

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

ОТЧЕТ

по лабораторной работе №2

по дисциплине «Параллельные алгоритмы»

**Тема: Реализация потокобезопасных структур данных с
блокировками**

Студент гр. 1304

Кардаш Я.Е.

Преподаватель

Сергеева Е.И.

Санкт-Петербург

2024

Цель работы.

Изучить реализацию потокобезопасных структур данных с блокировками с «грубой» и «тонкой» блокировкой, используя шаблон «производитель - потребитель».

Задание.

Реализовать итерационное (потенциально бесконечное) выполнение подготовки, обработки и вывода данных по шаблону “производитель-потребитель” (на основе лаб. 1 (части 1.2.1 и 1.2.2)). Количество производителей и потребителей должно быть изменяемым. Обеспечить параллельное выполнение потоков обработки готовой порции данных, подготовки следующей порции данных и вывода предыдущих полученных результатов. Использовать механизм “условных переменных”.

2.1 Использовать очередь с “грубой” блокировкой.

2.2 Использовать очередь с “тонкой” блокировкой

Выполнить тестирование п. 2.1 и 2.2, убедиться в корректности результатов.

Сравнить производительность 2.1. и 2.2 в зависимости от количества производителей и потребителей.

Выполнение работы.

В ходе работы программы производитель генерирует две матрицы заданного размера и записывает их в очередь в виде кортежа `std::tuple`.

Потребитель забирает этот кортеж, перемножает матрицы и записывает результат в файл.

Перемножение матриц выполняется по блокам, на каждый блок выделяется свой поток. Размеры матриц и блоков для удобства установлены таким образом, что при перемножении выделяется 4 потока.

Каждая очередь ограничена размером в 5 элементов

Очередью с «грубой» блокировкой будем считать такую очередь, которая применяет мьютекс для защиты всей структуры данных `std::queue`. Очередью с

«тонкой» блокировкой будем считать такую очередь, которая применяет мьютекс для каждого хранящегося элемента данных в `std::queue`. Каждый из производителей и потребителей делает по 10 операций генерации матриц и их перемножения и записи в файл соответственно.

Производительность для данного задания измеряется как время выполнения программы.

На рисунке 1 отображена зависимость времени от количества `producer` и `consumer` для грубой и тонкой блокировок

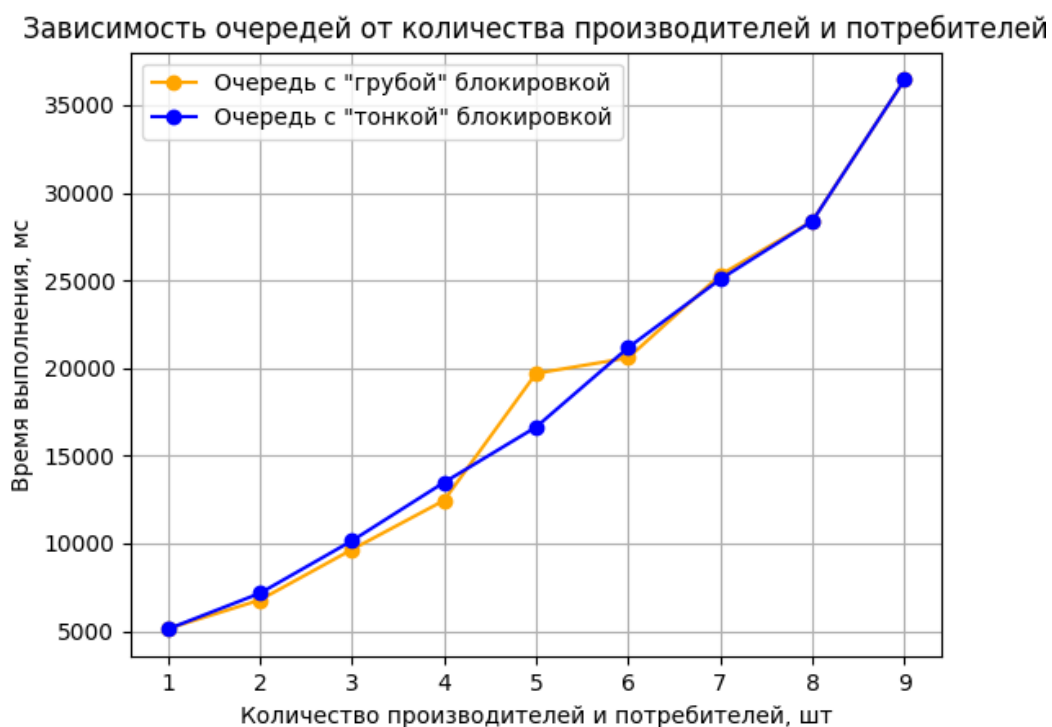


Рисунок 1.

По графику видно, что при приближении к предельной нагрузке (6 потоков) тонкая блокировка выигрывает в производительности, при превышении этого значения разница между грубой и тонкой блокировкой незначительна.

Выводы.

Было исследовано, изучена реализация потокобезопасных структур данных с блокировками с «грубой» и «тонкой» блокировкой, используя шаблон «производитель-потребитель». В результате сравнения очередей в зависимости

от количества потребителей и производителей установлено, что очередь с «грубой» блокировкой менее производительна чем очередь с «тонкой» блокировкой, так как при использовании «грубой» блокировки потоки-потребители будут ждать освобождения целой очереди и простаивать, в то время как при использовании «тонкой» блокировки потоки-потребители будут выполнять больше работы и меньше простаивать, так как гранулярность блокировок мельче и больше работы выполняется не под защитой блокировок.