**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 структур данных без блокировок**

|  |  |  |
| --- | --- | --- |
| Студент гр. 0304 |  | Аристархов И.Е. |
| Преподаватель |  | Сергеева Е.И. |

Санкт-Петербург

2023

**Цель работы.**

На практике освоить методы создания алгоритмов без блокировок.

**Поставленные задачи.**

Реализовать очередь, удовлетворяющую lock-free гарантии прогресса.

Протестировать доступ к реализованной структуре данных в случае нескольких потоков производителей и потребителей.

**Выполнение работы.**

Для создания очереди без блокировок использовалась Очередь Майкла и Скотта. Схема освобождения памяти была взята из книги «C++ Concurrency in Action».

Устройство очереди осталось таким же, как и в прошлой работе, за исключением того, что голова теперь указывает на фиктивный элемент, а настоящие данные в следующем узле.

Очередь Майкла и Скотта подразумевает использования механизма помощи в операциях pop и push. По сути, если одна операция обнаруживает промежуточное состояние объекта из-за работы другой операции, то она помогает его завершить.

С точки зрения освобождения памяти используется механизм сохранения узлов до момента, когда их возможно освободить. Удалять указатели можно только одним потоком, поэтому ожидается момент, когда лишь один поток будет находится в pop для проведения удаления.

**Анализ.**

В очереди требуется сохранить следующие инварианты:

* Хвост очереди всегда в конце и его next указатель пустой.
* Голова очереди всегда в начале и на нее никто не указывает.
* Голова всегда стоит либо на том же месте, что и хвост, либо хвост стоит дальше в очереди, чем голова.

Компьютер, на котором производились замеры, имел процессор 4 физическими ядрами и 8 потоками (по 2 потокам на ядро).

Т.к. данная программа потенциально может работать бесконечно, было решено дать ей работать 1 секунду и считать максимальное количество успешно проведенных операций (от генерации до вывода).

Производилось умножение матриц 40х40 с 5 потоками на умножение.

При работе получились следующие результаты:

|  |  |  |  |
| --- | --- | --- | --- |
| Количество работающих двойных пар | Грубая блокировка | Тонкая блокировка | Без блокировок |
| 1 | 132 | 141 | 183 |
| 5 | 375 | 302 | 347 |
| 10 | 403 | 305 | 451 |
| 20 | 454 | 205 | 331 |
| 50 | 286 | 323 | 229 |

Из таблицы видно, что очередь без блокировок иногда дает лучше результаты, чем очереди с блокировкой, на небольшом количестве поток, но начинает работать хуже с увеличением потоков.

Это ожидаемо, ведь очередь без блокировок гарантирует не быстродействие, а успешную работу очередей без блокировок за определенное количество повторений. К тому же производительность сильно зависит от реализации.

Еще одним фактором уменьшения скорости играет сборщик мусора, которому требуется небольшая нагрузка на pop для эффективной работы. Иначе он будет просто набирать мусор, который сможет освободить только при снижении нагрузки.

**Выводы.**

В результате выполнения лабораторной работы была разработана программа производящяя умножение матриц с применением паттерна «Производитель — Потребитель». Изучалась очередь без блокировок в сравнении с очередями с грубой и тонкой блокировками.

Опыты показали, что очередь без блокировок не гарантирует быстродействие и на самом деле часто уступает по производительности очередям с блокировками.