СЕРИАЛИЗАЦИЯ ТРАНЗАКЦИЙ В БАЗАХ ДАННЫХ

Александр ЛУПУ

Департамент Программной Инженерии и Автоматики, Группа TI-192, Факультет Вычислительной Техники и Микроэлектроники, Технический Университет Молдовы, Кишинёв, Республика Молдова

Автор корреспондент: Александр ЛУПУ, e-mail: lupu.alexandru@isa.utm.md

Научный руководитель: Дориан САРАНЧУК, DISA, FCIM, UTM

Аннотация. *Данная работа посвящена сериализации транзакций баз данных. В работе рассматриваются виды конфликтов, а также методы захвата и освобождения.*
Ключевые слова: *сериализация, транзакции, захват, освобождение.*

Введение

Одним из методов повышения производительности некоторых сериализованных проектов является сериализация транзакций. Предполагается, что основной задачей компонента SUBD, отвечающего за обработку транзакций, является предоставление такого механизма. Реальная автономия пользователя обеспечивается в системах, поддерживающих сериализацию транзакций [1].

Основным препятствием для реализации является выбор стратегии сериализации для ряда транзакций, что, несомненно, уменьшит их параллельность. Первое, что приходит на ум, как правило, считается выполнением сделок позже. Однако существует несколько сложностей, которые операторы различных транзакций могут выполнять в любом порядке при сохранении серийности. Примеры включают простое чтение операций и операций, которые не конфликтуют друг с другом на объектах базы данных.

Виды конфликтов

Между транзакциями существуют следующие виды конфликтов: [1]
W-W - транзакция 2 пытается изменить объект, измененной не закончившейся транзакцией 1;
R-W - транзакция 2 пытается изменить объект, прочитанной не закончившейся транзакцией 1;
W-R - транзакция 2 пытается читать объект, измененной не закончившейся транзакцией 1.

Практические методы сериализации транзакций основывающиеся на учете этих конфликтов

Методы «привлечения» и «освобождения» объектов, созданных по инициативе сделок, используются для обеспечения серийной транзакции: сделка «притягивает» объект, что, в свою очередь, приводит к его блокированию для других сделок и освобождает его только после завершения собственной сделки [2].

В этом случае результаты объекта по чтению одной транзакцией несовместимы с результатами того же объекта с чтением другой транзакции по записи, а последствия первого объекта различными транзакциями по записям не совместимы.

Считается, что более широко используемый подход в SUBD, основанный на архитектуре «клиент-сервер», является решением, которое достигает соответствия двухфазному протоколу объектов DB. В целом, протокол устанавливается тем фактом, что объект будет помещен в соответствующий объект, прежде чем какой-либо вид операции будет выполнен над объектом базы данных от имени транзакции. В согласовании с сведениями протоколом выполнение транзакции разбивается на 2 фазы:

1) транзакции – скопление захватов;
2) высвобождение захватов.

Самая большая проблема с последующим двухфазным процессом заключается в определении того, что квалифицируется как объект для achvata. Верховные вариации в реляционных базах данных, вероятно, включают: [3]

Файл - это физическая сущность, которая может хранить отношения и индексы; таблица - это сущност, используемая в экономическом анализе, которая основана на большом количестве записей для указанной связи; запись - это простое физическое вещество, найденное в базе данных; страница данных - физическая субстанция, содержащая записи одной или нескольких отношений, индекса или индексной информации.

Наименьшие потери будут включены в диапазон, так как чем больше объект, тем меньше будет каталога системы. Если вы выберете определенные элементы, чтобы назначить для файла или отношения, строка призраков будет устранена. Однако вероятность транзакционных аварий возрастает, и их допустимый уровень параллельной производительности падает больше всего, когда они используются для больших объектов. Чтобы увидеть события в ситуациях, когда их нет, синхронизированный сценарий намеренно ухудшается во время выполнения при усилении объекта.

Согласно этой аббревиатуре, транзакция на самом деле является завершенным блоком ссылок на основу данных и некоторые связанные с ними операции, для которых обеспечивается удовлетворение четырех требований, таких как так называемые квазиты: Атомность: Транзакционная деятельность разделяет неразрывный блок данных на начало и конец [2].

Либо вся эта единица производится, либо она вообще не генерируется. Возвращение к первоначальному состоянию происходит в том случае, если в процессе сделки возникает ошибка; последовательность: все предметы присутствуют и находятся в согласованном состоянии при заключении сделки;

Для предотвращения перекрестного влияния между транзакциями координируется то, чтобы все типы приложений имели одновременный доступ к разделенным объектам.

Основными режимами синхронизационных захватов являются:

Монопольный режим – X (эксклюзивный), представляющий собой монополистическое прикосновение объекта и необходимое для выполнения манипуляций касания, расширения и трансформации. Кооперативный режим — S (Shared), представляющий собой отделяемое прикосновение объекта и необходимый для выполнения операции чтения объекта. Захват объекта одной транзакцией чтения несовместим с захватом другой трансакцией того же объекта записи, а захваты первого объекта различными транзакционными записями являются не совместимыми. Однако захват объекта несколькими транзакциями чтения совместим, то есть несколько транзакций могут декларировать объект [2]. Критерии сопоставимости захватов 1-го объекта различными транзакциями изображены в таблице 1:

Таблица 1

Правила совместимости захватов одного объекта разными транзакциями

	X	S
-	да	да
X	нет	нет
S	нет	да

Вероятные состояния элемента перечислены в первой колонке как синхронизационные диапазоны. Когда указано, «-» обозначает состояние объекта, для которого в основном нет формулировки. Транзакция, которая запрашивает волна синхронизации объекта BD, останавливается до тех пор, пока волна от этого объекта не будет выпущена, потому что она в настоящее время используется другой транзакцией в неконкурентном режиме. Обратите внимание, что текст «нет» в нашей таблице фактически соответствует упомянутым более вероятным событиям транзакций по доступу к объектам базы данных. WW, RW и WR Сравнительность S-снимков аналогична тому, чем событие RR не является в реальности.

Методы сериализации транзакций

Сериализация транзакций имеет два фундаментальных различия: отличия, основанные на синхронизации объектов базы данных, и отличия на основе использования временных ярлыков. Экспозиция инстанций транзакций и их устранение порождают сущность обеих различий. Ниже мы рассмотрим эти различия относительно молодым взглядом.

На самом деле, существует два типа оптимизма - жизненно важный оптимизм и пессимизм - для каждого из этих различий [1].

При использовании пессимистических исторических подходов, где эпизоды бывают частыми, инциденты замечаются и обращаются с ними, как только они возникают. (actual or understood). Основой жизненно-утверждающих подходов является представление о том, что все операции трансформации базы данных производят результаты, которые хранятся в рабочей памяти транзакций. Только на этапе фиксации транзакции база данных подвергается действительным изменениям. Он проверяется, чтобы увидеть, были ли какие-либо инциденты с другими транзакциями в течение этого времени [1].

Дальше мы ограничимся рассмотрением больше популярных пессимистических видов способов сериализации транзакций. Пессимистические способы относительно элементарно модифицируются в собственные жизнеутверждающие варианты.

Типы временных целей. Для использования временных целей был разработан иной метод сериализации транзакций, который выполняется безупречно в соответствии с требованиями редких транзакционных событий и не требует создания понимания текущей транзакции. Вышеприведенное иллюстрирует основное понятие метода: в случае, если транзакция T1 началась до транзакции T2, система обеспечивает сопоставимый режим исполнения, как если бы T1 был завершен до начала T2.

Для каждой транзакции T, которая соответствует времени, начинающемуся с T, требуется краткосрочный знак t. Транзакция T записывает временную отметку и вид операции (читать или изменять) для каждой операции, выполненной на объекте r.

Транзакция T1 имеет следующие эффекты перед выполнением операции на объекте r:

Найдя объект, вы можете убедиться, что транзакция T все еще активна. T1 обозначает объект r и завершает его действие, если T завершен.

Транзакция T1 проверяет конфликты операций, если транзакция T не завершена. Если между операциями нет конфликта, транзакция T1 завершает свою операцию, а объект r остается временным тегом с малой значимостью.

При столкновении двух операций T1 и T, если t(T)> t (T1) (т.е. транзакция T более «молодая», чем трансакция T), T фиксируется, и T1 продолжает работать; если t/T t(T1) (то есть транзакции T старше, чем T1), T1 получает новую временную отметку, и она начинается заново.

Потенциально более частые транзакции, чем при использовании транзакций синхронизации, являются одним из недостатков метода временных тегов.

Конфликт сделок ориентирован более прямо как результат, поэтому. Несмотря на это, производство массивных маркеров времени столкновения с соотношением достигнутых успехов в распределенных системах недостаточно элементарно. (this is a different enormous science). Тем не менее, эти недостатки компенсируются в распределенных системах тем, что нет необходимости решать заторможенность; как мы уже отмечали, построение понимания ожиданий в распределённых системах довольно дорогое.

Заключение

В заключении можно отметить, что механизм сериализации транзакций является основной функцией компонента СУБД, ответственного за управление транзакциями.

Библиография

1. Методы захвата и освобождения – [online] [дата обращения 02.03.2023], Доступно: e-educ.ru/bd28.html
2. Сериализация. транзакций – [online] [дата обращения 02.03.2023], Доступно: https://studfile.net/preview/5133502/page:46/
3. Сериализация транзакций с использованием синхронизационных захватов – [online] [дата обращения 02.03.2023], Доступно: https://studme.org/411523/informatika/serializatsiya_tranzaktsiy_ispolzovaniem_sinhronizatsionnyh_zahvatov_blokirovok