

Зимняя школа по параллельному программированию 2019

## **Профилирование фрагментированных программ**

Исполнитель: Патрушева Елизавета

НГТУ, ФПМИ, 3 курс

Руководитель: Перепелкин В. А.

01.02.2019

## **План доклада**

- Постановка задачи
- Идея решения
- Реализация
- Тестирование
- Заключение

## **Постановка задачи**

Профилирование — это специальный режим исполнения фрагментированных программ, при котором сохраняется дополнительная информация об их исполнении. При профилировании фрагментированной программы профилировочный код вставляется во фрагменты кода, таким образом профилирование осуществляется вне зависимости от того, на какой исполнительской системе осуществляется исполнение. Целью работы является создание профилировщика, выполняющего эти функции.

Требования к решению:

- Интерфейс, совместимый с языком С
- Возможность статической линковки с тестом

## **Идея решения**

Для получения профилировочной информации в начале и в конце каждого фрагмента кода вставляется вызов функции профилировщика, фиксирующей начало и конец выполнения фрагмента вычислений. Профилировщик собирает эту информацию и сохраняет в файлы. Скрипт для обработки выполняет статистический анализ полученной информации и представляет ее в виде графиков.

## Реализация

Функции профилировщика, фиксирующие начало и конец выполнения фрагмента вычислений реализованы средствами языка С. Получаемая в результате информация включает в себя:

- имя исполняемой функции;
- время начала и конца выполнения фрагмента кода;
- время процесса, затраченного на исполнении кода;
- номер узла и идентификатор потока, на котором выполнялся фрагмент.

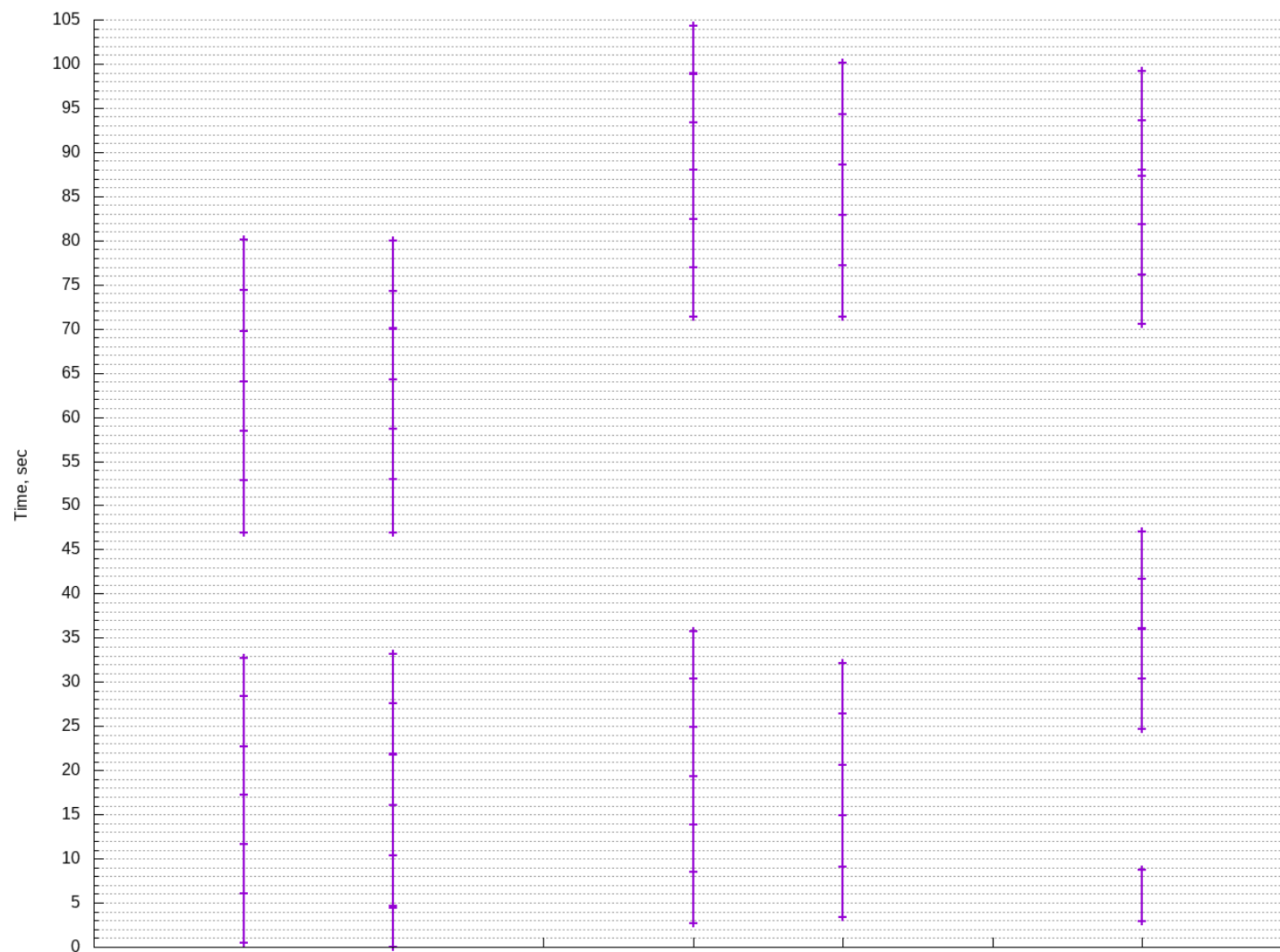
## Реализация

Функция, выполняющая обработку полученной информации, реализована средствами языка Си. В результате ее работы пользователь получает следующие данные (для каждого отдельного узла):

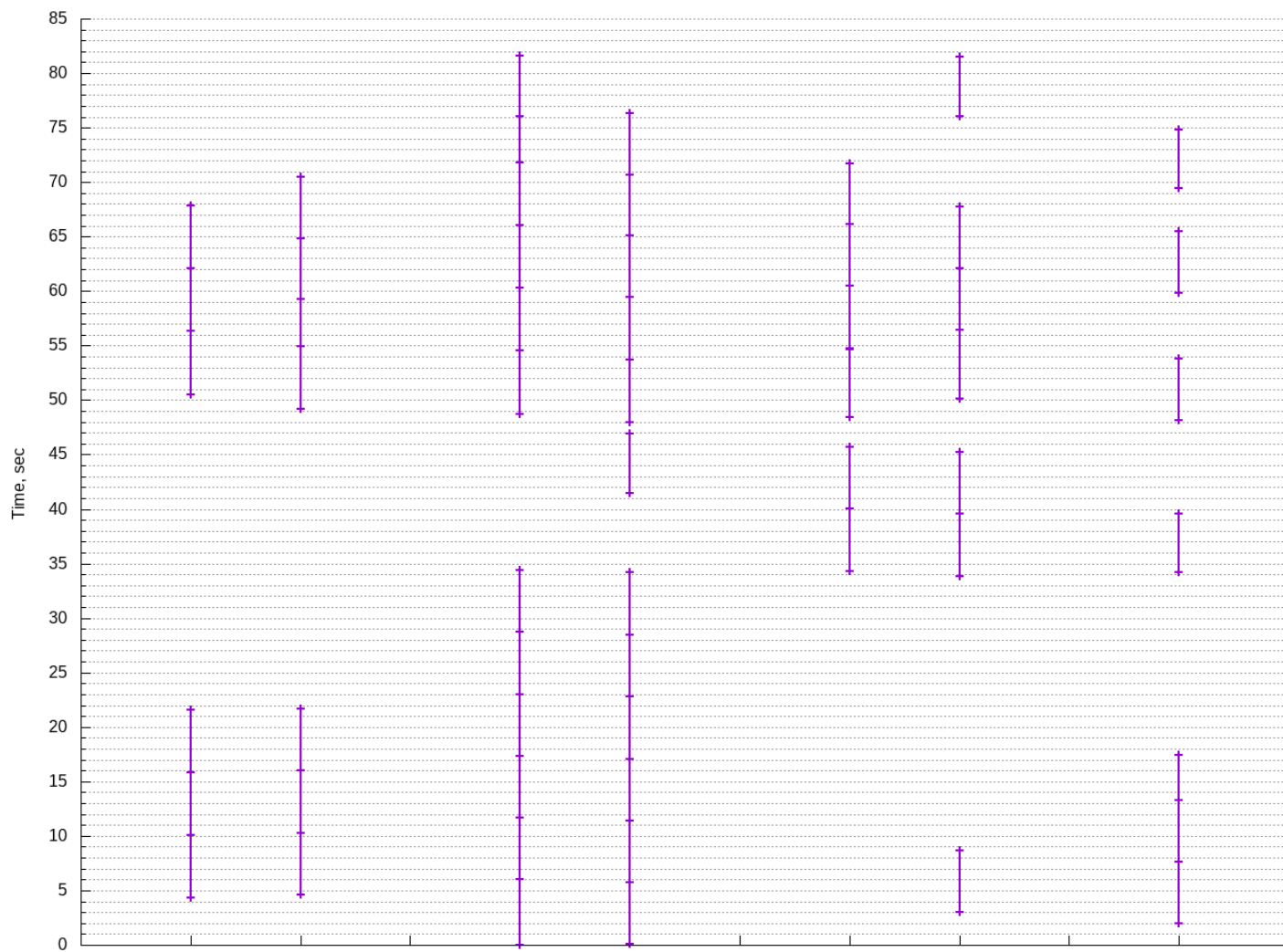
- полезная нагрузка узла в секундах;
- общее время работы узла в секундах;
- загруженность узла в процентах.

Построение графиков на основе полученной информации реализовано при помощи утилиты `gnuplot`.

# Тестирование

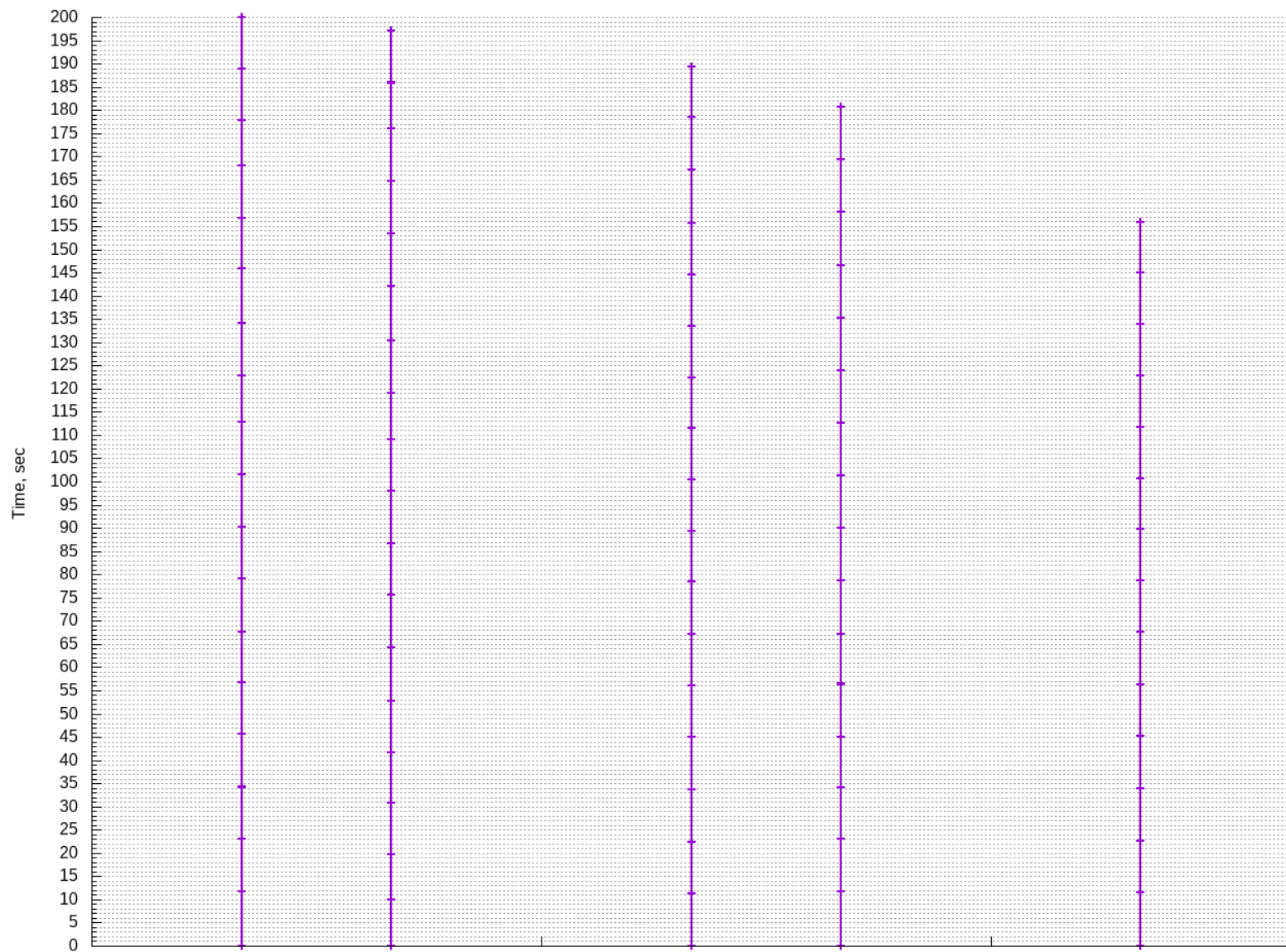


# Тестирование

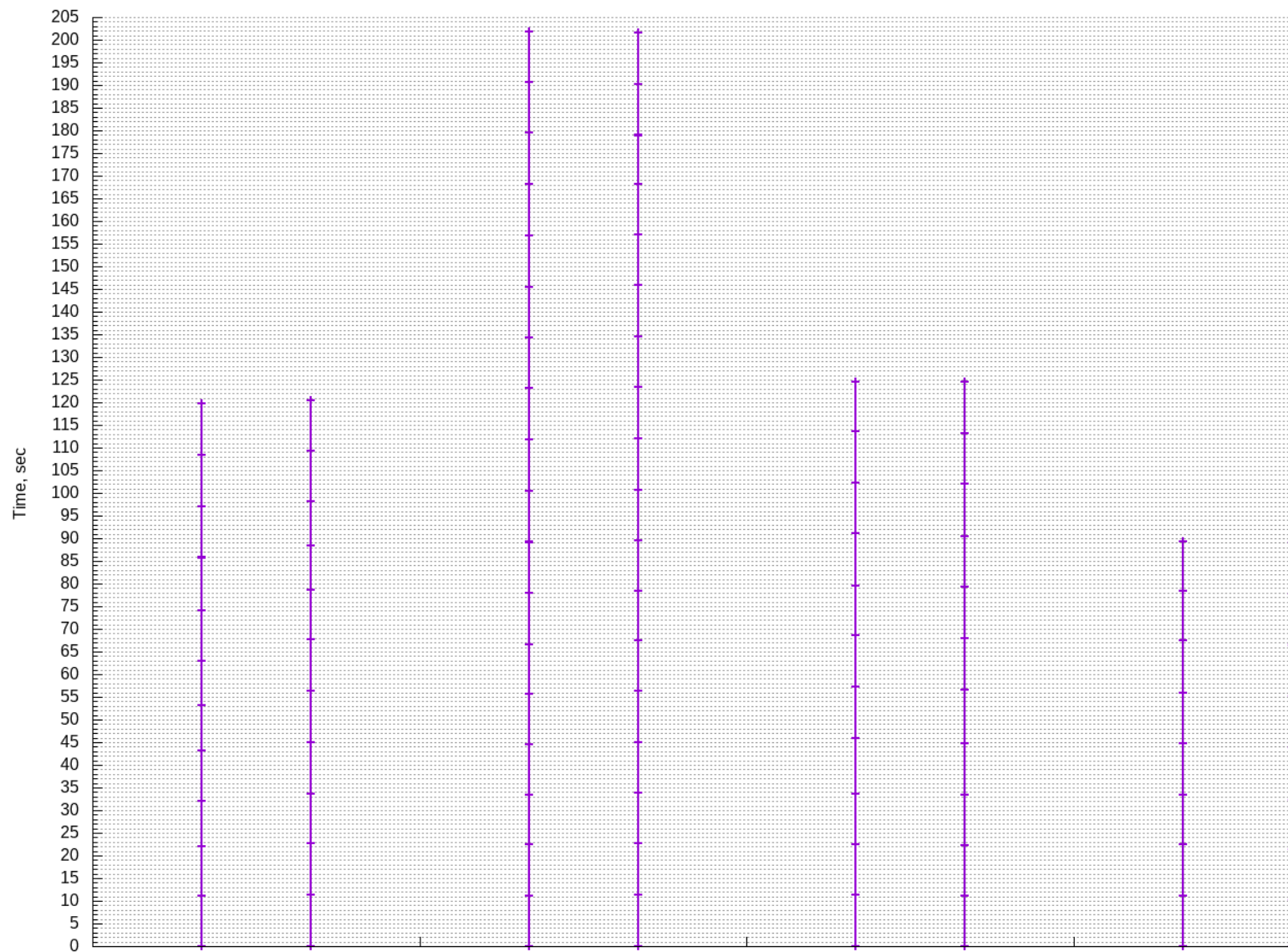




# Тестирование



# Тестирование



## **Заключение**

Был реализован профилировщик, выполняющий функции сбора информации, ее статистической обработки и визуализации. Все требования к решению были соблюдены.

Наглядность полученных результатов позволяет получить более полное представление о результатах работы фрагментированной программы.