Акименко Андрій Миколайович – кандидат фізико-математичних наук, доцент, професор кафедри інформаційних систем в економіці, Чернігівський національний технологічний університет (вул. Шевченка, 95, м. Чернігів, 14027, Україна).

Акименко Андрей Николаевич – кандидат физико-математических наук, доцент, профессор кафедры информационных систем в экономике, Черниговский национальный технологический университет (ул. Шевченко, 95, г. Чернигов, 14027, Украина).

Akymenko Andrii – PhD in Mathematical, Associate Professor, Professor of Information Systems in Economics Department, Chernihiv National University of Technology (95 Shevchenka Str., 14027 Chernihiv, Ukraine).

E-mail: anakim2@gmail.com

Бивойно Тарас Павлович – старший викладач кафедри інформаційних систем в економіці, Чернігівський національний технологічний університет (вул. Шевченка, 95, м. Чернігів, 14027, Україна).

Бивойно Тарас Павлович – старший преподаватель кафедры информационных систем в экономике, Черниговский национальный технологический университет (ул. Шевченко, 95, г. Чернигов, 14027, Украина).

Bivoyno Taras – Senior Lecturer of Information Systems in Economics Department, Chernihiv National University of Technology (95 Shevchenka Str., 14027 Chernihiv, Ukraine).

E-mail dec-fpo@ukr.net

УДК 004.658:004.4'41:004.82

Ірина Бальченко

ПРОБЛЕМИ РОЗРОБЛЕННЯ НЕОДНОРІДНИХ РОЗПОДІЛЕНИХ СИСТЕМ УПРАВЛІННЯ БАЗАМИ ДАНИХ

Ирина Бальченко

ПРОБЛЕМЫ РАЗРАБОТКИ НЕОДНОРОДНЫХ РАСПРЕДЕЛЕННЫХ СИСТЕМ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

Iryna Balchenko

ISSUES OF THE DEVELOPMENT OF HETEROGENEOUS DISTRIBUTED DATABASE MANAGEMENT SYSTEMS

Проведено аналіз сучасних методів та технологій, що застосовуються під час вирішення задач створення та використання неоднорідних розподілених систем управління базами даних. Виокремлено задачі фрагментації даних і розподіленої компіляції запитів та можливі шляхи їх розв'язання.

Ключові слова: *неоднорідні розподілені бази даних, фрагментація, розподілені запити, розподілена компіляція.*

Рис.: 1. Бібл.: 7.

Проведен анализ современных методов и технологий, которые используются при решении задач создания и использования неоднородных распределенных систем управления базами данных. Выделены задачи фрагментации данных и распределенной компиляции запросов и возможные пути их решения.

Ключевые слова: *неоднородные распределенные базы данных, фрагментация, распределенные запросы, распределенная компиляция.*

Рис.: 1. Библ.: 7.

The analysis of modern methods and technologies, which are used in solving problems of creation and use of heterogeneous distributed database management systems are shown. Tasks of data fragmentation and distributed compiling of queries and possible way of its solutions are selected.

Key words: *heterogeneous distributed databases, a fragmentation, distributed queries, a distributed compilation.*

Fig.: 1. Bibl.: 7.

Постановка проблеми. Розподілена база даних (РБД) – набір логічно пов'язаних між собою розділених даних, які фізично розподілені по різних вузлах мережі. РБД можуть працювати під управлінням однакових і неоднакових СУБД. У першому випадку говорять про однорідні розподілені системи, у другому – про неоднорідні.

Неоднорідні системи зазвичай виникають у тих випадках, коли сервери, що вже експлуатують свої власні системи з базами даних, з часом інтегруються в розподілену систему [1]. Неоднорідні системи включають два або більше продукти управління даними, що суттєво відрізняються (наприклад, реляційні СУБД від різних постачальників, таких, як PostgreSQL, MySQL, MS SQL Server, Oracle).

Основним завданням інтеграції неоднорідних РБД є надання користувачам інтегрованої системи глобальної схеми БД, представленої в деякій моделі даних, і автоматичне

перетворення операторів маніпулювання БД глобального рівня в оператори, зрозумілі відповідним локальним СУБД.

Аналіз досліджень і публікацій. Про створення РБД говорять та пишуть давно [1–3]. Але до цього часу проблема не вирішена остаточно, зокрема, актуальною є задача зменшення часу відгуку на запит користувача до РБД, а також задача створення неоднорідної системи управління РБД (СУРБД), що здатна оперувати запитами до локальних СУБД різних типів.

Щоб забезпечити оптимальний баланс загрузки серверів БД та зменшити час відгуку на запит користувача, фрагменти РБД розподіляються по вузлах мережі, при цьому використовуються прийняті критерії розміщення. Коли критерії розміщення даних призводять до перевантаження серверів або виникають інші причини (наприклад, зміна структури мережі), необхідно провести реорганізацію бази. Така реорганізація проводиться досить рідко, що тягне за собою неефективне функціонування РБД до проведення наступних планових оптимізаційних заходів. Тому важливо, щоб система управління РБД могла керувати розміщенням фрагментів даних по серверах БД одночасно з виконанням основних функцій.

Ідеологія побудови розподілених баз даних за принципом «зверху вниз», для яких спочатку визначається глобальна схема, а потім проводиться розподіл об'єктів бази даних може бути застосована при побудові неоднорідних СУРБД тільки в разі використання механізму трансляції мов локальних СУБД різних типів.

Виділення не вирішених раніше частин загальної проблеми. Більшість авторів під проектуванням РБД розуміють фрагментацію та розміщення, тобто розділення БД на фрагменти та прийняття рішення про те, де будуть зберігатися ці фрагменти. Проектування схем фрагментації та розміщення відношень ґрунтується на інформації про способи та методи використання РБД. Методи використання залежать від стратегії виконання запитів, яка, у свою чергу, повинна враховувати схеми фрагментації та розміщення.

Інша мета, що виникає перед розробником – досягти прозорого доступу, що являє собою дещо більше, ніж звичайне забезпечення доступу до віддалених СУБД та їх базам даних. Для організації взаємодії між різними типами СУБД необхідно забезпечити перетворення переданих повідомлень, для чого кожен з вузлів повинен мати можливість формулювати запити на мові тієї СУБД, яка використовується на локальному вузлі, або система повинна взяти на себе виконання всіх необхідних перетворень.

Мета статті. Проаналізувати наявні вимоги до неоднорідних розподілених баз даних, методи фрагментації та технології розподіленої компіляції запитів, запропонувати вдосконалення відомих методів та технологій.

Виклад основного матеріалу. Серед багатьох властивостей, яким, за К. Дейтом [2], повинна задовольняти розподілена база даних, виділимо такі:

– Локальна автономність. Локальні дані належать локальним вузлів і управляються адміністраторами локальних БД.

– Безперервне функціонування. Видалення або додавання вузла не повинно вимагати зупинки системи в цілому.

– Незалежність від фрагментації. Доступ до даних не повинен залежати від наявності або відсутності фрагментації і від типу фрагментації.

– Оброблення розподілених запитів. Система повинна автоматично визначати методи виконання з’єднання (об'єднання) даних.

– Незалежність від типу СУБД. СУРБД повинна бути здатною функціонувати поверх різних локальних СУБД, можливо, з різними моделями даних (вимога гетерогенності).

Опишемо метод і технологію фрагментації даних і виконання розподілених запитів відповідно до представлених вимог.

Фрагментація даних.

Основною метою фрагментації є звуження простору пошуку при виконанні запиту. Прийнято виділяти дві базові стратегії фрагментації даних: горизонтальна фрагментація (ГФ) та вертикальна фрагментація (ВФ). ВФ – це поділ атрибутів на групи, ГФ – поділ відношення на підмножини таким чином, що кожна підмножина містить повний набір атрибутів. Фрагментація описує розбиття відношень в БД. Розбиття відношень на мінімальні, неділені надалі фрагменти дозволить розмістити їх з максимально можливою гнучкістю. Для отримання такого набору фрагментів для кожного відношення R на підставі транзакцій визначається набір мінтерм предикатів і групи атрибутів. Причому предикати описують застосування ГФ, а групи атрибутів – ВФ. Групи атрибутів виходять шляхом розділення безлічі атрибутів, що використовуються у транзакціях.

Схема розміщення вказує місце розташування фрагментів. Вона описується матрицею D, рядками якої є сервера БД, а стовпцями – сформований набір фрагментів. Одиниця в комірці матриці D_{ij} означає наявність фрагмента й у вузлі j. Для задоволення властивості повноти схеми розміщення кожен фрагмент повинен перебувати хоча б в одному вузлі (у кожному стовпчику повинна бути хоча б одна одиниця).

У [4] доведено, що набір мінтерм предикатів має властивості повноти і достатності. За визначенням, для груп атрибутів ці властивості очевидні. Таким чином, описана фрагментація має властивості повноти і відновлюваності.

Запропонований підхід передбачає використання алгоритмів динамічного управління розміщенням фрагментів даних на серверах БД залежно від частоти звернення до фрагментів даних і навантаження на самі сервери. Проведення таких перерозподілів дозволить, по-перше, оперативно реагувати на зміни умов, по-друге, забезпечити більш високу ефективність реакції на запити користувачів протягом їх роботи з РБД в порівнянні з традиційною технологією. Для цього необхідно вирішити ряд підзадач:

1. Збір даних про стан та ефективність роботи системи.
2. Визначення фрагментів даних для переміщення.
3. Реалізація операції реорганізації структури РБД.

Загалом на першому етапі повинні бути отримані такі дані: кількість запитів до серверу та до кожного фрагмента даних, чисельність віддалених запитів до кожного фрагмента даних від кожного сервера БД, час виконання запитів та розмір запитів від кожного сервера БД, обсяг завантаження кожного сервера.

Переміщення даних являє собою процес прийняття рішення про місце зберігання даних з метою мінімізації цільової функції при виконанні запитів. Задачу динамічного управління структурою РБД слід формулювати так: для цієї логічної схеми БД, безлічі запитів і конфігурації серверів описати схему фрагментації, схему розміщення фрагментів і стратегії виконання кожного запиту таким чином, щоб оптимізувати цільову функцію:

$$\zeta = \sum_{i=1}^{n}\sum_{j=0}^{m}(\tilde{c}_t(t)\cdot\tau_{ij}^{D}/\tau_{ij}^{1}+\tilde{c}_g(t)\cdot\tilde{\omega}_{ij}(t))\cdot\tilde{\phi}_{ij}(t)\to\min, \quad (1)$$

де n – кількість серверів; m – кількість запитів; φ_{ij} – частота виникнення j-го запиту в і-му вузлі; τ_{ij}^{D} – час відгуку на запит, виконаний у РБД; τ_{ij}^{l} – розрахунковий час відгуку на запит, виконаний локально на вузлі і за умови наявності в ньому всіх необхідних фрагментів; ω_{ij} – коефіцієнт використання ресурсів під час оброблення j-го запиту, що породжений в і-му вузлі; c_t и c_g – коефіцієнти важливості часу відгуку та готовності транзакції, що лежать в межах від 0 до 1.

У [5] значення коефіцієнтів визначаються проектувальником на підставі вимог до РБД. У цьому підході пропонується використовувати експертні оцінки у вигляді нечітких лінгвістичних змінних на часовій осі. Пошук рішення задачі оптимізації чіткої функції

від нечітких змінних можна здійснити використовуючи методи нечіткого лінійного та нелінійного програмування та інтерактивні методи побудови компромісних рішень [6].

Розподілена компіляція запитів.

Розглянемо загальну схему розподіленої компіляції.

Будемо називати головним вузлом той вузол мережі, в якому ініційований процес компіляції запиту SQL, і додатковими вузлами – ті вузли, які залучаються до цього процесу у ході його виконання. На самому грубому рівні процес компіляції можна розбити на такі фази.

У головному вузлі проводиться граматичний розбір запиту SQL з побудовою внутрішнього подання запиту у вигляді дерева. На основі інформації з локального каталогу головного вузла і віддалених каталогів додаткових вузлів проводиться заміна імен об'єктів, що фігурують у запиті, на їх системні ідентифікатори.

Як спосіб трансляції запитів різних СУБД, можна запропонувати використовувати граматики мов найбільш популярних СУБД у формі близькій до РБНФ, а також ANTLR [7] – генератор парсерів, що дозволяє за описом граматики створювати парсер на одному з цільових мов програмування. Нижче представлений фрагмент граматики мови T-SQL:

```
grammar tsql;

tsql_file
    : sql_clauses? EOF
    ;

sql_clauses
    : sql_clause+
    ;

sql_clause
    : dml_clause
    | ddl_clause
    | cfl_statement
    | another_statement
    ;
```

```
dml_clause
    : delete_statement
    | insert_statement
    | select_statement
    | update_statement
    ;
insert_statement
    : with_expression?
      INSERT (TOP '(' expression ')'
PERCENT?)?
      INTO?       (ddl_object       |
rowset_function_limited)
      insert_with_table_hints?
      ('(' column_name_list ')')?
      output_clause?
      insert_statement_value
      for_clause? option_clause? ';'?
```

Отриманий низхідний аналізатор необхідно доповнити семантичними діями з використанням механізму Listeners або Visitors (рис.).

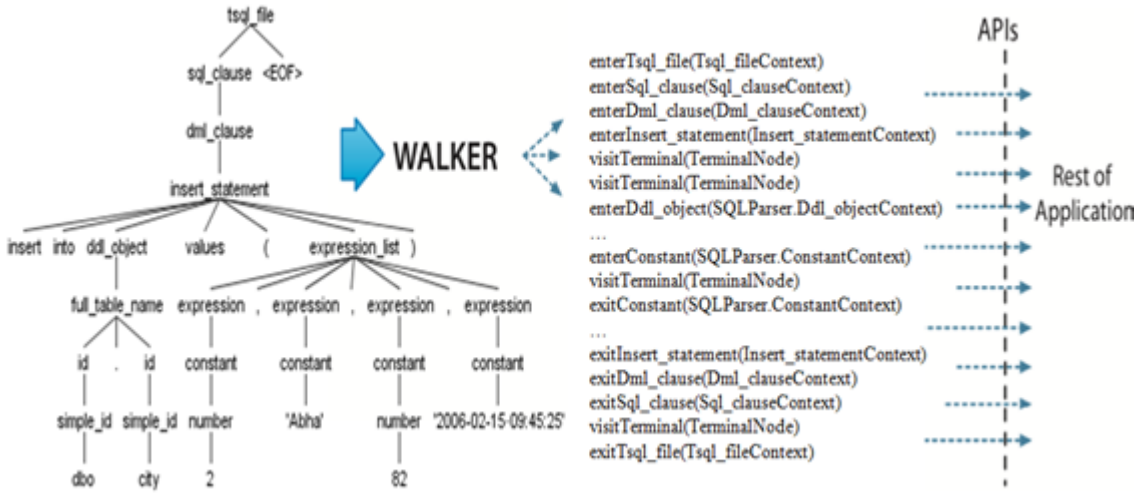

Рис. 1. Механізм Listeners з можливістю доповненням синтаксичними діями при обході дерева

У головному вузлі генерується глобальний план виконання запиту, в якому враховується лише порядок взаємодій вузлів при реальному виконанні запиту. Глобальний план відображається в перетвореному відповідним чином дереві запиту.

У разі якщо у глобальному плані виконання запиту беруть участь додаткові вузли, проводиться його декомпозиція на частини, кожну з яких можна виконати в одному вузлі (наприклад, локальна фільтрація відносини відповідно до заданого в умові вибірки предикату обмеження). Відповідні частини запиту (у внутрішньому поданні) розсилаються в додаткові вузли.

У кожному вузлі, що бере участь у глобальному плані виконання запиту (головному і додаткових) виконується завершальна стадія виконання компіляції. Ця стадія включає, по суті, дві останні фази процесу компіляції запиту: оптимізацію і генерацію машинних кодів. Проводиться перевірка прав користувача, від імені якого проводиться компіляція, на виконання відповідних дій; відбувається опрацювання представлень бази даних; здійснюється локальна оптимізація оброблюваної частини запиту відповідно до наявних індексів; нарешті, проводиться генерація коду.

Висновки. Наявні методи та технології побудови неоднорідних розподілених систем мають за мету створення інтегрованого середовища шляхом поєднання існуючих баз даних і не забезпечують динамічного управління розміщенням фрагментів даних на серверах БД.

Запропоновано використовувати оптимізацію розміщення фрагментів даних, зважаючи на нечіткість даних відносно часової осі. Крім того, для забезпечення прозорого доступу до СУБД різних типів запропоновано використовувати технологію трансляції мов локальних СУБД за допомогою побудови парсеру синтаксично керуючого перекладу.

Таким чином, описані проблеми, методи та технології побудови неоднорідних федеративних систем управління базами даних розширюють наявні можливості та дозволяють наблизитися до побудови «ідеальних» розподілених СУБД.

Список використаних джерел

1. *Sheth A. P.* Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases / Amit P. Sheth, James A. Larson // ACM Computing Surveys. – 1990. – Vol. 22, no. 3,– Pp. 183–236.

2. *Бобрешов-Шишов Д. И.* Динамическое управление структурой распределенной базы данных / Д. И. Бобрешов-Шишов, Л. А. Саяркин, И. А. Шаров // Молодой ученый. – 2015. – № 7. – С. 51–53.

3. *Дейт К. Дж.* Введение в системы баз данных / К. Дж. Дейт. – 8-ое издание. – М. : Вильямс, 2005. – 1328 с.

4. *Ceri S.,* Negri M., Pelagatti G. Horizontal data partitioning in database design // ACM SIGMOD international conference on Management of data. Orlando, Florida. – 1982. – Pp. 128–136.

5. *Новосельский В. Б.* Метод автоматизации проектирования распределенной реляционной базы данных / В. Б. Новосельский // Программные продукты и системы. – 2008. – № 3. – С. 45–48.

6. *Зайченко Ю. П.* Исследование операций: нечеткая оптимизация : учеб. пособие / Ю. П. Зайченко. – К. : Высшая школа, 1991. – 191 с.

7. *Parr T.* The Definitive ANTLR4 Reference / Terence Parr. – Dallas, Texas • Raleigh, North Carolina: The Pragmatic Programmers. – 2013. – 328 p.

Бальченко Ірина Володимирівна – кандидат технічних наук, асистент кафедри програмної інженерії, Чернігівський національний технологічний університет (вул. Шевченка, 95, м. Чернігів, 14027, Україна).
Бальченко Ирина Владимировна – кандидат технических наук, ассистент кафедры информационных технологий и программной инженерии, Черниговский национальный технологический университет (ул. Шевченко, 95, г. Чернигов, 14027, Украина).
Balchenko Iryna – PhD in Technical Sciences, Assistant of Information Technologies and Program Engineering Department, Chernihiv National University of Technology (95 Schevchenka Str., 14027 Chernihiv, Ukraine).
E-mail: phiona06@yandex.ua