

Боротьба з Очікуваннями у Microsoft SQL Server

Олександр Тунік
кафедра програмної інженерії
Харківський національний університет
радіоелектроніки
Харків, Україна

Олександр Саманцов
кафедра програмної інженерії
Харківський національний університет
радіоелектроніки
Харків, Україна
oleksandr.samantsov@nure.ua

Wait Time Prevention in Microsoft SQL Server

Oleksandr Tunik
Department of Software Engineering
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine

Oleksandr Samantsov
Department of Software Engineering
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
oleksandr.samantsov@nure.ua

Анотація—У роботі висвітлюється питання очікувань та блокування у Microsoft SQL Server. Розглянуто джерела очікувань та надано рекомендації щодо зменшення часу, що витрачається на них

Abstract— The paper describes problem of waits and blocks in Microsoft SQL Server. The sources of waits are considered and recommendations for reducing the time spent on them are given

Ключові слова—Microsoft Sql Server, база даних, очікування, блокування

Keywords— Microsoft Sql Server database, waits, blockings

I. ВСТУП

При створенні баз даних до них виставляється ряд вимог. Серед них можна визначити наступні:

- Вимоги цілісності - дані мають бути повними та такими, що не протирічать один-одному.
- Багаторазове використання даних.
- Швидкий пошук та отримання інформації по запитах користувача.
- Простота оновлення даних.
- Зменшення надмірності даних.
- Захист даних від несанкціонованого доступу та їх знищення [1].

Ці вимоги є комплексними, та вимагають ексклюзивного доступу до даних. Саме тому ми не можемо позбутися очікувань при розробки баз даних.

Метою роботи є висвітлення очікувань у базі даних Microsoft SQL Server та розробка загальних методів боротьби з ними.

II. MICROSOFT SQL SERVER

Microsoft SQL Server — комерційна система керування базами даних, що розповсюджується корпорацією Microsoft. Мова, що використовується для запитів - Transact-SQL, створена спільно Microsoft та Sybase. Transact-SQL є реалізацією стандарту ANSI / ISO щодо структурованої мови запитів SQL із розширеннями. Використовується як для невеликих і середніх за розміром баз даних, так і для великих баз даних масштабу підприємства [2].

SQL Server надає засоби для створення та управління реляційними базами даних. Також SQL Server 2012 підтримує засоби аналітичної обробки даних (Analysis Services), засоби звітності (Reporting Services), а також безліч засобів, що спрощують розробку додатків.

З огляду на те, що на Microsoft SQL Server працюють критичні бізнес-додатки, підприємства висувають дуже жорсткі вимоги по продуктивності, відмовостійкості та безпеки самої СКБД. Серед вимог ми можемо виокремити наступні:

- Інтерактивне шифрування баз даних. Microsoft SQL Server підтримує прозоре шифрування баз даних і журналів транзакцій.
- Віддзеркалення БД (Database Mirroring). Технологія зеркалювання передбачає наявність одного активного сервера, з яким працюють всі клієнти, і одного дзеркального сервера, на який відбувається



передача всіх журналів транзакцій і їх застосування до дзеркальної БД. При відмові активного сервера, відбувається перемикання на дзеркальний сервер.

- Розподіл ресурсів (Resource Governor). Microsoft SQL Server дозволяє розподіляти ресурси і призначати пріоритет для різних завдань. Ця функція дозволяє уникнути випадків різкого падіння продуктивності всього сервера, і переривання роботи всіх користувачів при виконанні однієї складної задачі [2].

Таким чином, ми можемо зробити висновок, що Microsoft SQL Server є надійним інструментом, що повністю відповідає вимогам бізнесу, та використовується у бізнес-додатках.

III. ДЖЕРЕЛА БЛОКУВАНЬ У MICROSOFT SQL SERVER

Компонент Microsoft SQL Server - Database Engine використовує такі механізми для гарантії цілісності транзакцій і підтримки узгодженості баз даних, коли кілька користувачів звертаються до одних і тих же даних в один і той же час.

- Блокування- Кожна транзакція запитує блокування різних типів ресурсів, наприклад рядків, сторінок або таблиць, від яких ця транзакція залежить. Блокування не дає іншим транзакціям змінювати ресурси, щоб уникнути помилок в транзакції, що запросила блокування. Кожна транзакція звільняє свої блокування, якщо більше не залежить від блокування ресурсу.
- Управління версіями рядків - Якщо використовується рівень ізоляції на основі управління версіями, компонент Database Engine зберігає версії кожної зміненої рядки. Додатки можуть вказати, що транзакція буде використовувати версії рядків для перегляду даних, що існували до її початку або до початку запиту, замість того, щоб захищати всі операції читання блокуванням. При управлінні версіями рядків ймовірність того, що операція читання блокуватиме інші транзакції, значно знижується [3].

Блокування та управління версіями рядків не дають користувачам зчитувати незафіксовані дані і не дають декільком користувачам змінювати одні й ті ж дані в один і той же час. Без цих механізмів запити до таких даних могли б повертати непередбачені результати, наприклад дані, які ще не були зафіксовані в базі даних. Таким чином може бути гарантовано, що усі дані, що зчитують користувачі є повними та не суперечать самі собі.

Додатки можуть вибирати рівні ізоляції транзакцій, які визначають рівень захисту транзакції від змін, внесених іншими транзакціями. Для окремих інструкцій Transact-SQL можуть бути задані вказівки таблиць для подальшого уточнення поведінки, що відповідає вимогам програми.

IV. ОСНОВНІ ТИПИ ОЧІКУВАНЬ У MICROSOFT SQL SERVER

Можна виділити наступні найбільш популярні види очікувань у Microsoft SQL Server

- **SXPACKET** - Коли паралельна операція створюється для запиту, з'являється кілька потоків для одного запиту. Кожен запит має справу з різним набором даних. Через низку причин один або кілька потоків відстають створюючи очікування SXPACKET. Є потік організатор / координатор (потік 0), який чекає виконання всіх інших потоків і являє результат виконання на стороні клієнта. Тільки після того, як всі потоки завершують своє виконання і передадуть всі дані потоку 0, запит може вважатися завершеним. Очікування організатором інших потоків і є SXPACKET очікування. Ми можемо помітити, що очікування SXPACKET не скасується негативно на час роботи запиту, проте під час його підвищується навантаження на сервер. Також причинами виникання цього виду очікування може бути розміщення даних на дисках з різною швидкістю читання, велика фрагментація даних та помилки при оцінюванні об'єму даних, що має опрацювати Microsoft SQL Server.
- **SOS_SCHEDULER_YIELD** - виникає, коли планувальник добровільно віддає час для виконання іншому запиту. Як тільки вивільниться квант часу - запит поновить виконання. Виникає найчастіше у системах з невеликою, у порівнянні з кількістю одночасних запитів, процесорів.
- **PAGEIOLATCH** - виникає, коли задача очікує даних від жорсткого диска. Довгий час очікування під час роботи запиту може означати на проблему з дисковою системою. Також, може виникати при конкуренції при доступі до файлів транзакцій та тимчасових файлів. Цей тип очікування поділяється на наступні підтипи: PAGEIOLATCH_DT – блокування, що очікує на видалення. PAGEIOLATCH_EX – блокування, що очікує на ексклюзивний доступ. PAGEIOLATCH_KP – блокування, що очікує на режим утримання. PAGEIOLATCH_NL – використовується лише для ознайомлення. Не підтримується у сучасних версіях Microsoft SQL Server. PAGEIOLATCH_SH – блокування, що очікує на загальний доступ. PAGEIOLATCH_UP – блокування, що очікує на доступ у режимі оновлення.
- **PAGELATCH** - виникає, коли задача очікує даних від оперативної пам'яті. Скоріш за все ця сторінка пам'яті досить часто використовується Microsoft SQL Server, також велика кількість цих блокувань, може вказувати на конкуренцію. при доступі до тимчасових файлів. Очікування поділяється на наступні підтипи: PAGELATCH_DT – очікування на блокування у режимі видалення. PAGELATCH_EX – очікування на блокування у ексклюзивному режимі. PAGELATCH_KP – очікування на блокування у режимі утримання. PAGELATCH_SH – очікування на блокування у режимі загального доступу. PAGELATCH_UP –



очікування на блокування у режимі оновлення даних.

- RESOURCE_SEMAPHORE – очікування на отримання кванту пам'яті для виконання операції. Виникає коли об'єм пам'яті є недостатнім для виконання усіх паралельних запитів, або при помилці планувальника запитів при використанні сортування, хеш-операцій тощо.
- PREEMPTIVE_OS – очікування на відповідь від операційної системи, що працює у режимі витісняючої багатозадачності. Найпопулярніші підтипи цього очікування -
 PREEMPTIVE_OS_WRITEFILE – очікування на Windows-функцію WriteFile, що записує дані у пристрій вводу-виводу.
 PREEMPTIVE_OS_LOOKUPACCOUNTSID – очікування на Windows-функцію LookupAccountSid, що отримує ім'я облікового запису, що створив об'єкт безпеки.
 PREEMPTIVE_OS_CRYPTACQUIRECONTEXT - очікування на Windows-функцію CryptAcquireContext, що отримує конкретний криптографічний контейнер з конкретним криптографічним сервісом.
 PREEMPTIVE_OS_CRYPTIMPORTKEY - очікування на Windows-функцію CryptImportKey, що імпортує криптографічний ключ з бінарного контейнеру до криптографічного сервісу.
 PREEMPTIVE_OS_DISCONNECTNAMEDPIPE - очікування на Windows-функцію DisconnectNamedPipe, що від'єднує серверну частину іменованого каналу від клієнта.
 PREEMPTIVE_OS_DELETEFILE - очікування на Windows-функцію DeleteFile, що видаляє вказаний файл.
- WRITELOG - має місце при очікуванні завершенні запису журналу транзакцій. Цей вид очікувань зазвичай спостерігається в високо транзакційних базах даних. Коли дані змінюються вони пишуться в кеш лог і кеш буферу. Це очікування виникає, коли лог кеш скидається на диск [4].

V. БОРОТЬБА З ОЧІКУВАННЯМИ

Проаналізувавши найбільш популярні види очікувань ми можемо запропонувати наступні методи боротьби:

- Зменшення рівня паралелізації запитів. Таким чином ми зможемо зменшити навантаження на сервер та час CXPACKET очікувань.
- Створення належних індексів, для прискорення доступу до даних та зменшення доступу до диску.
- Розміщення тимчасових файлів, файлів транзакцій, та файлів з даними на різних фізичних дисках, таким чином не буде виникати боротьба за читання даних.

- Постійне оновлення статистики таблиць та індексів, таким чином планувальник буде знати на який об'єм даних потрібно очікувати.
- Регулярна дефрагментація баз даних, таблиць, індексів для прискорення доступу до даних та зменшення операцій вводу-виводу.
- Постійний контроль за фізичними можливостями серверу. Сервер має мати достатню кількість процесорів для обробки даних, об'єм оперативної пам'яті для розміщення даних, та досить швидкі жорсткі диски.

Використовуючи ці методи можливо зменшити час очікування при роботі з Microsoft SQL Server, та пришвидшити доступ до даних.

VI. ПРАКТИЧНІ РЕЗУЛЬТАТИ БОРОТЬБИ З ОЧІКУВАННЯМИ

Для аналізу успішності боротьби з очікуваннями, проводився ряд вимірювань за допомогою скрипта, що зображений на Рис. 1.

```

WITH [Waits] AS
(
    SELECT
        [wait_type] AS [wait_type],
        [wait_time_ms] / 1000.0 AS [Wait],
        ([wait_time_ms] - [signal_wait_time_ms]) / 1000.0 AS [Resource],
        ([signal_wait_time_ms] / 1000.0) AS [Signal],
        [blocking_session_id] AS [Blocking],
        100.0 * ([wait_time_ms] / SUM([wait_time_ms]) OVER ([Partition] BY [wait_type])) AS [Percentage],
        COUNT(*) OVER ([Partition] BY [wait_type]) AS [Count]
    FROM sys.dm_os_wait_stats
    WHERE [waiting_tasks_count] > 0
)
SELECT
    [W1].[wait_type] AS [WaitType],
    CAST(SUM([W1].[Wait]) AS DECIMAL(10,2)) AS [Wait_S],
    SUM([W1].[Count]) AS [WaitCount],
    CAST(SUM([W1].[Percentage]) AS DECIMAL(10,2)) AS [Percentage],
    COUNT(*) AS [W]
FROM [Waits] AS [W]
GROUP BY [W].[wait_type]
ORDER BY [W].[Count] DESC

```

Рис. 4. Скрипт для збору інформації щодо очікувань

Таким чином, ми можемо виміряти дві важливі характеристики: кількість очікувань, що відбулось з моменту останнього оновлення статистики та який сумарний час запит чекав на вивільнення ресурсу. Вимірювання до та після внесених запропонованих змін проводилось після тижня однотипної роботи з базою. Внесення змін вимагало перезавантаження сервісу MS Sql Server, що скидає статистичні дані. Отримані дані можна побачити у таблиці 1.

ТАБЛИЦЯ II. Вплив внесених змін

Тип очікування	До внесення змін		Після внесення змін	
	Час очікування (с)	Кількість очікувань	Час очікування (с)	Кількість очікувань
CXPACKET	678594,16	205652236	685380,1	195369624
LATCH_EX	65827,06	39073941	50028,56	29696195
RESOURCE_SEMAPHORE	40597,26	1612	39785,31	1547



SOS_SCHEDULER_YIELD	31175,85	30575657	30240,57	30269900
PAGEIOLATCH_SH	24740,29	11833540	14844,17	8165142
PAGEIOLATCH_EX	8561,24	3945839	5564,8	2525336
PAGELATCH_UP	1955,06	1106758	1368,54	675122
WRITELOG	942,63	300792	603,28	192506
PAGELATCH_EX	821,71	6361920	534,11	4453344
PAGELATCH_SH	807,7	2724135	508,85	1770687
PAGEIOLATCH_UP	399,72	12532	279,8	8521
PREEMPTIVE_OS_WRITEFILE	290,71	113644	183,14	69322
LATCH_SH	166,75	7141	133,4	5427
PREEMPTIVE_OS_CRYPTACQUIRE_CONTEXT	17,42	35416	16,89	36478
PREEMPTIVE_OS_CRYPTIMPORTKEY	12,19	35409	11,58	34346
LATCH_UP	1,65	2	1,27	1

З наведеної таблиці можна побачити, що внесені зміни позитивно вплинули на зменшення часу очікувань, які пов'язані з роботою жорсткого диска. Однак, час очікувань, які пов'язані з роботою пам'яті та процесору знизився не настільки значно через апаратні обмеження системи.

VII. ВИСНОВКИ

Проаналізовані джерела блокувань та основні типи очікувань у Microsoft SQL Server.

Розглянуті методи зменшення часу очікування при роботі з Microsoft SQL Server, які пришвидшують доступ до даних. Наведені дані про результати використання цих методів свідчать про ефективність їх застосування для характерних видів очікувань, пов'язані з роботою жорсткого диска.

ЛІТЕРАТУРА REFERENCES

- [1] К. Дж Дейт., Введение в системы баз данных. М.: Издательский дом "Вильямс", 2005.
- [2] D. Petkovic Microsoft SQL Server 2016: A Beginner's Guide. Rosenheim: McGraw-Hill Education, 2016
- [3] M. Chmel, V. Muzny SQL Server 2017 Administrators Guide. Birmingham: Packt Publishing Ltd, 2017
- [4] W. Assaf, R. West, S. Aelterman, M. Cumutt SQL Server 2017 Administration Inside Out. Houston: Microsoft Press, 2017.

