wordlen.py \
8 -input /input \
9 -output /output \
10 -mapper mapper_avg_wordlen.py \
11 -reducer reducer_avg_wordlen.py
12
13
14 echo "✓ Результат:"
15 hdfs dfs -cat /output/part-00000

```

Рисунок 3 Скрипт для запуска

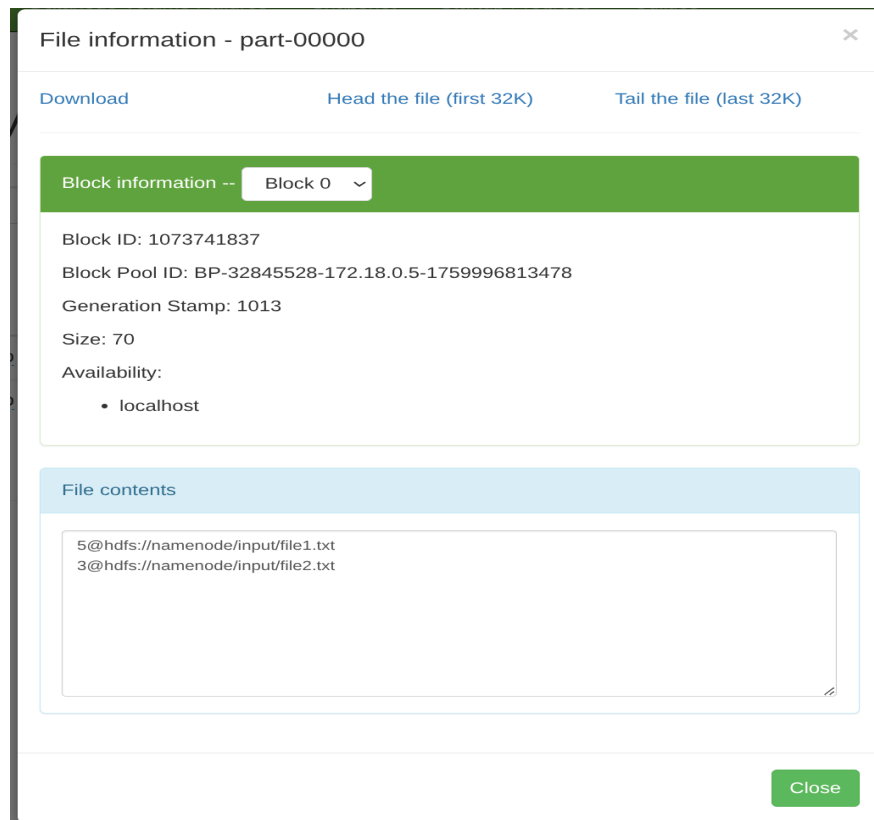


Рисунок 4 Результаты работы в графическом интерфейсе

Листинги программ:

mapper_avg_word_len.py:

```
#!/usr/bin/env python
# mapper_avg_wordlen.py (Python 2)

import sys
import os
import re

for line in sys.stdin:
    line = line.strip()
    words = re.findall(r'\w+', line)
    if not words:
        continue
    filename = os.environ.get('map_input_file', 'unknown')
    for word in words:
        print "{0}\t{1}".format(filename, len(word))
```

reducer_avg_wordlen.py:

```
#!/usr/bin/env python
# reducer_avg_wordlen.py (Python 2)

import sys
```

```
current_file = None
total_len = 0
word_count = 0
for line in sys.stdin:
    line = line.strip()
    filename, length = line.split('\t')
    length = int(length)

    if current_file and filename != current_file:
        avg = float(total_len) / word_count if word_count else 0
        print "{0}@{1}".format(int(avg), current_file)
        total_len = 0
        word_count = 0
    current_file = filename
    total_len += length
    word_count += 1
if current_file:
    avg = float(total_len) / word_count if word_count else 0
    print "{0}@{1}".format(int(avg), current_file)
```

Вывод: в ходе лабораторной работы были получены практические навыки по работе с MapReduce.