

Метод керування потоками даних в клієнт-серверних програмних засобах за умов невизначеності, на основі локальної моделі керованого процесу

М.І. Гученко¹, М.М. Іванова¹, П.П. Костенко¹, О.Г. Славко¹

Abstract – Factors affected on to the speed of data accessing in CIS is considered. Adaptive system of the data stream control in CIS is considered. The algorithm for data streams control in the workplace automated control system under conditions of low a priori information, based on controlled process local model, is presented.

Key words – Workplace Automated Control System (WACS), SQL-query optimization, CIS based on RDBMS.

I. ВСТУП

Одним з головних засобів зберігання та обробки інформації виступають системи управління базами даних, що стали ваговою частиною сучасних комп'ютерних інформаційних систем, зокрема автоматизованих систем управління. Особливе місце в таких системах займають автоматизовані системи управління робочим місцем (АСУ РМ). Активно застосовуються дані системи в медичних, юридичних закладах, бухгалтерських, економічних, статистичних відділах. Наразі розроблено спектр вітчизняного та іноземного програмного забезпечення (система автоматизації медичного закладу амбулаторного типу «Нейрон» ТОВ АЛЮР, автоматизована інформаційна система «Поликлініка» ТОВ ПАРУС, система автоматизації медичного закладу «ArchiMed» MLS IT Systems-Ukraine та ін.), що використовує в своїй організації АСУ РМ. Прискорення та максимальна автоматизація роботи оператора (лікаря, економіста тощо), ось головний чинник використання АСУ РМ. Тому постійно актуальною проблемою є максимізація швидкодії КІС на базі СУБД.

II. АНАЛІЗ ПОПЕРЕДНІХ ДОСЛІДЖЕНЬ

На швидкість отримання інформації в КІС впливають декілька чинників: програмна реалізація клієнтських частин КІС, апаратні характеристики ПК, структура БД, структура SQL-запитів до БД тощо. Прискорення швидкодії КІС вбачається в оптимізації зазначених чинників.

До клієнтських частин КІС частіше застосовується методи рефакторингу [1] ніж оптимізації, це обумовлено значною насиченістю програмного коду. Будь оптимізація програмних кодів призводить до погіршення їх логічного та синтаксичного розуміння, тому оптимізація на цьому рівні проводиться за крайньої необхідності.

Оптимізація структури БД передбачає реорганізацію інформації в базі даних на логічному рівні [2]. Актуальним є використання реляційних баз даних, основаних на реляційній моделі даних, описаній доктором Е. Коддом [3]. Робота БД базується на математичній і логічній моделі, описаній в теорії реляційної моделі даних [4]. Головними перевагами цієї теорії вважається: сувора логічна формалізація, незалежність даних та можливість оптимізації структури бази даних за допомогою нормалізації. Така оптимізація призводить до конструювання «дуже» складних запитів, отримання даних значно уповільнюється, через, головним чином, велику кількість з'єднань таблиць.

Говорячи про оптимізацію запитів в реляційних СУБД [5], зазвичай мають на увазі такий спосіб обробки запитів, коли за початковим поданням запиту шляхом його перетворень виробляється процедурний план його виконання, найбільш оптимальний за наявних у базі даних керуючих структурах. Відповідні перетворення початкового подання запиту виконуються спеціальним компонентом СУБД – оптимізатором. Оптиміальність виробленого ним плану запиту носить умовний характер: план оптимальний відповідно до критеріїв, закладеними в оптимізатор, що породжує можливі відмінності від реальної оптимальності.

Метою роботи є розробка методу оптимізації SQL-запитів, за умов невизначеності структури та характеристик КІС на основі локальної моделі керованого процесу.

III. МАТЕРІАЛИ І РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ

Внутрішній оптимізатор SQL-запитів БД володіє всіма необхідними даними для коректної оптимізації плану виконання запиту (метадані БД, індекси, статистика тощо.). Єдиним слабким місцем даної системі є правила оптимізації. Оптимізатор обмежений в своїй роботі одним запитом користувача, що призводить до генерації обмеженої кількості варіантів оптимізації. Вибір не оптимального плану виконання запиту обумовлюється некоректною формою запиту користувача. Модифікація структури запиту, на боці клієнта, реалізується технологією рефакторингу, що описана в роботі [6]. Рефакторинг SQL додатків – запитів кропіткий процес, що потребує періодичного застосування, внаслідок динамічного ускладнення структури та об'єму БД. Тому

¹ Кременчуцький національний університет імені Михайла Остроградського, вул. Першотравнева, 20, Кременчук, 39600, УКРАЇНА, E-mail: ppkostenko@gmail.com

постає проблема оптимізації структури SQL – запитів в режимі реального часу до їх виконання на сервері СУБД.

Можливим вирішенням є використання адаптивної системи керування. Проте мізерний об'єм апріорної інформації, суттєва поточна інформаційна невизначеність та необхідність керування в режимі реального часу призводить до труднощів застосування статистичних, інтелектуальних та робастних системи керування. В [7] показано, що шляхом проведення експерименту в режимі реального часу можливо отримати модель керованого процесу. Подібна локальна модель відображає в даний момент часу динаміку об'єкта керування та збурень, що діють на нього.

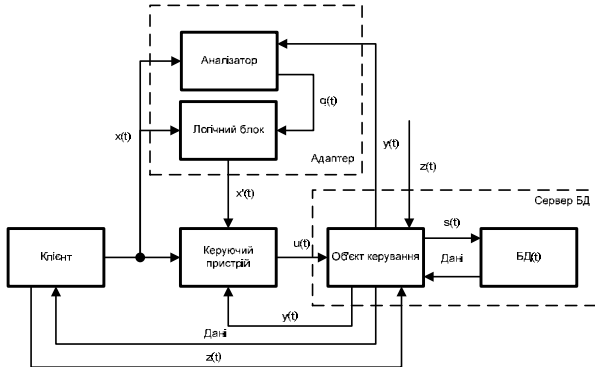


Рис.1. Структурно-функціональна схема системи керування

Система керування показана на рис. 1. Вхідний сигнал $x(t)$ – дискретна величина SQL_{ik}^t , де $t \in \{Select, Insert, Update, Delete\}$, $t \in \{Select, Insert, Update, Delete\}$, N – множина варіантів SQL – запитів певного типу, $t \in \{Select, Insert, Update, Delete\}$ – показник адаптованості i -го типу запиту; $x'(t)$ – вхідний сигнал оптимізації; $q(t)$ – дискретний сигнал включення оптимізації; $u(t)$ – сигнал керування, оптимізований SQL-запит; $z(t)$ – збурення, $z(t) \in \{Insert, Update, Delete\}$; $y(t)$ – час виконання запиту $y(t) \in \mathbb{R}$, $s(t)$ – PreSQL-запит.

Метод оптимізації представлено на рис. 2. Принцип оптимізації оснований на побудові локальної моделі керованого процесу. Для формування розуміння структури та завантаження БД, після відпрацювання запиту клієнта проводиться свідоме погіршення умов вибору даних. Час відпрацювання погіршеного запиту та запиту клієнта виступає як критерій оптимізації запита, та дозволяє виключати некоректні варіанти оптимізації з подальших циклів.

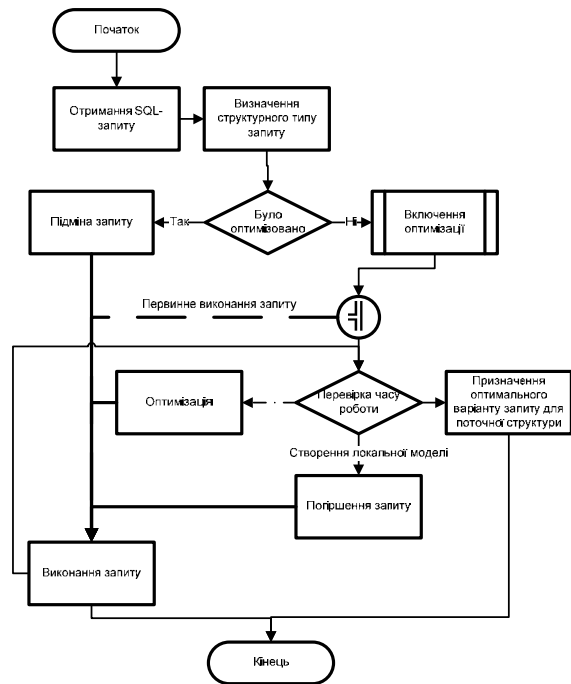


Рис.2. Метод оптимізації SQL-запитів на основі локальної моделі керованого процесу.

IV. ВИСНОВКИ

Запропоновано метод оптимізації SQL-запитів на основі локальної моделі керованого процесу. Формування локальної моделі керованого процесу дозволяє оцінити можливі варіанти оптимізації запитів та розробити автономну систему оптимізації (незалежну від виду СУБД, структури БД та зовнішніх збурень), що залежить лише від лексичних особливостей мови SQL.

СПИСОК ПОСИЛАНЬ

- [1] Скотт В. Емблер, Прамодкумар Дж. Садаладж Рефакторинг баз даних: еволюционное проектирование = Refactoring Databases: Evolutionary Database Design (Addison-Wesley Signature Series). — М.: «Вильямс», 2007. — С. 368.
- [2] Коломейчук В.В., Розробка та дослідження бази даних для систем обробки статистичної інформації, Математичні машини і системи, 2009, № 4 с. 89-95.
- [3] Codd, E.F. (1970). "A Relational Model of Data for Large Shared Data Banks". Communications of the ACM 13 (6): 377–387.
- [4] Мейер М. Теория реляционных баз данных. – Москва: Мир, 1998. – 608 с.
- [5] A. Deshpande, Z. Ives, and V. Raman. Adaptive query processing. Foundations and Trends in Databases, 1(1):1–140, 2007.
- [6] Фаро С., Паскаль Л. Рефакторинг SQL приложений. – Пер. с англ. – СПб: Символ-Плюс, 2009. – 336 с., ил.
- [7] Гученко, М.І. Активно-резонансний принцип керування. [Текст] / М.І. Гученко // 16-а Міжнародна конференція з автоматичного управління «Автоматика-2009». Тези доповідей. Чернівці, 22-25 вересня, 2009. – С. 59-61.