

УДК 002.53:004.65

ВЫЧИСЛИТЕЛЬНЫЕ МОДЕЛИ ПАРАЛЛЕЛЬНЫХ ТРАНЗАКЦИОННЫХ СЕРВЕРОВ

Ж. Ндикумагенге,¹

аспирант

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»

Рассматривается проблема разработки вычислительных моделей для транзакционных серверов данных, основанных на транзакционной концепции, для решения задачи безопасного хранения данных в случае параллельного доступа к данным и отказов.

Ключевые слова — база данных, вычислительная модель, параллельный процесс, распределенная система, транзакция, сервер данных.

Введение

Управление транзакциями играет ключевую роль в информационных системах и, в частности, в системах распределенных баз данных (БД). Первоначально транзакционная концепция развивалась в контексте систем управления БД как парадигма обеспечения параллельного доступа к разделенным БД и устранения отказов. Анализ оперативной обработки транзакций в реальном времени в таких предметных областях, как обмен банковских средств, туристический бизнес показывает, что применение транзакционной концепции может включать современный бизнес-процесс, который имеет дело с распределенными системами потенциально крупных масштабов, с межоперационными компонентами и транзакционными серверами данных. Трехуровневая архитектура распределенной системы позволяет отделить транзакционные серверы данных от прикладных программ (приложений), посылающих запросы для инициализации транзакций. Под такой архитектурой ответственность за соблюдение АСИД-свойств (Атомарность, Согласованность, Изоляция и Долговечность) лежит на транзакционных серверах данных [1]. АСИД-контракт между прикладной программой и транзакционными серверами данных требует, чтобы про-

грамма динамически разграничила транзакции, а также получила желательный результат и успешное или неуспешное завершение транзакций. Менеджер запросов приложения может инициировать большое число транзакций [1]. При такой ситуации положение транзакции может находиться под угрозой при запуске большого числа конкурирующих транзакций, вызывая состояние, которое приводит к блокировке к отказу в выполнении (Lock Thrashing) [2]. Теория и алгоритмы параллельного выполнения транзакций для транзакционных серверов данных, базирующиеся на модели страницы (модель читатель/писатель) и модели объекта, позволяют адекватно и эффективно решить задачи управления параллельных процессов распределенной системы.

Постановка задачи

Архитектура системы БД имеет 6 уровней. Языковой и интерфейсный уровень образуют интерфейсы прикладных разработчиков и прикладного программного интерфейса (SQL), уровень запросов и оптимизации, отвечающий за детальное разбиение запросов на элементарные операции (действия) и применяющий решения об оптимизации их выполнения за счет, в частности, применения к ним механизма многопоточной обработки. Уровень выполнения запросов отвечает за выбор и подготовку методов и алгоритмов (выборка, сортировка, слияние и т. д.), необходимых уровню оптимизации запросов. Уровень доступа следит за управлением индексирования, доступностью и манипуляцией сохраняемых данных,

¹ Научный руководитель — кандидат технических наук, доцент, заместитель заведующего кафедрой автоматизированных систем обработки информации и управления Санкт-Петербургского государственного электротехнического университета «ЛЭТИ» В. В. Цехановский.

а организация сохраняемых данных на страничных дисках лежит на уровне сохранения. Он отвечает также за операции ввода/вывода с диска и кэширование памяти (рисунок).

Данные в системах БД сохраняются в виде записи строкового типа последовательных байтов и состоят из полей или столбцов. Записи сохраняются на дисках в виде страниц размером 16 КБ. Страница — минимальная единица данных, переводимая между диском и оперативной памятью — служит единицами кэширования в памяти. Диск может читать или писать одну или много страниц (блок) только один раз. Доступ к данным осуществляется сканированием записей данных по специфицированному ключу выбора. Существует много разработанных целым поколением исследователей структур данных (дерево поиска, хеш-таблицы, В+ дерева и т. п.), облегчающих осуществление таких типов операций. Трехуровневая архитектура системы распределенных БД позволяет выполнять параллельные транзакции и маскирует параллельности и возможные отказы от разработки приложений (или прикладных программ). Данная возможность обусловлена абстрактной концепцией, названной транзакцией, служащей интерфейсным контрактом между прикладной программой и транзакционными серверами данных. Прикладная программа специфицирует границы транзакций, указывает начало и конец вызова при выполнении транзакций. Транзакционные серверы данных обеспечивают соблюдение АСИД-свойства запускаемых транзакций [1]. Однако транзакционные серверы данных должны иметь компоненты данных, обеспечивающие параллельную обработку и восстановление и обладающие высокой производительностью, а также для выполнения большого числа транзакций выдавать выходные результаты за очень короткое время. Приложения для обработки транзакций

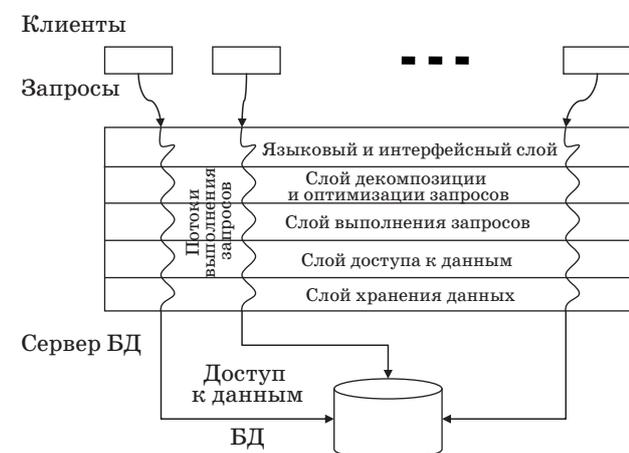
(транзакционные серверы данных) относятся к классу критических для бизнеса или иной деятельности. Транзакционное приложение занимается обновлением информации в БД. Нагрузка на такое приложение обычно предсказуема, значительную долю в ней занимают запросы на обновление. Для управления процессами используются мониторы транзакций. Они координируют поток информации между различными клиентами, иницилирующими запросы, и приложением обработки транзакций, которое отвечает на эти запросы [3].

Модель страницы

Вычислительные модели для транзакционных серверов данных реализуются разными способами в зависимости от выбранного уровня абстракции, от среды программирования или используемого приложения. Принцип разделения (применения) моделей обусловлен методологией, позволяющей отражать главные и значимые компоненты [1]. Единый подход, в котором вычислительные модели будут использоваться, характеризуется следующей стратегией:

- 1) определение элементарных операций на объекты, являющихся неделимыми, атомарными и изолированными от других операций;
- 2) определение (формирование) последовательности элементарных операций или частичный порядок элементарных операций в совокупность операций, обеспечивающих АСИД-свойства;
- 3) образование расписаний транзакций, как абстрактное понятие параллельного выполнения набора частичного порядка элементарных операций, за счет формирования ряда познаваемых индивидуальных транзакций;
- 4) выбор из синтаксических корректных расписаний тех, которые удовлетворяют АСИД-свойствам;
- 5) разработка протоколов или алгоритмов создания корректных расписаний, которые по вызовам прикладных программ запускают и выполняют операции динамически.

Эти шаги значимы для любой виртуальной транзакционной модели, разработка которой идет от базовых (атомарных) операций к протоколам. Модель страницы обеспечивает эффективный доступ к страницам данных и отображает все операции высшего уровня в операции чтения и записи на страницах. Каждая операция чтения или записи предполагает неделимую операцию независимо от того, имеет ли это место в кэш-памяти или на диске. Операции на странице представляют собой строгую форму абстракции. Модель страницы определяет синтаксические корректные транзакции. Формально сервер дан-



■ Словесная архитектура системы БД

ных содержит конечное множество $D = \{x, y, z, \dots\}$ неделимых и идентифицируемых элементов с неделимыми операциями чтения и записи. Отношение полного порядка определяется моделью страницы транзакции в виде конечной последовательности шагов (действий) типа $r(x)$ или $w(x)$:

$$t = p_1 p_2 \dots p_n,$$

где $n < \infty$; $p_i \in \{r(x), w(x)\}$, $1 \leq i \leq n$; $x \in D$, x — элемент (с индексом и без него) из D ; r — операция чтения, а w — операция записи. Простая транзакция имеет вид

$p_j = r(x)$ — шаг j читает элемент данных x ;

$p_j = w(x)$ — шаг j пишет элемент данных x .

Такая формализация определяет некую семантику для отдельной транзакции, такой, что две разные транзакции имеют разные шаги, но возможные одни и те же типы шагов [1].

Семантическая интерпретация транзакции на уровне модели страницы базируется на $r(x)$ и $w(x)$ операции.

Если $p_j = r(x)$, то j -й шаг данной транзакции является операцией чтения, элемент x получает текущее значение v_j :

$$v_j = x.$$

Если $p_j = w(x)$, то j -й шаг данной транзакции является записью, текущее значение элемента x зависит от сочетания значений v_j , прочитанных до шага j :

$$x = f_j(v_{j_1}, v_{j_2}, \dots, v_{j_k}),$$

x — возвращаемое значение такое, что $\{v_{j_1}, v_{j_2}, \dots, v_{j_k}\} = \{j_r | p_{j_r} \text{ шаг чтения и } j_r < j\}$.

Все значения v_j , $1 \leq i \leq k$, считанные до j -го шага транзакции t , используются как параметры в функцию f_j .

Формально модель страницы — транзакция — есть пара $t = (\Omega, \rho)$, где Ω — конечное множество шагов вида $r(x)$ или $w(x)$, $x \in D$ и $\rho \subseteq \Omega \times \Omega$ есть отношение частичного порядка на множестве Ω такое, что если $\{p, q\} \subseteq \Omega$, где p и q имеют доступ к одному и тому же элементу данных и хотя бы одна из этих двух операций является операцией записи, то $p \rho q \vee q \rho p$.

В данной модели любые пары операций, одна из которых операция записи, должны быть упорядочены. В каждой транзакции каждый элемент данных может быть прочитан или записан максимум один раз и чтение элемента данных невозможно после записи.

Модель объекта

Модель объекта дает возможность представления вызовов любых методов доступных объектов

в виде абстрактных типов данных. Реализация объекта и его операции требуют запроса некоторых типов объектов нижнего уровня. Графическое представление транзакций имеет древовидную структуру с их вызываемыми операциями в качестве вершин [1]. Конечные вершины дерева транзакций являются элементарной операцией в смысле модели страницы.

Это необходимо, чтобы обеспечить модульность структуры деревьев, отражение всех соответствующих аспектов операционного выполнения, строгость рассуждений о параллелизме между транзакциями.

Дерево транзакции строится так, чтобы его «листья» (конечные вершины) были элементарными операциями чтения и записи на страницах. Транзакция t — конечное дерево помеченных вершин.

Формализация модели объекта транзакции проводится следующим образом:

- идентификацией транзакции как метки корневого узла (вершины) дерева;
- именами и параметрами вызываемых операций как метки внутренних вершин;
- моделью страницы как метки конечных вершин (листьев);
- отношением частичного порядка « ρ » на множестве Ω конечных вершин таким, что для всех конечных вершин операции p и q , где p имеет вид $w(x)$ и q имеет вид $r(x)$ или $w(x)$ или наоборот, мы имеем $p \rho q \vee q \rho p$.

Для изучения и исследования множественных транзакций, выполняемых параллельно, постановка задачи должна формироваться объединением вовлеченных операционных деревьев транзакций в «лес» вложенных транзакций (могут быть распределенные и/или параллельные транзакции). При этом необходим просмотр частичного порядка выполнения операций конечных вершин и неявно полученного частичного порядка выполнения относительно высокоуровневых операций. Частичный порядок выполнения операций определяется объединением конечных вершин всех вызванных деревьев. Из слияния структур всех деревьев можно получить одно дерево транзакций, на основе которого лучше изучается внутреннее упорядочение вершин операций исходных деревьев, получившихся из частичного порядка выполнения операций конечных вершин. Этот подход позволит нам исследовать не только «следы» операций на уровне конечных вершин, но также параллелизм или последовательные упорядочения в среде операций высокого уровня. Суждение параллелизма на различных древесных уровнях действительно необходимо для эксплуатации объектной модели ради оптимизации производительности [1]. Вложен-

ные транзакции (распределенные и/или параллельные) важны для распределенных систем, потому что они представляют естественный способ распределения транзакций по нескольким компьютерам и превращают работу исходной транзакции в логическую форму [4].

Заключение

Транзакции играют важную роль в управлении вычислительными процессами распределенной системы БД, являясь связующим звеном между прикладной программой и рядом транзакционных серверов данных, обеспечивая объединение множества запросов на эти серверы в логические единицы. Транзакционная концепция об-

легчает работу разработчиков прикладных программ и снижает стоимость обслуживания. Высокий параллелизм, большая пропускная способность и быстрота восстановления от отказов являются важными функциональными требованиями для современных распределенных информационных систем. Модели страницы и объекта являются не только теоретическим обоснованием данной проблематики, но и прикладным средством управления современных информационных систем. Абстрактное основание данных моделей может использоваться для получения конкретных алгоритмов параллельной обработки конкурентных транзакций и фактически жизнеспособных, ориентированных системных решений.

Литература

1. Weikum G., Vossen G. Transactional information systems, Theory, algorithms and practice of concurrency control and recovery. — San Francisco: Morgan Kaufmann Publishers, 2002. 853 p.
2. Philip A. B., Newcomer E. Principles of Transactions processing for the systems professional. — San Francisco: Morgan Kaufmann Publishers, 1997. 357 p.
3. Гома Х. Проектирование систем реального времени, параллельных и распределенных приложений. — М.: ДМК Пресс, 2002. — 704 с.
4. Таненбаум Э., ван Стен М. Распределенные системы. Принцип и парадигмы. — СПб.: Питер, 2003. — 789 с.

УВАЖАЕМЫЕ АВТОРЫ!

С сентября 2009 г. основные элементы статей, размещенные на платформе РУНЭБ, индексируются в крупнейшей поисковой системе Интернета Google.

На сайте РУНЭБ (<http://www.elibrary.ru>) доступна новая услуга — «обсуждение статьи». Авторы и читатели теперь могут вступить в диалог и ответить на вопросы и комментарии друг друга.