

С.В. Калиниченко¹
 П.Е. Котиков²
 А.А. Свиначук³

S.V. Kalinichenko
 P.E. Kotikov
 A.A. Svinarchuk

**ПРИМЕНЕНИЕ ПРОТОКОЛА
 ДВУХФАЗНОГО ПОДТВЕРЖДЕНИЯ
 ТРАНЗАКЦИЙ В РАСПРЕДЕЛЕННЫХ
 БАЗАХ ДАННЫХ**

**THE APPLICATION OF THE TWO-
 PHASE CONFIRMATION
 OF TRANSACTIONS PROTOCOL
 IN DISTRIBUTED DATABASES**

Данная статья посвящена проблеме организации распределенных запросов и транзакций в распределенных гетерогенных базах данных. Показаны некоторые сложности управления параллельным выполнением и возможные их решения. Главное внимание в статье уделено наиболее сложным механизмам обеспечения устойчивой и надежной работы баз данных в автоматизированных системах – механизму обеспечения целостности и непротиворечивости данных и механизму распределенных транзакций и распределенных запросов. Показана исключительная важность четкого управления параллельным выполнением, что позволяет избежать тупиков и противоречий в многопроцессорных операциях. Решение проблемы видится в использовании протокола двухфазного подтверждения транзакций. В статье показана его реализация в две фазы выполнения.

Ключевые слова: гетерогенные базы данных, транзакции, распределенные запросы, непротиворечивость данных, целостность данных.

This paper is devoted to the problem of distributed queries and transactions in distributed heterogeneous databases. Some of the complexity of managing the parallel execution and the possible solutions are highlighted. The main attention in article is given the most difficult mechanisms to ensure the stable and reliable operation of databases in automated systems, – guaranteeing the integrity and consistency of data and to the mechanism of distributed transactions and distributed queries. Here is shown the critical importance of clear governance parallel implementation that avoids dead-ends and contradictions in multiprocessor operations. The solution of this problem lies in the Protocol is two-phase confirmation of transactions. The article shows its implementation in two phases of execution.

Keywords: heterogeneous databases, transactions, distributed queries, data consistency.

Проблема организации распределенных запросов и транзакций в распределенных гетерогенных базах данных сложилась достаточ-

¹ Кандидат технических наук, заместитель начальника кафедры, Военно-космическая академия им. А.Ф. Можайского.

© Калиниченко С.В., 2017.

² Кандидат технических наук, преподаватель кафедры, Военно-космическая академия им. А.Ф. Можайского.

© Котиков П.Е., 2017.

³ Кандидат технических наук, старший преподаватель кафедры, Военно-космическая академия им. А.Ф. Можайского.

© Свиначук А.А., 2017.

но давно. Она тесно связана с исключительной сложностью ряда механизмов обеспечения целостности и непротиворечивости данных, а также со сложностью реализации распределенных транзакций и распределенных запросов.

Одной из наиболее ценных возможностей при применении гетерогенных распределенных баз данных является возможность выполнения распределенных запросов [1; 4]. Такую возможность выполнять распределенный запрос предоставляет только полностью распределенная база данных, поскольку здесь возможно:

1) разбить базу данных на несколько фрагментов;

2) ссылаться на один или более таких фрагментов из одного запроса, используя прозрачность фрагментации;

3) обеспечить возможность в одном запросе обращаться к физически разделенной таблице.

Центральное место занимает прозрачность транзакций. Во всех случаях так называемая прозрачность транзакций остается свойством системы управления распределенной базы данных (БД), которое гарантирует, что все транзакции БД будут обеспечивать целостность и непротиворечивость базы данных. В случае распределенной базы данных транзакции БД могут обновлять данные, хранящиеся на различных компьютерах, объединенных в сеть. Свойство прозрачности транзакций будет гарантировать, что транзакция будет завершена только в том случае, если на всех сайтах базы данных, вовлеченных в транзакцию, будут завершены все части транзакции. В системах распределенных баз данных для управления транзакциями и обеспечения целостности и непротиворечивости базы данных, как правило, используется сложный механизм распределенных транзакций и распределенных запросов.

Также сложным становится само управление параллельным выполнением в распределенной среде. Несмотря на известность этого факта, на практике он не привлекает должного внимания, вплоть до возникновения проблем при проектировании [2; 5].

Вне зависимости от того, является ли транзакция распределенной или нет, она формируется на базе одного или нескольких запросов. Основное отличие нераспределенных транзакций от распределенных состоит только в том, что последние могут обновлять или запрашивать данные на нескольких удаленных сайтах сети. Распределенная транзакция (distributed transaction) позволяет ссылаться на несколько различных локальных или удаленных сайтов. При этом каждый простой запрос может ссылаться только на один удаленный сайт, а транзакция в целом может ссылаться на несколько сайтов, поскольку каждый запрос сам по себе может ссылаться на различные сайты. Распределенный запрос (distributed request) позволяет получать данные от нескольких удаленных сайтов. Поскольку каждый запрос может получать доступ к данным, расположенным более чем на одном сайте, транзакция может получать доступ к нескольким сайтам [3].

Размещение и разбиение данных должно быть прозрачным для конечного пользователя. Свойство прозрачности транзакций гарантирует, что все транзакции можно рассматривать как централизованные, а также оно обеспечивает их

сериализуемость, т.е. одновременное выполнение транзакций. Это значит, что независимо от того, являются они распределенными или нет, база данных будет переходить из одного устойчивого состояния в другое.

Исключительно важным становится механизм управления параллельным выполнением из-за возможной противоречивости [5]. Управление параллельным выполнением становится особенно значимым в среде распределенной базы данных, поскольку многоместные (на нескольких сайтах) и многопроцессорные операции с большей вероятностью могут привести к противоречивости данных и тупикам, чем односторонние (выполняющиеся на одном сайте) системы [5]. Компонент TP (процессор транзакций) системы управления распределенной БД должен гарантировать, что все части транзакции на всех сайтах будут завершены до того, как последний оператор COMMIT завершит всю транзакцию в целом. Предположим, что каждая операция транзакции подтверждалась локальным процессором данных (DP), но один из DP не смог записать результаты транзакции. Это может привести к противоречивому состоянию БД и неизбежным проблемам с целостностью, поскольку невозможно отменить уже записанные данные.

Решение проблемы состоит в использовании протокола двухфазного подтверждения транзакции (two-phase commit protocol) [3; 5]. Централизованной базе данных необходимо только один процессор данных (DP). Все операции с базой данных проводятся на одном сайте, и последовательность операций сразу становится известной. Распределенные базы данных позволяют транзакциям осуществлять доступ к данным на нескольких сайтах. В этих случаях завершающий оператор COMMIT не должен выполняться до тех пор, пока каждый сайт не завершит свою часть транзакции. Протокол двухфазного подтверждения транзакции требует, чтобы каждая запись в журнале транзакций процессора данных выполнялась до фактического обновления фрагмента. В этом и есть его главная роль. Протокол двухфазного подтверждения транзакции требует применения протокола «выполнить – отменить – повторить» и протокола упреждающей записи. Протокол DO – UNDO – REDO используется процессором данных для отката транзакций назад (roll back) или отката транзакций вперед (roll forward) на основе записей. Чтобы гарантировать, что операции DO – UNDO – REDO смогут обеспечить корректное выполнение операций при крахе системы, используется протокол упреждающей записи. Протокол упреждающей записи (write-ahead protocol) принуждает

фиксировать в журнале запись данных для постоянного хранения перед фактическим выполнением этой операции. Протокол двухфазного подтверждения транзакции определяет операции между двумя типами узлов: узел-координатор (coordinator) и один или более подчиненных узлов-субординаторов (subordinates), или когорт (cohorts). Протокол реализуется в две фазы.

Фаза 1. Подготовка

1. Координатор посылает сообщение PREPARE TO COMMIT (подготовка к завершению) всем субординаторам.

2. Субординаторы получают сообщение, записывают информацию в журнал транзакций в соответствии с протоколом упреждающей записи и посылают координатору уведомление YES/PREPARED TO COMMIT (да/завершение подготовлено) или NO/NOT PREPARED (нет/завершение не готово).

3. Координатор убеждается, что все узлы готовы к завершению, или в противном случае отменяет действие.

Если все узлы сообщили, что они готовы к завершению (PREPARED TO COMMIT), транзакция переходит в фазу 2. Если один или более узлов отвечают, что они не готовы (NO или NOT PREPARED), координатор распространяет среди всех субординаторов сообщение ABORT (прекращение).

Фаза 2. Последний оператор COMMIT

1. Координатор оповещает всех субординаторов, рассылая сообщение COMMIT, и ожидает ответа.

2. Каждый субординатор, получив сообщение COMMIT, обновляет базу данных в соответствии с протоколом DO.

3. Субординаторы отвечают координатору сообщением COMMITED (завершено) или NOT COMMITED (не завершено).

Если один или более субординаторов не выполнили операцию завершения, координатор рассылает сообщение ABORT и тем самым инициирует операцию UNDO (отмену всех изменений).

Цель протокола двухфазного подтверждения транзакции состоит в обеспечении корректного завершения всеми узлами своих частей транзакции; в противном случае транзакция отменяется. Если один или более узлов не выполняют операцию завершения, то необходимая информация по восстановлению БД будет находиться в журнале транзакций и база данных может быть восстановлена с помощью протокола DO – UNDO – REDO.

Таким образом, реализация возможности организации распределенных запросов к рас-

пределенным базам данных связана не только с безусловным выполнением требований прозрачности, но и с необходимостью решения проблемы управления параллельным выполнением в распределенной среде.

Возможным решением является использование протокола двухфазного подтверждения транзакций.

Литература

1. Колбина О.Н. Современные и теоретические аспекты управления распределенными базами данных / Е.П. Истомин, О.Н. Колбина // Информационные технологии и системы: управление, экономика, транспорт, право : сборник научных трудов. – СПб. : ООО «Андреевский издательский дом», 2011. – Вып. 1 (9).

2. Колбина О.И. Применение распределенных баз данных в геоинформационных системах прогнозирования георисков / Е.П. Истомин, О.Н. Колбина, Е.М. Зоринова // Сборник трудов Международной научно-практической конференции «Инфогео-2013». – СПб. : ООО «Андреевский издательский дом», 2013.

3. Миков А.И., Замятина Е.Б. Распределенные системы и алгоритмы / А.И. Миков, Е.Б. Замятина. – М. : ИНТУИТ, 2008. – 287 с.

4. Разработка и развитие методов, моделей и систем геоинформационного управления пространственно-распределенными объектами : отчет о НИР / Е.П. Истомин, А.Г. Соколов, О.Н. Колбина. – СПб. : Российский государственный гидрометеорологический университет, 2013. – 102 с.

5. Роб П., Корнел К. Системы баз данных: проектирование, реализация и управление / пер. с англ. – 5-е изд., перераб. и доп. – СПб. : БХВ-Петербург, 2004. – 1040 с.

6. Котиков П.Е. Репликация данных между серверами баз данных в среде геоинформационных систем / П.Е. Котиков, А.А. Нечай // Вестник Российского нового университета. Серия «Сложные системы: модели, анализ и управление». – 2015. – Выпуск 1. – С. 90–94.

7. Лохвицкий В.А. Подходы к построению системы автоматизированной интеграции информации в базу данных для ее современной актуализации / В.А. Лохвицкий, С.В. Калинин, А.А. Нечай // Мир современной науки. – 2015. – № 3.

8. Котиков П.Е. Подход к определению пространственно-временных изменений данных / П.Е. Котиков, А.А. Нечай // Научный альманах. – 2015. – № 7 (9). – С. 688–690.