



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 № 4 по дисциплине «Анализ алгоритмов»

Тема Параллельные вычисления на основе нативных потоков

Студент Бугаков И. С.

Группа ИУ7-54Б

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

Москва, 2025

СОДЕРЖАНИЕ

| | |
|---|-----------|
| ВВЕДЕНИЕ | 3 |
| 1 Входные и выходные данные | 4 |
| 2 Преобразование входных данных в выходные | 5 |
| 3 Тестирование | 7 |
| 4 Примеры работы программы | 8 |
| 5 Описание исследования | 9 |
| ЗАКЛЮЧЕНИЕ | 11 |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ | 12 |

ВВЕДЕНИЕ

Зачастую в приложениях возникает необходимость выполнять некоторые действия параллельно. При этом исполняемый параллельно код должен иметь доступ общим для всей программы ресурсам. В связи с этим возникает необходимость в многопоточном коде. Кроме того, при выполнении на многопроцессорной машине такой код действительно будет выполняться параллельно, значительно ускоряя процесс вычислений [3].

Цель данной лабораторной работы: получить навык организации параллельных вычислений на основе нативных потоков.

Задачи лабораторной работы:

- 1) разработать ПО осуществляющее выгрузку данных со страниц указанного ресурса;
- 2) провести тестирование разработанного ПО;
- 3) провести замеры времени выгрузки содержимого страниц при различном количестве потоков управления.

1 Входные и выходные данные

Входные данные программы:

- заглавная страница ресурса — `sostra.ru`;
- количество потоков управления;
- количество выгружаемых страниц.

Выходные данные: директория с полученными страницами рецептов.

2 Преобразование входных данных в выходные

Для загрузки страниц была написана функция представленная в листинге 2.1, использующая библиотеку curl [1], для получения ссылок на другие страницы использовались функции приведенные в листинге 2.2.

Листинг 2.1 — Функция загрузки страниц

```
1 string load_page(const string &url, set<string> &urls, queue<string> &
   _urls_to_handle) {
2     CURL *curl = curl_easy_init();
3     string result;
4     if (curl) {
5         curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, curl_write);
6         curl_easy_setopt(curl, CURLOPT_URL, url.c_str());
7         curl_easy_setopt(curl, CURLOPT_WRITEDATA, &result);
8         CURLcode res_code = curl_easy_perform(curl);
9         curl_easy_cleanup(curl);
10        prepare_hyperlinks(result, urls, _urls_to_handle);
11    }
12    return result;
13 }
```

Листинг 2.2 — Функции выделения ссылок

```
1 set<string> extract_hyperlinks(const string &text) {
2     static const std::regex hl_regex("href=\"/(.*?)\"", std::
       regex_constants::icase);
3     return {std::sregex_token_iterator(text.begin(), text.end(), hl_regex
       , 1), std::sregex_token_iterator{}};
4 }
5
6 void prepare_hyperlinks(const string &text, set<string> &urls, queue<
   string> &_urls_to_handle) {
7     set<string> links = extract_hyperlinks(text);
8     for (auto it = links.begin(); it != links.end(); ) {
9         if (it->find(".ru") == -1 && it->find(".com") == -1 && it->find(".
       org") == -1 &&
10        it->substr(0, 12) == string("recipe-book/")) {
11         string new_url = string(BASE_URL) + (*it);
12         set_urls_mutex.lock();
13         if (urls.find(new_url) == urls.end()) {
14             urls.insert(string(BASE_URL) + (*it));
15             set_urls_mutex.unlock();
```

```
16         queue_urls_mutex.lock();
17         _urls_to_handle.push(new_url);
18         queue_urls_mutex.unlock();
19     } else
20         set_urls_mutex.unlock();
21     ++it;
22 }
23 }
24 }
```

3 Тестирование

Для проверки работоспособности было вручную проверено содержимое выгруженных страниц. Все тесты были успешно пройдены.

4 Примеры работы программы

На рисунке 4.1 приведена директория с выгруженными страницами, представленными HTML файлами. Всего получено 191 страница рецептов.

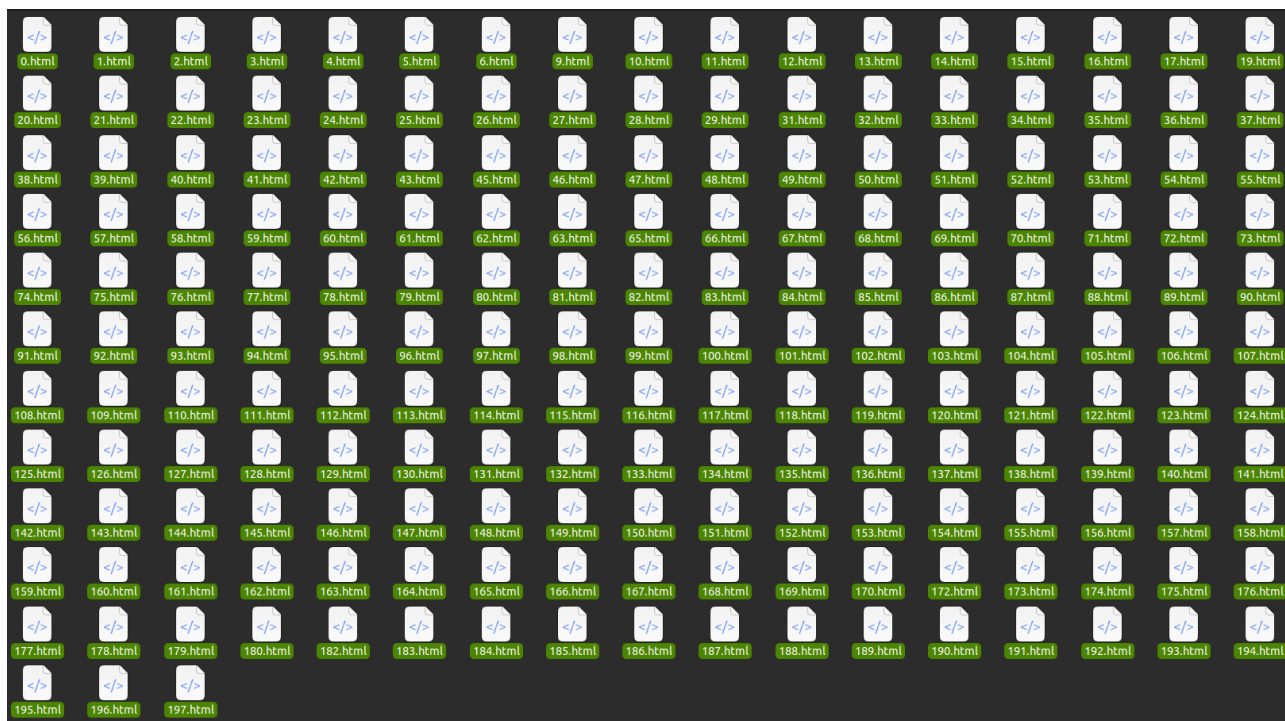


Рисунок 4.1 — Директория с полученными страницами

На рисунке 4.2 приведена одна из выгруженных страниц с рецептами..

Вьетнамский блинчик Нэм (NEM)



Приготовление

Вам понадобится

- Фарш (свиной или куриный)
200 г
- Гривы шампиньоны
150 г
- Бобовая вермишель «Фунчоза» марки SenSoyPremium
50 г (1 пучок)
- Рисовая бумага марки SenSoyPremium
10 шт. (1 упаковка)
- Соевый соус ORIGINAL марки Sen Soy Premium
- Рисовый уксус марки Sen Soy Premium
1 ч.л.
- Рыбный соус марки Sen Soy Premium
1 ч.л.
- Морковь
½ шт.
- Лайм
½ шт.
- Лук репчатый
1 шт.
- Растительное масло для жарки
- Для украшения (сладкий перец, стебель сельдерея, лайм)

Рисунок 4.2 — Страница с рецептом

5 Описание исследования

Исследовалась зависимость скорости выгрузки в терминах числа страниц в единицу времени от количества потоков. Замеры проводились для 1, 2, 4, 8, 16, 32 потоков. Выгружалось 191 страница, по три раза для каждого числа потоков, итоговое значение усреднялось.

Исследования проводились на машине со следующими характеристиками:

- процессор Intel(R) Core(TM) i5-10210U, тактовая частота 1.60 ГГц;
- оперативная память: 16 ГБ;
- операционная система: Ubuntu 22.04.4 LTS.

Результаты замеров приведены в таблице 5.1.

Таблица 5.1 — Среднее число загружаемых страниц в единицу времени при различном числе потоков

| Число потоков | Среднее количество выгружаемых страниц в единицу времени |
|---------------|--|
| 1 | 0.476693 |
| 2 | 1.11428 |
| 4 | 1.66893 |
| 8 | 2.41893 |
| 16 | 1.21408 |
| 32 | 1.98875 |

На рисунке 5.1 приведен график зависимости среднего числа выгружаемых страниц от числа потоков. Для визуализации данных использовалась библиотека matplotlib [2].

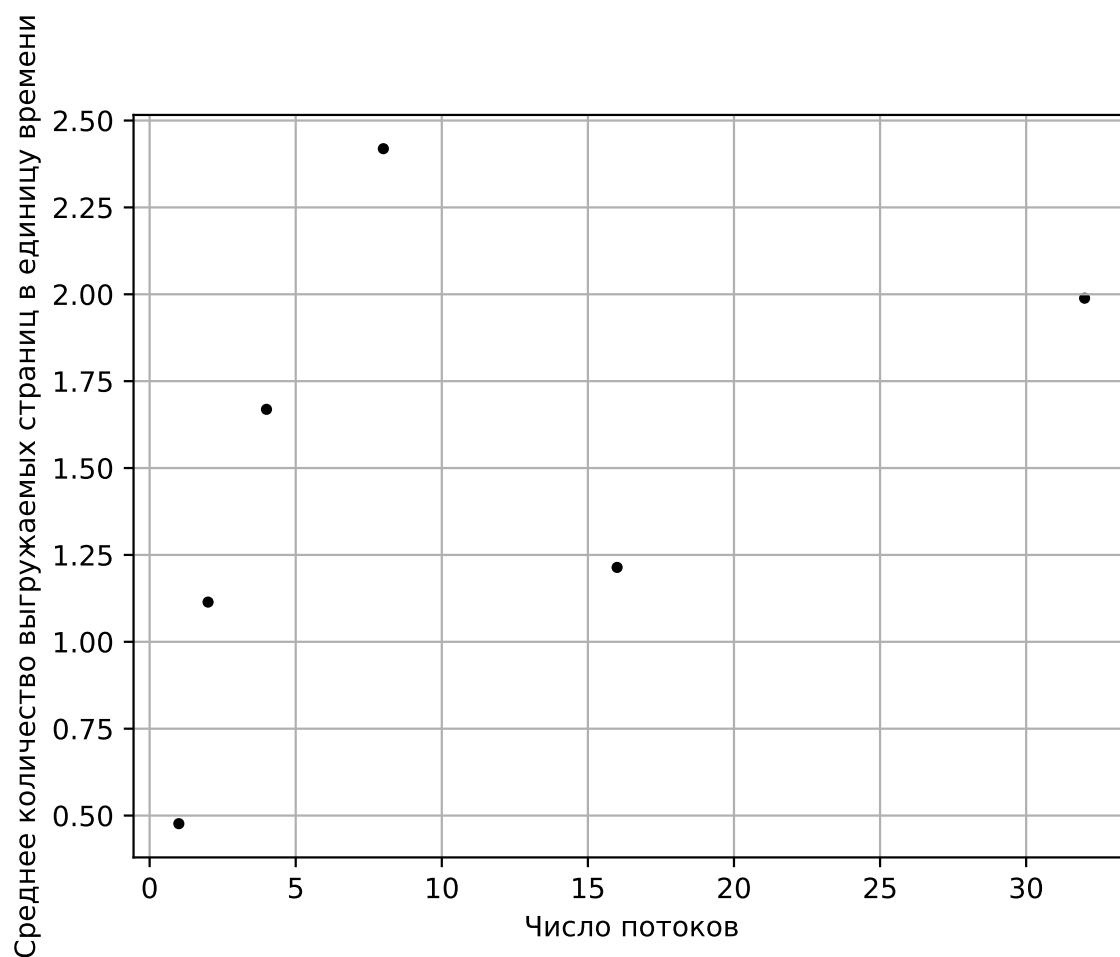


Рисунок 5.1 — Зависимость среднего числа выгружаемых страниц от числа потоков

Из результатов исследования следует, что использование потоков для параллельного выполнения задач ускоряет работу приложения. Однако, при превышении количеством потоков числа логических ядер, дальнейшего ускорения не происходит.

ЗАКЛЮЧЕНИЕ

Целью данной работы была организация параллельных вычислений на основе нативных потоков.

В ходе лабораторной работы были выполнены следующие задачи:

- 1) разработано ПО осуществляющее выгрузку данных со страниц указанного ресурса;
- 2) проведено тестирование разработанного ПО;
- 3) проведены замеры времени выгрузки содержимого страниц при различном числе потоков.

По итогам анализа было выявлено, что параллельное выполнение различными потоками задач ускоряет работу программы. Однако, при превышении числом потоков числа логических ядер процессора, дальнейшего ускорения не происходит.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. curl documentation. Режим доступа: <https://curl.se/libcurl/>. Дата обращения 24.12.2024.
2. matplotlib. Режим доступа: <https://matplotlib.org/>. Дата обращения 24.12.2024.
3. Э. Бос Х. Таненбаум. *Современные операционные системы*. Питер, 2015.