Выполнил: Бойко Илья Группа: ПИН-б-о-22-1

1. Цель работы Освоить модель программирования MapReduce на практике. Получить навыки реализации алгоритмов обработки данных с использованием парадигмы MapReduce на языке Python для запуска в Hadoop Streaming.

2. Используемый стек технологий Язык программирования: Python 3.8.10

Фреймворк: Hadoop Streaming

Платформа: Ubuntu Linux 20.04 LTS

Библиотеки: стандартные библиотеки Python

3. Теоретические сведения MapReduce — это модель и фреймворк для параллельной обработки больших наборов данных на кластере. Вычисления разделяются на две основные фазы:

Map (Отображение) — обработка входных данных и генерация промежуточных пар ключ-значение

Reduce (Свёртка) — агрегация промежуточных значений по ключам

Shuffle & Sort — неявный этап группировки значений по ключам

4. Ход выполнения работы

- 4.1. Подготовка окружения bash

Установка Python

sudo apt install python3 python3-pip

Установка Java для Hadoop

sudo apt install openjdk-8-jdk

Настройка переменных окружения

export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64 export HADOOP_HOME=~/hadoop

- 4.2. Создание тестовых данных
- Файл input.txt:

hello world hello data goodbye data big data analytics hello hadoop mapreduce programming

- Файл grades.txt:

101,math,92 101,physics,85 102,math,90 102,physics,95 103,math,80 103,physics,88 104,math,75 104,physics,82

- 4.3. Реализация задачи 1: WordCount
- Код mapper_wc.py:

#!/usr/bin/env python3 import sys

def main(): for line in sys.stdin: line = line.strip() words = line.split() for word in words: print(f"{word}\t1")

if **name** == "**main**": main()

- Код reducer_wc.py:

#!/usr/bin/env python3 import sys

def main(): current_word = None current_count = 0

```
for line in sys.stdin:
    line = line.strip()
    word, count = line.split('\t', 1)

    try:
        count = int(count)
    except ValueError:
        continue

    if current_word == word:
        current_count += count
    else:
        if current_word:
            print(f"{current_word}\t{current_count}")
        current_word = word
        current_count = count

if current_word:
    print(f"{current_word}\t{current_count}")
```

if **name** == "**main**": main()

- 4.4. Реализация задачи 2: Средняя оценка студентов
- Код mapper_avg.py:

#!/usr/bin/env python3 import sys

def main(): for line in sys.stdin: line = line.strip() if not line: continue

```
    try:
        student_id, subject, grade = line.split(',')
        grade = float(grade)
        print(f"{student_id}\t{grade}")
    except ValueError:
        continue
```

if **name** == "**main**": main()

- Код reducer_avg.py:

#!/usr/bin/env python3 import sys

def main(): current_student = None grades = []

```
for line in sys.stdin:
    line = line.strip()
    student_id, grade = line.split('\t', 1)

    try:
        grade = float(grade)
    except ValueError:
        continue

    if current_student == student_id:
        grades.append(grade)
    else:
        if current_student and grades:
            average = sum(grades) / len(grades)
            print(f"{current_student}\t{average:.1f}")

        current_student = student_id
        grades = [grade]

if current_student and grades:
    average = sum(grades) / len(grades)
    print(f"{current_student}\t{average:.1f}")
```

if **name** == "**main**": main()

- 4.5. Локальная отладка
- Тестирование WordCount:

cat input.txt | python3 mapper_wc.py | sort | python3 reducer_wc.py

- Результат:

analytics 1 big 1 data 3 goodbye 1 hadoop 1 hello 3 mapreduce 1 programming 1 world 1

- Тестирование средней оценки:

cat grades.txt | python3 mapper_avg.py | sort | python3 reducer_avg.py Результат:

101 88.5 102 92.5 103 84.0 104 78.5

- 4.6. Настройка и запуск Hadoop
- Подготовка HDFS:

Форматирование HDFS

hdfs namenode -format

Запуск Hadoop

start-dfs.sh start-yarn.sh

Создание директорий в HDFS

hadoop fs -mkdir -p /user/student/input_data

Загрузка данных в HDFS

hadoop fs -put input.txt /user/student/input_data/ hadoop fs -put grades.txt /user/student/input_data/

- 4.7. Запуск задач в Hadoop Streaming Запуск WordCount:

```
hadoop jar $HADOOP_HOME/share/hadoop/tools/lib/hadoop-streaming-*.jar
-files mapper_wc.py,reducer_wc.py
-input /user/student/input_data/input.txt
-output /user/student/output_wc
-mapper "python3 mapper_wc.py"
-reducer "python3 reducer_wc.py"
```

- Проверка результатов:

hadoop fs -cat /user/student/output_wc/part-00000

- Запуск вычисления средней оценки:

```
hadoop jar $HADOOP_HOME/share/hadoop/tools/lib/hadoop-streaming-*.jar
-files mapper_avg.py,reducer_avg.py
-input /user/student/input_data/grades.txt
-output /user/student/output_avg
```

-mapper "python3 mapper_avg.py"
-reducer "python3 reducer_avg.py"

- Проверка результатов:

hadoop fs -cat /user/student/output_avg/part-00000 5. Результаты выполнения

- 5.1. Результаты WordCount Слово Количество analytics 1 big 1 data 3 goodbye 1 hadoop 1 hello 3 mapreduce 1 programming 1 world 1