

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

Кафедра ВТ

**ОТЧЕТ
по лабораторной работе №1
по дисциплине «Параллельные алгоритмы и структуры
данных»
Тема: Атомарные снимки, блокировки**

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	4
Список использованных источников	4

1. ЗАДАНИЕ

Реализовать алгоритм Remote Core Blocking. Разработать простой тест. Оценить эффективность.

2. ВЫПОЛНЕНИЕ РАБОТЫ

Алгоритм Remote Core Locking (RCL) написан на языке C11 для ОС Linux x64 на основе публикации [1].

RCL – это алгоритм блокировки, обеспечивающий высокую производительность на многоядерных системах за счет исполнения критических секций на специализированных ядрах, заменяя захват блокировки на вызов процедуры. Такой подход повышает эффективность использования L1 и L2 кэшей процессора.

Кроме того был реализован тест, запускающий N потоков, каждый из которых инкрементирует общую переменную M раз на величину S . По завершению тестовая программа сверяет полученное значение переменной с ожидаемым ($N \cdot M \cdot S$). Измеряется время с момента запуска потоков до их завершения.

На графике ниже показана зависимость времени выполнения от числа потоков для двух реализаций тестовой программы: с использованием RCL и мьютексов. Из графика следует, что с ростом contention, когда все больше потоков соревнуются за доступ к критической секции, RCL начинает выигрывать реализацию на мьютексах. При малом числе потоков мьютексы оказываются быстрее из-за накладных расходов RCL.

Параметры стенда представлены в таблице ниже:

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

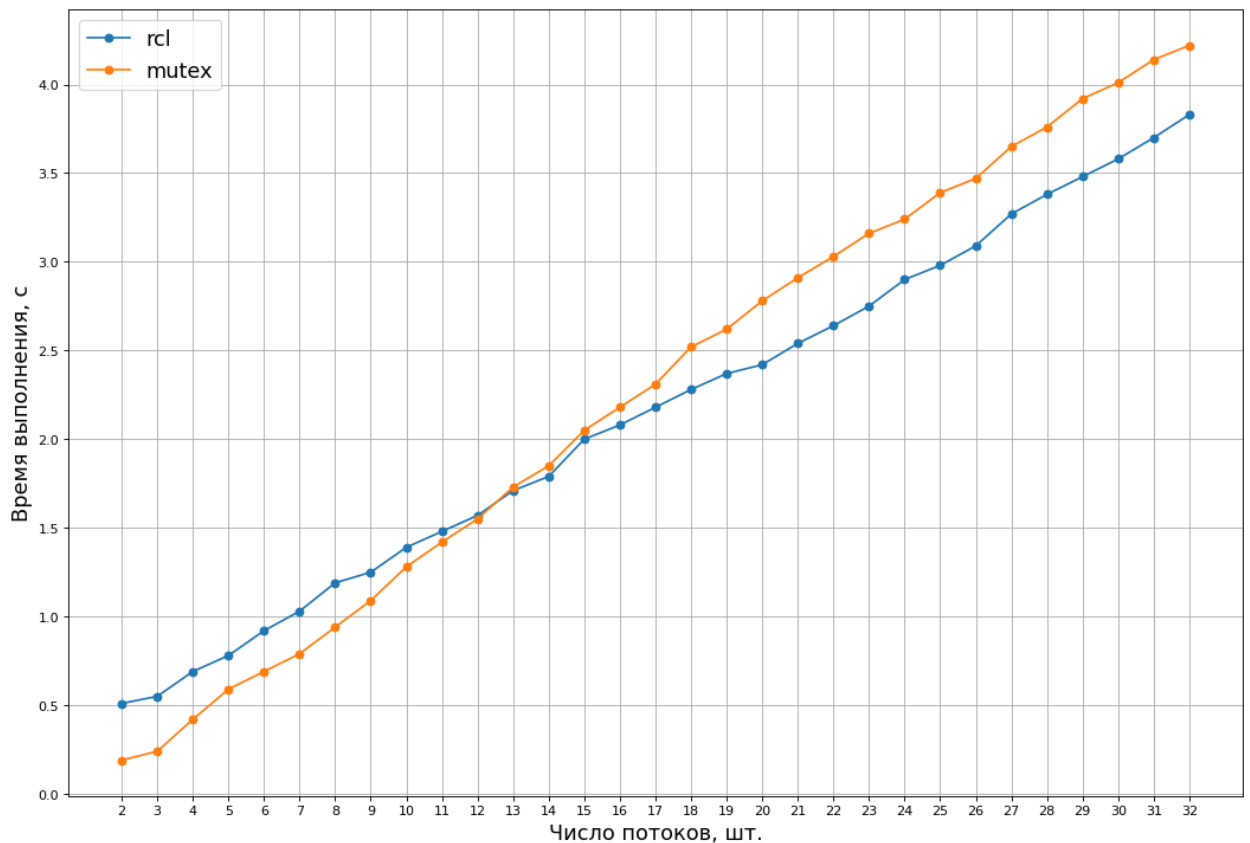


Рис. 2.1

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 и Servicing Thread.

Степан Репин реализовал остальные части библиотеки: Manager Thread, Backup Thread, инициализацию, запуск. А также выполнил отладку.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. J.-P. Lozi, F. David, G. Thomas, J. Lawall, G. Muller. Remote Core Locking: Migrating Critical-Section Execution to Improve the Performance of Multithreaded Applications // USENIX ATC 12. - 2012. - P. 65-76