

ИССЛЕДОВАНИЕ ВЗАИМОБЛОКИРОВОК В РАСПРЕДЕЛЕННОЙ БАЗЕ ДАННЫХ, «МЕТОД ОБЕРМАКА»

Куценко Ирина Львовна,

к.ф.-м.н, доцент, Российский Университет Дружбы Народов (РУДН)
nicky2004@list.ru

Аннотация. Данная статья посвящена изучению проблем параллелизма и обнаружению взаимоблокировок в распределенной системе базы данных. В связи с этим, изучается и применяется метод выявления взаимоблокировки и условия его разрешения цикличности. Далее представлены несколько транзакций с указанием агента на каждом узле и реализуется построение графа ожидания. Решение этой задачи смоделировано с использованием метода Обермака, который позволяет разрешить проблему цикла графа ожидания, при помощи итерационного подхода для решения задачи параллельного доступа на каждом узле.

Ключевые слова: Распределенная система базы данных, транзакция, параллелизм, граф ожидания, взаимоблокировка.

THE DEADLOCKS RESEARCH IN A DISTRIBUTED DATABASE, “THE OBERMAK METHOD”

Kutsenko Irina Livovna,

Ph.D. of physico-mathematical sciences, Russian Peoples' Friendship University

Abstract. This paper is concerned with the problem of deadlocks in distributed database system concurrency. There are some methods of study and apply to identify the deadlock and conditions to permit the cycling. There are certain transactions, indicating the agent on each node of the graph and implemented to build expectations. This problem is modeled using the Obermak method, which allows to solve the problem of the graph cycle expectations, with iterative approach for solving the problem of concurrent access on each node.

Keywords: distributed database system, transaction, concurrency, Wait-For Graph – WFG, deadlock.

1. Управление параллельным доступом

Процесс организации одновременного выполнения в базе данных различных операций доступа, гарантирующий предотвращение их влияния друг на друга, является управление параллельным доступом. [1]

Чтобы обеспечить конкурентное вмешательство в базу данных, гарантируя согласованный вид для каждой транзакции, обычно используется блокировка.

Определение 1.1. Блокировка – это процедура, используемая для управления параллельным доступом к данным.[1]

Когда некоторая транзакция получает доступ к базе данных, механизм блокировки позволяет (с

целью удовлетворения получения некорректных результатов) запретить попытки получения доступа к этим же данным со стороны других транзакций. В результате получается цикличное состояние ожидания, которое называется взаимоблокировка. [1]

Система выявления взаимоблокировок базируется на поиске циклов в локальном ожидающем графе.

Определение 1.2. Граф ожидания (Wait-For Graph - WFG) – это отражение зависимости транзакций друг от друга. [1]

Точки графа используются для представления транзакций, с ориентировочными ребрами, показывающими какую транзакцию ожидает данная транзакция. Когда циклы выявлены в TWFG, они разбиваются путем выбора транзакции, которая вовлечена в цикл и заставляет транзакцию «упасть»

(обычно позволяет транзакции сделать рестарт). Эта операция становится более комплексной, когда TWFG распределен между множеством узлов распределенной базы данных.

Далее поставим задачу описать распределенный метод выявления взаимоблокировки.

Для правильной итерации метода во всех узлах и правильной передачи информации допускается, что выявление взаимоблокировок осуществляется в каждом узле после того, как выполняется связь между двумя выявителями блокировок.

Рассмотрим 5 транзакций со следующими характеристиками:

Транзакция T1 запускается на узле А и создает агента на узле В.

Транзакция T2 запускается на узле С и создает агента на узле А.

Транзакция T3 запускается на узле А и создает агента на узле С.

Транзакция T4 запускается на узле В и создает агента на узле С.

Транзакция T5 запускается на узле С.

Построим граф, который показывает траекторию движения заданных транзакций (см. Рис. 1):

В распределенной системе, где транзакция выполняется много раз, граф на рисунке 2 (Рис. 2) также становится распределенным. Разделение графа на рисунке 2 (Рис. 2) показано на рисунке 3 (Рис. 3).

Минимум каналов связи между агентами каждой транзакции показывается на рисунке 4. (Рис. 4).

К примеру транзакция 3 может располагаться в узле А, и подать запрос который приведет к перемещению в узел С. Канал связи был установлен между двумя узлами. С агентом в узле С, хранящего ответ на запрос из узла А Транзакция 3 в узле А не может продолжиться до того, как агент в узле С не даст ответ.

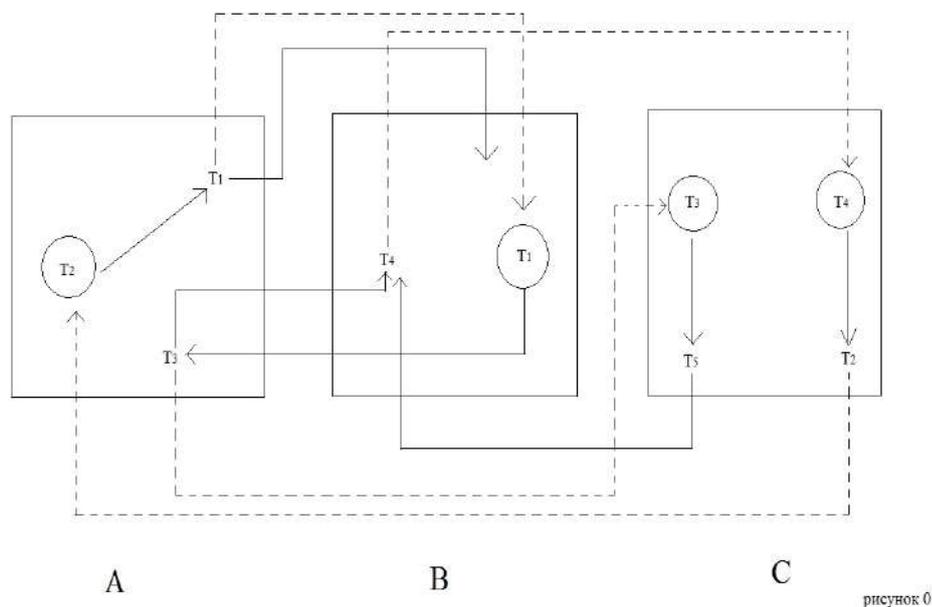


Рис. 1. Граф, показывающий движение транзакций.

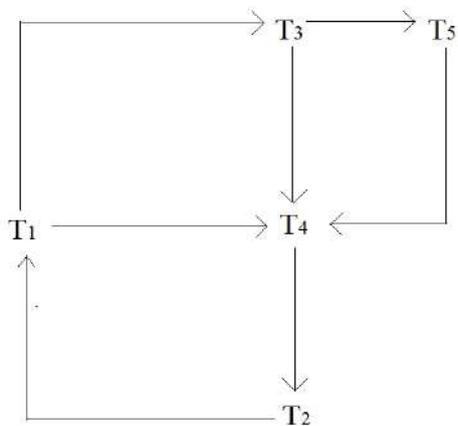
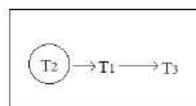
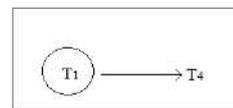


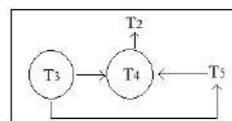
рисунок 1



А



В



С

рисунок 2

Рис. 2. Упрощенный граф ожидания.

Рис. 3. Разделение графа.

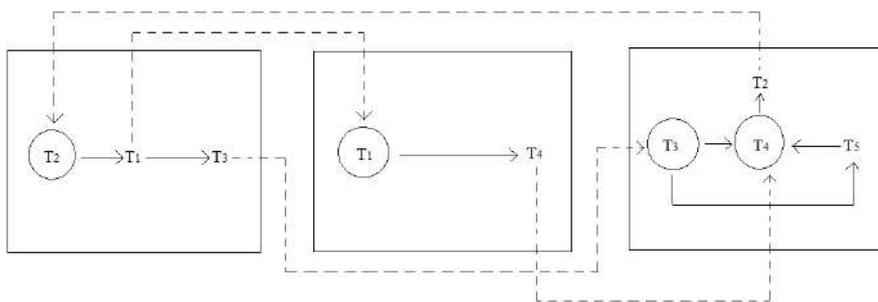


рисунок 3

Рис. 4. Минимум каналов связи между агентами каждой транзакции.

2. Процесс выявления взаимоблокировок

Для решения поставленной задачи воспользуемся методом выявления взаимоблокировок [3].

Используя рисунок 3 как пример, с начальным допущением, что выявления блокировок не производилось ранее, будет выполняться следующее приложение распределенного метода выявления взаимоблокировок. Первая итерация выявления взаимоблокировки, которая отражена на рисунке 5, будет проявляться в активности и положении в каждом узле. (Рис. 5).

Так как это первая итерация метода выявления взаимной блокировки в каждом узле нет начальных

последовательностей, которые могли бы быть добавлены к локальному ожидающему графу в каждом узле.

В узле А T(EX) показан, как ожидание для транзакции T2. Касаемо установления канала связи вместе с агентом, представляющим транзакцию T2 в этом узле. Транзакции T3 и T1 показаны как ожидающие для T(EX), потому что агенты, представляющие эти транзакции в узле А, ожидают получения сообщения.

Два цикла будут выявлены в узле А, и они показаны в колонне «ЦИКЛЫ». Оба цикла содержат специальную точку графа T(EX), и пронумерованы в порядке транзакция-ожидание-транзакция. Циклы, пронумерованные в этой колонне, всегда будут показывать как начало, так и окончание транзакции.

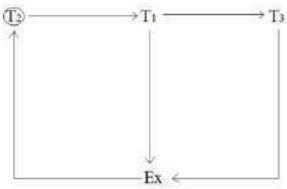
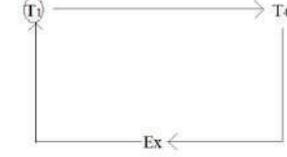
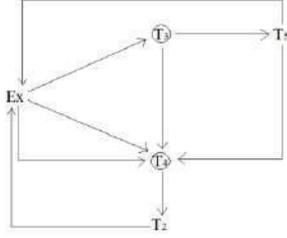
№ узла	Послед. откуда	Локальный граф ожидания	Циклы	Послед. куда
A	-		<p>Ex, T2, T1, Ex</p> <p>Ex, T2, T1, T3, Ex</p>	<p>Ex, T2, T1</p> <p>в B</p>
B	-		<p>Ex, T1, T4, Ex</p>	-
C	-		<p>Ex, T3, T5, Ex</p> <p>Ex, T3, T5, T4, T2, Ex</p> <p>Ex, T4, T2, Ex</p> <p>Ex, T3, T4, T2, Ex</p>	<p>Ex, T3, T5, T4, T2</p> <p>Ex, T4, T2 в A</p> <p>Ex, T3, T4, T2</p>

Рис. 5. Первая итерация выявления взаимоблокировки.

Так как лексическое упорядочение транзакции ожидающей со стороны T(EX) меньше, чем транзакции, которая ожидает T(EX) в узле В, то никакой Последовательности не будет отправлено из узла В.

Оба и узел А и узел С имеют потенциальные циклы, но по причине лексического упорядочения транзакций вне циклов, они будут отправлять Последовательности. Узел А будет отправлять Последовательность T(EX),T2,T1 узлу В, потому что транзакция, ожидающая T(EX), ожидает получения сообщения из узла В.

Хотя узел С содержит четыре потенциальных

цикла, только Последовательности EX,T3,T5,T4,T2; EX,T4,T2; EX,T3,T4,T2; будет послана узлу А.

Аналогичным образом должны быть проведены следующие итерации. Последняя итерация распределенного метода выявления отвзаимоблокировки показаны на рисунке 6 (Рис. 6).

С последней итерацией, все циклы взаимоблокировок разрушены. И окончательное состояние, в котором нет дальнейших выявлений взаимоблокировок, имеет место. Каждые следующие итерации (давая допущение, что оставшиеся транзакции не вызовут конфликтов) не показывают потенциальные циклы взаимоблокировок.

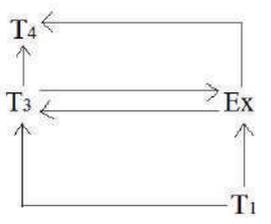
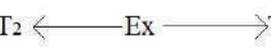
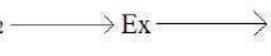
№ узла	Послед. откуда	Локальный граф ожидания	Циклы	Послед. куда
A	Ex, T3, T4, T2 * Ex, T4, T2 * (Из С)		(Ex, T3, Ex)	—
B	Victim= T1 (Из С)		—	—
C	Victim= T4 (из В)		—	—

Рис. 6. Последняя итерация распределенного метода выявления взаимных блокировок.

Список литературы

1. Томас Коннолли «Базы Данных»
2. К. Дж. Дейт «Введение в системы баз данных»
3. Рон Обермак «Распределенный метод выявления взаимоблокировок»
4. Мартин Грабер «SQL»