

Управление ресурсами и блокированием в централизованных базах данных

Ленкин Алексей Викторович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

Целью исследования является изучение проблем блокировок ресурсов при исполнении транзакций в базах данных и пути реализации максимально возможного уровня параллельной работы. Методами исследования является поиск и анализ русскоязычных и англоязычных источников по теме исследования, изучение документации MS SQL. Итогом исследования является описание тестового стенда для реализации ручных блокировок.

Ключевые слова: MS SQL, блокировки, транзакции, параллелизм, централизованные базы данных

Resource and block management in centralized databases

Lenkin Aleksei Viktorovich

Sholom-Aleichem Priamursky State University

Student

Abstract

The aim of the study is to study the problems of resource blocking when executing transactions in databases and ways to implement the highest possible level of parallel work. The research methods are the search and analysis of Russian-language and English-language sources on the research topic, the study of MS SQL documentation. The result of the study is a description of the test bench for the implementation of manual locks.

Keywords: MS SQL, locks, transactions, concurrency, centralized databases

При работе с базами данных для обеспечений целостности информации, выполняемые команды для взаимодействия с ней, как правило, требуется организовывать в транзакции. Такой подход обеспечивает лучшую параллельность и позволяет пользователю откатывать изменения сделанные одной или несколькими транзакциями.

Обычно при работе транзакции на затронутые последней ресурсы базы данных накладываются так называемые блокировки. Они осуществляются с помощью автоматических алгоритмов и механизмов. Такой подход обеспечивает целостность исполнения транзакции и некоторую параллельность при работе, но в больших и/или сложных структурах баз данных такой подход лишь замедлит работу с ними. Решением этой

проблемы может быть использование собственного метода блокировок, подходящего под конкретную ситуацию или структуру, определенного уровня изоляции транзакций.

Исследованиями в данной теме занимались следующие авторы. Ю.Г. Пулатов [1] провел сравнение методов оптимистической и пессимистической блокировки при параллельном доступе к ресурсу базы данных. Мониторинг блокировок в Oracle был продемонстрирован В. Михеичевым [2]. Г.А. Черноморов [3] получил рабочую модель функционирования корпоративной информационной системы централизованной архитектуры с учетом блокировок транзакций. В. Михеичев [4] демонстрирует взаимоблокировки Deadlock-сессий в Oracle. Причины блокировок в системе 1С: Предприятие и методы борьбы с ними тщательно рассмотрены К.В. Мазеиным, А.С. Поняшовой, П.П. Безносовым, А.А. Козырь, С.В. Храмовым [5].

Для исследования были выбраны базы данных MS SQL, так как они обладают огромным функционалом для работы с транзакциями и имеют встроенные механизмы управления блокировками.

В базах данных MS SQL поддерживаются и работают с общими типами блокировок.

Режим блокировки учитывает различные типы блокировок, которые могут быть применены к ресурсу, который должен быть заблокирован [5]:

- Исключительная (X);
- Разделяемая (S);
- Обновления (U);
- Намерения (I);
- Схемы (Sch);
- Групповое обновление (BU).

Для ручного управления блокировками в MS SQL существует механизм уровня изоляции, который присваивается нужной транзакции и выдает при её работе соответствующие блокировки и метод табличных указаний. Начнём с уровня изоляций.

Уровни изоляции указаны в таблице 1 [6].

Таблица 1. Уровни изоляций транзакций в базах данных MS SQL.

Уровень изоляции	Грязное чтение	Невоспроизводимое чтение	Фантомные данные	Параллелизм (несколько пользователей)
ReadUncommitted	Да	Да	Да	Наилучший
ReadCommitted	Нет	Да	Да	Хороший
Snapshot	Нет	Нет	Нет	Хороший
RepeatableRead	Нет	Нет	Да	Слабый
Serializable	Нет	Нет	Нет	Очень слабый

Так как обычно транзакции выполняются моментально, и отследить в это время вручную какие ресурсы были заблокированы и как именно,

существуют автоматизированные встроенные программные средства SQL Server Profiler [7].

Данное программное обеспечение позволяет создать трассировку и визуально отслеживать изменения в данных БД (рис. 1).

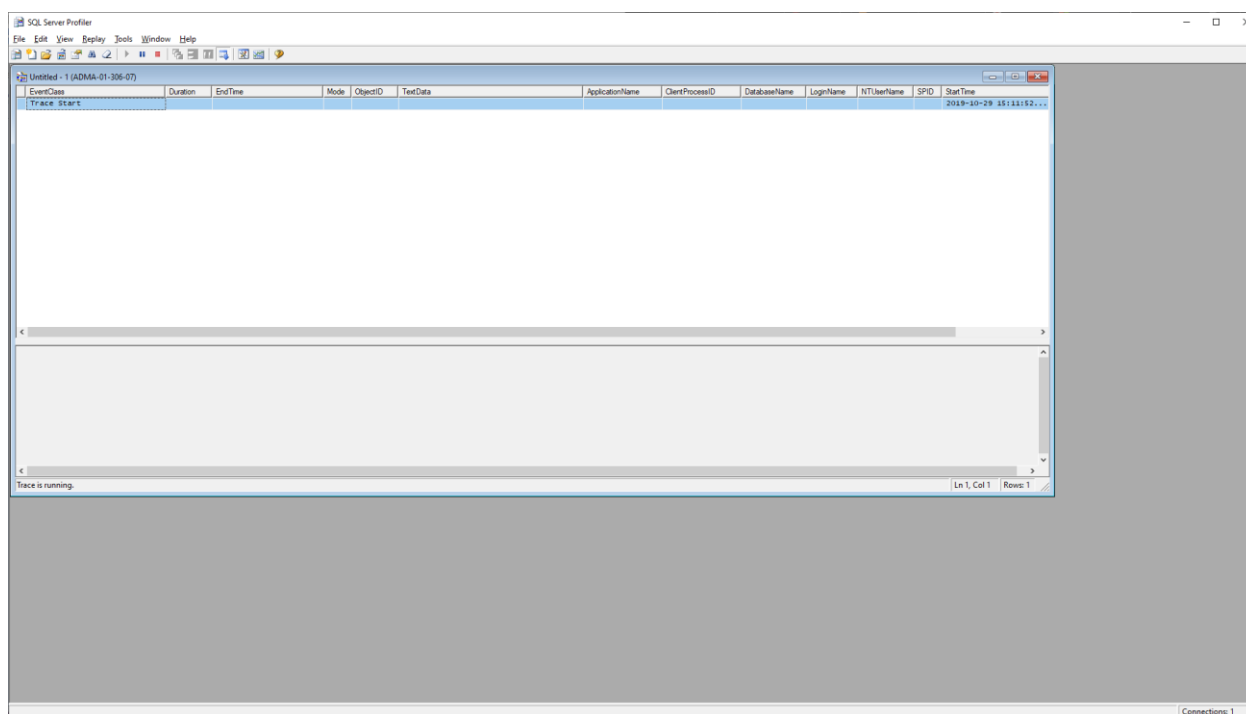


Рисунок 1. Главное окно программы SQL Server Profiler в режиме трассировки

При правильной настройке фильтров можно видеть каждую наложенную на данные блокировку и отследить цепочку блокировок.

Использование в данном исследовании этого программного обеспечения позволит точно сказать, изменяются ли при разном уровне изоляции транзакции блокировки, а также отмечать возможные ошибки, которые могут возникнуть в процессе выполнения транзакций.

Был создан экспериментальный прототип программы для проверки работы уровня изоляций транзакций в C# и отслеживания блокировок с помощью SQL Server Profiler. Листинг программы приведен ниже:

```
#region Using directives

using System;
using System.Collections.Generic;
using System.Text;
using System.Data;
using System.Data.SqlClient;

#endregion
namespace ConsoleApplication3
{
```

```
class Program
{
    static void Main(string[] args)
    {
        SqlConnection con = new
SqlConnection("Server=localhost;Database=University;Integrated
Security=True;");

        SqlCommand cmd1 = new SqlCommand(
"INSERT INTO dbo.Kurs (Name_krs) VALUES ('8-й курс')",
con);

        SqlTransaction transaction1 = null;

        try
        {
            con.Open();
            transaction1 =
con.BeginTransaction(IsolationLevel.ReadUncommitted);
            cmd1.Transaction = transaction1;
            cmd1.ExecuteNonQuery();
            transaction1.Commit();
        }
        catch
        {
            transaction1.Rollback();
        }
        finally
        {
            con.Close();
        }
    }
}
```

Алгоритм выполнения программы следующий:

1. Происходит соединение с базой данных.
2. Создается переменная, содержащая любую SQL команду.
3. Создается транзакция.
4. Устанавливается уровень изоляции транзакции.
5. Транзакция запускается на выполнение и в случае возникновения ошибок отменяется.

После запуска транзакции «INSERT INTO dbo.Kurs (Name_krs) VALUES ('8-й курс')» с уровнем изоляции «IsolationLevel.ReadUncommitted» был получен результат на рис. 2. Где видны как разделяемые блокировки, так и исключительные

Lock:Acquired	2 - DATABASE	3 - S
Lock:Acquired	5 - OBJECT	8 - IX
Lock:Acquired	6 - PAGE	8 - IX
Lock:Acquired	7 - KEY	15 - RangeI-N
Lock:Acquired	7 - KEY	5 - X
Lock:Released	7 - KEY	5 - X
Lock:Released	6 - PAGE	8 - IX
Lock:Released	5 - OBJECT	8 - IX
Lock:Released	2 - DATABASE	3 - S

Рисунок 2. Транзакция «INSERT INTO dbo.Kurs (Name_krs) VALUES ('8-й курс')» с уровнем изоляции «IsolationLevel.ReadUncommitted»

Но так как уровень изоляции был на уровне, который предполагает отсутствие любых блокировок можно сделать следующие выводы:

1. Уровни изоляций транзакций в базах данных MS SQL не работают в полной мере
2. Автоматические блокировки не учитывают описания уровня изоляции при работе с транзакцией

Следовательно следующим решением для ручного управления блокировками можно считать «Табличные указания»

Табличные указания (табличные подсказки) – переопределяют поведение оптимизатора запросов по умолчанию на время выполнения инструкции языка обработки данных (DML). Для этого указываются способ блокировки, один или более индексов, операция обработки запроса, например сканирования таблицы или поиска в индексе, или другие параметры. Табличные указания задаются в предложении FROM инструкции DML и относятся только к таблицам и представлениям, на которые ссылается это предложение.

Табличные указания задаются к запросу с помощью конструкции WITH (<table_hint>), где <table_hint> – одно из указаний выбранного уровня блокировки, опишем самые важные из них:

1. PAGLOCK – блокировка страницы, вместо блокировки строки или индексов по умолчанию.
2. ROWLOCK – блокировка строк, вместо блокировки страниц или таблиц.
3. TABLOCK – блокировка всей таблицы.
4. NOLOCK – отсутствие любых блокировок.

Попробуем использовать ту же SQL команду, но уже без создания транзакции, используем блокировку страницы, она будет выглядеть следующим образом:

```
INSERT INTO dbo.Kurs WITH (PAGLOCK) (Name_krs) VALUES ('8-й курс')
```

Результат работы команды можно увидеть в SQL Server Profiler (рис. 3).

Lock:Acquired	2 - DATABASE	3 - S
Lock:Released	2 - DATABASE	3 - S
Lock:Acquired	2 - DATABASE	3 - S
Lock:Released	2 - DATABASE	3 - S
Lock:Acquired	5 - OBJECT	8 - IX
Lock:Acquired	6 - PAGE	5 - X
Lock:Released	6 - PAGE	5 - X
Lock:Released	5 - OBJECT	8 - IX

Рисунок 3. Результат выполнения команды на вставку с табличным указанием PAGLOCK

Из выполнения этой команды видно, что блокировка вручную выполняется успешно и, даже если такой уровень изоляции не поддерживается базами данных MS SQL, табличные указания всё равно дают больший функционал над созданием механизма ручных блокировок

Таким образом, была получена стартовая программа для тестирования ручных блокировок и отслеживания базовых механизмов блокировок. Далее тестирование системы должно проводиться с несколькими машинами и сервером, чтобы проверить, как табличные указания будут взаимодействовать с конкурирующими транзакциями.

Также надо будет сделать предположение, какая блокировка нужна под конкретный тип данных и вид запроса, а после сравнить скорость и стабильность работы при ручных и автоматических блокировках

Библиографический список

1. Пулатов Ю.Г. Сравнение методов оптимистической и пессимистической блокировки при параллельном доступе к ресурсу базы данных // Решетневские чтения. 2018. Т. 2. С. 289-290.
2. Михеичев В. Мониторинг блокировок в Oracle. Методы предупреждения и автоматического устранения // Системный администратор. 2015. № 4 (149). С. 30-34.
3. Черноморов Г.А. Модель функционирования корпоративной информационной системы централизованной архитектуры с учетом блокировок транзакций // Известия высших учебных заведений. Северо-Кавказский регион. Технические науки. 2003. № 2. С. 33-40.
4. Михеичев В. Взаимоблокировки Deadlock-сессий в Oracle // Системный администратор. 2016. № 3 (160). С. 38-42.
5. Мазеин К.В., Поняшова А.С., Безносков П.П., Козырь А.А., Храмов С.В. Причины блокировок в системе 1С: Предприятие и методы борьбы с ними // Аллея науки. 2017. Т. 1. № 12. С. 247-254.
6. All about locking in SQL Server // SQLShack URL: <https://www.sqlshack.com/locking-sql-server/> (дата обращения: 24.06.20).
7. Уровни изоляции // Professor Web URL: <https://professorweb.ru/my/sql->

- server/2012/level3/3_16.php (дата обращения: 24.06.20).
8. Приложение SQL Server Profiler // Microsoft. Документация по SQL URL: <https://docs.microsoft.com/ru-ru/sql/tools/sql-server-profiler/sql-server-profiler?view=sql-server-ver15> (дата обращения: 24.06.20).
 9. Табличные указания (Transact-SQL) // Microsoft. Документация по SQL URL: <https://docs.microsoft.com/ru-ru/sql/t-sql/queries/hints-transact-sql-table?view=sql-server-ver15> (дата обращения: 26.03.20).