

РАЗРАБОТКА АДАПТИВНЫХ К ЗАПРОСАМ И ДАННЫМ СТРУКТУР ПОИСКА

Хлытин Г. А.¹

Науч. руководители – Аксенов В. Е.¹, Шалыто А.А.¹

¹Университет ИТМО

xloutin@yandex.ru

Работа выполнена в рамках НИР №224088 «Многопроцессорные структуры с использованием информации о данных».

Введение

В современных СУБД и in-методу хранилищах упорядоченный поиск по ключу является базовой операцией, а реальная рабочая нагрузка часто оказывается неравномерной: небольшая доля ключей запрашивается существенно чаще остальных, при этом данные со временем изменяются из-за вставок и удалений. Классические деревья поиска обеспечивают предсказуемое время доступа, но не используют знание о распределении ключей и частотах обращений, тогда как обучаемые индексы стремятся аппроксимировать распределение ключей и предсказывать позицию элемента, ускоряя поиск на чтениях. Недостаток прямолинейных обучаемых подходов — сложность поддержания актуальности модели при обновлениях, поскольку требуется переобучение или компромиссные механизмы буферизации вставок и «ленивых» удалений. С другой стороны, самоподстраивающиеся структуры (например, семейство splay-подходов) уменьшают стоимость доступа к «горячим» данным за счет перестроений, но не используют числовую природу ключей и их распределение. Научная проблема состоит в разработке такой структуры поиска, которая одновременно (1) использует обучаемую модель распределения ключей, (2) адаптируется к перекосу запросов, и (3) сохраняет эффективность при динамических обновлениях данных.

Основная часть

Предлагается EIndex — самоподстраивающийся обучаемый индекс, ориентированный на адаптацию как к данным (распределению ключей), так и к запросам (частотам обращений). В основе EIndex лежит идея разбиения отсортированного множества ключей на сегменты, для каждого из которых хранится компактная модель, предсказывающая приблизительную позицию ключа внутри сегмента; после предсказания выполняется локальный поиск в ограниченной окрестности, что позволяет сохранять корректность ответа при контролируемой ошибке модели. Для поддержки вставок новые элементы направляются в локальные структуры (корзины) около соответствующих ключевых границ сегмента, а удаления выполняются «лениво» (через пометки), что снижает стоимость обновлений до момента перестроения.

Ключевое отличие EIndex — способ адаптации верхнего уровня структуры к скошенной нагрузке: вместо статической или редко перестраиваемой организации сегментов используется splay-дерево моделей, которое перемещает часто используемые сегменты ближе к корню и тем самым уменьшает среднюю стоимость доступа к «горячим» диапазонам ключей. Чтобы снизить накладные расходы на повороты, применяется рандомизированная оптимизация: операция splay выполняется не при каждом обращении, а лишь с заданной вероятностью, что на практике уменьшает стоимость перестроений при сохранении эффекта самоадаптации. Для адаптации к изменениям распределения ключей (вследствие накопленных вставок/удалений и деградации исходного разбиения) EIndex предусматривает периодические перестроения, при которых выполняется упорядоченный обход элементов и пересборка сегментов с

использованием эффективного алгоритма разбиения распределения на линейные сегменты, рассчитанного на регулярное применение. В результате EIndex стремится поддерживать одновременно: актуальное разбиение на сегменты (адаптация к данным) и выгодное расположение «популярных» сегментов в структуре навигации (адаптация к запросам).

Экспериментальная проверка на синтетических и SOSD наборах ключей и нагрузках (включая равномерные и зипфианские распределения запросов, а также сценарии «только чтения» и смешанные операции) показывает, что EIndex способен улучшать пропускную способность относительно базовых вариантов обучаемых индексов на скошенных нагрузках и демонстрирует конкурентоспособность по сравнению с традиционными подходами.

Выводы

Разработан EIndex — адаптивная к запросам и данным структура поиска, сочетающая обучаемое предсказание позиции ключа с самоорганизацией верхнего уровня на основе splay-дерева моделей и вероятностного splay для снижения накладных расходов. Практическое использование результатов возможно в качестве in-memory индекса для упорядоченных ключей в приложениях, где наблюдается перекоп по обращениям и присутствуют обновления. Для внедрения предлагается оформить EIndex как библиотечный модуль на Java и подобрать параметры вероятностного splay и порогов перестроения под профиль конкретной системы.

Литература

1. A Survey of Learned Indexes for the Multi-dimensional Space / A. Al-Mamun [и др.]. — 2024. — DOI: 10.48550/ARXIV.2403.06456.
2. Albers S. Randomized splay trees: Theoretical and experimental results // Information Processing Letters. — 2002. — Февр. — Т. 81, №4. — С. 213–221. — ISSN 0020-0190. — DOI: 10.1016/s0020-0190(01)00230-7.
3. Bayer R., McCreight E. Organization and maintenance of large ordered indices // Proceedings of the 1970 ACM SIGFIDET (now SIGMOD) Workshop on Data Description, Access and Control - SIGFIDET '70. — ACM Press, 1970. — С. 107. — DOI: 10.1145/1734663.1734671.
4. Cormen T. H., Leiserson C. E., Rivest R. L. Introduction to Algorithms. — The MIT Press, McGraw-Hill Book Company, 1989. — ISBN 0-262-03141-8.
5. FINEdex: a fine-grained learned index scheme for scalable and concurrent memory systems / P. Li [и др.] // Proceedings of the VLDB Endowment. — 2021. — Окт. — Т. 15, №2. — С. 321–334. — ISSN 2150-8097. — DOI: 10.14778/3489496.3489512.
6. Learned Lock-free Search Data Structures / G. Bhardwaj [и др.]. — 2023. — DOI: 10.48550/ARXIV.2308.11205.
7. Levy C. C., Tarjan R. E. A Foundation for Proving Splay is Dynamically Optimal. — 2019. — DOI: 10.48550/ARXIV.1907.06310.
8. Perl Y., Itai A., Avni H. Interpolation search—a log log N search // Communications of the ACM. — 1978. — Июль. — Т. 21, №7. — С. 550–553. — ISSN 1557-7317. — DOI: 10.1145/359545.359557.
9. RadixSpline: a single-pass learned index / A. Kipf [и др.] // Proceedings of the Third International Workshop on Exploiting Artificial Intelligence Techniques for Data Management. — ACM, 06.2020. — С. 1–5. — (SIGMOD/PODS '20). — DOI: 10.1145/3401071.3401659.
10. Slastin A., Alistarh D., Aksenov V. Efficient Self-Adjusting Search Trees via Lazy Updates. — 2023. — DOI: 10.48550/ARXIV.2310.05298.
11. Sleator D. D., Tarjan R. E. Self-adjusting binary search trees // Journal of the ACM. — 1985. — Июль. — Т. 32, №3. — С. 652–686. — ISSN 1557-735X. — DOI:

- 10.1145/3828.3835.
12. SOSD: A Benchmark for Learned Indexes / A. Kipf [и др.]. — 2019. — DOI: 10.48550/ARXIV.1911.13014.
 13. The Case for Learned Index Structures / T. Kraska [и др.]. — 2017. — DOI: 10.48550/ARXIV.1712.01208.