

УДК 681.32

І.А. Лебедева, Я.Ю. Стасєва, Р.Ю. Жермельова

Харківський університет Повітряних Сил ім. І. Кожедуба

ФОРМАЛІЗАЦІЯ ПРОЦЕСУ ПОДАННЯ ЗАПИТІВ У СИСТЕМІ УПРАВЛІННЯ БАЗАМИ ДАНИХ

Розглянута формалізація процесу подання запитів на оброблення інформації в системі управління базами даних (СУБД) у локальних обчислювальних мережах (ЛОМ). Наданий математичний опис властивостей транзакцій.

локальна обчислювальна мережа, запит, система управління базами даних, транзакція

Вступ

Інтенсивне використання в останнє десятиліття у ЛОМ власних кластерів глобальних реляційних баз даних (РБД) зумовило необхідність широкого розвитку засобів забезпечення сумісного використання розподілених даних [1, 2]. Найважливіше функціональне навантаження лягає у зв'язку з цим на засоби управління мережними розподіленими процесами і ресурсами.

Проблематика реляційних БД ґрунтується на дослідженні і реалізації моделі даних та процедур маніпулювання даними із збереження цілісності РБД. З позицій безпосередньої роботи з РБД (користувач, розробник) і з позицій надання відповідного програмно-технічного середовища (розробник, постачальник) проблематика РБД у ЛОМ може бути представлена таким чином:

- інтерфейс користувача;
- мова запитів до БД;
- оптимізація запитів у формі програмного середовища (оптимізатор запитів);
- механізми виконання запитів;
- фізичні реалізації у вигляді організації файлових систем, кластеризації та індексування;
- організація введення – виведення і масової дискової пам'яті;
- комп'ютери і комунікації.

Найбільший інтерес викликають такі проблеми, як оптимізація запитів у формі програмного середовища, механізми виконання запитів, організація масової дискової пам'яті.

Оброблення запитів заповнює прогалину між БД і файловими системами комп'ютерів та може бути реалізовано за допомогою оптимізатора і механізму виконання. Перший транслює запит, виражений мовою запитів високого рівня (наприклад, на SQL), у послідовність операцій (наприклад, в операції реляційної алгебри), які використовуються в механізмі виконання запитів і у файловій системі. Мета оптимізації запитів полягає у знаходженні оптимального плану оцінки запитів, який мінімізує відповідні витрати на

його виконання, а саме: часові (час очікування користувача РБД, час оброблення системою в цілому, час введення/виведення даних, мережевий час), а також витрати на зовнішню пам'ять і всі ресурси.

Крім того, формалізація запитів включає і суто формальні методи доступу до даних, що фізично зберігаються, такі, як: пошук, прямий пошук.

Механізм виконання запиту подається набором операторів виконання запитів, алгоритмами синхронізації цих операторів і включає такі рівні:

- операційна система;
- мережна взаємодія;
- паралельні і/або розподілені обчислення.

У зв'язку з цим стає очевидним, що власне дані вже не можуть повністю визначати проблематику РБД. Зручною абстракцією для опису суті і напрямів розв'язання багатьох реальних задач у таких СУБД, як Oracle, Informix, Sybase, є використання транзакцій [3]. Тому метою даної статті є розглядання транзакції як основного засобу реалізації процесу подання запитів.

Результати досліджень

Впровадження транзактної організації оброблення інформації як динамічних, так і розподілених варіантів реляційних БД дозволяє знизити вартість робочих станцій користувачів, зокрема, наведені витрати на зовнішню пам'ять (ЗП), і підвищити ефективність використання обчислювальних ресурсів ВК. Всі елементи оброблення даних, таким чином, пов'язані з поняттям транзакції. Транзакція звичайно розглядається як програмна одиниця, що виконується в повному обсязі і реалізує з СУБД користувачів та додатків. Під транзакцією розуміють основну інформаційно-операційну одиницю подання запитів та інформаційних обмінів у базах даних, що реалізується у вигляді одного коректного програмного модуля.

Структуру транзакції представляє кортеж загального вигляду:

<BOT, Read/Write, Commit/Abort EOT>

де **BOT** – початок транзакції (Begin Of Transaction); **Read/Write** – оператори введення/виведення;

Commit/Abort – оператори виконання або припинення транзакції; **ЕОТ** – кінець транзакції (End Of Transaction).

Виконання транзакцій пов'язується з деякими послідовностями операцій над інформаційними ресурсами. Нехай $T = \{T_i\}_{i=1}^k$ – множина транзакцій, які можуть бути одночасно задіяні системою, $F = \{f_s\}_{s=1}^n$ – множина можливих операцій, що виконуються транзакціями. Транзакція T_i характеризується відповідною їй послідовністю операцій, де l_i – кількість операцій транзакції T_i ; $f_{ij} \in F \forall j = \overline{1, l_i}$.

Алгоритми управління транзакціями повинні задовольняти вимоги, що стоять перед системою підтримки транзакцій, головні з яких полягають у забезпеченні основних властивостей транзакцій.

Основними властивостями транзакцій є:

- атомарність;
- ізольованість;
- довго тривалість;
- узгодженість (несуперечність).

Основні властивості транзакцій частково описані в різній літературі [4 – 6]. Проте в жодному з наведених джерел не дано строгого математичного опису цих властивостей. **Математичний опис** перших трьох властивостей транзакцій наведений нижче.

Ці властивості порівнянні з набором аксіом, обов'язковість виконання яких впливає з необхідності забезпечення несуперечності інформаційних ресурсів при паралельному функціонуванні процесів.

Властивість 1: атомарність. Властивість атомарності означає успішне виконання операцій транзакції. Поставимо у відповідність кортежу τ_i булевий вектор $V = (b_i^1, b_i^2, \dots, b_i^{l_i})$: $b_i^j \in \{0,1\} \forall j = \overline{1, l_i}$, причому

$$b_i^j = \begin{cases} 1, & \text{якщо операція } f_{ij} \text{ виконана успішно;} \\ 0, & \text{у протилежному випадку.} \end{cases}$$

Властивість атомарності T_i вимагає, щоб або були виконані всі операції, або не виконана жодна з них, тобто:

$$\bigwedge_{j=1}^{l_i} b_i^j \vee \bigwedge_{j=1}^{l_i} \bar{b}_i^j = 1.$$

Таким чином, якщо транзакція переривається, то атомарність вимагає, щоб часткові результати, проведені транзакцією, були анульовані. Отже, атомарність означає неподільність транзакції, тобто або транзакція виконується повністю, або викидається з процесу оброблення.

Властивість 2: ізольованість. Ізольованість розуміється в тому смислі, що в процесі виконання транзакція ніяк не впливає на інші транзакції. Таким чином, транзакція, що не завершилася, не може передавати свої результати іншим транзакціям.

Властивість 3: довготривалість. Властивість довго тривалості можна подати у вигляді:

$$\forall i, t : t_i < t < t_{i+1} : \sigma(t) = \text{const},$$

де t_i – моменти часу, у які відбуваються зміни глобального стану ресурсів (БД) $\sigma(t)$; $\sigma(t)$ – дискретний процес зміни глобального стану ресурсів (БД).

Властивість довготривалості означає, що результат транзакції після її завершення не може бути змінений жодним чином. Іншими словами, якщо відбулося завершення транзакції, тобто всі її операції виконані успішно, то система повинна гарантувати, що результати виконання операцій транзакцій не будуть втрачені.

Властивість 4: узгодженість. Суть цієї властивості в наступному. Стан бази даних може змінюватися транзакціями в процесі актуалізації (оновлення даних), у процесі вставки даних (insert) або в процесі знищення даних (delete) таким чином, що кожна транзакція переводить базу даних з одного цілісного стану в інший.

Цілісність вважається основною властивістю бази даних, і обмеження на цілісність визначаються моделлю БД, її схемою і забезпечують узгодженість всієї її інформації, що зберігається в БД. Інформація в БД вважається узгодженою, якщо всі її інформаційні об'єкти й операції з цими об'єктами визначені явним чином при початковому формуванні структури БД. У процесі виконання транзакцій база даних на якийсь час втрачає свою узгодженість і відновлює її тільки після закінчення транзакції, тому властивість узгодженості транзакцій розуміється як властивість, завдяки якій у результаті маніпуляцій з транзакціями вся база даних залишається узгодженою.

Слід відзначити, що при виконанні транзакції (у тому числі і плоскої) можливо виникнення неузгодженості як самих транзакцій, так і всієї бази в цілому. Подібна неузгодженість транзакцій може виникнути в результаті самих рядових помилок введення – виведення, а неузгодженість бази даних може виникнути через відмови в програмно-технічному середовищі виконання. У першому випадку виникає завдання відновлення транзакцій, у другому випадку – завдання аварійного відновлення системи.

Висновок

Основні аспекти, пов'язані з управлінням транзакціями в системах управління базами даних, повністю залежать від характеристик даних і необхідних прикладних процесів. Таким чином, з урахуванням математичного опису наведених властивостей транзакцій, вона є найбільш зручною абстракцією для формалізації процесу подання запитів на оброблення інформації в СУБД, яка обслуговує кластерні фрагменти розподіленої бази даних.

Список литературы

1. Конолли Т., Бегг К., Страчан А. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. – М.-К.-СПб.: Изд. дом «Вильямс», 2000. – 1120 с.
2. Якубайтис Э.А. Информационно-вычислительные сети. – М.: Финансы и статистика, 1984. – 232 с.
3. Цейтлин Н.А., Акимова Ю.А., Литвинов Ю.С. Статистический анализ моделируемых параметров обработки оперативных транзакций // Системы информационного воздействия. – Х.: НАНУ, ПАНИ, ХВУ. – 1996. – С. 93-99.
4. Азаров С. Краткий очерк современной проблематики баз данных // Компьютеры + Программы. – К.: Колмиздат. – 1997. – № 3 (36). – С. 30-35.
5. Зиновьев Э.В. Управление сетевыми информационными процессами и ресурсами. – Рига: Зинатне, 1987. – 303 с.
6. Urman S. *JSQL. Programming.* – New York: McGraw-Hill, 2005. – 788 p.

Надійшла до редакції 2.08.2006

Рецензент: д-р техн. наук, проф. В.А. Краснобаев, Харківський університет Повітряних Сил ім. І. Кожедуба, Харків.