НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Кафедра автоматизації проектування енергетичних процесів і систем

Лабораторна робота № 1

З дисципліни: «Оброблення надвеликих масивів даних»

**Виконав:**

студент 5 курсу

групи ТР-02мп, ТЕФ

Круглий Д.В.

**Перевірила:**

проф. Федорова Н. В.

Київ – 2020

**Тема:** використання шаблонів проектування MapReduce.

**Мета:** створити модель кластеризації даних за допомогою технології MapReduce.

**Теоретичні відомості**

MapReduce – найбільш відома технологія паралельної обробки даних великих розмірностей. Особливістю є розділення логіки програми на підзадачі розбиття даних та об’єднання проміжних результатів з можливістю необмеженого горизонтального масштабування шляхом додавання обчислювальних вузлів до кластеру [1].

Дана технологія дозволяє ефективно вирішувати різноманітні задачі аналізу даних, але не виправдана для задач пошуку відповідно до запиту. Схема функціонування MapReduce задачі показана на рис. 1.



Рисунок 1 – Схема функціонування MapReduce задачі

Робота MapReduce складається з двох кроків: розділення та згортання. На кроці map виконується попередня обробка вхідних даних. Для цього один з комп’ютерів отримує вхідні дані задачі, розділяє їх на частини і передає іншим комп’ютерам для обробки. На кроці reduce виконується згортання попередньо оброблених даних. Головний вузол отримує відповіді від робочих вузлів і на їх основі формує результат – розв’язок задачі [1].

Для прикладу опишемо реалізацію задачі підрахунку кількості кожного слів у файлі засобами MapReduce.

Дану програму реалізовано мовою програмування С++.

**Лістинг програмного коду**

#include <algorithm>

#include <limits>

#include <string>

#include "stdint.h"

#include "hadoop/Pipes.hh"

#include "hadoop/TemplateFactory.hh"

#include "hadoop/StringUtils.hh"

using namespace std;

class WordCountMapper : public HadoopPipes::Mapper {

public:

WordCountMapper(HadoopPipes::TaskContext& context) {

}

// функція отримує як паарметр строку, виводить формат: word,"1"

void map(HadoopPipes::MapContext& context) {

//отримання строки тексту

string line = context.getInputValue();

vector< string > words =

HadoopUtils::splitString(line, " ");

for (unsigned int i = 0; i < words.size(); i++) {

context.emit(words[i], HadoopUtils::toString(1));

}

}

};

class WordCountReducer : public HadoopPipes::Reducer {

public:

WordCountReducer(HadoopPipes::TaskContext& context) {

}

void reduce(HadoopPipes::ReduceContext& context) {

int count = 0;

while (context.nextValue()) {

count += HadoopUtils::toInt(context.getInputValue());

}

context.emit(context.getInputKey(), HadoopUtils::toString(count));

}

};

//головна функція

int main(int argc, char \*argv[]) {

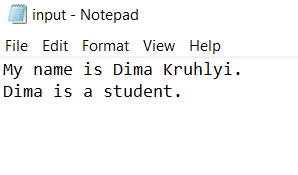
return HadoopPipes::runTask(HadoopPipes::TemplateFactory<

WordCountMapper,

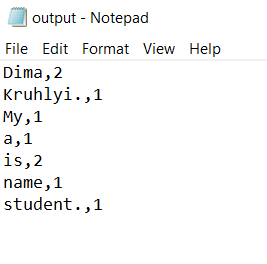
WordCountReducer >());

}

**Вхідні дані**



**Вихідні дані**



**Висновок:** виконавши дану лабораторну роботу, я ознайомився з роботою технології MapReduce та реалізував програму для підрахунку слів з використанням даної технології.

**Посилання:**

1. MapReduce [Електронний ресурс]: // wikipedia– Режим доступу до ресурсу: https://ru.wikipedia.org/wiki/MapReduce