Методики разработки, ориентированные на повышение параллельности

Докладчик, должность

Цели доклада

- Тема блокировок и борьбы с ними неоднократно поднималась в методических материалах, статьях на ИТС и докладах на мероприятиях
- Однако эти материалы разбросаны по различным источникам и не систематизированы
- Цели доклада:
  - Собрать всю информацию по методикам разработки многопользовательских приложений в одном месте
  - Обобщить знания по теме
- Рассматриваем только клиент-серверный вариант
- Тема настолько обширна, что в одном докладе невозможно ее подробно осветить
- Методики находятся в стадии разработки

Работа в многопользовательской среде

- **Ресурс – совокупность взаимосвязанных данных**
- **Нельзя параллельно модифицировать один и тот же ресурс, поскольку это приведет к получению непредсказуемого и/или неверного результата:**
  - **Практически ни один ресурс не является атомарным, то есть неразделимым на составные части.** Например, запись регистра может содержать несколько полей, иметь соответствующие индексы и т.д.
  - **Ни одна операция не может быть выполнена за нулевое время**
  - **Всегда существует вероятность того, что две параллельные сессии будут одновременно модифицировать один и тот же ресурс**
- **При этом могут возникнуть следующие ситуации:**
  - **В зависимости от того, в какой момент времени пересеклись два конкурирующих процесса, может быть получено разное итоговое состояние данных, то есть данные становятся непредсказуемыми**
  - **Некоторые составные части ресурса могут оказаться утерянными, то есть данные потеряют свою целостность**
  - **Составные части ресурса могут придти в несогласованное состояние, то есть данные окажутся противоречивыми**

Работа в многопользовательской среде

- **Многопользовательская среда - это компромисс:**
  - **Между требованиями предсказуемости, целостности и непротиворечивости данных**
  - **И требованием параллельности работы**
- **Иначе говоря, параллельность возможна не всегда:**
  - **Операции могут выполняться параллельно если они не модифицируют один и тот же ресурс**
  - **Модификация одного ресурса может выполняться только последовательно**
- **Блокировки – механизм обеспечения последовательности операций разных пользователей над общими данными.** С помощью блокировок пользователи «выстраиваются в очередь» к общему ресурсу, то есть параллельная работа временно превращается в последовательную

Основные сведения о блокировках

- **Пример блокировки:**
  - **Пользователь А посылает команду на изменение некоторого ресурса**
  - **Система автоматически блокирует ресурс до начала выполнения команды**
  - **Действие по изменению ресурса длится конечное, но ненулевое время. Все это время ресурс остается заблокирован**
  - **В это время пользователь Б пытается изменить тот же самый ресурс**
  - **Ресурс заблокирован пользователем А, поэтому пользователь Б ставится в очередь, то есть ожидает снятия блокировки, установленной пользователем А**
  - **Процесс модификации ресурса, запущенный пользователем А завершается**
  - **Система автоматически снимает блокировку с модифицированного ресурса**
  - **Система устанавливает блокировку от имени пользователя Б и разрешает ему начать операцию над ресурсом**

Основные сведения о блокировках

- **Блокируются только разные сессии. Сессия никогда не блокирует сама себя**
- **Соответствующая блокировка устанавливается системой автоматически при любом обращении к данным**
- **Срок жизни блокировки:**
  - **Блокировка живет до конца транзакции, в которой она установлена**
- **Транзакция:**
  - **Явная транзакция.** Создается при помощи метода глобального контекста НачатьТранзакцию()
  - **Скрытая транзакция.** Любые операции модификации данных выполняются в транзакции. Транзакция автоматически открывается перед началом операции и автоматически завершается по ее окончании
- **Блокировка отрицательно влияет на показатели производительности и масштабируемости системы:**
  - **Увеличивается время отклика**
  - **Уменьшается пропускная способность**
  - **Возможны сообщения о превышении времени ожидания блокировки (lock request time out)**
  - **Возможны конфликты взаимных блокировок (deadlock)**
- **По отношению к бизнес-логике приложения блокировки можно разделить на две категории:**
  - **Необходимые. Блокировки, которые обеспечивают предсказуемость, целостность и непротиворечивость данных**
  - **Избыточные. Ненужные с точки зрения бизнес-логики приложения**
- **Технически блокировки (необходимые и избыточные) ничем не отличаются**
  - **Любая блокировка оказывает отрицательное влияние на параллельность**
  - **Необходимая блокировка нужна для решения задач, стоящих перед приложением**
  - **Отрицательное влияние избыточной блокировки ничем не оправдано**

Стратегия разработки многопользовательского приложения

- По отношению к необходимым блокировкам:
  - Проектировать систему таким образом, чтобы минимизировать количество общих ресурсов, которые могут быть заблокированы при параллельной работе
  - Реализовать систему таким образом, чтобы обеспечить наличие всех необходимых блокировок:
    - *Использование транзакций*
    - *Использование опции ДЛЯ ИЗМЕНЕНИЯ*
  - Минимизировать влияние необходимых блокировок на производительность системы, то есть уменьшать время блокировки
  - Исключить возможность возникновения конфликта взаимных блокировок (deadlock)
- По отношению к избыточным блокировкам:
  - При разработке системы следовать рекомендациям, позволяющим избежать избыточных блокировок
  - Проводить тестирование системы в многопользовательском режиме с целью поиска и устранения избыточных блокировок

В каком случае происходит блокировка

Возможность параллельного выполнения операций над одним и тем же ресурсом в двух разных сессиях 1С:Предприятия		Сессия 2			
		Чтение вне транзакции	Чтение в транзакции	Чтение ДЛЯ ИЗМЕНЕНИЯ	Запись
Сессия 1	Чтение вне транзакции	+	+	+	+
	Чтение в транзакции	+	+	+	-
	Чтение ДЛЯ ИЗМЕНЕНИЯ	+	+	-	-
	Запись	+	-	-	-

Что при этом блокируется

- Все затронутые операцией данные (на уровне записей таблиц)
  - При выполнении запроса будут заблокированы все прочитанные записи, а не только те, которые были получены по условию запроса
- При работе с необъектными данными будет дополнительно заблокирован диапазон значений, соответствующий условию выборки и соседние с этим диапазоном записи
- В некоторых случаях могут быть заблокированы все ресурсы данного типа (то есть, таблица целиком)

Проектирование системы с учетом необходимых блокировок

- Программист не устанавливает блокировки в явном виде
- Блокировки устанавливаются автоматически при обращении к данным информационной базы
  - Косвенное управление блокировками осуществляет 1С:Предприятие
  - Непосредственное управление блокировками осуществляется SQL сервером
- Способы управления блокировками, имеющиеся в распоряжении разработчика:
  - Использование транзакции
  - Использование опции запроса ДЛЯ ИЗМЕНЕНИЯ
- Требуется осторожность при использовании транзакции и опции ДЛЯ ИЗМЕНЕНИЯ:
  - Они используются для того, чтобы заблокировать нужные ресурсы, то есть уменьшить параллельность работы пользователей
  - Можно использовать только в том случае, когда это оправдано с точки зрения бизнес-логики

Проектирование системы с учетом необходимых блокировок

- **Пример блокировки, необходимой с точки зрения бизнес-логики:**
  - **Контроль остатков товара на складе при оперативном проведении документа, списывающего товар**
  - **Два пользователя одновременно проводят документы, списывающие один и тот же товар с одного и того же склада**
  - **Списание товара производится в два этапа:**
    - ***Проверка наличия необходимого количества товара на складе***
    - ***Списание проданного товара***
  - **Если не заблокировать остаток, то система может позволить списать больше товара, чем имеется на складе**
  - **Операция должна быть изолирована от возможного вмешательства параллельно работающих пользователей**
  - **Блокировка должна быть установлена во время выполнения проверки и снята после окончания списания**
  - **Для этого операция выполняется в транзакции**

Проектирование системы с учетом необходимых блокировок

- **Обобщенная ситуация:**
  - **Необходимо выполнить некоторую операцию над данными либо получить из одних (первичных) данных другие (вторичные):**
    - ***Прочитать данные***
    - ***Выполнить преобразование данных***
    - ***Записать данные либо создать на их основе другие данные***
  - **На все время операции исходные данные должны быть заблокированы для того, чтобы предотвратить возможность их изменения в процессе расчета**
  - **Для этого следует выполнять всю операцию в единой транзакции**

Проектирование системы с точки зрения минимизации блокировок

- **Пример избыточной блокировки:**
  - **Необходимо запретить двум пользователям одновременно списывать один и тот же товар с одного и того же склада**
  - **Однако необходимо разрешить одновременно списывать разные товары или товары с разных складов**
- **Проектное решение должно обеспечить правильность работы с точки зрения бизнес-логики и позволить избежать избыточной блокировки:**
  - **Неправильное решение. Хранить в системе остатки в разрезе товара,** то есть создать регистр накопления «ОстаткиТоваров» с одним измерением: «Товар». В этом случае при списании одного товара с разных складов будет возникать избыточная (с точки зрения бизнес-логики) блокировка
  - **Правильное решение. Хранить остатки в разрезе товаров и складов,** то есть создать регистр накопления «ОстаткиТоваров» с двумя измерениями: «Товар» и «Склад»

Общая методика проектирования

- **Выделить из всей совокупности данных системы оперативные данные – информацию, которая будет одновременно вводиться большим количеством пользователей. Как правило, это будут некоторые виды документов**
- **Разделить оперативный и неоперативный режим работы этих документов**
- **Свести к необходимому минимуму набор регистров, которые читаются и записываются в оперативном режиме**
- **Требования к регистрам:**
  - **Учитывать особенности структур хранения данных регистров**
  - **Правильно спроектировать измерения.** Данные в регистрах хранятся в разрезе измерений. Измерения должны быть подобраны так, чтобы обеспечить возможность массового параллельного ввода данных
  - **Правильно задать индексы регистров и формулировать условия запросов.** Для всех условий запросов должен быть подходящий индекс
  - **Избегать ситуаций, при которых регистр может оказаться пустым.** Пустой регистр – потенциальный источник избыточной блокировки. Если регистр не является необходимым, то не следует его использовать в оперативном режиме
- **Требования к оперативному режиму проведения:**
  - **Функциональность должна быть сведена к необходимому минимуму.** Остальные действия вынести в неоперативный режим или в регламентные операции
  - **Не перемещать границу последовательности при оперативном проведении документа.** Все бизнес-процессы, которые требуют движения границы последовательности (например, партионный учет) вынести в неоперативный режим.
  - **Избегать обращения к данным, которые не являются необходимыми для получения результата.** Необходимо помнить, что все эти данные будут заблокированы от записи

Работа с транзакциями

- Использовать транзакцию только в том случае, когда это действительно необходимо:
  - С точки зрения бизнес-логики задачи необходимо изолировать последовательность действий с данными от возможного вмешательства со стороны параллельно работающих пользователей
  - Все данные, которые были затронуты с момента начала транзакции, будут оставаться заблокированными до конца транзакции
- Помнить, что проведение документа автоматически производится в транзакции
- Осмотрительно использовать опцию ДЛЯ ИЗМЕНЕНИЯ в запросах:
  - Все данные, прочитанные с опцией ДЛЯ ИЗМЕНЕНИЯ будут заблокированы не только на запись, но и на чтение
  - Использовать ДЛЯ ИЗМЕНЕНИЯ только в тех случаях, когда это необходимо:
    - *Запрос на чтение данных в транзакции с целью их модификации*
  - Не использовать «открытую» форму опции (без указания списка таблиц)
    - *В этом случае будут заблокированы записи всех таблицы, используемые в запросе*
  - Указывать только те таблицы, которые будут изменены в данной транзакции
- Избегать чтения ненужных данных при работе в транзакции. Все прочитанные данные будут заблокированы от записи
- Оптимизировать все выполняемые запросы по скорости для того, чтобы уменьшить общее время выполнения транзакции
- Недопустимо использование в транзакциях интерактивных операций, требующих ответной реакции пользователя (предупреждения, вопросы и т.д.)

Индексы и условия запросов

- Отсутствие индексов, необходимых для отбора данных может привести к чтению большего количества данных, чем необходимо для выполнения операции. Все прочитанные данные будут заблокированы
- Необходимо внимательно отнестись к условиям запросов, выполняемых в транзакции:
  - У объектов метаданных должны быть созданы индексы, подходящие для отбора по условиям запросов
  - Если такого индекса не будет, то SQL сервер будет вынужден сканировать в поисках данных всю таблицу или некоторую ее часть и, соответственно, заблокирует все прочитанные записи, а не только те записи, которые удовлетворяют условиям запроса

Блокировки на пустых регистрах

- Имеется регистр, в котором по каким-то причинам нет записей (то есть он не используется в текущей работе)
- Происходит обращение к этому регистру в транзакции
  - Чтение с опцией ДЛЯ ИЗМЕНЕНИЯ
  - Удаление по любому условию
- При выполнении этих операций SQL сервер заблокирует таблицу регистра целиком, то есть все подобные операции будут выполняться строго последовательно

Блокировки при обмене данными

- Выгрузка изменений блокирует внесение новых изменений для объектов данного типа
- Рекомендуется делать выгрузку во время минимальной загрузки системы (в нерабочее время)
- Если это невозможно, то осуществлять выгрузку короткими транзакциями:
  - Разбивать выгрузку на несколько транзакций
  - Делать выгрузку в одной транзакции, но достаточно часто

Конфликты блокировок

- Конфликт взаимной блокировки из-за разного порядка обращения к ресурсам:
  - Сессия 1 открывает транзакцию и обращается к ресурсу А, что приводит к его блокировке
  - Сессия 2 открывает транзакцию и обращается к ресурсу Б, что приводит к его блокировке
  - Сессия 1 пытается обратится к ресурсу Б, но не может этого сделать, так как он уже заблокирован Сессией 2. Сессия 1 становится в очередь
  - Сессия 2 пытается обратится к ресурсу А, но не может этого сделать, так как он уже заблокирован Сессией 1. Сессия 2 становится в очередь
  - Сессии ждут друг друга и не могут завершить свои транзакции
  - Менеджер блокировок SQL сервера обнаруживает неразрешимый конфликт блокировок и снимает одну из транзакций
- Рекомендации: всегда соблюдать один и тот же порядок при обращении к ресурсам

Конфликты блокировок

- **Конфликт взаимной блокировки при попытке записи после чтения:**
  - **Сессия 1 открывает транзакцию и читает ресурс А, чем блокирует его от записи (но не от чтения)**
  - **Сессия 2 открывает транзакцию и читает ресурс А, чем блокирует ресурс от записи**
  - **Сессия 1 хочет изменить ресурс А, но не может этого сделать, так как ресурс заблокирован Сессией 2**
  - **Сессия 2 пытается сделать то же самое, но не может из-за блокировки, установленной Сессией 1**
  - **Получаем неразрешимый конфликт**
- **Типичный пример такого конфликта: проверка остатков при записи движений в регистр во время проведения документа.** Документ сначала проверяет доступность товара (читает таблицу остатков регистра), а затем изменяет значение остатков (пишет в таблицу остатков регистра). Два таких документа с высокой вероятностью попадают в неразрешимый конфликт блокировок
- **Рекомендации. Если предполагается сначала прочитать ресурс в транзакции, а затем его обновить, то необходимо использовать опцию ДЛЯ ИЗМЕНЕНИЯ**

Избыточные блокировки, возникающие по вине SQL сервера

- **Реальные блокировки могут не совпадать с тем, что написано в приложении:**
  - **Блокировки устанавливает SQL сервер**
  - **Блокируются не те ресурсы, к которым обращалось приложение, а те, которые в действительности были затронуты при выполнении запроса**
  - **Если условие отбора не полностью удовлетворяется имеющимися индексами, то заблокированными может оказаться существенно больше записей, чем необходимо, вплоть до всей таблицы**
  - **Такие ситуации могут возникать по разным причинам:**
    - ***Ошибка разработчика (отсутствие необходимого индекса)***
    - ***Устаревшие статистики SQL сервера***
    - ***Эскалация блокировок, выполняемая SQL сервером для оптимизации своей работы***
- **Рекомендации:**
  - **Вероятность возникновения таких ситуаций уменьшается, если следовать рекомендациям в ходе проектирования и разработки**
  - **Рекомендуется регулярно выполнять реиндексацию информационной базы**
  - **Рекомендуется регулярно обновлять статистики: sp_updatestats**

Тестирование работы приложения в многопользовательском режиме

- Не всегда можно избежать избыточных блокировок путем следования рекомендациям во время проектирования и разработки
- Необходимо полноценное тестирование работы приложения в многопользовательском режиме
- Пример: тестирование масштабируемости УПП

Спасибо за внимание!