УДК 004.021

УНИВЕРСАЛЬНЫЙ МЕХАНИЗМ КОЛЛАБОРАТИВНЫХ ОПЕРАЦИЙ ДЛЯ МНОГОПОТОЧНЫХ СТРУКТУР ДАННЫХ НА ЯЗЫКЕ JAVA

Коротченко Д.С. (СПбГУ)
Научный руководитель – PhD, доцент Аксенов В.Е. (ИТМО)

Введение. Из-за всё возрастающей нагрузки в современных программных системах весомую роль играют многопоточные структуры данных, помогая обрабатывать запросы к такой системе параллельно. Различные оптимизации работы с такими структурами широко исследуются в настоящее время [1]. Во многих многопоточных структурах данных некоторые операции выполняются в блокирующем на запись (модификацию) режиме, при этом исполнение такой операции может занимать значительное время. Чаще всего выполнение таких операций блокирует до своего завершения все потоки, запросившие операцию модификации структуры. Примером такой блокирующей операции может служить процесс перестройки структуры HashMap. В такой структуре все элементы разделены на корзины и при превышении определенного порога количество корзин увеличивается, а элементы перемещаются между корзинами. Такой процесс обычно подразумевает блокировку всех корзин в структуре на запись, создание нового набора корзин, перекладывание элементов структуры из старых корзин в новые, а только после этого разблокировку структуры и выполнение всех заблокированных операций модификации уже на новом наборе корзин. Другой пример такой операции — снятие снимка текущего состояния структуры. Самый простой способ получить корректный снимок — заблокировать все операции модификации, сделать снимок (например, путём создания поэлементной копии структуры), а затем разблокировать и завершить все операции модификации.

Основная часть. Одной из идей для ускорения выполнения таких операций является её распараллеливание. Для более эффективного распараллеливания могут быть предложены разные варианты [2], однако все они исходно подразумевают в конечном итоге использование выделенного пула потоков для выполнения распараллеленной операции. Это не самый эффективный подход, так как потоки, ожидающие возможности выполнить модифицирующие операции, продолжат простаивать в заблокированном режиме на время выполнения операции.

Коллаборативный подход к выполнению таких операций подразумевает иное решение. При коллаборативном подходе такие операции представляются в виде задач, которые могут быть поделены на более мелкие задачи. В случае, если один из потоков пытается выполнить модифицирующую операцию, то вместо того, чтобы в заблокированном состоянии ожидать завершения выполнения операции, поток присоединяется к выполнению задач, при необходимости разделяя одну из имеющихся задач на несколько и забирая одну из новых получившихся задач. Такой подход позволяет ускорить завершение операции, и тем самым уменьшить время блокировки операций модификации, а главное — использовать потоки, которые заблокированы из-за выполнения операции, для её более быстрого завершения. Однако в то же время приводит к накладным расходам на создание задач и на их разделение на более мелкие задачи, а также на управление распределением задач между потоками.

Для многих операций уже были предложены подобные механизмы, к примеру, такой подход применим к перестроениям различных структур (например, HashMap [3] или деревьев поиска [4]), и одной из целей работы являлось создание универсального механизма, позволяющего легко начать исполнять различные операции в коллаборативном режиме. Такой механизм был разработан, после чего был применен к некоторым популярным операциям, например:

1. Задача снятия снимка текущего состояния широко исследуется [5], а применение коллаборативного подхода выглядит здесь естественным. Как для такой задачи будет выглядеть задача и процесс разбиения на меньшие задачи? Исходной задачей является копирование в результирующую структуру всех элементов оригинальной структуры. При

разделении на более мелкие задачи достаточно разделить элементы, которые необходимо скопировать, на несколько групп, которые можно обрабатывать независимо, после чего обработка каждой из групп и становится отдельной (более мелкой) задачей.

2. Коллаборативный подход также был применён к операции перестройки HashMap. Исходная задача в этой операции состоит в перемещении элементов из всех исходных корзин в соответствующие этим элементам новые корзины, а разбиение на меньшие задачи заключается в разделении исходных корзин, относящихся к текущей задаче, на группы, которые будут обработаны независимо, каждая в своей задаче.

Выводы. Был предложен и реализован универсальный механизм коллаборативных операций, который может быть применён к различным многопоточным структурам данных. Также были проведены эксперименты с различными структурами данными, показавшие преимущества использования коллаборативных операций по сравнению с классическим подходом.

Список использованных источников:

1. Matthew Rodriguez, Michael Spear. Optimizing Linearizable Bulk Operations on Data Structures // In Proceedings of the 49th International Conference on Parallel Processing (ICPP'20). Association for Computing Machinery, New York, NY, USA — 2020 — Article 24, 1–10.
2. Anton A. Malakhov. Per-bucket concurrent rehashing algorithms // 2015 — URL: https://arxiv.org/abs/1509.02235
3. Tobias Maier, Peter Sanders, Roman Dementiev. Concurrent hash tables: fast and general?(!) // In Proceedings of the 21st ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP'16), Barcelona, Spain — 2016 — Article 34.
4. Trevor Brown, Aleksandar Prokopec, Dan Alistarh. Non-blocking interpolation search trees with doubly-logarithmic running time // In Proceedings of the 25th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP'20), San Diego, California, USA — 2020 — 276–291.
5. Guy E. Blelloch, Yuanhao Wei. VERLIB: Concurrent Versioned Pointers // In Proceedings of the 29th ACM SIGPLAN Annual Symposium on Principles and Practice of Parallel Programming (PPoPP'24). Association for Computing Machinery, New York, NY, USA, — 2024 — 200–214.