

індивідуальних інформаційних потреб студентів та визначення послідовності їх вивчення.

1. J. V. Rogushina, A. Y. Gladun, O. V. Anishchenko, S. M. Pryima, (2024) Semantic Support of Personal Learning Trajectory Development, in: UkrPROG 2024 International Scientific and Practical Programming Conference, CEUR Vol-3806, p.487-505.
2. A.G. Barto, R. S. Sutton, (2018). Reinforcement Learning: An Introduction, The MIT Press: Cambridge, Massachusetts, 2018.
3. B. Jang, M. Kim, G. Harerimana, J. W. Kim, (2019). Q-learning algorithms: A comprehensive classification and applications. IEEE access, 7, 133653-133667.
4. Гришанова І.Ю., Рогушина Ю.В., (2024) Використання методу Q-learning для побудови та оптимізації маршрутів композитного вебсервісу. Проблеми програмування, № 4.

ПЕРСПЕКТИВИ ВПРОВАДЖЕННЯ ІНДЕКСУ НА БАЗІ СУФІКСНОГО ДЕРЕВА ДЛЯ ПОШУКУ ПІДРЯДКІВ У СУБД ВЕЛИКОГО РОЗМІРУ Зважій Д.В

PROSPECTS OF IMPLEMENTING A SUFFIX TREE-BASED INDEX FOR SUBSTRINGS SEARCHING IN LARGE DMBS Zvazhii D. V.

This study explores the advantages and disadvantages of implementing a suffix tree-based index to optimize substring search operations in a DBMS when working with large datasets. Experimental evaluations of the time complexity for substring search operations using suffix trees and database management systems like Elasticsearch, PostgreSQL, MySQL, and ClickHouse have been conducted. Based on the obtained results, the hypothesis regarding the potential efficiency of implementing a suffix tree-based index to optimize substring search operations in a DBMS is confirmed.

Вступ:

Сучасні інформаційні системи стикаються зі стрімким зростанням обсягів даних, що створює нові виклики для їх зберігання та обробки. Ефективний доступ до даних, особливо під час виконання пошукових операцій, стає критично важливим для підтримки продуктивності ІТ-систем. Реляційні та нереляційні системи керування базами даних (СУБД), такі як *PostgreSQL*, *MySQL*, *ClickHouse*, *Elasticsearch*, пропонують різноманітні інструменти для швидкого пошуку та індексації. Проте індекси, що зазвичай використовуються, зокрема на основі В-дерев [1] або інвертованих індексів [2], не завжди оптимально підходять для пошуку підрядків у великих текстових масивах. Це часто призводить до збільшення часу обробки запитів та обмеження масштабованості системи.

Суфіксне дерево — це структура даних [3], яка може значно скоротити час виконання пошуку підрядків, роблячи її потенційно ефективним рішенням для покращення роботи СУБД з великими текстовими обсягами. У нашій роботі ми прагнули експериментально підтвердити гіпотезу, що впровадження суфіксного дерева як індексу для сучасних СУБД дозволить підвищити швидкість виконання операцій пошуку підрядків.

Методологія:

Під час експериментів було використано набори даних обсягом до двох мільйонів рядків. Для побудови суфіксного дерева був застосований алгоритм Укконена [4], відомий своєю лінійною часовою складністю. Експерименти проводилися для індексу на базі суфіксного дерева та на декількох типах СУБД — *Elasticsearch*, *PostgreSQL*, *MySQL* та *ClickHouse* — щоб порівняти ефективність пошукових операцій: “пошук рядка (повне співпадіння)”, “пошук підрядка”, “підрядок не входить у рядок”, “починається з”, “не починається з”, “закінчується на” та “не закінчується на”. Кожен запит перевірявся на серії унікальних підрядків, щоб уникнути значного впливу надто швидких або надто повільних результатів. Для запобігання впливу кешування на результати, кожна серія запитів виконувалась ізольовано. Для цього перед кожною серією запитів створювалася нова таблиця з тестовим набором даних. Тестова реалізація суфіксного дерева не підтримувала кешування, тому це дозволяло мінімізувати кількість операцій побудови дерева.

Висновки:

У статті опублікованій в журналі “Кібернетика та системний аналіз” [5] представлені результати практичних експериментів з використання індексу на основі суфіксного дерева в порівнянні з можливостями сучасних СУБД. Основною метою дослідження було підтвердити або спростувати гіпотезу, що впровадження індексу із суфіксним деревом може підвищити часову ефективність операцій пошуку рядків. Отримані результати підтвердили цю гіпотезу та показали потенціал використання суфіксних дерев у сучасних системах керування базами даних.

Найкращі результати індекс на базі суфіксного дерева продемонстрував для операцій “пошук рядка (повне співпадіння)”, “пошук підрядка”, “починається з”, “закінчується на”. Найближчим до показників суфіксного дерева був *Elasticsearch*, проте в середньому різниця становила 1,5 мс та 3 мс та більше в залежності від типу запиту. Однак результати для операцій “підрядок не входить у рядок”, “не починається з”, “не закінчується на” для індексу на основі суфіксного дерева були гіршими, ніж у *Elasticsearch* та *ClickHouse*. Проте оцінки складності наведені у статті [5] вказують на значний потенціал для оптимізації цих операцій. Експеримент також виявив слабкі сторони індексу на базі суфіксного дерева, зокрема час, необхідний для його побудови, та обсяг пам’яті, потрібний для зберігання отриманої структури даних. Дані тестування показують, що при побудові суфіксного дерева за алгоритмом Укконена [4] час, необхідний для вставки нових даних, постійно зростає. Однак це може бути несуттєвим, якщо задача допускає побудову індексу у фоновому режимі.

Література

1. Comer D. Ubiquitous B-Tree. *ACM Comput. Surv.* 1979. Vol. 11. P. 121–137. <https://doi.org/10.1145/356770.356776>.
2. Baeza-Yates R., Ribeiro-neto B. and Mills D. Modern Information Retrieval. *Addison-Wesley Longman Publishing*. 1999.
3. Weiner P. Linear pattern matching algorithms. Proc. 14th Annual IEEE Symposium on Switching and Automata Theory (SWAT 1973) (15–17 October 1973, Iowa City, Iowa, USA). Iowa City, 1973. P. 1–11. <https://doi.org/10.1109/SWAT.1973.13>.
4. Ukkonen E. On-line construction of suffix trees. *Algorithmica*. 1995. Vol. 14., N 3. P. 249–260. <https://doi.org/10.1007/BF01206331>.
5. Кібернетика та системний аналіз. URL: <http://www.kibernetika.o>

СЕКЦІЯ 1. КОМП’ЮТЕРНІ НАУКИ