

РЕАЛИЗАЦИЯ ПОТОКОБЕЗОПАСНЫХ СПИСКОВ С ПРОПУСКАМИ БЕЗ ИСПОЛЬЗОВАНИЯ БЛОКИРОВОК

А. Р. ОМЕЛЬНИЧЕНКО, В. А. СМЕРНОВ, А. А. ПАЗНИКОВ

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

Для того чтобы эффективно использовать многопроцессорные вычислительные системы первоначально необходимо задумываться о правильной синхронизации параллельных потоков при доступе к глобальным структурам данных. В настоящее время при разработке параллельных программ для вычислительных систем с общей памятью активно развиваются модели, алгоритмы и программное обеспечение без использования блокировок (lock-free) [1]. Масштабируемость, меньшее время простоя потоков, отсутствие взаимных блокировок (deadlock) и закликиваний (livelock) являются существенными преимуществами такого подхода. Высокая скорость выполнения операций на структурах с неблокирующей синхронизацией [2] не является приоритетным качеством, существующая гарантия продвижения системы в целом куда важнее.

Под неблокирующей синхронизацией понимается механизм взаимодействия нескольких потоков при одновременном доступе к разделяемой памяти, при котором между этими потоками не возникает конфликтов, и все получают свою долю операций. При блокирующей синхронизации если несколько потоков пытаются совершить операции над разделяемой памятью, доступ распределяется таким образом, что активным потоком считается лишь один, остальные ждут своей очереди. С точки зрения производительности такой подход не всегда выгоден, могут возникать ситуации с долгим переключением с одного потока на другой, или некоторые потоки могут вообще не получить свою часть вычислительных операций.

В данной работе для реализации lock-free алгоритмов были выбраны списки с пропусками (skip lists). Выбор данной структуры обоснован схожими показателями производительности с бинарными деревьями поиска [3]. Применение механизмов неблокирующей синхронизации к спискам с пропусками может послужить хорошей альтернативой различным деревьям, широко используемым в настоящее время, таким как красно-черное дерево, АВЛ-дерево, В-дерево.

Списки с пропусками представляют из себя многоуровневый связный список. На нижнем уровне все элементы связаны друг с другом, в каждом следующем уровне элемент появляется с некоторой вероятностью p . Список постоянно сбалансирован, поэтому время на поиск необходимого элемента значительно сокращается.

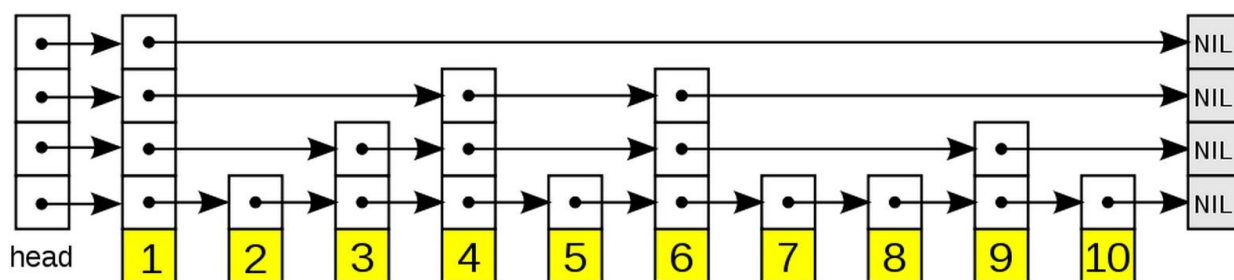


Рис. 1. Структура списков с пропусками

Для проведения экспериментов были выбраны следующие скиплисты с соответствующими способами синхронизации:

- Optimistic skip list – lock (спинлок)
- Fraser skip list – read-modify-write
- ConcurrentSkipListMap – read-modify-write
- No hot spot skip list – read-modify-write
- Rotating skip list – read-modify-write
- Elastic skip list – transaction

Спинлок является более низкоуровневой реализацией мьютекса. Если поток/процессор хочет получить доступ к разделяемой памяти, в переменную атомарно записывается состояние «занято», и до тех пор пока состояние спинлока не изменится, другие потоки/процессоры не могут получить доступ.

Под read-modify-write синхронизацией подразумевается применение различных атомарных операций, таких как CAS (compare and swap), LL/SC(load link / store conditional) и др. В данных экспериментах применялась операция CAS. При изменении области данных у этих данных имеется особый идентификатор, после того как поток поработал над разделяемой памятью, сравниваются (compare) два значения этих идентификаторов, если они разные, то изменения вносятся (swap).

Использование транзакционной памяти подразумевает выделение участков кода в атомарные транзакции, которые выполняются параллельно, организуя таким образом неблокирующую синхронизацию.

Для проведения экспериментов применялся бенчмарк Synchrobench. Время работы 5с. Каждый скиплист тестировался при различных уровнях нагрузки. На рисунке 2а) представлены результаты запуска бенчмарка при нулевом проценте обновляющих операций (вставка, удаление), а на 2б) – результаты при 50% .

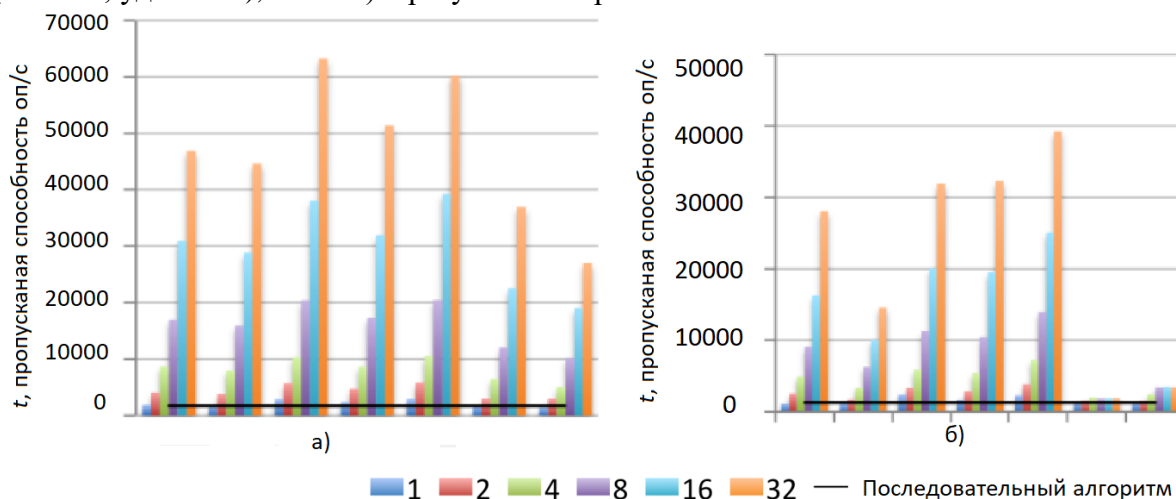


Рис. 2 Результаты запуска бенчмарка для скиплистов.

Списки с пропусками демонстрируют высокую производительность, сравнимую с деревьями. Вполне вероятно, что такие же деревья с неблокирующей синхронизацией могут превзойти данные результаты, но сложности, возникающие при имплементации lock-free механизмов к деревьям, заставляют задуматься о поиске более простых альтернатив, которыми и являются скиплисты. Для оптимальной работы рекомендуется установка вероятностной характеристики p в диапазоне $[0,25;0,5]$ в зависимости от решаемых задач.