

МЕТОД РАТИФИКАЦИИ, ОБЕСПЕЧИВАЮЩИЙ СЕРИАЛИЗУЕМОСТЬ ПАРАЛЛЕЛЬНО ВЫПОЛНЯЕМЫХ ТРАНЗАКЦИЙ В РАСПРЕДЕЛЕННЫХ СИСТЕМАХ

Багирова Севиндж, Алиев Алекпер

Бакинский Государственный Университет

Аннотация

Данная работа посвящена проблемам обеспечения сериализуемости параллельно выполняемых транзакций в распределенных системах. Анализируются различные методы управления параллелизмом. Предлагается алгоритм ратификации параллельно выполняемых транзакций, и доказывается утверждение о результативности предложенного алгоритма.

Abstract

This paper is devoted to the problems of ensuring serializability of parallel transactions in distributed systems. Various methods of managing concurrency are analyzed. An algorithm for the ratification of parallel transactions is proposed, and the statement on the effectiveness of the proposed algorithm is proved.

Введение

Теоретические исследования в области распределённых систем обработки информации и их применения на практике развиваются быстрыми темпами. Одним из наиболее развитых видов распределённых систем являются системы Распределённой Базы Данных (РБД) [1,2]. Эти системы могут быть представлены как интегрированные системы баз данных, состоящие из отдельных локальных баз данных, географически распределённых и соединённых с помощью компьютерных сетей. Основные задачи в управлении системы РБД могут быть разделены на три различные группы как управление *параллелизмом* (concurrency control), оптимизация выполнения запросов (query processing optimization) и обеспечение надёжности (reliability).

Фундаментальной задачей при проектировании системы управления РБД является корректное управление параллельного доступа множества различных пользователей к РБД. Другими словами, главной задачей является разработка соответствующего алгоритма управления параллелизмом транзакций. Алгоритм управления параллелизмом предназначен для обеспечения корректного выполнения каждой транзакций, производимой в системе и предупреждения конфликтов в РБД. Управление параллелизмом занимается предотвращением конфликтов, появляющихся в результате некорректного параллельного обращения к одним и тем же данным. При этом важной является обеспечение *непротиворечивости* (consistency) и *целостности* (integrity) РБД. Далее будем использовать просто термин «целостность» для обозначения обоих понятий.

Система управления транзакциями

Общепринятым методом обработки информации в распределенных системах является транзактная обработка. Каждая транзакция переводит систему из одного целостного состояния в другое целостное состояние. Последовательное выполнение транзакций также переводят систему в целостное состояние. Когда транзакции выполняются параллельно, то при этом возможно, нарушение целостности РБД. Поэтому необходимо создать такую систему управления, которая обеспечивала бы целостность РБД. Такой системой является система управления транзакциями, главной задачей которой является управление параллелизмом транзакций в рамках всей системы РБД.

К системе управления транзакциями предъявляются ряд требований: атомарность транзакций; статистическая справедливость параллельно выполняемых транзакций;

сериализуемость распределенного плана выполнения транзакций; производительность системы; отказоустойчивость системы и т.д. Одним из важных требований является обеспечение сериализуемости. Некоторый распределенный план выполнения транзакций называется сериализуемым, если он эквивалентен, по крайней мере, одному последовательному (сериальному) плану. Признаком сериализуемости распределенного плана выполнения транзакций является ацикличность соответствующего графа зависимостей.

Методы управления параллелизмом

Методы управления параллелизмом призваны обеспечивать сериализуемость транзакций, как можно меньше ограничивая при этом их параллельное выполнение. Все алгоритмы управления транзакциями подразделяются на два класса: пессимистические и оптимистические. Существуют 3 метода управления параллелизмом: метод блокировки; метод временных меток; метод ратификации.

Одним из первых предложенных методов управления параллелизмом являются методы блокировки. Основной идеей блокировок заключается в том, что если для выполнения некоторой транзакции необходимо, чтобы некоторый объект не изменялся без согласия этой транзакции, то этот объект должен быть заблокирован. В методе блокировки конфликты между транзакциями из-за несовместимости режимов блокировки приводят к потерям производительности. Когда транзакция «отказывается» в блокировке некоторого объекта, она должна ждать снятия блокировки с данного объекта другой транзакцией. В этом случае возможна ситуация кругового ожидания, приводящая к *тупику* (deadlock). Для устранения возникшего тупика необходимо абортить, по крайней мере, одну из транзакций. Это в свою очередь предполагает ликвидацию всех изменений в РБД, произведенных абортируемой транзакцией, и снятие всех блокировок с занятых ей объектов. Основным недостатком пессимистического подхода являются простои во время тупиков и растрата машинных ресурсов на их обнаружение и разрешение.

Упорядочение по временным меткам - это один из пессимистических подходов, посредством которого последовательность выполнения транзакций подчиняется некоторой приоритетной дисциплине, в частности соответствие с их временными метками. Временная метка - это уникальное число, которое присваивается транзакции или объекту и выбирается из монотонно возрастающей последовательности, которая часто является функцией от времени. В распределенной системе, где каждый узел имеет свой собственный уникальный идентификационный номер, временная метка образуется конкатенацией значений локального времени и идентификационного номера. Поскольку каждая транзакция имеет уникальную метку, то упорядочение по временным меткам является сериализуемым. При этом обнаруживается недостаток данного подхода, обусловленный необходимостью хранения и обработки большего объема информации, связанной с временными метками. Хотя при этом не возникает тупиковых ситуаций, однако, откаты при обнаружении нарушения непротиворечивости РБД аналогичны откатам при обнаружении тупиковых ситуаций, свойственным методам блокирования.

Основная идея метода ратификации - максимизировать объем выполняемой работы. Эти алгоритмы дают шанс работать всем транзакциям, но разрешают благополучно завершиться только тем из них, чье завершение не влечет нарушения целостности РБД. С точки зрения менеджера управления транзакций при использовании оптимистического протокола выполнение транзакции делится на три концептуальные фазы: *чтение, проверка, запись*. В течение фазы чтения транзакция работает параллельно с другими транзакциями без каких-либо ограничений, но все измененные данные записываются в личную рабочую память транзакции, а не в базу. Когда транзакция завершает свою работу менеджер инициирует фазу проверки транзакции на наличие конфликтов с другими транзакциями. Такая проверка называется *ратификацией* (Validation). Если проверка окончилась удачно (т.е. конфликтов не обнаружено), то сделанные транзакцией изменения переносятся в базу и становятся видимыми другим

транзакциям. В противном случае, сама транзакция или конфликтующие с ней транзакции аннулируются, т.е. результаты выполнения всех конфликтующих транзакций стираются из личной рабочей памяти, а сами эти транзакции подлежат последующему рестарту. Фаза ратификации является решающей для обеспечения целостности, так как именно в этой фазе анализируется наличие конфликтов и затем принимается решение о фиксации или об откате всех или некоторых транзакций.

Таким образом, первый метод, метод блокировки, является индикационным, т.е. применение его обеспечивает обнаружение несериализуемого распределенного плана. Если таковой обнаружен, то производится откат одной или нескольких транзакций. Второй метод, метод временных меток, является превентивным, т.е. не допускает образования несериализуемого распределенного плана, вновь поступающие транзакции, которые могут вызвать несериализуемость, откатываются. Третий метод, метод ратификации, также является превентивным, его преимущество в том, что откатываются только те транзакции, которые действительно (а не потенциально) образуют несериализуемый распределенный план.

Алгоритм ратификации

Идея алгоритма основана на включения в систему управления транзакциями механизма прямой проверки некоторого множества транзакций на отсутствие цикла в соответствующем этому множеству графе зависимостей. Назовем это множество «испытываемым». Если такой цикл отсутствует и возникновение его в будущем также невозможно, проверенное множество считается ратифицированным, и транзакции, принадлежащие этому множеству, могут выполняться в том порядке, в котором они стоят в очередях. При наличии цикла одна из транзакций, инцидентная ему, откатывается и через некоторое время стартует вновь. В работе в рамках прямого метода предложен один из алгоритмов ратификации, который будем называть алгоритмом растущего дерева. Он основан на дискретизации процесса инициации транзакций и построении в каждом узле дерева зависимостей между транзакциями, которое в процессе роста определяет наличие или отсутствие цикла в графе зависимостей. Дискретизация процесса инициации происходит для того, чтобы циклы в графе зависимостей не образовывались за конечное время. Дискретизация заключается в следующем. Временная ось всех узлов делится на равные отрезки, называемые интервалами дискретности. Начальные отрезки каждого интервала дискретности называются интервалами инициации. Эта величина интервала дискретности подбирается эмпирически и должна удовлетворять определенному неравенству. Алгоритм для множества узлов, иницировавших транзакции в одном и том же интервале инициации, заканчивается, если все транзакции, иницированные в этом интервале, либо ратифицированы, либо ликвидированы в процессе отката.

Заключение

Таким образом, одним из важных требований, предъявляемым к системе управления транзакциями, является сериализуемость транзакций. В данной работе рассматриваются проблемы обеспечения сериализуемости параллельно выполняемых транзакций в распределенных системах. Анализируются различные методы управления параллелизмом, и дается их сравнительный анализ. Предлагается алгоритм ратификации, обеспечивающий сериализуемость параллельно выполняемых транзакций в распределенных системах, и доказывающее утверждение о результативности предложенного алгоритма.

Список использованных источников

1. Schneider F.B. Paradigms for distributed programs // Lecture Notes in Computer Science – 1985. - № 190. – P.431-480.
2. Ozsu M., Valduries P. Principles of Distributed Database Systems, Prentice-Hall, 1999.