

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

ОТЧЕТ
по лабораторной работе №3
по дисциплине «Параллельные алгоритмы»
Тема: Реализация параллельной структуры данных с тонкой
блокировкой.

Студент гр. 9304

Попов Д.С.

Преподаватель

Сергеева Е.И.

Санкт-Петербург

2022

Цель работы.

Реализовать корректную работу потоков используя шаблон “производитель-потребитель”.

Задание.

Обеспечить структуру данных из лаб.2 как минимум тонкой блокировкой (*сделать lock-free). Протестировать доступ в случае нескольких потоков-производителей и потребителей. Сравнить производительность со структурой с грубой синхронизацией (т.е. с лаб.2).

В отчёте сформулировать инвариант структуры данных.

Выполнение работы.

Изменению подвергся класс *CustomQueue*, который теперь представляет из себя *lock-free* структуру, содержащую указатели на первый и последний элемент и контейнер для хранения данных *Node*.

Метод работы *push*:

1. Создает новую ячейку *Node* с данными.
2. Заходит в бесконечный CAS цикл.
3. Получает текущий адрес хвоста.
4. Пытается добавить в хвост новый элемент:
 1. Если получилось, то обновляет значение хвоста структуры и выходит из цикла.
 2. Если нет, значит другой поток уже обновил хвост, пытается сдвинуть хвост и повторяет попытку вставки элемента (возвращаемся в пункт 2).

Метод работы *pop*:

1. Заходит в бесконечный CAS цикл.
2. Получает значения хвоста, головы и следующего элемента головы.
3. Проверяет совпадают ли хвост и голова:
 1. Если совпадают:

1. Проверяет следующий элемент головы, если его нету, то возвращаемся к началу цикла (пункт 1)
2. Если следующий элемент головы есть, значит другой поток не успел подвинуть указатель, пытается сдвинуть самостоятельно и возвращается к началу цикла (пункт 1).
2. Если голова и хвост не совпадают, значит очередь не пуста, пытается обновить значение головы, если получилось, то возвращает данные.

Исследование зависимости между количеством потоком, размерами входных данных и параметрами вычислительной системы.

Исследование для одного потребителя и производителя.

Таблица 1 - Сравнение размера входных данных и времени вычисления для одного потока при грубой синхронизации структуры:

Время вычисления(миллисек.)	Размер входных данных
29	1000 x 1000
269	5000 x 5000
1022	10000 x 10000

Таблица 2 - Сравнение размера входных данных и времени вычисления для одного потока при тонкой синхронизации структуры:

Время вычисления(миллисек.)	Размер входных данных
27	1000 x 1000
682	5000 x 5000
2806	10000 x 10000

Исследование для нескольких потребителей и производителей.

В таблице 3 представлено общее время работы программы в зависимости от кол-ва производителей и потребителей при грубой синхронизации структуры:

Один производитель и два потребителя (миллисек.)	Два производителя и один потребитель (миллисек.)	Два производителя и два потребителя (миллисек.)	Размер генерируемых данных
1099	1085	659	6 * (5000*5000)
7157	7029	4685	12 * (10000*10000)

В таблице 4 представлено общее время работы программы в зависимости от кол-ва производителей и потребителей при тонкой синхронизации структуры:

Один производитель и два потребителя (миллисек.)	Два производителя и один потребитель (миллисек.)	Два производителя и два потребителя (миллисек.)	Размер генерируемых данных
2987	7343	1775	6 * (5000*5000)
24151	14316	16092	12 * (10000*10000)

Выводы.

В ходе выполнения лабораторной работы была реализована программа на языке программирования C++ для попарного сложения матриц, использующая в качестве структуры данных *lock-free* очередь.