

**МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И.УЛЬЯНОВА (ЛЕНИНА)**

Кафедра ВТ

**ОТЧЕТ
по лабораторной работе №2
по дисциплине «Параллельные алгоритмы и структуры
данных»
Тема: Линейные разделяемые структуры данных**

Студенты гр. 8307

Николаев Д.Е.

Репин С.А.

Преподаватель

Пазников А.А.

Санкт-Петербург
2022

СОДЕРЖАНИЕ

1. Задание	3
2. Выполнение работы	3
3. Разделение работы в бригаде	5
Список использованных источников	5

1. ЗАДАНИЕ

Реализовать lockless стек на основе алгоритма Elimination back-off stack.

2. ВЫПОЛНЕНИЕ РАБОТЫ

Алгоритм Elimination back-off stack написан на языке C++20 для ОС Linux x64 на основе публикации [1].

Elimination back-off – структура данных типа стек, имеющая свойства неблокируемости, линерализуемости и масштабируемости. В простом lockless стеке операция CAS может быть очень затратной и слабо масштабируемой с ростом числа потоков. Применение специальной back-off стратегии – алгоритма, исполняющегося после неудачного CAS и призванного разгрузить процессор – позволяет устранить этот недостаток. В данном случае в качестве такой стратегии используется специальный массив, содержащий дескрипторы выполняемых разными потоками операций, что дает возможность в некоторых случаях взаимно поглотить операции pop и push – не совершать их в реальности.

Кроме того был реализован тест, запускающий N потоков, каждый из которых выполняет push- и pop-операции над стеком, порядка десятков тысяч раз. По завершению тестовая программа сверяет полученное состояние стека с ожидаемым. Измеряется время с момента запуска потоков до их завершения.

На графиках ниже показана зависимость времени выполнения от числа потоков для двух реализаций тестовой программы: с использованием Elimination back-off stack и стека std::stack с мьютексом. Из графика следует, что с ростом contention, когда все больше потоков соревнуются за доступ к критической секции, время выполнения обеих реализаций сначала сравниваются, а затем EBS начинает обгонять реализацию на мьютексе (сравниваются на 15-16 потоках, EBS обгоняет с 25). А также кривая времени для lockless реализации намного более гладкая. Таким образом, применение Elimination back-off stack оправдано при большом количестве потоков.

Параметры стенда представлены в таблице ниже:

Название	Значение
ОС	Arch Linux (Linux 5.16 x64)
Процессор	Intel Core i7 11700
Число ядер	16 (8)

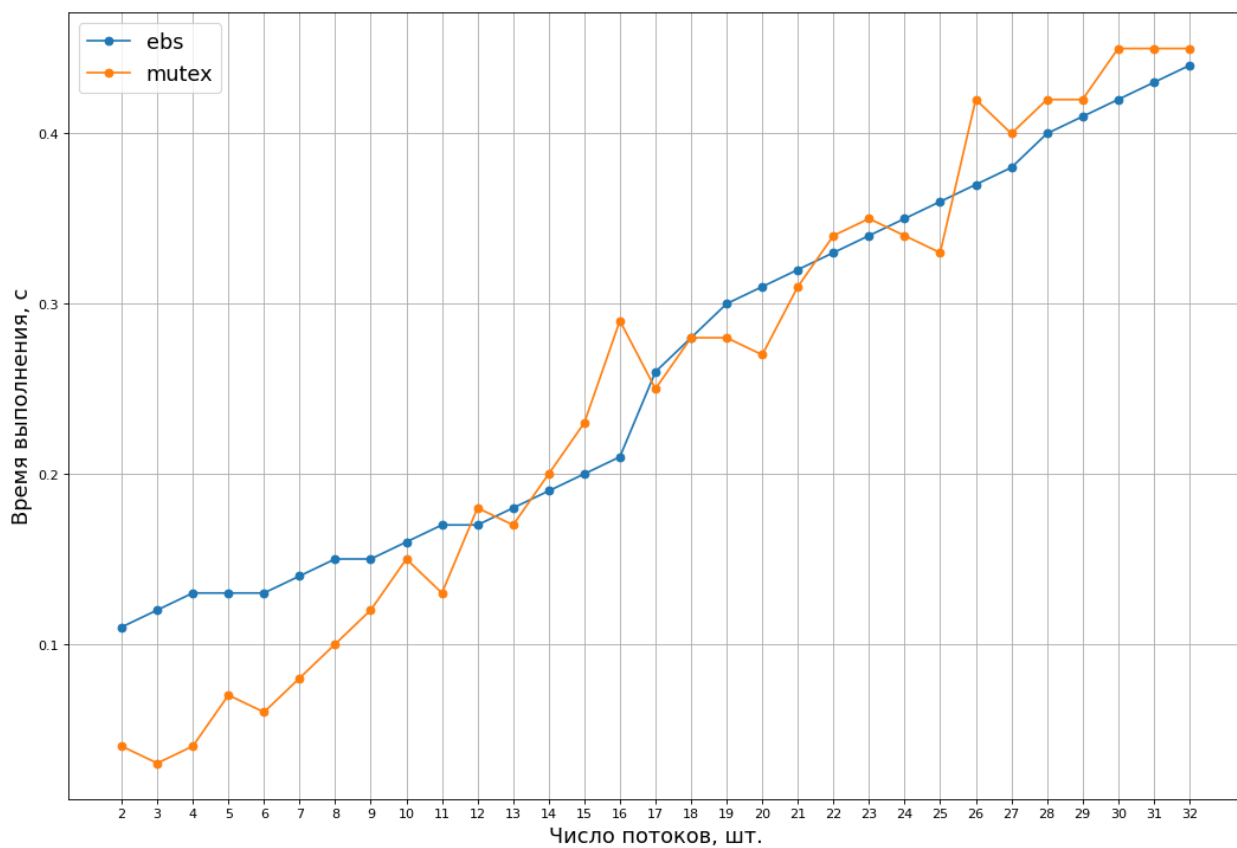


Рис. 2.1 График для 2-32 потока

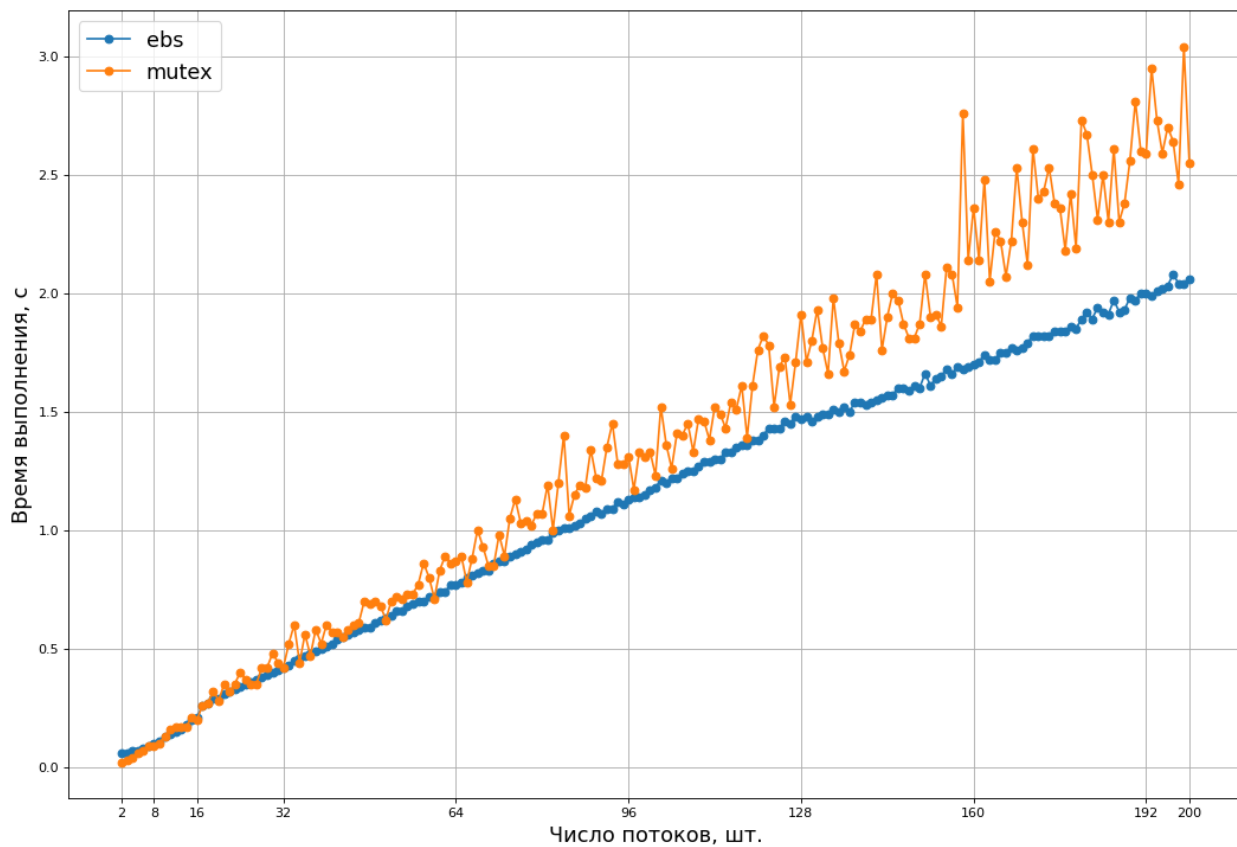


Рис. 2.2 График для 2-200 потоков

3. РАЗДЕЛЕНИЕ РАБОТЫ В БРИГАДЕ

Вместе выяснили и обсудили суть алгоритма, особенности его реализации. Определились с интерфейсом библиотеки и ее тестированием.

Дмитрий Николаев выполнил разработку тестовой утилиты, скриптов для анализа результатов. Выполнил отладку и рефакторинг. Сделал реализацию на мьютексах.

Степан Репин реализовал Elimination back-off stack.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. D. Hendler, N. Shavit, L. Yerushalmi. A Scalable Lock-free Stack Algorithm
// In Proceedings of the sixteenth annual ACM symposium on Parallelism
in algorithms and architectures (SPAA '04). Association for Computing
Machinery. - 2004. - P. 206–215.