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»

Отчёт по лабораторной работе №5
по курсу «Анализ алгоритмов»

Тема Организация параллельных вычислений по конвейерному принципу

Студент Талышева О.Н.

Группа ИУ7-55Б

Преподаватели Волкова Л.Л., Строганов Ю.В.

Москва — 2024 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В современных вычислительных системах организация параллельных вычислений играет ключевую роль в повышении производительности программного обеспечения. Одним из эффективных методов структурирования параллельных вычислений является конвейерный принцип, при котором задача делится на несколько последовательных этапов. Каждый этап выполняется в отдельном потоке, позволяя системе одновременно обрабатывать разные части задачи на разных уровнях конвейера. Такой подход особенно полезен для задач с последовательными стадиями обработки данных, поскольку он позволяет распределить выполнение между несколькими ядрами процессора, минимизируя время простоя и повышая общую производительность программы.

Цель работы: получение навыка организации параллельных вычислений по конвейерному принципу.

Для достижения этой цели были поставлены следующие задачи:

1. анализ предметной области;
2. разработка алгоритма обработки данных;
3. создание ПО реализующего разработанный алгоритм;
4. исследование характеристик созданного ПО.

1 Входные и выходные данные

Входными данными для ПО является папка data, содержащая html файлы со сказанными кулинарными страницами. Выходные данные — база данных с таблицами, содаржащими рецепты, ингредиенты для них и шаги по приготовлению. Также программа делает замеры времени для получения максимального, минимального, среднего и медианного времён нахождения в очередях 2 и 3 и обработки на трёх стадиях.

2 Преобразование входных данных в выходные

Программа запускает 5 потоков, каждый из которых выполняет свою последовательную часть обработки заявки. Первый поток создаёт задачи с путями к файлам из папки data и помещает их в первую очередь. Второй поток берёт заявки из первой очереди, читает файлы и дополняет заявки недостающими данными о рецептах, затем помещает задачу во вторую очередь. Третий поток берёт заявку из второй очереди, очищает выбранные из файла данные от html символов и помещает задачу в третью очередь. Четвёртый поток берёт заявки из третьей очереди, записывает данные из неё в таблицы базы данных и помещает задачу в четвёртую очередь. Пятый поток берёт

заявки из четвёртой очереди и вычисляет максимальное, минимальное, среднее и медианное времена нахождения в очередях 2 и 3 и обработки на предыдущих трёх стадиях, попутно уничтожая заявки. Таким образом программа реализует параллельные вычисления по конвейерному принципу. [2]

3 Примеры работы программы

На рисунке 1 представлен пример работы программы с получившейся базой данных (рецепты были взяты с сайта kedem.ru и загружены программой из лабораторной работы №4).

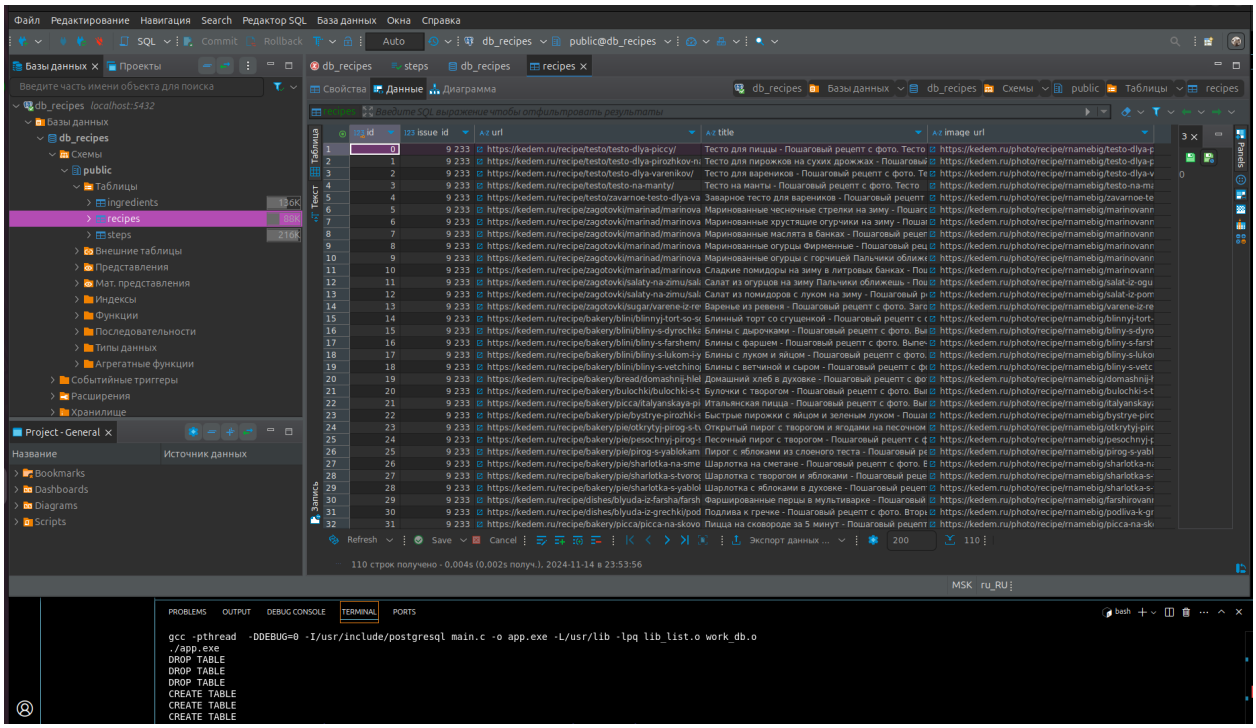

Рисунок 1 – Пример работы программы.

4 Тестирование

Выполнено тестирование реализованной основной части программы по методологии чёрного ящика. В таблице 1 представлено описание тестов. Все тесты пройдены успешно.

Таблица 1 – Функциональные тесты

№ теста	Входные данные	Ожидаемые выходные данные	Успешность теста
1	Папка с файлами, структура данных корректна	Заявки успешно добавлены в очередь 1, корректная обработка данных	Успешно
2	Пустая папка	Нет заявок для обработки, программа завершает работу без ошибок	Успешно
3	Некорректный формат данных в файле (HTML)	Программа игнорирует файл, продолжает обработку остальных задач	Успешно
4	Папка с файлами, корректная структура данных	Заявки корректно проходят через очереди, данные записываются в БД	Успешно
5	Большое количество файлов (параллельная обработка)	Все файлы обрабатываются в многозадачном режиме, результаты записываются в БД	Успешно
6	Проблемы с подключением к БД	Программа выводит ошибку подключения и завершает работу	Успешно
7	Недостаток памяти (искусственно)	Программа корректно обрабатывает ошибку выделения памяти	Успешно
8	Неожиданное завершение потока	Программа завершает работу с выводом ошибок при некорректной работе потока	Успешно

5 Описание исследования

Были проведены замеры времени для получения максимального, минимального, среднего и медианного времён нахождения в очередях 2 и 3 и обработки на трёх стадиях (см листинг 1 и график 2).

```
Device 1
tmin = 0.165504 ms, tmax = 0.413699 ms, tavg = 0.214917 ms, tmed =
    0.211162 ms

Device 2
tmin = 0.001872 ms, tmax = 0.022422 ms, tavg = 0.007080 ms, tmed =
    0.006392 ms

Device 3
tmin = 7.072097 ms, tmax = 58.647235 ms, tavg = 23.602610 ms, tmed =
    22.469280 ms

Queue 2
tmin = 0.002048 ms, tmax = 0.084550 ms, tavg = 0.007184 ms, tmed =
    0.003553 ms

Queue 3
tmin = 124.725575 ms, tmax = 2680.544291 ms, tavg = 1456.930559 ms, tmed =
     1466.566185 ms
```

Листинг 1 – Результаты замеров времени на разном количестве потоков

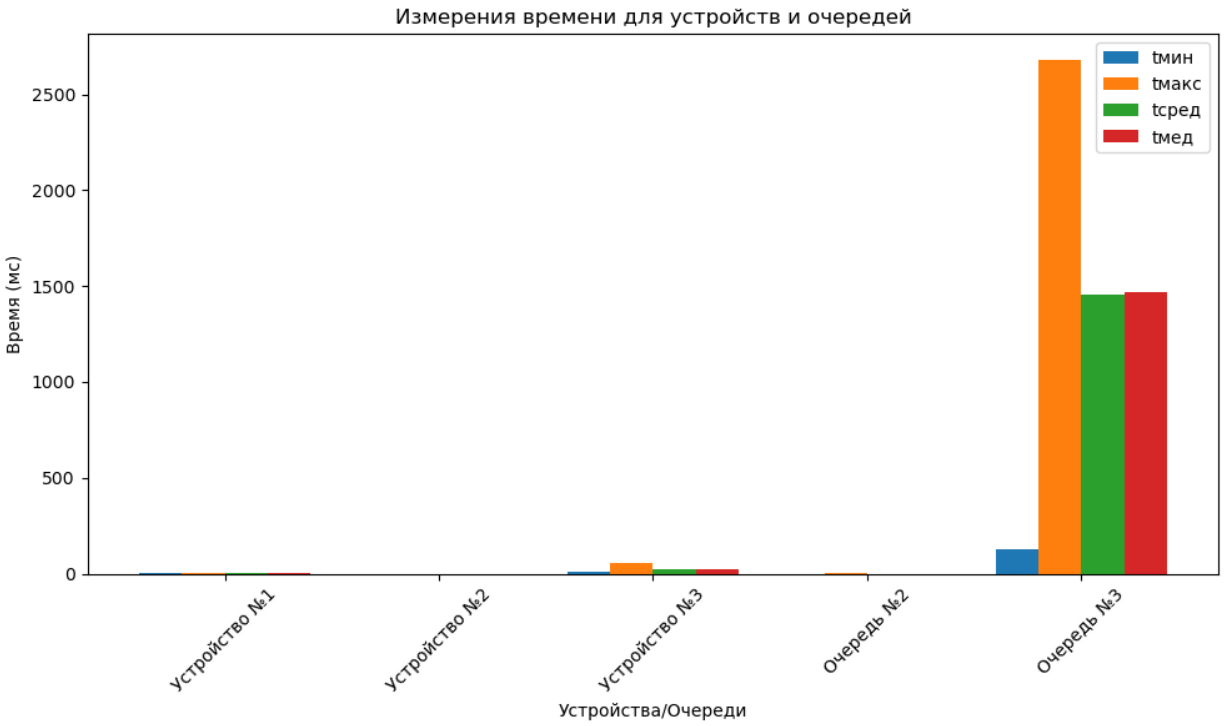

Рисунок 2 – График замеров времени для устройств и очередей

Выводы

Процесс обработки данных через потоки на первых двух стадиях (чтение данных из файла и извлечение необходимого подмножества данных) происходит достаточно быстро, с минимальными задержками. Однако на третьей стадии, связанной с записью извлечённых данных в хранилище (в данном случае в базу данных), возникает значительное замедление. Это объясняется тем, что операция записи в базу данных требует существенно большего времени по сравнению с операциями чтения и обработки данных на предыдущих этапах. Как следствие, время нахождения данных в очереди 3 превышает время, затраченное на их обработку в очереди 2, что приводит к задержкам на последней стадии обработки.

ЗАКЛЮЧЕНИЕ

В результате выполнения лабораторной работы были получены навыки организации параллельных вычислений по конвейерному принципу.

В частности:

1. был проведён анализ предметной области;
2. разработан алгоритм обработки данных;
3. создано ПО реализующее разработанный алгоритм;
4. исследованы характеристики созданного ПО.

В ходе лабораторной работы был рассмотрен, спроектированн и запрограммирован алгоритм парсинга страниц и записи данных в базу с помощью нативных потоков по конвейерному принципу.

Проведённые замеры времени обработки данных через потоки на разных стадиях показали, что врями чтение данных и их обработка на первых двух устройствах происходит быстрее, чем запись этих данных в хранилище. В связи с этим время нахождения в очереди к потоку загружающему данные в базу также больше, чем в очереди к потоку, занимающемуся выборкой данных.

Таким образом, лабораторная работа позволила не только освоить принципы организации параллельных вычислений по конвейерному принципу, но и на практике выявить узкие места в процессах обработки данных. Особенно это касается этапа записи данных в хранилище, который оказался значительно более ресурсоёмким и времязатратным по сравнению с другими этапами. Кроме того, проведённые эксперименты продемонстрировали эффективность использования многозадачности для ускорения обработки данных, что делает данное решение перспективным для применения в реальных системах, требующих обработки больших объёмов информации в режиме реального времени.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] Методы парсинга данных на C++ и Python. Tetraquark. [Электронный ресурс]. URL: https://tetraquark.ru/archives/47 (дата обращения: 22.10.2024).

[2] AppMaster, Конвейерное программирование, доступно по ссылке: https://appmaster.io/ru/glossary/konveiernoe-programmirovanie, дата обращения: 14 ноября 2024.