

Исследование различных методов разработки неблокирующих структур данных

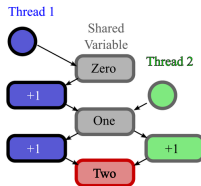
Семен Пьянков, 576 группа
Научный руководитель: Бабичев С.Л., к.ф.-м.н.

Кафедра информатики
Московский Физико-Технический Институт
(национальный исследовательский университет)

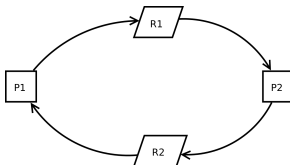
25 июня 2019 г.

Проблемы параллельного программирования

- Race conditions



- Deadlock

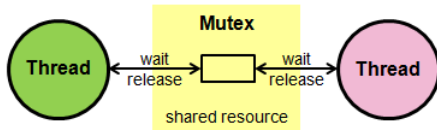


- Live lock

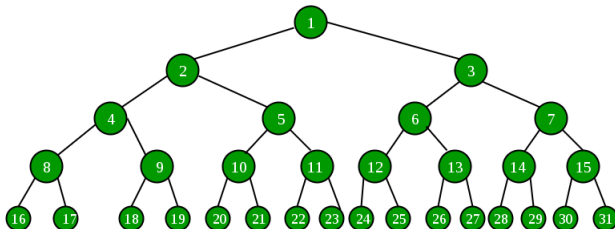
Различные подходы к построению структур данных

Блокирующие структуры данных

- Mutex



- Гранулярность

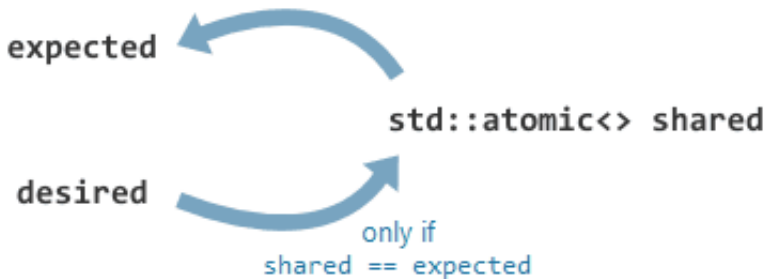


- Возможны большие затраты времени

Неблокирующие структуры данных

Оптимистический подход

- Оптимизм — надеемся, что другие потоки не успели изменить данные по этому же указателю
- Операция CAS



Неблокирующие структуры данных

Способы решения проблемы ABA

Программный механизм

Hazard Pointers

Библиотека libcds)

Аппаратный механизм

Транзакционная память

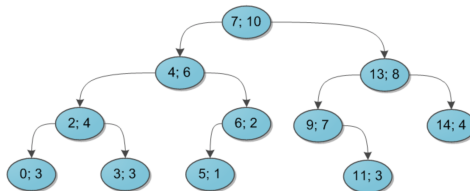
Intel TSX, RTM

Исследуемые структуры данных

- Стек, основанный на однонаправленном списке



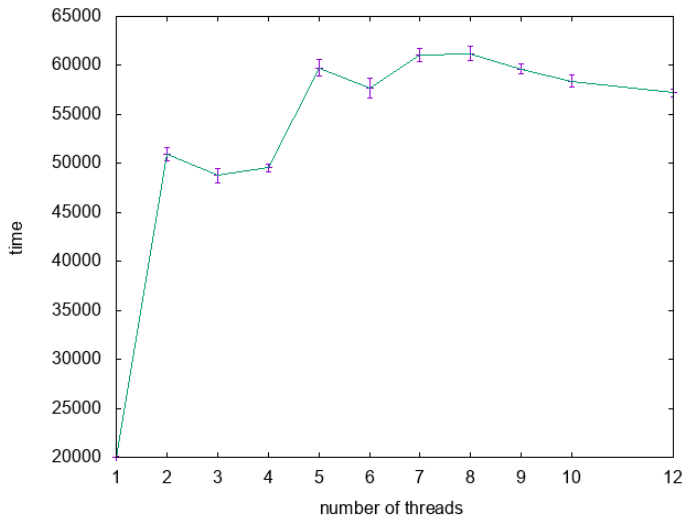
- Декартово дерево (Treap)



- Реализации стека с помощью mutex, CAS, Hazard Pointers, RTM
- Реализации Treap с помощью mutex и RTM
- Intel TSX RTM: XBEGIN, XEND, XABORT и XTEST
- Intel Core i7 8700B, Coffee Lake с поддержкой Intel TSX
- C++11 с поддержкой `std::thread`, компилятор g++ 8.3.0
- Замеры времени с помощью `rdtsc`
- Вставка и удаление элементов из случайно сгенерированных данных

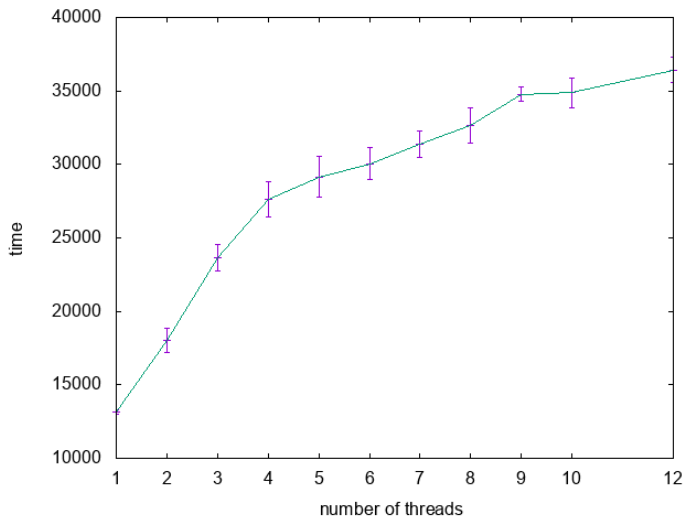
Результаты исследования

Стек с блокировками



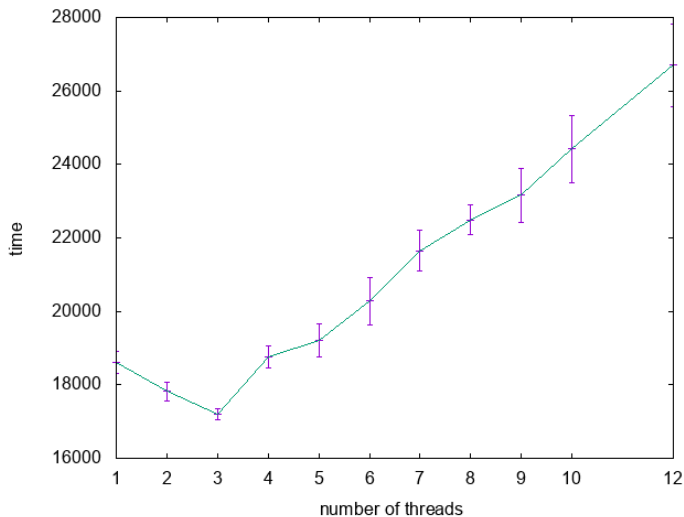
Результаты исследования

Стек с CAS



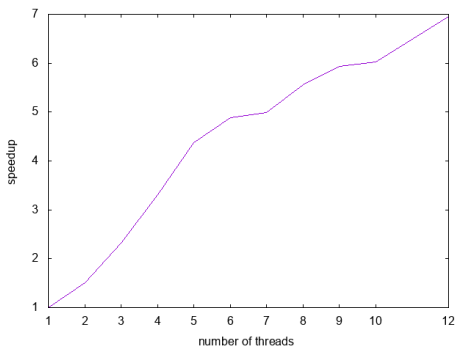
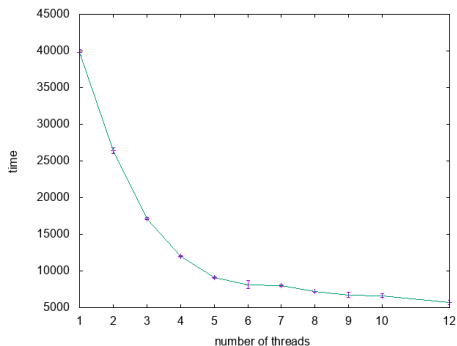
Результаты исследования

Стек с Hazard Pointers



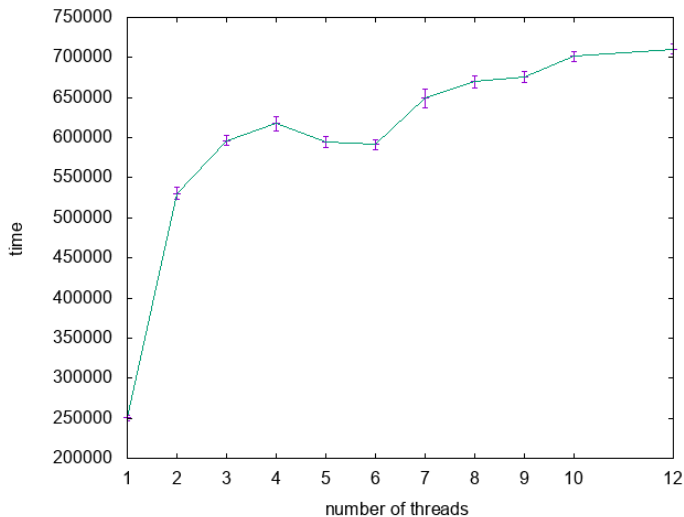
Результаты исследования

Стек с RTM



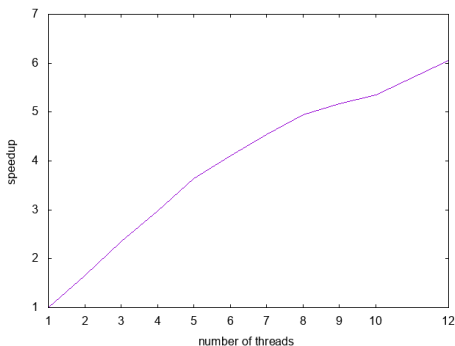
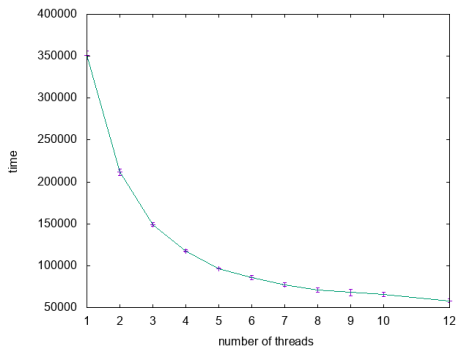
Результаты исследования

Треар с блокировками



Результаты исследования

Treap с RTM



- Изучены различные подходы к построению параллельных структур данных
- Построены различные реализации структур данных стек и Treap
- Было проведено сравнение времени работы построенных структур данных
- Транзакционная память позволяет построить наиболее быстрые неблокирующие структуры данных, что и было продемонстрировано на примере Treap

Спасибо за внимание!