УДК 004.051

ВНЕДРЕНИЯ СЕКЛОКА В КОНКУРЕНТНУЮ ХЕШ ТАБЛИЦУ С КУКУШКИНЫМ ХЕШИРОВАНИЕМ

Давыдов А.В. (Университет ИТМО), **Малахов А.А.** (ООО "Техкомпания Хуавей")
Научный руководитель – доцент, PhD Аксенов В.Е.
(Университет ИТМО)

Введение. В современном мире многопоточных процессоров конкурентные хеш таблицы активно используются во многих областях — например в базах данных. Одной из задач в проектировании хеш таблиц является оптимизация их под вид нагрузки, под которой они будут использоваться. В очень большом количестве ситуаций количество читающих операций значительно превышает количество пишущих, например, данные социальной сети(список друзей, записи на стене и т. д.) читаются в разы чаще чем изменяются. Для такого вида нагрузки хочется иметь оптимальное решение.

Мы будем рассматривать оптимизацию одной из самых эффективных конкурентных хеш таблиц — конкурентной хеш таблицы с кукушкиным хешированием. В ней используется схема с двумя индексами бакетов для элемента: основным и альтернативным. Для синхронизации операций с этими бакетами в оригинальном алгоритме используется спинок. И пишущие и читающие операции берут блокировки на бакеты и работают с ними. Для пишущих операций альтернатив этому нет, но для читающих операций является возможным использование какой-либо read-write блокировки вместо спинлока для значительного ускорения случаев, когда количество читающих операций велико.

Основная часть. В качестве read-write блокировки был выбран секлок.

Секлок — read-write блокировка имеющая внутри uint, который инкрементируют потоки(если uint четен то блокировка свободна, иначе занята), но читающие операции могут считывать его эпоху(значение), если она четна, то читать значение в бакете, и если эпоха после этого равна считанной до этого, то чтение из бакета успешно. Таким образом чтение не блокирует другие операции.

Однако секлок накладывает на решение и ограничение — нельзя деаллоцировать память, так как во время чтения, другой поток может инициировать перехеширование всей таблицы и деаллоцировать старые бакеты. Таким образом мы теряем возможность сокращать размер таблицы, но зачастую это и не требуется, как пример можно привести использование хештаблиц в базах данных. Ввиду изложенного выше ограничения, вектор бакетов был заменен на сегментированный вектор, расширение которого не требует реаллокации памяти.

Практически единственным накладным расходом в предложенном решением является то, что для доступа к бакету в сегментированном векторе необходимо вычисление индекса и два прохода по памяти вместо одного у обычного. Для решения этой проблемы хранение в ходе алгоритма индексов блокировок/бакетов было заменено на хранение их указателей, что позволило добиться минимизации накладных расходов до крайне малых значение.

Выводы. Секлок интегрирован в хеш таблицу с кукушкиным хешированием, корректность нового решения подтверждена тестами, а его обоснованность — бенчмарками, в которых новое решение показывает следующие результаты в зависимости от характера нагрузки.

При значительном доминировании модифицирующих операций решение незначительно хуже оригинала, в силу приведенных выше накладных расходов.

При незначительном доминировании модифицирующих операций решение сравнивается по производительности с оригинальным.

При дальнейшем увеличении количества чтений решение начинает доминировать над оригинальным алгоритмом, превосходя его в разы.

Список использованных источников:

1. Xiaozhou Li, David G. Andersen, Michael Kaminsky, Michael J. Freedman. Algorithmic Improvements for Fast Concurrent Cuckoo Hashing // EuroSys '14: Proceedings of the Ninth European Conference on Computer Systems. 2014
2. GitHub [Электронный ресурс] // URL: https://github.com/efficient/libcuckoo

Давыдов А.В. (автор) Подпись

Аксенов В.Е. (научный руководитель) Подпись