МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**
Кафедра математической кибернетики и компьютерных наук

АНАЛИЗ И СРАВНЕНИЕ РАБОЧИХ НАГРУЗОК БАЗ ДАННЫХ НА ДИСКЕ И В ОПЕРАТИВНОЙ ПАМЯТИ

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

Студентки 4 курса 411 группы
направления 02.03.02 — Фундаментальная информатика и информационные технологии
факультета КНиИТ
Гаврилкиной Елизаветы Петровны

Научный руководитель
старший преподаватель _____ М. И. Сафрончик

Заведующий кафедрой
к. ф.-м. н., доцент _____ С. В. Миронов

Саратов 2023

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Выпускная квалификационная работа посвящена технологии In-Memory OLTP. Актуальность работы, в первую очередь, обусловлена тем, решение проблем производительности баз данных является сегодня крайне актуальной задачей, так как объемы информации неуклонно растут вместе с требованиями к скорости их обработки. Задача оптимизации одна из ключевых в отрасли баз данных и одним из популярных решений стала технология In-Memory — это набор концепций хранения данных, когда они сохраняются в оперативной памяти приложения, а диск используется для бэкапа. Так как максимального повышения быстродействия можно добиться за счет памяти.

Целью данной работы является сравнительный анализ технологии In-Memory OLTP и привычной технологии баз данных на диске. Особенности практического применения технологии In-memory для повышения производительности базы данных, за счет ее миграции в оперативную память.

Задачи, реализованные в ВКР:

- Создать БД на диске, реализовать бизнес-логику с помощью триггеров и хранимых процедур.
- Проанализировать рабочую нагрузку базы данных на диске для выяснения возможной миграции ее в память.
- Перенос БД в оперативную память и анализ рабочих нагрузок этой БД в ОП.
- Сравнение показателей с целью выявления возможного повышения производительности.

Бакалаврская работа включает в себя введение, две главы, заключение, список из 21 источника и семь приложений. Глава «Технология In-Memory» описывает теоретическую часть работы. Рассматривается технология In-Memory OLTP, ее применение, нюансы ее использования на практике, достоинства и недостатки. Так же были рассмотрены среда и средства разработки базы данных. В главе «Реализация базы данных и анализ и сравнение рабочих нагрузок» была описана практическая часть диплома. Определен общий объем работ, проводится анализ рабочих нагрузок таблиц на диске и в памяти, наглядно иллюстрируется получившийся результат разработки. Рассматривается разработка программы для MS SQL Managemant Studio 18, а так же проиллюстрированы полные отчеты о проведенных тестах.

1 Технология In-Memory

Данные, хранящиеся в оперативной памяти, доступны намного быстрее, чем данные, хранящиеся на диске. В In-Memory OLTP данные хранятся в виде оптимизированных таблиц, индексов и процедур обработки, что ускоряет работу с ними [1].

В ядре обработки в памяти баз данных OLTP реализован новый механизм оптимизации управления конкурентным доступом с помощью многоверсионности. Данная схема организации обеспечивает более высокое быстродействие и большую степень масштабируемости, поскольку не допускает применения блокировок или других процессов, препятствующих функционированию процессора с максимальной скоростью.

Цель данного процесса состоит в том, чтобы сократить число команд, которые процессор должен выполнить в ходе выполнения запроса. Ядро обработки в памяти базы данных OLTP может обеспечить повышение производительности от 10 до 30 раз [2].

Так как многоядерные процессоры используют общий кеш, и обычные алгоритмы параллельного выполнения основных операций баз данных эффективно работают только при малых объемах данных, не переполняющих этот кеш, т. е. для нагрузок типа OLTP [4].

Технология In-Memory OLTP использует столбцы с плотной упаковкой, основываясь на том, что в памяти уже отсортированных данных поиск значительно быстрее, чем в случае с традиционными строками. Платформа предоставляет оптимизированные структуры, такие как хранилище данных (Memory-Optimized Table), конструкции структур в памяти (Memory-Optimized Index) и хранилище процедур (Memory-Optimized Stored Procedure), которые позволяют работать с данными намного быстрее, чем традиционные способы обработки SQL-запросов.

Для обеспечения устойчивости базы данных при использовании таблиц In-Memory OLTP, в SQL Server предусмотрен механизм создания точек контроля для свободных таблиц In-Memory OLTP (как отдельного объекта базы данных) и соответствующих логов.

In-Memory OLTP позволяет использовать не только полностью in-memory таблицы, но также и гибридные таблицы, которые могут хранить данные как в оперативной памяти, так и на диске. Такие гибридные таблицы также назы-

ваются таблицами с оптимизацией для памяти (memory-optimized table with durability).

1.1 Хранение данных

Технология In-Memory OLTP использует другой подход к хранению данных в отличие от дисковых таблиц.

В таблицах, оптимизированных для памяти, данные хранятся в CHECKPOINT FILE PAIRS.

CHECKPOINT FILE PAIRS состоит из двух файлов:

— DATA FILE – содержит новые версии записей;

— DELTA FILE – содержит информацию об удаленных записях.

При создании новой записи добавляется запись в DATA FILE. При удалении добавляется запись в DELTA FILE. А при изменении записи старая версия записи добавляется в DELTA FILE, а новая записывается в DATA FILE.

При загрузке данных SQL Server начинает читать данные из DATA FILE, используя их как фильтры, соответственно записи, которые были удалены, в память не загружаются.

OLTP-операции в памяти интегрированы с SQL Server для эффективной работы в любых областях, таких как разработка, развертывание, управление и поддержка. База данных может содержать как объекты в памяти, так и объекты на диске [5] [6].

1.2 Транзакции

В SQL Server 2016 при использовании in-memory OLTP транзакции работают на основе оптимистической блокировки. Это означает, что вместо того чтобы блокировать данные для других транзакций, система просто проверяет, не изменялись ли данные с момента последнего обращения к ним. Если данные были изменены другой транзакцией, текущая транзакция будет отменена.

Для поддержки транзакций в in-memory OLTP в SQL Server 2016 используется новый объект - natively compiled stored procedure (скомпилированная хранимая процедура). Такие процедуры компилируются в машинный код, что позволяет значительно ускорить их выполнение и обеспечить более низкий уровень задержек при обработке транзакций.

Кроме того, SQL Server 2016 поддерживает транзакции в распределенной среде. Использование распределенных транзакций может быть полезно, когда

необходимо координировать работу нескольких баз данных или серверов.

1.3 Индексы

Использование индексов в In-Memory OLTP имеет свои особенности:

— Индексы должны быть определены во время создания таблицы.
— Индексы должны быть непрерывными.
— Индексы могут быть компактными.
— Индексы не могут быть совместными.
— Индексы могут быть хешированными [7].

Активно используются специальные индексы для таблиц данных:

— Кластерные индексы;
— Внешние индексы;
— Первичные индексы;
— Хэш индексы (Hash);
— Некластерные индексы, которые специально создаются для сканирования и оптимизируются для памяти. [8]

Каждая операция CREATE TABLE для таблицы, оптимизированной для памяти, должна включать индекс либо явно посредством INDEX, либо неявно посредством ограничения PRIMAY KEY или UNIQUE.

Избегайте большого количества некластеризованных индексов, если таблица используется в приложении оперативной обработки транзакций (online transaction processing, OLTP) или часто обновляется. Большое количество индексов для таблицы влияет на производительность инструкций INSERT, UPDATE и DELETE, так как все индексы должны обновляться при изменении данных в таблице [9].

1.3.1 HASH индексы

Рассмотрим совершенно новый тип индексов – HASH индексы. Создание такого индекса осуществляется с помощью внутренней HASH функции, которая является детерминированной и единственной для всего MS SQL Server,а это значит, что несколько значений ключей индекса не могут быть связаны с одним HASH индексом,иначе появляется конфликт HASH-а.Если будет большое количество конфликтов, то это отрицательно отразиться на операции чтения.

Хэш-индекс состоит из массива указателей. Каждый элемент массива называется хэш-контейнером:

— Каждый контейнер имеет размер 8 байт, в которых хранится адрес списка ссылок на записи ключей.
— Каждая запись представляет собой значение ключа индекса, к которому добавляется адрес соответствующей строки в оптимизированной для памяти таблице.
— Каждая запись указывает на следующую запись в списке ссылок на записи, связанных с текущим контейнером.

1.4 Процедуры и триггеры

Для работы с оптимизированными в памяти таблицами необходимо использовать оптимизированные для них процедуры и триггеры. Отличительной особенностью их является возможность из компиляции в собственном коде.

Хранимые процедуры Transact-SQL, которые отмечены в программе как NATIVE COMPILATION, компилируются в собственном коде. Это означает, что процедура компилируется в машинный код для эффективного выполнения бизнес-логики, критической с точки зрения производительности [11].

Такие процедуры отличаются от стандартных, у них есть ряд особенностей:

— код процедуры определяется один раз, далее ее можно изменить только через пересоздание;
— внутри процедуры транзакция определяется как BEGIN ATOMIC, что определяет свои требования;
— объекты, на которые ссылается процедура, не могут быть изменены при наличие данных процедур;
— нельзя просмотреть актуальный план данных процедур;
— нельзя получить статистику выполнения данных процедур;
— для соединения таблиц внутри хранимой процедуры используется только NETEDLOOP;
— не используется параллелизм;
— план выполнения Native Compiled процедуры определяется в момент ее создания, в SQL Server 2016 для перекомпиляции процедуры можно использовать процедуру sp recompile [12].

В In-Memory OLTP могут использоваться триггеры и хранимые процедуры, как и в традиционной SQL Server. Однако, для использования In-Memory OLTP некоторые требования к описанию и использованию этих объектов могут

отличаться от обычной SQL Server.

В целом, использование триггеров и хранимых процедур в In-Memory OLTP может повысить производительность запросов и обработку данных, но при этом они должны быть оптимизированы для ожидаемых условий использования и соответствовать требованиям In-Memory OLTP.

1.5 Безопасность

In-Memory OLTP в SQL Server 2016 использует механизм прозрачного шифрования данных (Transparent Data Encryption, TDE) для защиты данных в памяти. Это означает, что все данные, хранящиеся в памяти, автоматически шифруются, причем дешифровка происходит только на уровне процесса, где они хранятся. Таким образом, данные в памяти защищены от внешнего доступа и эксплуатации.

Шифрование файлов базы данных выполняется на уровне страницы. Страницы в зашифрованной базе данных шифруются до записи на диск и дешифруются при чтении в память. Прозрачное шифрование данных не увеличивает размер зашифрованной базы данных [13].

К преимуществам использования именно этого способа защиты данных, кроме выигрыша в памяти и скорости шифровки, относятся:

— Основное преимущество прозрачного шифрования данных в In-Memory OLTP состоит в том, что оно обеспечивает высокий уровень защиты данных, хранящихся в памяти сервера.
— In-Memory OLTP в SQL Server 2016 может использоваться с минимальной настройкой.
— Совместимость с другими функциональными возможностями.

Но при этом есть ряд недостатков этого метода, которые ограничивают использование технологии In-Memory OLTP:

— Отрицательной стороной средства TDE является то, что оно требует дополнительных накладных расходов.
— Высокие затраты на хранение.
— Недостатком TDE является то, что он сильно загружает процессор и память при каждом обращении к базе данных.
— Ограничения на обработку данных.
— TDE также шифрует базу данных tempdb, что может повлиять на производительность всех прочих баз в пределах одного экземпляра SQL Server.

— Шифрование на уровне столбца требует дополнительного программирования.
— Функция прозрачного шифрования данных не обеспечивает шифрование каналов связи, что способствует дополнительным рискам потери данных.
— Шифрование не гарантирует полной безопасности.

При выборе стратегии шифрования ваших баз данных рекомендуется тщательно взвешивать все за и против TDE и шифрования уровня столбцов [14].

1.6 Достоинства и недостатки технологии In-Memory OLTP

Технология In-Memory OLTP имеет ряд преимуществ в обработке больших объемов данных. Рассмотрим основные из них:

— Высокая производительность.
— Масштабируемость.
— Лучшая поддержка критических приложений.
— Работа с данными в реальном времени.

Все эти достоинства в сумме дают значительный прирост производительности, который может достигать разницы в 30 раз.

Не смотря на все преимущества OLTP-систем, практика использования OLTP-систем показала неэффективность их применения для полноценного анализа информации. Аналитические запросы к БД очень трудно формулируются и крайне неэффективно выполняются, поскольку содержат в себе представления, объединяющие большое количество таблиц [15]. Поэтому при проектировании систем анализа стараются максимально упростить схему БД и уменьшить количество таблиц, участвующих в запросе [15] [16].

С практической точки зрения все эти недостатки могут привести к сбоям системы OLTP и аппаратным сбоям, которые в свою очередь могут привести к простою системы, и соответственно повлиять на проведение онлайн-транзакции.

С технической точки зрения тоже есть ряд неблагоприятных факторов:

— Требует больших объемов памяти.
— Высокие требования к процессору.
— Лимитированный размер базы данных.
— Требуется дополнительная группировка данных.
— Увеличение времени на создание резервных копий и восстановление данных.

1.7 Практическое применение технологии In-Memory OLTP

Для использования технологии In-memory OLTP в SQL Server 2016 необходимо выполнить следующие шаги:

- Установить и настроить SQL Server 2016 с функциональностью In-memory OLTP. При установке SQL Server 2016 можно выбрать опцию установки In-memory OLTP, которая добавит соответствующие функции в БД.
- Создать таблицы в памяти (In-memory). Для этого нужно использовать новый тип таблицы 'memory-optimized', который будет храниться в оперативной памяти (RAM).
- Настроить индексы в памяти для оптимизации производительности и увеличения скорости запросов.
- Перенести данные из традиционных таблиц в таблицы In-memory. Для этого можно использовать инструменты импорта данных SQL Server.
- Настроить администрирование и мониторинг In-memory OLTP. Необходимо использовать специализированные инструменты для мониторинга использования ресурсов памяти и процессора.
- Тестирование и оптимизация производительности. Необходимо проводить тестирование производительности и настраивать настройки для достижения максимальной производительности. [17] [18] [19].

1.8 Выбор среды и средства разработки

В качестве основной среды разработки был выбран SQL Server Management Studio 18.

Для визуализации базы данных при построении был использован интернет-ресурс dbdesigner.

Был применен стандартный метод SQL Server - PowerShell, для работы с которым были установлены модули SQL Server и SQL Server PowerShell [18].

С помощью стандартных отчетов, встроенных в SQL Management Studio 18 проводится проанализируем дисковые таблицы в построенной базе и таблиц, хранимых в памяти.

2 Реализация базы данных и анализ и сравнение рабочих нагрузок

Для реализации сравнительного эксперимента и последующего анализа его результатов были реализованы база данных с двумя видами таблиц: таблицы с хранением на диске, таблицы с хранением в памяти. К каждому виду таблиц был прописан ряд триггеров и процедур, по которым производился замер производительности. По данным полученных замеров произведено сравнение по времени.

Далее были созданы такие же таблицы, триггеры и процедуры, но уже оптимизированные в памяти. Для достоверности результата используемые триггеры и процедуры были одинаковы для обеих баз.

Сравнивались два способа обработки запроса, начнем с самого простого способа, замерим время работы дисковой процедуры на дисковых таблицах и процедуры в памяти на таблицах, оптимизированных для памяти.

Было замерено время выполнения каждой процедуры и каждого триггера для наших таблиц, записанное в отдельный файл полученные значения и построим диаграмму, по которой видно, что производительность выше у таблиц, расположенных в памяти, при выполнении одних и тех же запросов на идентичных таблицах с одинаковым количеством строк.

С помощью стандартных отчетов, встроенных в SQL Management Studio 18, проанализированы дисковые таблицы в созданной экспериментальной базе [21].

Раздел о миграции в отчете содержит таблицу со сведениями о трудностях, связанных с преобразованием этой таблицы базы данных в оптимизированную для памяти таблицу. Чем выше оценка трудности, тем сложнее преобразовать таблицу.

Таким образом, если рассмотреть полученные тесты, то видно, что самое большое число блокировок у таблицы, содержащей в себе больше всего строк и внешних ключей, ее оценка миграции выше других. Это подтверждает, что производительность метода OLTP напрямую зависит от объема дисковых таблиц.

ЗАКЛЮЧЕНИЕ

В рамках данной работы было изучено большое количество различных источников в поисках решения поставленной задачи. Изучены различные способы работы с базами данных посредством SQL Server 2016, Sql Management Studio 18. Была создана база данных, заполненная сгенерированными данными.Также был изучен процесс создания таблиц, оптимизированных для памяти, а также работа с ними. Были разработаны различные триггеры и процедуры, которые применялись для измерения производительности для каждой из баз. А также применялись встроенные средства SQL Server 2016 для выполнения анализа нагрузки баз.

Был произведен сравнительных анализ производительности дисковых таблиц и таблиц в памяти. Был зафиксирован прирост производительности при использовании таблиц, хранимых в памяти, в 3-12 раз, что дает значительный выигрыш при обработке больших данных в промышленных масштабах. Также было выявлено, что запросы к таблицам, имеющим сложные соединения, требуют больше времени для оптимизации, что может тормозить обработку текущих транзакций. Также при любом сбое на сервере или его перезагрузке системе потребуется время для заполнения оперативной памяти данными дисковой базы заново, что усложняет работу в реальном времени.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 *Кэмпбелл, Л.* Базы данных / Л. Кэмпбелл, Ч. Мейджорс. — СПб.: Питер, 2020. — 304 с.

2 Майкл Оти. Революция технологий обработки в памяти [Электронный ресурс]. — URL: https://www.osp.ru/winitpro/2014/02/13039445 (дата обращения: 10.04.2023). Загл. с экр. Яз. Рус.

3 Усовершенствования In-memory OLTP в SQL Server 2016 [Электронный ресурс]. — URL: https://nerohelp.com/11005-in-memory-oltp-sql-server-2016.html (дата обращения: 15.04.2023). Загл. с экр. Яз. Рус.

4 *Новиков, Б.А.* Основы технологий баз данных: учебное пособие / Б.А. Новиков, Е.А. Горшкова, Н.Г. Графеева. — 2-е изд. М.: ДМК Пресс, 2020. — 582 с.

5 Алгоритм должен быть устойчив к большим файлам не помещающимся целиком в оперативную память [Электронный ресурс]. — URL: https://kompyutery-programmy.ru/komplektuyuschie/algoritm-dolzhen-byt-ustojchiv-k-bolshim-fajlam-ne-pomeschayuschimsya-celikom-v-operativnuyu-pamyat.html (дата обращения: 17.04.2023). Загл. с экр. Яз. Рус.

6 Введение в таблицы, оптимизированные для памяти [Электронный ресурс]. — URL: https://learn.microsoft.com/ru-ru/sql/relational-databases/in-memory-oltp/introduction-to-memory-optimized-tables?view=sql-server-ver15 (дата обращения: 20.04.2023). Загл. с экр. Яз. Рус.

7 Руководство по архитектуре и разработке индексов SQL Server и Azure SQL [Электронный ресурс]. — URL: https://learn.microsoft.com/ru-ru/sql/relational-databases/sql-server-index-design-guide?view=sql-server-ver165 (дата обращения: 22.04.2023). Загл. с экр. Яз. Рус.

8 Все, что необходимо знать про индексы MS SQL [Электронный ресурс]. — URL: https://otus.ru/journal/vse-chto-neobhodimo-znat-pro-indeksy-ms-sql/ (дата обращения: 22.04.2023). Загл. с экр. Яз. Рус.

9 Индексы для оптимизированных для памяти таблиц [Электронный ресурс]. — URL: https://learn.microsoft.com/ru-ru/sql/relational-databases/in-memory-oltp/indexes-for-memory-optimized-tables?view=sql-server-

ver1/viewfallbackfrom=aps-pdw-2016-au7 (дата обращения: 29.04.2023). Загл. с экр. Яз. Рус.

10 *Осетрова, И.С.* Разработка баз данных в MS SQL Server 2014 / И.С. Осетрова. — СПб.: Университет ИТМО, 2016. — 114 с.

11 Собственная компиляция таблиц и хранимых процедур [Электронный ресурс]. — URL: https://learn.microsoft.com/ru-ru/sql/relational-databases/in-memory-oltp/native-compilation-of-tables-and-stored-procedures?view=sql-server-ver16viewfallbackfrom=aps-pdw-2016 (дата обращения: 01.05.2023). Загл. с экр. Яз. Рус.

12 Создание хранимых процедур, скомпилированных в собственном коде [Электронный ресурс]. — URL: https://learn.microsoft.com/ru-ru/sql/relational-databases/in-memory-oltp/creating-natively-compiled-stored-procedures?view=sql-server-ver15 (дата обращения: 02.05.2023). Загл. с экр. Яз. Рус.

13 Прозрачное шифрование данных (TDE) [Электронный ресурс]. — URL: https://learn.microsoft.com/ru-ru/sql/relational-databases/security/encryption/transparent-data-encryption?view=sql-server-ver15 (дата обращения: 08.05.2023). Загл. с экр. Яз. Рус.

14 *Абдулхассан, Ф.Х.* Прозрачное шифрование данных (TDE) // Молодой ученый. – 2014. — № 8 (67). — С. 122-125.

15 *Барсегян, А.А.* Методы и модели анализа данных: OLAP и Data Mining / А.А. Барсегян, М.С. Куприянов, В.В. Степаненко, И.И. Холод. — СПб.: БХВ-Петербург, 2004. — 336 с.

16 OLTP [Электронный ресурс]. — URL: https://translated.turbopages.org/proxyu/en-ru.ru.7edccf04-6476a05c-b5c5c83c-74722d776562/https/corporatefinanceinstitute.com/resources/data-science/oltp/ (дата обращения: 10.05.2023). Загл. с экр. Яз. Рус.

17 Microsoft SQL Server. Таблицы в памяти (in-memory tables) [Электронный ресурс]. — URL: http://sqlcom.ru/optimizationquery/in-memory-tables-simple/ (дата обращения: 11.05.2023). Загл. с экр. Яз. Рус.

18 In-Memory tables. Таблицы в памяти - просто. [Электронный ресурс]. — URL: https://dbasimple.blogspot.com/2016/10/in-memory-tables.html (дата обращения: 14.05.2023). Загл. с экр. Яз. Рус.

19 Как создать таблицу, оптимизированную для памяти? Технология In-Memory OLTP в Microsoft SQL Server [Электронный ресурс]. — URL: https://info-comp.ru/obucheniest/679-create-table-in-memory-oltp.html (дата обращения: 16.05.2023). Загл. с экр. Яз. Рус.

20 SQL Server PowerShell [Электронный ресурс]. — URL: https://learn.microsoft.com/ru-ru/sql/powershell/sql-server-powershell?view=sql-server-2017 (дата обращения: 20.05.2023). Загл. с экр. Яз. Рус.

21 Определение, должна ли таблица или хранимая процедура быть перенесена в In-Memory OLTP [Электронный ресурс]. — URL: https://learn.microsoft.com/ru-ru/sql/relational-databases/in-memory-oltp/determining-if-a-table-or-stored-procedure-should-be-ported-to-in-memory-oltp?view=sql-server-ver16viewfallbackfrom=azure-sqldw-latest (дата обращения: 21.05.2023). Загл. с экр. Яз. Рус.