

ПОСТРЕЛЯЦИОННАЯ СУБД CACHE'

АКУЛОВ Олег, МАКИДОН Мария
Coordonator: САРАНЧУК Дориан

Технический Университет Молдовы

Аннотация: Переход современных пользователей информационных технологий к наибольшему использованию веб сервисов поставил перед разработчиками ряд проблем, которые невозможно разрешить с помощью реляционных СУБД прежде всего из-за весьма характерной для современного бизнеса необходимости усложнять и часто вносить изменения в структуры данных. Также, современные высоконагруженные приложения изменили требования к СУБД — сегодня необходимы эффективные технологии создания специализированных решений с быстрым временем реакции при обработке больших массивов данных.

Ключевые слова: база данных, СУБД, Cache', объектно ориентированное программирование.

1. Введение

Cache - это высокопроизводительная постреляционная база данных, а также развитая система управления базой данных (СУБД). Основным отличием БД Cache от РБД и ООБД является универсальность представления данных в Cache, реализуемая с помощью т.н. единой архитектуры данных. В рамках этой архитектуры существует единое описание объектов и таблиц, отображаемых непосредственно в многомерные структуры ядра базы данных, ориентированного на обработку транзакций. Имеющиеся серверы Cache Objects и Cache SQL предоставляют в распоряжение разработчика все наиболее популярные интерфейсы, посредством которых и унаследованные реляционные и новые объектно-ориентированные прикладные системы получают равноправный доступ к данным. Кроме реляционного и объектного доступов к данным, разработчику предлагается прямой доступ к данным (Cache Direct), т.е. доступ непосредственно к многомерным структурам ядра. Архитектура Cache приведена на рис. 1.

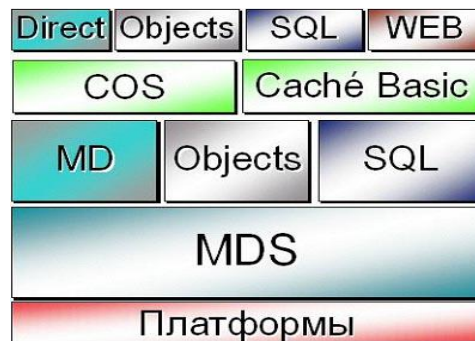


Рисунок 1 Архитектура постреляционной СУБД Cache

В Cache' реализована концепция Единой архитектуры данных. К одним и тем же данным, хранящимся под управлением Многомерного Сервера Данных Cache' есть три способа доступа: прямой, объектный и реляционный:

1. **Cache' Direct Access** - прямой доступ к данным, обеспечивает максимальную производительность и полный контроль со стороны программиста. Разработчики приложений получают возможность работать напрямую со структурами хранения. Использование этого типа доступа накладывает определенные требования на квалификацию разработчиков, но понимание структуры хранения данных в Cache' позволяет оптимизировать хранение данных приложения и создавать сверхбыстрые алгоритмы обработки данных.
2. **Cache' SQL** - реляционный доступ, обеспечивающий максимальную производительность реляционных приложений с использованием встроенного SQL. Cache' SQL соответствует

стандарту SQL 92. Кроме этого, разработчик может использовать разные типы триггеров и хранимых процедур. Все это позволяет Cache' успешно конкурировать с реляционными СУБД. Даже без использования прямого и объектного доступа приложения на Cache' работают быстрее за счет высокой производительности Многомерного Сервера Данных.

3. **Cache' Objects** - объектный доступ, для максимальной продуктивности разработки при использовании Java, Visual C++, VB и других ActiveX-совместимых средств разработки, таких как PowerBuilder и Delphi. В Cache' реализована объектная модель в соответствии с рекомендациями ODMG (Группа управления объектными базами данных – Object Database Management Group). В Cache' полностью поддерживаются наследование (в том числе и множественное), инкапсуляция и полиморфизм.

Cache' позволяет комбинировать три типа доступа, оставляя разработчику свободу выбора. Например, при реализации биллинговой системы объектный доступ может использоваться при описании бизнес-логики приложения и создания пользовательского интерфейса с помощью объектно-ориентированных средств разработки (VB, Delphi, C++), реляционный доступ – для совместимости с другими системами и интеграции с инструментами построения отчетов и аналитической обработки данных (Seagate Info, Cognos, Business Objects). Прямой доступ обеспечивает максимальную производительность и может быть использован для реализации таких операций, в которых применение обычных хранимых процедур, основанных на SQL, не может обеспечить необходимую производительность. В биллинге к таким операциям относятся закрытие периода, массовая загрузка данных (CDR). Использование прямого доступа для реализации подобных операций позволяет увеличить производительность на 1-2 порядка.

Независимость хранения данных от способа их представления дает разработчику возможность создавать приложения на языках программирования, поддерживающих как реляционную технологию (ODBC, JDBC), так и объектную (Java, C++, XML и т.д.). При этом нет необходимости создавать промежуточную среду для конвертации из одного формата представления данных в другой - Cache произведет необходимую конвертацию автоматически.

Объектно-ориентированные базы данных (ООБД) по сравнению с традиционными БД обеспечивают следующие преимущества:

- в ООБД хранятся не только данные, но и методы их обработки, инкапсулированные в одном объекте;
- ООБД позволяют обрабатывать мультимедийные данные;
- ООБД допускают работу на высоком уровне абстракции;
- ООБД позволяют пользователям создавать структуры данных любой сложности.

В конце 1997 года компания InterSystems Corp. выпустила постреляционную СУБД Cache'. СУБД. Термин постреляционная СУБД обозначает принадлежность Cache' к СУБД нового поколения. Имеется в виду не только аспект времени, сколько ряд технологических преимуществ: единая архитектура данных и полная поддержка Cache' объектно-ориентированных технологий.

Так как реляционная модель не всегда подходит для описания сложных предметных областей, транзакционная модель данных Cache' позволяет оптимизировать данные на уровне хранения, поддерживать объектную модель и сложные типы данных. Все эти возможности значительно упрощают создание сложных систем.

Кроме этого, существует возможность значительно увеличить производительность системы, используя полную поддержку Cache' объектно-ориентированных технологий. При этом можно изменять и структуру базы, и способ работы с Cache' из клиентской части приложения.

2. Описание преимуществ Cache

В отличие от ранних многомерных СУБД, которые были оптимизированы для создания аналитических систем, Cache' ориентирована на системы обработки транзакций (Online Transaction Processing). В основе Cache' лежит транзакционная многомерная модель данных (TMDM), которая позволяет хранить и представлять данные так, как они чаще всего используются.

Многомерный сервер данных снимает многие ограничения, накладываемые реляционными СУБД, которые хранят данные в двумерных таблицах, ведь если реляционная модель состоит из большого количества таблиц, что необходимо при работе со сложными структурами данных, это существенно усложняет и замедляет выполнение сложных транзакций и ведет к хранению излишней информации.

Многомерный сервер данных Cache' предназначен для обработки транзакций в системах с большими и сверхбольшими БД и большим количеством одновременно работающих пользователей. Многомерный сервер данных Cache' позволяет разработчикам получить великолепную производительность, отказавшись от хранения избыточных данных и таблиц.

Отличительной особенностью СУБД Cache' является независимость хранения данных от способа их представления, что реализуется с помощью, так называемой, единой архитектуры данных Cache'. В рамках данной архитектуры существует единое описание объектов и таблиц, отображаемых непосредственно в многомерные структуры ядра базы данных, ориентированных на обработку транзакций. К одним и тем же данным, хранящимся под управлением многомерного сервера данных Cache' есть три способа доступа: прямой, объектный и реляционный.

Как только определяется класс объектов, Cache' автоматически генерирует реляционное описание данных этого класса в формате SQL. Подобным же образом, как только в словарь данных поступает DDL-описание в формате реляционной базы данных, Cache' автоматически генерирует реляционное и объектное описание данных, устанавливая тем самым доступ в формате объектов.

Все операции по редактированию проводятся только с одним описанием данных. Это позволяет сократить время разработки, сэкономить вычислительные ресурсы и приложения будут работать значительно быстрее.

Cache' хранит данные в виде многомерных разреженных массивов - глобалей. Уникальная транзакционная многомерная модель данных позволяет избежать проблем, присущих реляционным СУБД, оптимизируя данные на уровне хранения.

В основе Cache лежит реализация более простой, чем реляционная, модели, называемой по имени своих атомарных элементов глобалами (или если быть точным полное название Global Persistent Variables или просто globals). Глобаль - многомерный массив с разреженной индексацией. Ее элементы - узлы определяют значения индексов, причем элементы могут быть не содержащими данных, а индексы нет.

Глобалы не имеют схемы, допускают динамическое добавление столбцов, используют разреженное хранение значений столбцов. На уровне глобалов можно использовать при желании блокировки, транзакции, делать распределенное хранение и партиционирование.

На уровне глобалов не существует привычного для реляционных баз данных декларативного языка запросов. Запросы определяются алгоритмическим способом — выполнение запроса сводится к исполнению кода, написанного на языке Cache Object Script, который предоставляет достаточный набор простых, эффективных операций для работы с данными, хранимыми в глобалах.

Cache' Object Script. COS - полнофункциональный язык, который имеет все необходимые механизмы для работы с данными с помощью любого способа доступа. С помощью COS разработчик создает методы классов, триггеры, хранимые процедуры, различные служебные программы.

Уникальность Cache Object Script как языка программирования в том, что это пожалуй единственный язык, в синтаксисе которого явно введена конструкция для указания того, где хранится переменная — в памяти или грубо говоря на диске. Представьте, что в таких традиционных платформах как Java или .NET была бы такая возможность — во многом проблемы с преодолением среды между программой и базой данных просто бы не было.

Отсутствие такой конструкции для универсальных языков программирования после работы с Cache кажется странным — ведь естественно предположить, что код работает не только с переменными в памяти, но и с хранимыми переменными. При этом вам не надо заранее определять структуры в базе данных — вы просто работаете с ними так же как с переменными в языках с нестрогой типизацией.

В Cache можно использовать следующие классы (рис.2):

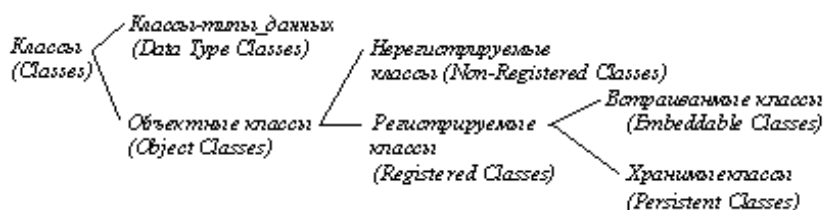


Рисунок 2. Типы классов СУБД Cache

Объектный доступ к данным обеспечивает:

- Быстрое и реалистичное моделирование сложных структур данных;
- Полную поддержку OO-методов проектирования и разработки, включая инкапсуляцию;
- Реализацию встраиваемых объектов, ссылок, коллекций, отношений;
- Быструю разработку приложений;
- Высокое быстродействие;
- Масштабируемость.

3. Web технологии в Cache

Когда речь заходит о создании сложных высокопроизводительных веб-приложений, то можно с уверенностью утверждать, что Web технологии, реализованные в Cache, удовлетворяют всем современным требованиям, когда быстрое развитие и приспособляемость так же важны, как быстродействие и масштабируемость.

Cache может автоматически создавать документы XML и соответствующие им схемы и определения данных (DTD) на основе классов Cache.

Все это обеспечивает быструю разработку XML-совместимых приложений и удобный обмен XML данными между приложениями.

Так же для Cache существует мастер создания Web-сервисов. Любой метод Cache может быть объявлен, как web-сервис. Cache автоматически создает WSDL описание и при вызове сервиса возвращает результат в виде сообщения SOAP.

Протокол DCP (Distributed Cache Protocol), технология DNM (Dynamic Namespace Mapping) и интерфейс Weblink позволяют пользователям эффективно работать в сети благодаря управлению кэш-памятью, прозрачному доступу к сетевым БД и возможности выполнения Web-приложений.

СУБД Cache' обеспечивает не только реализацию основных возможностей объектно-ориентированной технологии, но и позволяет во многом облегчить переход с реляционной технологии на объектную, а также может выступать в роле шлюза к реляционным базам данных.

Заключение

InterSystems Cache – пост-реляционная объектная СУБД, предоставляющая неограниченные возможности для разработки Web-решений и клиент-серверных приложений. Она обеспечивает великолепную производительность, богатые возможности масштабирования, кластеризации, зеркалирования и управления, возможность анализа данных в режиме реального времени, отказоустойчивость и надежность.

Помимо этого Cache' кроссплатформенная система, так как она поддерживает следующие операционные системы: Windows, Linux, основные реализации Unix и Open VMS.

Но при всех достоинствах современной объектной технологии разработки баз данных имеется несколько препятствий, которые удерживают разработчиков от принятия решения о переходе с реляционной технологии на объектную. Основным препятствием является значительный объем разработок, опирающихся на реляционные СУБД. Ведь при переходе на объектную технологию необходимо многое начинать с нуля, и поэтому возникает вопрос целесообразности такого перехода.

Кроме того, объектная технология, поддерживаемая в ряде постреляционных СУБД, не имеет развитого и стандартизированного языка генерации отчетов и анализа данных, каким является структурированный язык запросов SQL. Данные проблемы были решены при создании постреляционной СУБД Cache' от InterSystems.

Библиография

1. Кирстен В., Иррингер И., Рёриг Б., Шульте П. *СУБД Cache': объектно-ориентированная разработка приложений*. - СПб, "Питер", 2001.
2. *Руководство для разработчиков приложений в Cache'*. URL: <http://www.inftech.webservis.ru/it/database/cache/devguide/Model.html> (дата обращения 18.11.13)
3. Сиротюк Олег. *Постреляционная СУБД Cache'. Учебный курс*. InterSystems Corporation, 2002. URL: <http://www.ict.edu.ru/ft/003849/cache.pdf> (дата обращения 18.11.13).
4. *InterSystems Cache'. Высокопроизводительная объектная система управления базами данных*. URL: www.intersystems.ru (дата обращения 18.11.13).