

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Орловский государственный университет имени И.С. Тургенева»  
Физико-математический факультет  
Кафедра информатики

# Проектирование и разработка веб сервера с поддержкой FastCGI

Подготовил: Сафронов Алексей Александрович

Направление подготовки:

01.03.02 Прикладная математика и информатика

Направленность (профиль): Системное программирование и  
компьютерные технологии

Руководитель:

к.п.н., доцент Новиков В.С.

Орёл –  
2022

## **Актуальность работы.**

При быстрорастущем объёме информации в интернете и, соответственно количеству запросов, появилась необходимость не только наращивать количество серверов, но и использовать более совершенные подходы для поддержания удовлетворительной скорости обработки.

## **Цель работы.**

Разработка FastCGI приложения для обработки большого количества информации при сохранении удовлетворительной скорости работы во время пиковых нагрузок.

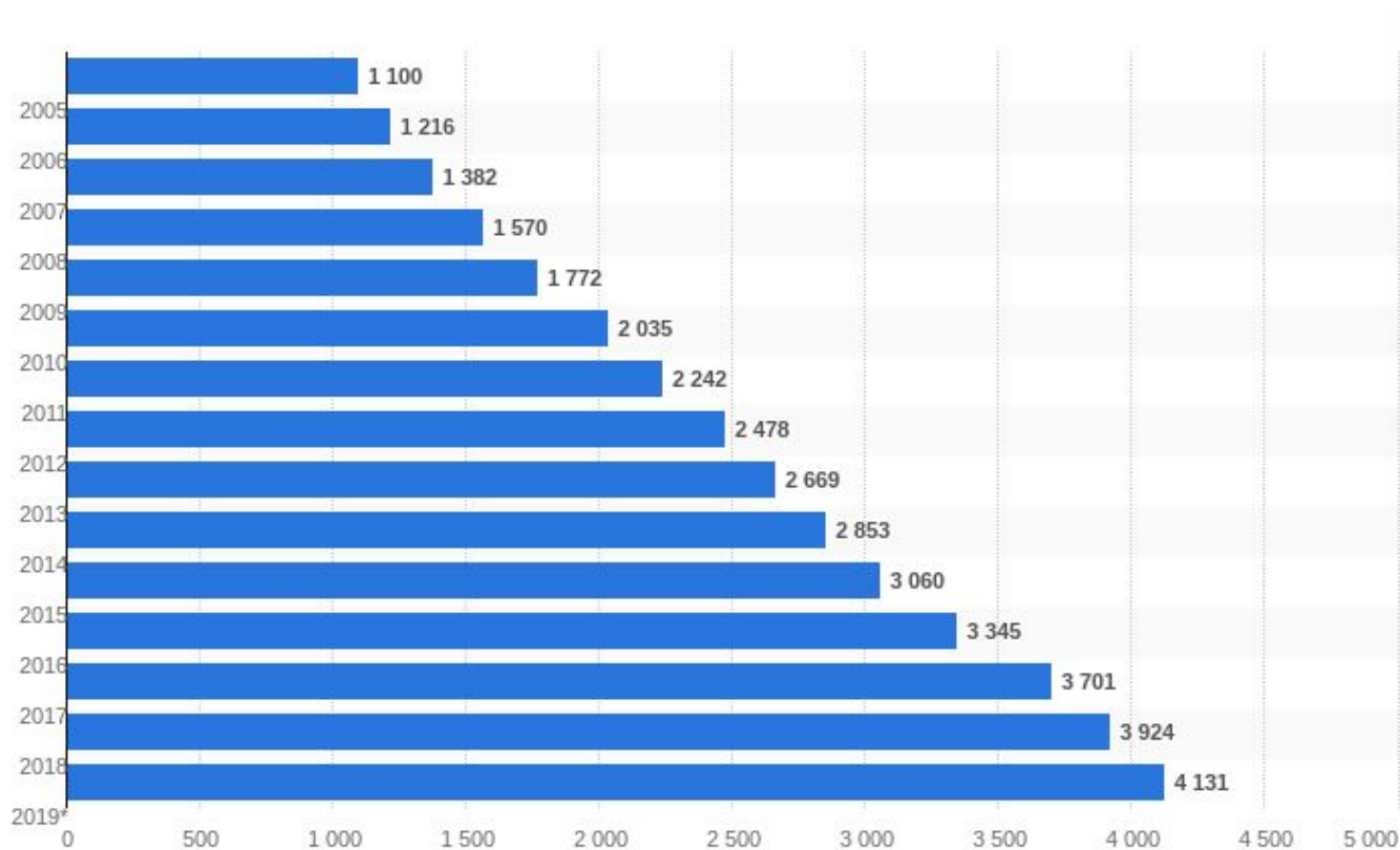
## **Задачи исследования.**

1. Развёртывание и конфигурирование сервера Apache2 для взаимодействия с CGI скриптами и FastCGI модулем.
2. Создать тестовую веб-страницу с некоторым объёмом данных с использованием Python 3.
3. Создать скрипт для проверки и анализа различных факторов, которые влияют на загрузку страницы.
4. Произвести исследования и анализы данных, полученных на предыдущем этапе и сделать выводы.

## Постановка задачи.

Установим и сконфигурируем веб сервер Apache2 с поддержкой CGI приложений при взаимодействии с помощью FastCGI модуля для обеспечения быстрой обработки большого количества данных и поддержки большого количества одновременных пользователей.

Для примера, ниже приведён график статистики до 2019 года как растёт объём уникальных пользователей в сети.



## **Программно-техническая реализация.**

Всё реализовывалось на операционной системе elementary OS. Из себя данная система представляет Linux на базе Ubuntu. Также был подготовлен модуль `fast_cgi` для Apache2 с помощью CMake. Сам веб сервер Apache2 был установлен и сконфигурирован с поддержкой обычных CGI и FastCGI приложений для возможности сравнить их работу. Для написания приложения использовался язык программирования Python 3.

Для поддержки FastCGI добавляем в конфигурацию веб сервера Apache2 данную команду:

```
LoadModule fastcgi_module /usr/lib/apache2/modules/mod_fastcgi.so
```

Данная команда загружает при старте сервера, собранный fast\_cgi модуль

В конфигурацию нашего сайта добавляем следующие строки:

- FastCgiServer /var/www/html/app.py -processes 4 -socket /tmp/fcgi.sock
- DirectoryIndex app.py
- Options +FollowSymLinks +ExecCGI
- AllowOverride All
- SetHandler fastcgi-script

Далее размещаем скрипт в нашем каталоге /var/www/html под названием app.py и после перезапуска веб сервера уже можем посмотреть его.

Теперь проанализируем технологии CGI и FastCGI.

Для этого был написан небольшой скрипт:

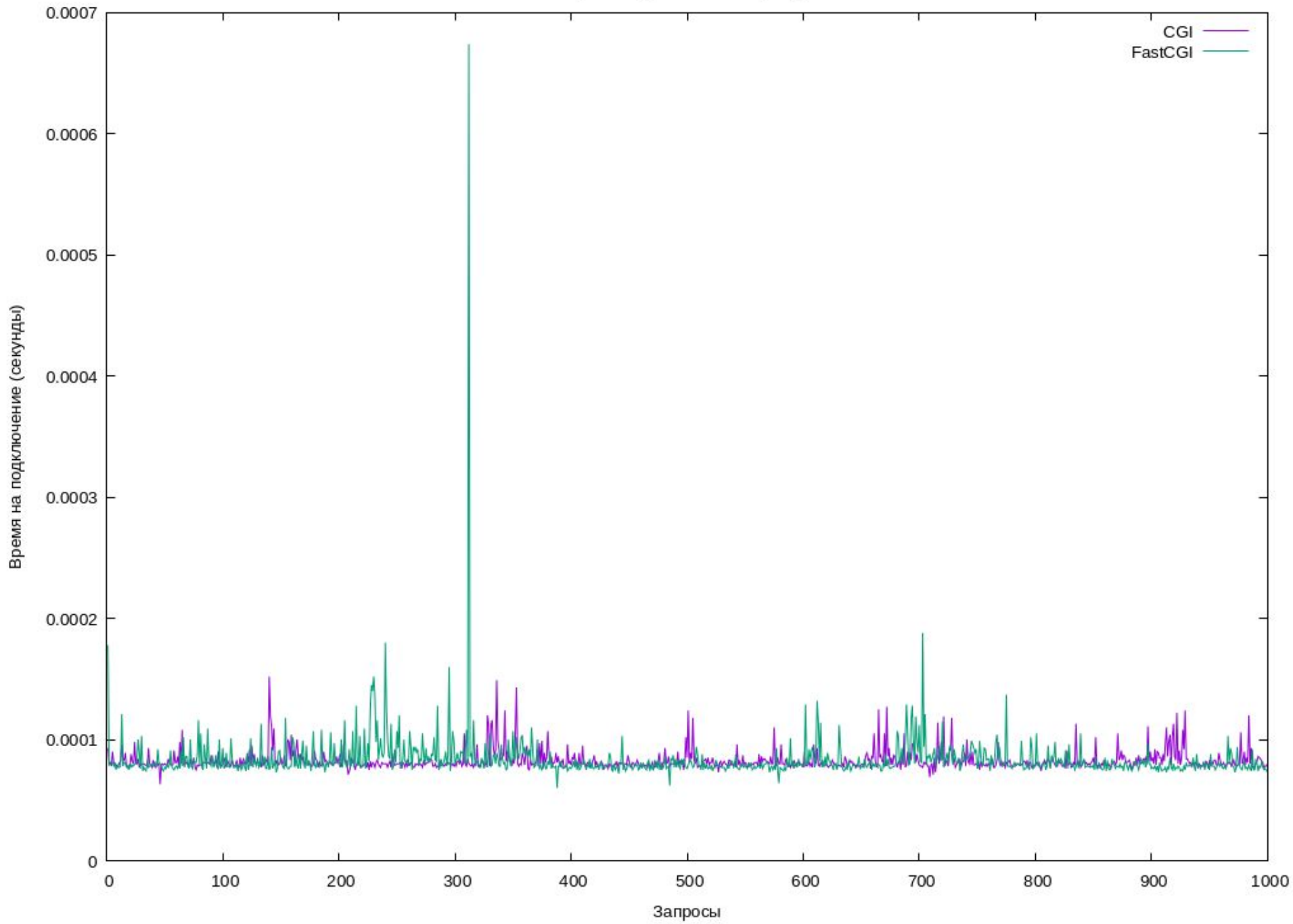
```
#!/bin/bash
i=0;
rm sas.txt
for i in {1..1000};
Do
curl -s -w "$i %{time_namelookup} %{time_connect} %{time_appconnect}
%{time_redirect} %{time_pretransfer} %{time_starttransfer} %{time_total}\n" >> sas.txt
-o /dev/null http://192.168.0.104;
Done
```

После того, как информация была собрана, пришло время для анализа.

Рассмотрим 4 стадии запроса и получения информации:

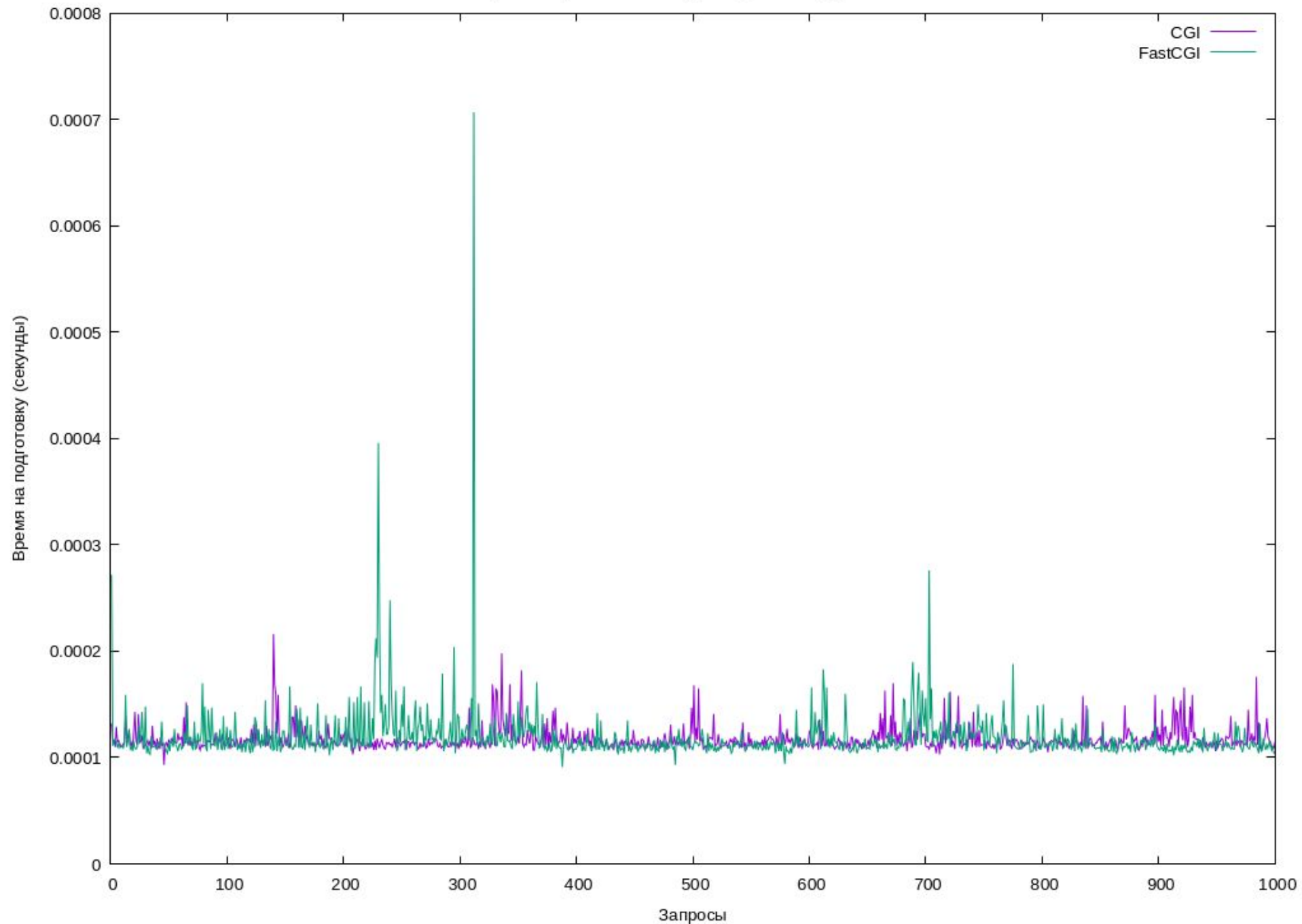
- Время на подключение к серверу
- Время на подготовку к обмену данными
- Время на отдачу данных
- Общее время ответа.

Время подключения к серверу

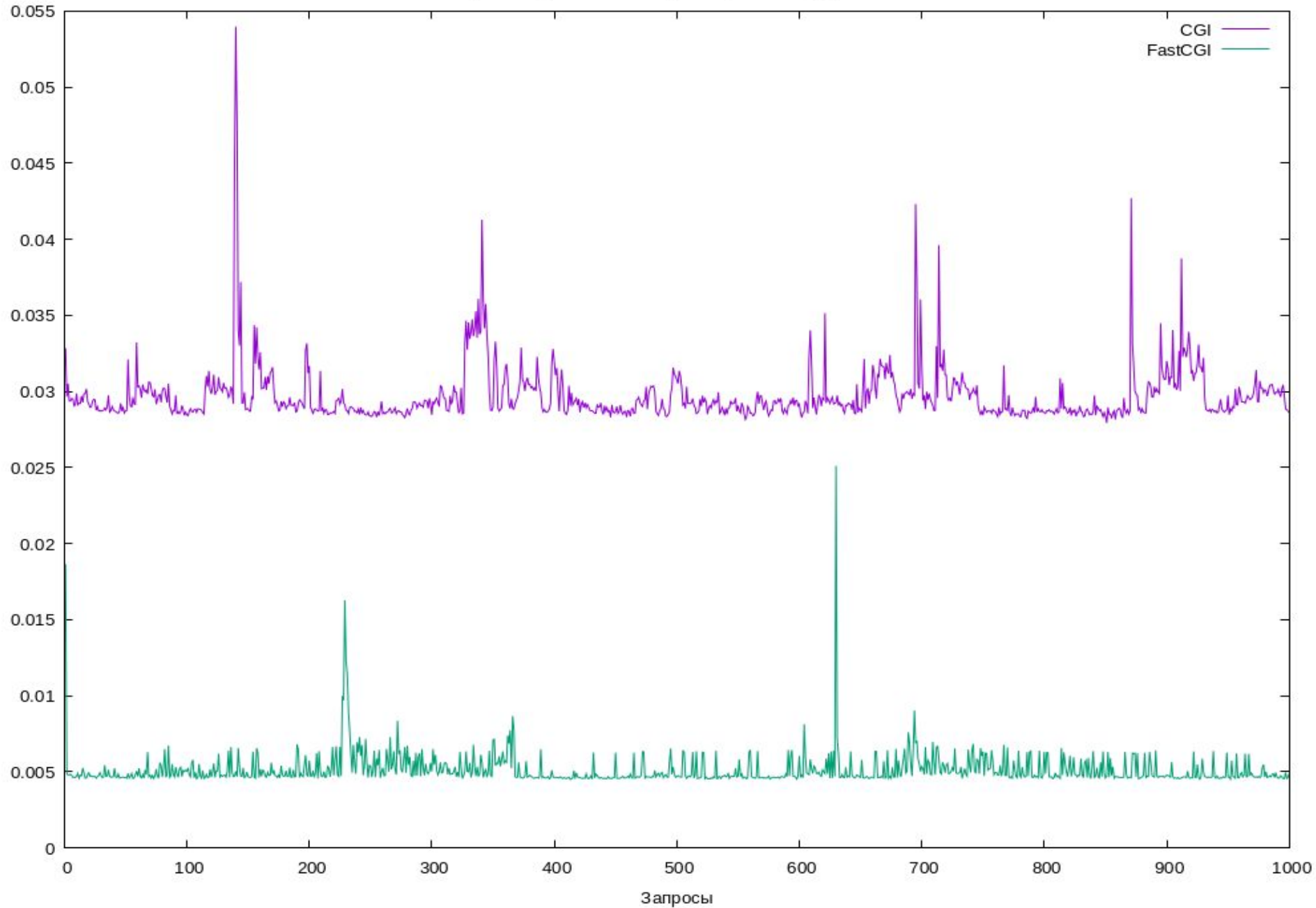




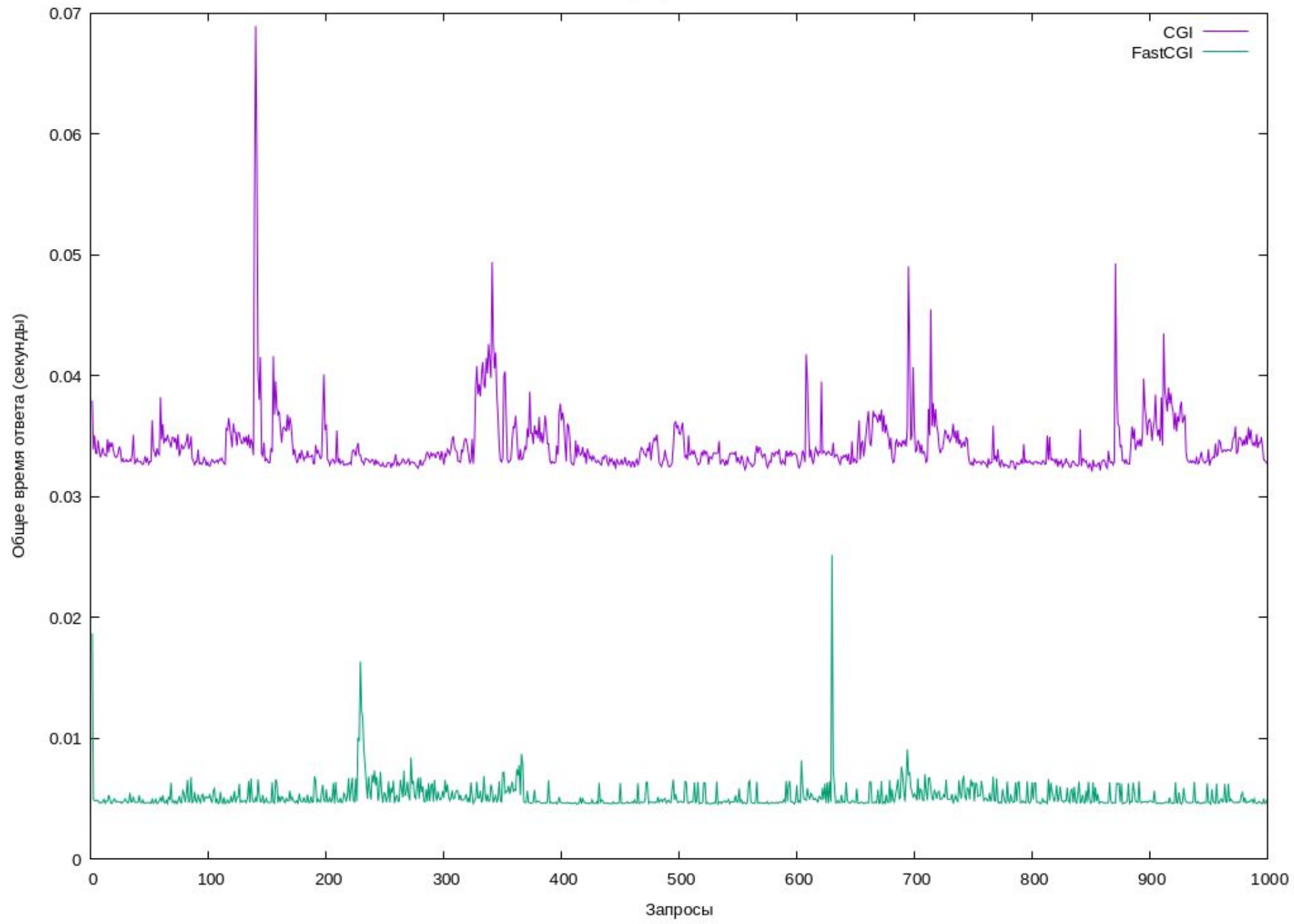
Время затраченное на подготовку к обмену данными



# Время на отдачу данных



Общее время ответа



# Результаты работы.

В работе получены следующие результаты:

1. Сконфигурирован веб сервер, поддерживающий FastCGI для работы с CGI скриптами.
2. Создано тестовое CGI приложение и скрипт для анализа скорости взаимодействия с ним.
3. Произведены исследования и анализ полученных данных, демонстрирующие различия в скорости работы CGI и FastCGI технологий.