

МЕТОД АЛГОРИТМІЧНОЇ ОПТИМІЗАЦІЇ ОБЧИСЛЕНЬ В БРОКЕРСЬКИХ БІЛІНГОВИХ СИСТЕМАХ

Вінницький національний технічний університет

Анотація

Авторами розглянуто проблему низької продуктивності рекурсивної обробки деревовидних структур ставок у PostgreSQL порівняно з Microsoft SQL Server. Запропоновано метод алгоритмічної оптимізації, що базується на одноразовій трансформації ієрархічних правил у денормалізовану плоску форму з використанням композитних типів та масивів. Це дозволяє замінити складну рекурсію вискоелективною операцією з'єднання таблиць, що суттєво пришвидшує розрахунок комісій у брокерських білінгових системах.

Ключові слова: PostgreSQL, Microsoft SQL Server, комп'ютерні науки, високонавантажені інформаційні системи, хмарні обчислення, оптимізація запитів, рекурсія, масиви, теорія множин, білінгові системи, деревовидні структури, денормалізація, композитні типи.

Abstract

The authors examine the issue of low performance in the recursive processing of tree-like rate structures within PostgreSQL compared to Microsoft SQL Server. An algorithmic optimization method is proposed based on a one-time transformation of hierarchical rules into a denormalized flat form using composite types and arrays. This approach allows replacing complex recursion with highly efficient table join operations, significantly accelerating

Keywords: PostgreSQL, Microsoft SQL Server, computer science, high-load information systems, cloud computing, query optimization, recursion, arrays, set theory, billing systems, tree structures, denormalization, composite types.

Сучасні брокерські білінгові системи потребують високої швидкості обробки великих масивів торговельних операцій для точного розрахунку комісійних нарахувань [1]. Процес обчислення базується на зіставленні параметрів кожної угоди з набором правил, які визначають тарифи для конкретних ринків, інструментів та учасників. Традиційно такі правила мають ієрархічну структуру, що відображає логіку успадкування умов від загальних до специфічних.

Ефективність функціонування сучасних фінансових платформ безпосередньо залежить від здатності систем керування базами даних оперативно опрацьовувати великі масиви транзакційних даних. Використання деревовидних структур для моделювання тарифних сіток є стандартом у галузі, оскільки дозволяє гнучко налаштувати умови для різних ринків та клієнтів. Однак при масштабуванні систем виникає критичне питання вибору СУБД, здатної підтримувати високу швидкість обчислень.

Актуальність дослідження зумовлена технологічним розривом між комерційними рішеннями, такими як Microsoft SQL Server, та вільним програмним забезпеченням. PostgreSQL демонструє значну деградацію продуктивності при виконанні рекурсивних запитів через обмеження планувальника та механізмів паралелізації. У той час як SQL Server опрацьовує масив даних за лічені хвилини, PostgreSQL може витратити години на аналогічні ітераційні обчислення, що робить перехід на відкриті платформи без спеціальної оптимізації неможливим [2;3].

Сутність запропонованого методу ґрунтується на фундаментальних принципах теорії алгоритмів та структур даних. Рекурсія, хоча і є елегантним способом опису ієрархічних зв'язків, у контексті реляційних

баз даних часто призводить до надмірної обчислювальної складності через необхідність послідовного виконання вкладених запитів. Для подолання цього обмеження застосовується перехід до лінійних структур даних, зокрема масивів та матричних представлень. Масиви в PostgreSQL дозволяють зберігати впорядковані набори умов у межах одного запису, що фактично перетворює вертикальну ієрархію на горизонтальну структуру. Використання матричного підходу (діагоналізації) дозволяє відобразити складні взаємозв'язки між різними параметрами (ринок, інструмент, клієнт) у вигляді проекції, де кожен елемент масиву відповідає конкретному рівню вкладеності правила.

З точки зору теорії множин, процес розрахунку ставки можна розглядати як пошук перетину множини характеристик транзакції з множиною умов, визначених тарифним планом. Традиційний рекурсивний підхід виконує цей пошук крок за кроком, що збільшує кількість операцій порівняння. Запропонована процедура денормалізації трансформує ієрархію у сукупність кортежів, де кожна гілка дерева стає елементом універсальної множини правил. Застосування композитних типів для зберігання значень та операторів порівняння дозволяє звести задачу до одноразової перевірки приналежності характеристик угоди до відповідних підмножин правил. Це дозволяє системі виконувати операцію JOIN як високоефективне фільтрування множин, що нівелює архітектурні недоліки рекурсії та забезпечує $O(1)$ або $O(\log n)$ доступ до потрібних параметрів замість лінійної залежності від глибини дерева.

Сутність оптимізації полягає у докорінній зміні стратегії обробки даних – переході від динамічного рекурсивного обходу дерева до використання статичних моделей.

Процес трансформації ієрархії передбачає «розгортання» вузлів у пласку таблицю, де кожен рядок представляє повний шлях від кореня до цільового тарифу. Ключовим технічним інструментом є використання композитних типів, що дозволяють об'єднувати оператор порівняння та значення в єдину логічну одиницю. Замість багаторазового проходження ієрархії для кожної угоди, система одноразово формує масиви умов для кожної гілки дерева. Це дозволяє повністю усунути рекурсію з основного циклу обчислень.

Алгоритм замінює ресурсомісткі ітерації на високоефективну операцію JOIN, де атрибути торговельної операції зіставляються з попередньо обчисленими масивами обмежень. Завдяки цьому PostgreSQL отримує можливість будувати оптимальні плани виконання, використовувати індексний пошук та ефективно розподіляти навантаження між процесорними ядрами. Використання модифікатора IMMUTABLE для функцій порівняння масивів додатково прискорює процес, дозволяючи планувальнику кешувати результати. Таким чином, складна логічна перевірка трансформується у швидку операцію перетину множин.

Алгоритмічна заміна рекурсії на операції з масивами та з'єднаннями забезпечує кратне зростання продуктивності, дозволяючи використовувати безкоштовні СУБД у високонавантажених білінгових системах із збереженням швидкодії на рівні комерційних аналогів.

Трансформація деревовидної моделі у форму масивів умов забезпечує логічну цілісність розрахунків та коректне зіставлення операцій, що мають складні обмеження. Це створює передумови для реалізації систем розрахунку ставок у режимі майже реального часу на базі безкоштовних технологій.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Іванчук Я.В., Яковчук П.Л. Балансування навантаження комп'ютерної мережі // Електронні інформаційні ресурси: створення, використання, доступ та управління: збірник матеріалів Міжнародної науково-практичної Інтернет-конференції (Суми/Вінниця, 20-21 листопада 2024 р.). – Суми/Вінниця: НІКО / КЗВО «Вінницька академія безперервної освіти», 2024. – С. 64-65.
2. Microsoft SQL Server Documentation. Recursive Queries Using Common Table Expressions. 2024. URL: <https://learn.microsoft.com/en-us/sql/t-sql/queries/with-common-table-expression-transact-sql>
3. PostgreSQL Documentation. 16.2024. URL: <https://www.postgresql.org/docs/16/index.html>

Іванчук Ярослав Володимирович, д.т.н., професор, професор кафедри комп'ютерних наук, Вінницький національний технічний університет, м. Вінниця, ivanchuck@vntu.edu.ua

Яковчук Павло Леонідович, група 123а, Факультет інтелектуальних інформаційних технологій та автоматизації

Yaroslav Ivanchuk, Doctor of Technical Sciences, Professor, Professor of the Department of Computer Science, Vinnytsia National Technical University, Vinnytsia, ivanchuck@vntu.edu.ua

Pavlo Yakovchuk, group 123a, Vinnytsia National Technical University, Vinnytsia. Faculty of Intelligent Information Technologies and Automation pyakovchuk@gmail.com