Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Лабораторная работа №5

по дисциплине «Анализ алгоритмов»

Тема Организация параллельных вычислений по конвейерному принципу

Студент Тузов Даниил Александрович

Группа ИУ7-52Б

Преподаватель Строганов Дмитрий Владимирович

Москва, 2024 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В 5 лабораторной работе рассматривается тема обработки задач по конвейерному принципу.

Параллелизм описывает последовательности действий, которые происходят одновременно [1]. Нередко в современных системах используется распараллеливание вычислений, которое может привести к росту временной эффективности программы.

Целью работы является разработка ПО, выполняющего обработку файлов с html-кодом рецептов с сайта [5] в конвейерном режиме. Для достижения поставленной цели необходимо решить следующие задачи:

— рассмотреть структуру сайта;

— разработать ПО, обрабатывающее файлы с html-кодом в конвейерном режиме: чтение, парсинг и запись в json файл;

— исследовать лог времен создания, поступления в очередь, обработки и уничтожения каждой задачи;

— сформировать базу данных по полученным json-файлам;

— обосновать полученные результаты.

1 Входные и выходные данные

Входными данными в программе является директория с файлами, содержащими HTML-код страницы с рецептом. Выходными данными является директория с файлами в формате json, в которых хранится информация о:

— названии;

— авторе;

— ингредиентах;

— шагах выполнения;

2 Преобразование входных данных в выходные

Программа написана на языке C++ с использованием объектно-ориентированного программирования. Класс-конвейер моделирует работу конвейера. Для этого внутри функции $execute$ создается 3 отдельных потока [2], отвечающих за обработку на каждой стадии и поток-диспетчер, отвечающий за создание задач. Пример кода представлен в листинге 2.1:

Листинг 2.1 — Класс-конвейер

```
class Conveyer {
public:
    Conveyer(int col_files);
    void execute();
    void generate();
    void first_process();
    void second_process();
    void third_process();
private:
    std::queue<std::shared_ptr<ConveyerTask>> _first_queue;
    std::queue<std::shared_ptr<ConveyerTask>> _second_queue;
    std::queue<std::shared_ptr<ConveyerTask>> _third_queue;
    std::thread _threads[4];
    std::vector<std::shared_ptr<ConveyerTask>> _pool_tasks;
    std::vector<std::string> _urls;
    std::shared_ptr<Logger> _logger;
    int _col_files;
};
```

После формирования директории с файлами в формате json запускается скрипт на языке Python [3] для формирования базы данных PostgreSQL [4]. Код приведен в листинге 2.2:

Листинг 2.2 — Скрипт-создающих таблицу в базе данных

```
import psycopg2
from os import listdir
from os.path import isfile, join

class Lab:
    def __init__(self):
        self.__conn = psycopg2.connect(dbname="postgres",
                                       user="postgres",
                                       password="postgres123",
                                       host="localhost")
        self.__cursor = self.__conn.cursor()

    def __del__(self):
        self.__cursor.close()
        self.__conn.close()
```

```
    def create_db_request(self):
        sql_cr_schema = """create schema if not exists aalab;"""
        self.__cursor.execute(sql_cr_schema)
        sql_cr_table = """
            create table if not exists aalab.parse (
                id integer,
                issue_id integer,
                url text,
                title text,
                author text,
                ingredients json,
                steps json
            );
            alter table aalab.parse add constraint id_pk primary key(id);"""
        self.__cursor.execute(sql_cr_table)
        files = [join("data_json", f) for f in listdir("data_json") if
           isfile(join("data_json", f))]
        for i in files:
            sql_add_file = "create temp table tmp ( tmp json );\n" + \
                f"copy tmp from 'D:\Study\AA_labs\lab_5\\{i}';\n" + \
                "insert into aalab.parse\n" + \
                "select (json_populate_record(null::aalab.parse,
                   json_array_elements(tmp))).* from tmp;\n" + \
                "drop table tmp;"
            self.__cursor.execute(sql_add_file)
        self.__conn.commit()

db = Lab()
db.create_db_request()
```

3 Тестирование

При тестировании программы на вход подавалось число задач, которые необходимо обработать, и директория с файлами с html-кодом. В результате работы формировалась директория с файлами в формате json . На следующем этапе выполнялось сравнение файла с html кодом с json-файлом полученном в результате работы, проверялось успешность парсинга.

Тестирования выполнялось для разного количества задач.

Все тесты успешно пройдены.

4 Примеры работы программы

На рисунке 4.1 представлен исходный файл:

```
itemprop="bestRating" content="10" /> <meta itemprop="ratingValue" content="0"><meta itemprop="ratingCount" content="0"></div><script
type="text/javascript">reyting_def(25,0,'reyting_stars');</script>
    <aside class="col-sm-3 col-xs-2 padding">
<div>
    <strong>Автор:</strong>
    <a href="/user/0.html" rel="author" class="clearfix">
    <img class="hidden-xs radius-hiden {{class}}" src="/img/avatar/0avatara.jpg" alt="Автор Администратор" style="max-width: 120px">
    <span itemprop="author" class="author clearfix">Администратор</span></a><br>
    <a href="/result/user/podpiska.php?id=0" rel="nofolow" class="btn btn-sm "></a>
</div>

        <b>Категория блюда</b><br>
        <div itemprop="recipeCategory"><div class="category"> Каши</div><div class="sub-category"> Гречневая крупа</div></div>
    </aside>

    <section class="col-sm-8 col-xs-9 margin text-right list-unstyled">
    <h2>Ингредиенты</h2>
    <ul id="ing">
       <li itemprop="ingredients"  class="ingredient">1 ст. гречневой крупы</li><li itemprop="ingredients"  class="ingredient">1/2 ст.
чернослива</li><li itemprop="ingredients"  class="ingredient">1/2 ст. сушеных груш</li><li itemprop="ingredients"  class="ingredient">4
ст. воды</li><li itemprop="ingredients"  class="ingredient">4 ст. л. сливок</li><li itemprop="ingredients"  class="ingredient">соль</li>
    </ul>
    </section>
        <section class="instructions col-sm-12 col-xs-12 marginbottom clearfix">
    <h2>Способ приготовления</h2>
    <ul itemprop="recipeInstructions" id="instruction">
        <li class="instruction">Замочите чернослив и груши на 1,5 часа.</li>
<li class="instruction">Затем отварите фрукты в этой же воде.</li>
<li class="instruction">Отвар процедите и сварите в нем рассыпчатую кашу.</li>
<li class="instruction">Готовую кашу украсьте отварными сухофруктами и сливками.</li>
    </ul>
```

Рисунок 4.1 — Файл, содержащий html-код страницы рецепта

На рисунке 4.2 представлен json-файл:

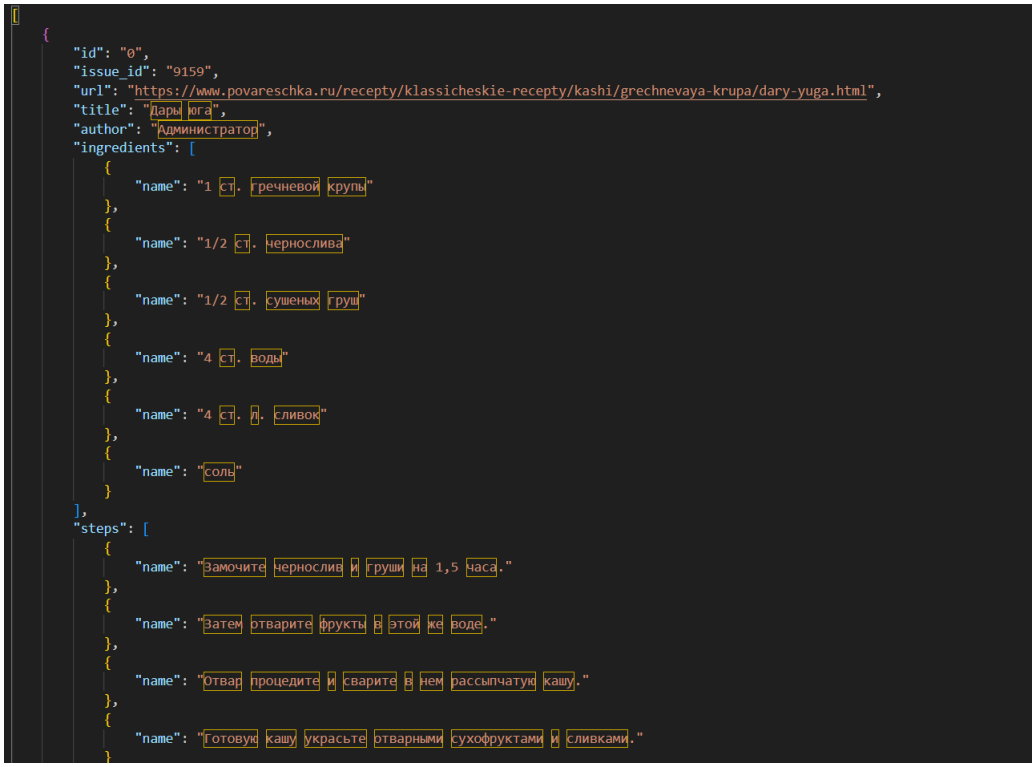
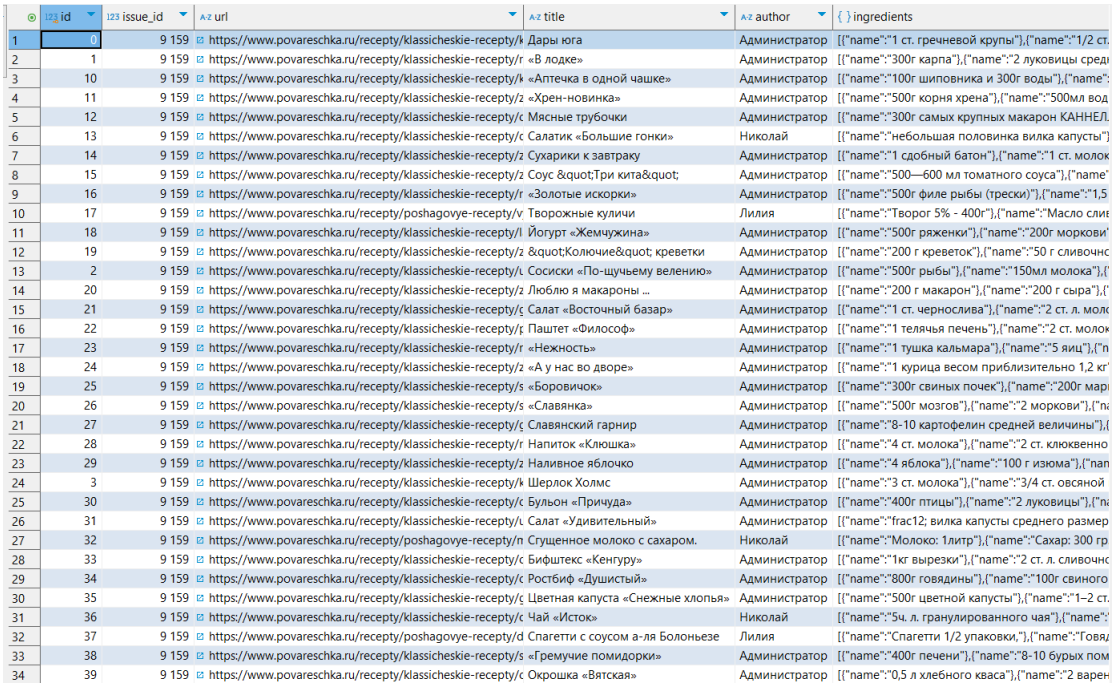


Рисунок 4.2 — Файл, содержащий json-код рецепта

На рисунке 4.3 представлена таблица в базе данных:



	id	issue_id	url	title	author	ingredients
1	0	9 159	https://www.povareschka.ru/recepty/klassicheskie-recepty/k	Дары юга	Администратор	[{"name":"1 ст. гречневой крупы"},{"name":"1/2 ст.
2	1	9 159	https://www.povareschka.ru/recepty/klassicheskie-recepty/r	«В лодке»	Администратор	[{"name":"300г карпа"},{"name":"2 луковицы средн
3	10	9 159	https://www.povareschka.ru/recepty/klassicheskie-recepty/k	«Аптечка в одной чашке»	Администратор	[{"name":"100г шиповника и 300г воды"},{"name":
4	11	9 159	https://www.povareschka.ru/recepty/klassicheskie-recepty/z	«Хрен-новинка»	Администратор	[{"name":"500г корня хрена"},{"name":"500мл вод
5	12	9 159	https://www.povareschka.ru/recepty/klassicheskie-recepty/c	Мясные трубочки	Администратор	[{"name":"300г самых крупных макарон КАННЕЛ
6	13	9 159	https://www.povareschka.ru/recepty/klassicheskie-recepty/c	Салатик «Большие гонки»	Николай	[{"name":"небольшая половинка вилка капусты"}
7	14	9 159	https://www.povareschka.ru/recepty/klassicheskie-recepty/z	Сухарики к завтраку	Администратор	[{"name":"1 сдобный батон"},{"name":"1 ст. молок
8	15	9 159	https://www.povareschka.ru/recepty/klassicheskie-recepty/z	Соус "Три кита"	Администратор	[{"name":"500—600 мл томатного соуса"},{"name"
9	16	9 159	https://www.povareschka.ru/recepty/klassicheskie-recepty/r	«Золотые искорки»	Администратор	[{"name":"500г филе рыбы (трески)"},{"name":"1,5
10	17	9 159	https://www.povareschka.ru/recepty/poshagovye-recepty/v	Творожные куличи	Лилия	[{"name":"Творог 5% - 400г"},{"name":"Масло слив
11	18	9 159	https://www.povareschka.ru/recepty/klassicheskie-recepty/I	Йогурт «Жемчужина»	Администратор	[{"name":"500г пряженки"},{"name":"200г моркови"
12	19	9 159	https://www.povareschka.ru/recepty/klassicheskie-recepty/z	"Колючие" креветки	Администратор	[{"name":"200 г креветок"},{"name":"50 г сливочно
13	2	9 159	https://www.povareschka.ru/recepty/klassicheskie-recepty/c	Сосиски «По-щучьему велению»	Администратор	[{"name":"500г рыбы"},{"name":"150мл молока"},{"
14	20	9 159	https://www.povareschka.ru/recepty/klassicheskie-recepty/z	Люблю я макароны ...	Администратор	[{"name":"200 г макарон"},{"name":"200 г сыра"},{"
15	21	9 159	https://www.povareschka.ru/recepty/klassicheskie-recepty/c	Салат «Восточный базар»	Администратор	[{"name":"1 ст. чернослива"},{"name":"2 ст. л. моло
16	22	9 159	https://www.povareschka.ru/recepty/klassicheskie-recepty/p	Паштет «Философ»	Администратор	[{"name":"1 телячья печень"},{"name":"2 ст. молок
17	23	9 159	https://www.povareschka.ru/recepty/klassicheskie-recepty/r	«Нежность»	Администратор	[{"name":"1 тушка кальмара"},{"name":"5 яиц"},{"n
18	24	9 159	https://www.povareschka.ru/recepty/klassicheskie-recepty/z	«А у нас во дворе»	Администратор	[{"name":"1 курица весом приблизительно 1,2 кг"
19	25	9 159	https://www.povareschka.ru/recepty/klassicheskie-recepty/s	«Боровичок»	Администратор	[{"name":"300г свиных почек"},{"name":"200г мар
20	26	9 159	https://www.povareschka.ru/recepty/klassicheskie-recepty/s	«Славянка»	Администратор	[{"name":"500г мозгов"},{"name":"2 моркови"},{"na
21	27	9 159	https://www.povareschka.ru/recepty/klassicheskie-recepty/c	Славянский гарнир	Администратор	[{"name":"8-10 картофелин средней величины"},{
22	28	9 159	https://www.povareschka.ru/recepty/klassicheskie-recepty/r	Напиток «Клюшка»	Администратор	[{"name":"4 ст. молока"},{"name":"2 ст. клюквенно
23	29	9 159	https://www.povareschka.ru/recepty/klassicheskie-recepty/z	Наливное яблочко	Администратор	[{"name":"4 яблока"},{"name":"100 г изюма"},{"nam
24	3	9 159	https://www.povareschka.ru/recepty/klassicheskie-recepty/k	Шерлок Холмс	Администратор	[{"name":"3 ст. молока"},{"name":"3/4 ст. овсяной
25	30	9 159	https://www.povareschka.ru/recepty/klassicheskie-recepty/c	Бульон «Причуда»	Администратор	[{"name":"400г птицы"},{"name":"2 луковицы"},{"na
26	31	9 159	https://www.povareschka.ru/recepty/klassicheskie-recepty/s	Салат «Удивительный»	Администратор	[{"name":"frac12; вилка капусты среднего размер
27	32	9 159	https://www.povareschka.ru/recepty/poshagovye-recepty/n	Сгущенное молоко с сахаром.	Николай	[{"name":"Молоко: 1литр"},{"name":"Сахар: 300 гр
28	33	9 159	https://www.povareschka.ru/recepty/klassicheskie-recepty/c	Бифштекс «Кенгуру»	Администратор	[{"name":"1кг вырезки"},{"name":"2 ст. л. сливочно
29	34	9 159	https://www.povareschka.ru/recepty/klassicheskie-recepty/c	Ростбиф «Душистый»	Администратор	[{"name":"800г говядины"},{"name":"100г свиного
30	35	9 159	https://www.povareschka.ru/recepty/klassicheskie-recepty/c	Цветная капуста «Снежные хлопья»	Администратор	[{"name":"500г цветной капусты"},{"name":"1–2 ст.
31	36	9 159	https://www.povareschka.ru/recepty/klassicheskie-recepty/c	Чай «Исток»	Николай	[{"name":"5ч. л. гранулированного чая"},{"name":
32	37	9 159	https://www.povareschka.ru/recepty/poshagovye-recepty/d	Спагетти с соусом а-ля Болоньезе	Лилия	[{"name":"Спагетти 1/2 упаковки,"},{"name":"Говя
33	38	9 159	https://www.povareschka.ru/recepty/klassicheskie-recepty/s	«Гремучие помидорки»	Администратор	[{"name":"400г печени"},{"name":"8-10 бурых пом
34	39	9 159	https://www.povareschka.ru/recepty/klassicheskie-recepty/c	Окрошка «Вятская»	Администратор	[{"name":"0,5 л хлебного кваса"},{"name":"2 варен

Рисунок 4.3 — Сформированная БД

5 Описание исследования

Все замеры проводились на ЭВМ, характеристики которой приведены ниже:

— процессор – 12th Gen Intel(R) Core(TM) i5-12450H 2.00 ГГц;

— оперативная память – 16,0 ГБ;

— тип системы – 64-разрядная операционная система, процессор x64;

— операционная система – Windows 11;

— версия ОС – 23H2;

— 12 логических ядер.

На вход подавалось 4 задачи. Лог, соответствующий процессу обработки этих задач, предствлен на рисунке 5.1:

```
Input count of file needed to read: 4
TASK: 1; TIME: 2644; COMM: task created
TASK: 1; TIME: 2646; COMM: in first queue
TASK: 2; TIME: 2646; COMM: task created
TASK: 2; TIME: 2646; COMM: in first queue
TASK: 3; TIME: 2647; COMM: task created
TASK: 3; TIME: 2647; COMM: in first queue
TASK: 4; TIME: 2647; COMM: task created
TASK: 4; TIME: 2648; COMM: in first queue
TASK: 1; TIME: 2648; COMM: at first station
TASK: 1; TIME: 2653; COMM: in second queue
TASK: 2; TIME: 2653; COMM: at first station
TASK: 1; TIME: 2654; COMM: at second station
TASK: 1; TIME: 2655; COMM: in third queue
TASK: 1; TIME: 2656; COMM: at third station
TASK: 1; TIME: 2657; COMM: processed
TASK: 2; TIME: 2659; COMM: in second queue
TASK: 3; TIME: 2659; COMM: at first station
TASK: 2; TIME: 2659; COMM: at second station
TASK: 2; TIME: 2660; COMM: in third queue
TASK: 2; TIME: 2660; COMM: at third station
TASK: 2; TIME: 2661; COMM: processed
TASK: 3; TIME: 2664; COMM: in second queue
TASK: 4; TIME: 2664; COMM: at first station
TASK: 3; TIME: 2664; COMM: at second station
TASK: 3; TIME: 2665; COMM: in third queue
TASK: 3; TIME: 2665; COMM: at third station
TASK: 3; TIME: 2666; COMM: processed
TASK: 4; TIME: 2669; COMM: in second queue
TASK: 4; TIME: 2669; COMM: at second station
TASK: 4; TIME: 2669; COMM: in third queue
TASK: 4; TIME: 2670; COMM: at third station
TASK: 4; TIME: 2671; COMM: processed
```

Рисунок 5.1 — Лог для четырех задач

Из рисунка 5.1 видно, что задачи действительно выполняются параллельно. На рисунке 5.2 представлена графическая интерпертация работы конвейера:

Рисунок 5.2 — Графическая интерпретация работы

Можно заметить, что задачи действительно выполняются в параллельном режиме.

На рисунке 5.3 представлены некоторые средние статистические показатели:

```
Average existance time: 17.75
Average first queue wait time: 9.25
Average second queue wait time: 0.25
Average third queue wait time: 0.5
Average first process time: 5.25
Average second process time: 0.75
Average third process time: 1
```

Рисунок 5.3 — Статистические показатели

Поскольку параметры ожидания в каждой очереди ненулевые, можно сделать вывод, что при конвейерной обработке возникают простои.

ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы поставленная цель была достигнута. Были решены все задачи:

1) рассмотрена структура сайта;
2) разработан класс, который выполняет парсинг файлов с html кодом и записывает результат в json формате;
3) реализован класс-конвейер, который выполняет обработку задач в параллельном режиме;
4) проведено исследование, в ходе которого выяснилось, что задачи выполняются параллельно, но иногда в процессе обработки возникают простои;
5) обоснованы полученные результаты и сделан вывод.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Энтони Уильямс, C++. Практика многопоточного программирования. — Город: Санкт-Петербург, Издательский дом «Питер», 2020. — 640 с.
2. Ковалев, Введение в многопоточность / [Электронный ресурс] // Режим доступа: https://rekovalev.site/ multithreading-3-cpp/#threads-create
3. Язык Python / [Электронный ресурс] // Режим доступа: https://docs.python.org/3/index.html
4. PostgreSQL / [Электронный ресурс] // Режим доступа: https://www.postgresql.org/
5. Сайт с рецептами / [Электронный ресурс] // Режим доступа: https://www.povareschka.ru/
6. Функция clock() / [Электронный ресурс] // Режим доступа: https://learn.microsoft.com/ru-ru/cpp/c-runtime-library/reference/clock?view=msvc-170