УДК 004.652.4

ПРОБЛЕМА ПРОИЗВОДИТЕЛЬНОСТИ КОРПОРАТИВНЫХ ХРАНИЛИЩ ДАННЫХ

СИРОТИНСКИЙ Виктор Викторович – Московский авиационный институт (государственный технический университет), доцент, к.э.н.
Тел.: 499 158-62-18; e-mail: iprokof@mail.ru

Victor V. SIROTINSKY – Moscow Aviation Institute (State Technical University), associate professor, candidate of science
Phone: 499 158-62-18; e-mail: iprokof@mail.ru

СЕМЕНОВ Дмитрий Владимирович – Московский авиационный институт (государственный технический университет), аспирант
Тел.: 8-929-613-47-82; e-mail: iprokof@mail.ru

Dmitry V. SEMENOV – Moscow Aviation Institute (State Technical University), postgraduate
Phone : 8-929-613-47-82. e-mail: iprokof@mail.ru

В статье затронуты вопросы построения корпоративных хранилищ данных. Рассмотрены распространенные модели систем управления базами данных. Описаны отличия OLTP-систем от систем, ориентированных на анализ данных. Систематизированы современные способы оптимизации производительности аналитических информационных систем.

Data warehouse foundation aspects are discussed in the article. Differences of OLTP-systems from the systems focused on the analysis of the data are described. Modern ways of optimization of productivity of analytical information systems are systematized.

Ключевые слова: хранилище данных, оптимизация, архитектура, витрины данных, OLTP-система, колоночные СУБД.

Key words: data warehouse, database performance, model of data warehouse, data mart, OLTP-system, columnar DBMS.

Более пятидесяти лет назад, а именно в 1968 г., была введена в эксплуатацию первая промышленная система управления базами данных (СУБД) фирмы IBM – система IMS. Данная СУБД принадлежала к классу иерархических. В IMS использовалась оригинальная и нестандартная терминология: «сегмент» вместо «запись», а под «записью баз данных (БД)» понималось все дерево сегментов. В файловой системе иерархическая база данных представляла собой корневую директорию и древовидную структуру поддиректорий и файлов.

Плохие показатели выполнения основных операций над данными нижних уровней иерархий, ориентация на определенные типы запросов, сложные для обычного пользователя логические связи информации сделали такие системы неконкурентоспособными.

В 1975 г. появился первый стандарт ассоциации по языкам систем обработки данных – Conference of Data System Languages (CODASYL). Стандарт определил ряд фундаментальных понятий в теории систем баз данных, и до сих пор являющихся основополагающими для сетевой модели данных, на основе которой появились сетевые базы данных.

Сетевые базы данных подобны иерархическим, за исключением того, что в них имеются указатели в обоих направлениях, которые соединяют родственную информацию.

Несмотря на то что эта модель решает некоторые проблемы, связанные с иерархической моделью, выполнение простых запросов остается достаточно сложным процессом.

Также, поскольку логика процедуры выборки данных зависит от физической организации этих данных, эта модель не является полностью незави-

симой от приложения. Другими словами, если необходимо изменить структуру данных, то нужно изменить и приложение.

В то же время компания IBM активно вела работы над созданием новой модели данных – реляционной. В 1970 году сотрудник компании IBM Э.Ф.Кодд издал работу «A Relational Model of Data for Large Shared Data Banks», которая считается первой работой по реляционной модели данных.

Такие модели характеризуются простотой структуры данных, удобным для пользователя табличным представлением и возможностью использования формального аппарата алгебры отношений и реляционного исчисления для обработки данных.

В 1981 г. Э.Ф. Кодд получил за создание реляционной модели и реляционной алгебры престижную премию Тьюринга Американской ассоциации по вычислительной технике.

В начале 80-х годов реляционная модель начала входить в моду. Борясь с недобросовестными поставщиками СУБД, которые утверждали, что их устаревшие продукты поддерживают реляционную технологию, Кодд опубликовал «12 правил Кодда», описывающих, что должна содержать реляционная СУБД. С тех пор реляционная модель является самой распространенной на рынке баз данных.

Когда реляционные базы данных достигли зрелости, были созданы OLTP-системы, автоматизирующие многие аспекты работы компаний. В дальнейшем информация, хранившаяся в OLTP (Online Transaction Processing – система обработки транзакций в реальном времени) и других информационных системах, стала использоваться для получения данных о реальном состоянии дел компании. Уже в конце 80-х и начале 90-х годов в результате попыток прогнозировать положение дел компании в ближайшем будущем возникают хранилища данных (ХД).

С появлением технологий хранилищ данных у крупных компаний появилась возможность консолидировать огромные объемы информации для последующего углубленного анализа. Именно ХД в условиях современного рынка становятся одним из важнейших средств формирования конкурентных преимуществ компании, повышения качества работы и эффективности оказываемых услуг.

Хранилище данных позволяет использовать углубленный анализ данных, или data mining («добывание данных»), – процесс выявления тенденций, трендов, взаимных корреляций данных, находящихся в ХД, с использованием средств искусственного интеллекта или статистического анализа. С помощью этой технологии обнаруживаются скрытые зависимости и тенденции, незаметные на первый взгляд. Но для работы алгоритмов метода необходимы большие объемы информации, в противном случае обнаруженные зависимости могут оказаться неверными.

При проектировании OLTP-систем используется техника диаграмм связи сущностей (E-R, entity relationship). Для оптимизации производительности используется нормализация данных. Нормализация предназначена для приведения структуры базы данных к виду, обеспечивающему минимальную избыточность. Наличие единственной копии данных позволяет избежать аномалий при обновлении, а также ускорить обработку запросов.

Основные различия в архитектурах построения OLTP-систем и ХД следующие:

1. Степень детализации хранимых данных – типичный запрос в OLTP-системе, как правило, выборочно затрагивает отдельные записи в таблицах, в то время как в системах ХД обратная ситуация: требуется выполнять запросы над большими количествами данных с применением агрегаций и группировок.

2. Качество данных в OLTP-системе практически всегда ниже, чем это требуется для хранилищ. Присутствует «человеческий фактор», т.к. при вводе оператором информации вероятность ошибки крайне велика, в дальнейшем это может привести к анализу противоречивой информации.

3. Количество обрабатываемых OLTP-системами данных существенно меньше, т.к. обрабатывается текущее состояние каких-либо параметров, например складских остатков. Хранилище данных аккумулирует в себе результаты многих лет деятельности компании по различным аспектам.

4. Характер запросов к данным также имеет свои отличительные особенности. В OLTP-системах чаще всего не предусмотрены нерегламентированные запросы, в то время как в ХД они являются допустимыми.

5. Характер вычислительной нагрузки на систему ХД изменяется в течение суток. Ночью информация из систем-источников извлекается, трансформируется и переносится в хранилище, а также создаются отчеты для аналитиков. Во время рабочего дня аналитики работают с отчетами и запускают нерегламентированные запросы. Нагрузка на сервер ХД во время создания отчетов и выполнения нерегламентированных запросов велика. Сервер OLTP-системы чаще всего загружен

равномерно, нагрузка пропорциональна количеству работающих пользователей.

6. Основное требование к OLTP-системам – обеспечить выполнение операций модификации над БД в реальном времени. Промедление в обработке банковских транзакций, например задержка в обслуживании банковских карточек клиентов, может поставить под сомнение репутацию банка.

Один из основоположников концепции хранилищ данных Ральф Кимбалл дал определение термину ХД: «Хранилище данных – программный комплекс, предназначенный для извлечения, очистки, проверки и загрузки данных из источников в базу данных с многомерной структурой, а также предоставляющий средства извлечения и анализа содержащейся в хранилище информации с целью помощи в принятии решений» [1].

Многомерная структура хранения данных может быть реализована с помощью многомерных БД или в системе управления реляционными БД с использованием схемы «звезда» (рис.1) или схемы «снежинка» (рис.2).

Модель данных «звезда» состоит из двух типов таблиц: одной таблицы фактов — центр «звезды» — и нескольких таблиц измерений по числу измерений в модели данных — лучи «звезды».

Так же, как и в схеме звезды, схема снежинки представлена централизованной таблицей фактов, соединенной с таблицами измерений. Отличием является то, что здесь таблицы измерений нормализованы с рядом других связанных измерительных таблиц.

Технология многомерных баз данных — ключевой фактор интерактивного анализа больших массивов данных с целью поддержки принятия

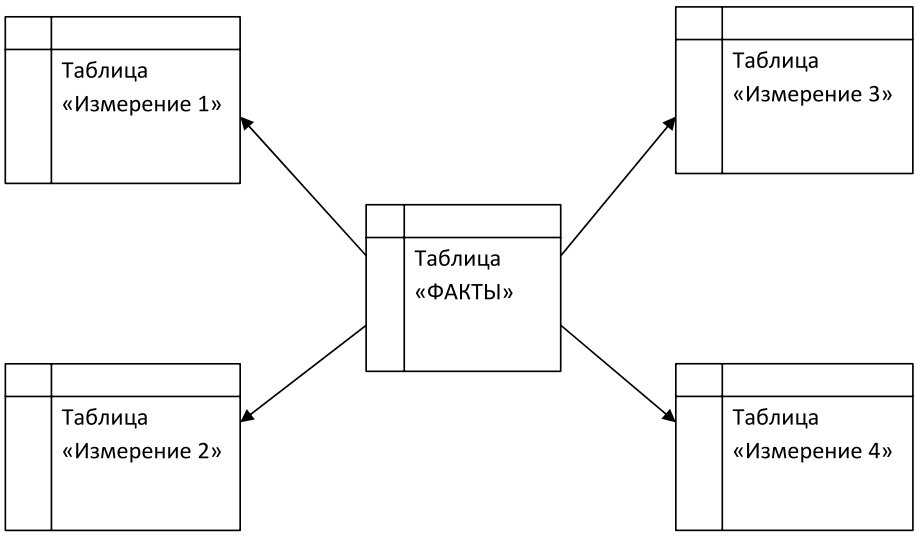

Рис. 1. Схема «звезда»

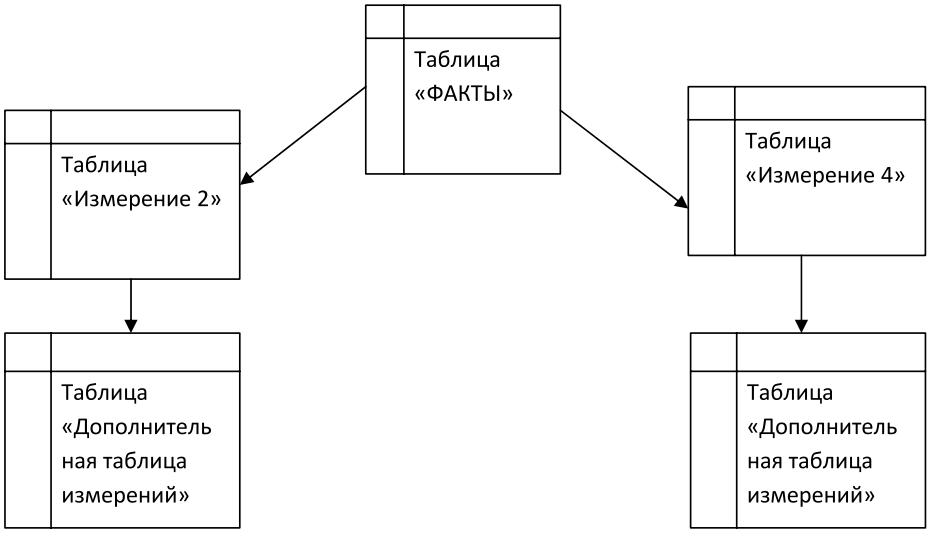

Рис. 2. Схема «снежинка»

решения. Подобные базы данных трактуют данные как многомерные кубы, что очень удобно именно для их анализа.

Многомерные модели рассматривают данные либо как факты с соответствующими численными параметрами, либо как текстовые измерения, которые характеризуют эти факты.

В процессе исследований в области ХД было обнаружено, что те же самые многочисленные преимущества хранилищ данных можно свести к подразделению или отрасли бизнеса и решению конкретной деловой задачи.

В 1991 году компания Forrester Research предложила концепцию витрин данных – тематических БД, содержащих информацию, относящуюся к отдельным аспектам деятельности организации.

Среди достоинств концепции можно перечислить следующие:

1) целевая БД максимально приближена к конечному пользователю;

2) для работы витрины данных не требуется больших вычислительных ресурсов;

3) пользователям доступны только необходимые данные;

4) пользователям доступны заранее агрегированные данные;

В качестве недостатков витрин данных можно назвать проблемы с контролем целостности и непротиворечивости хранимых в них данных.

Следующим этапов развития концепций хранилищ и витрин данных стало использование ХД в качестве единого интегрированного источника данных для витрин данных (рис. 3).

Сегодня такое решение стало стандартным способом организации доступа аналитиков компании к ХД.

В настоящее время можно выделить семь компаний – лидеров рынка, которые предлагают программные и программно-аппаратные комплексы для создания хранилищ данных: IBM, Microsoft, Netezza, Oracle, SAP, Sybase и Teradata.

Компании IBM, Microsoft, Oracle выпустили множество новых версий своих СУБД, добавляя все новые и новые опции, которые улучшают функциональность и увеличивают производительность.

Разработаны системы резервирования информации, новые способы сжатия данных, однако ни одна из систем не подверглась полному перепроектированию после ее исходного изготовления. Проблема рынка реляционных СУБД на данный момент заключается в том, что увеличивать производительность систем возможно наращиванием производительности серверов. За время существования реляционной концепции объемы хранимой информации увеличились на несколько порядков, да и мощность серверов невозможно увеличивать теми же темпами.

Можно выделить следующие известные способы оптимизации работы систем анализа данных:

1. Аппаратные способы повышения производительности.

Данный способ подразумевает использование наиболее современных процессоров, скоростных накопителей, больших объемов оперативной памяти, «быстрых» сетевых интерфейсов. Но такого роста тактовой частоты процессоров, как в 80-х и 90-х годах, уже не предвидится.

Радикальным способом решения проблемы аппаратного способа повышения производительности остается переход на другую аппаратную платформу. Например, с единичного сервера на

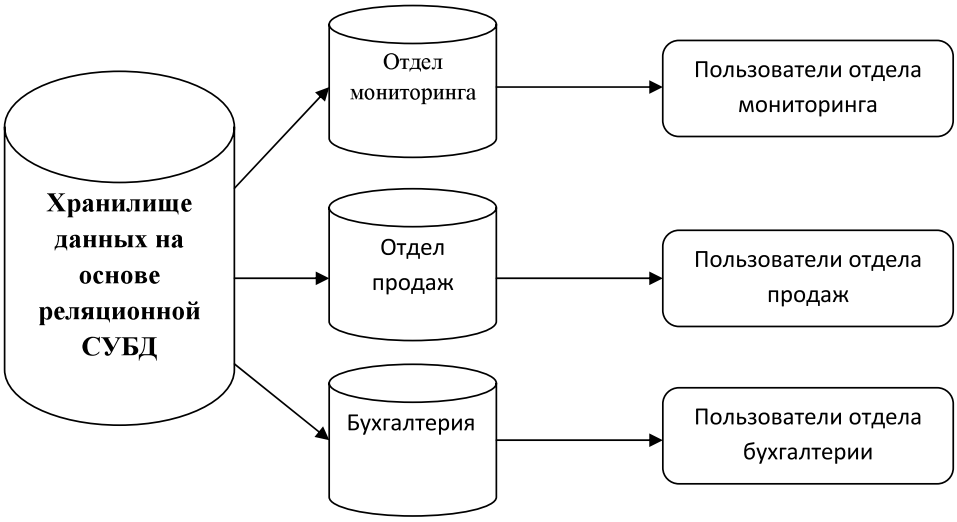

Рис. 3. Применение витрин данных совместно с ХД

распределенный кластер серверов. Производительность таких систем можно увеличивать практически линейно.

Огромную роль, хоть и косвенную, играет надежность и отказоустойчивость. При сбое в системе на какой-либо промежуток времени производительность системы чаще всего снижается, а иногда доступ к системе ограничивают.

2. Программные способы повышения производительности.

Появились новые структуры хранения данных и индексов, внедряется поддержка объектно-ориентированного подхода, используется многопоточность и распределение вычислений для сокращения времени обработки запроса.

В настоящее время большинство производителей СУБД используют метод организации хранения данных по строкам, где атрибуты записи хранятся на дисковом массиве последовательно. Этот подход удобен для OLTP-систем, т.к. запись информации на диск сервера происходит достаточно быстро. Но основная операция при работе с ХД – чтение, и традиционным способом ускорения получения отклика от СУБД является сжатие данных, что помогает снизить объем считываемых данных ценою увеличении нагрузки на процессоры сервера, а также использование кэширования, индексов и заранее подготовленных агрегированных данных.

Одним из кардинальных способов оптимизации работы ХД может стать СУБД с методом организации хранения данных по столбцам. Данный подход подразумевает отдельное хранение значений колонок таблицы, что может быть намного эффективней. СУБД считывает данные исключительно о тех столбцах, которые фигурируют в запросе. Например, при запросе к БД select поле1 from таблица where поле2=значение традиционная база данных будет последовательно считывать все строки таблицы или индекса и сравнивать значение колонки «поле2» с нужным значением. В случае с хранением данных по колонкам запрос затронет только данные о колонке «поле2» и на основании результатов поиска выберет данные «поля1».

В исследованиях производительности архитектуры колоночных СУБД приводятся данные о минимум 50-процентном увеличении производительности в обработке больших объемов данных.

Дополнительным преимуществом хранения данных по столбцам является высокая компрессия данных, т.к. данные в колонках чаще всего однотипны и нередко повторяются.

3. Административные способы повышения производительности.

Данные способы подразумевают сбор от всех возможных компонентов системы информации о производительности с последующим анализом. Целью является нахождение «узких» мест в системе, которые тратят вычислительные ресурсы неэффективно. В ряде случаев необходимо распределить нагрузку на ХД во времени. Чаще всего используют доступ по расписанию. Также возможен перенос нагрузки некоторых групп пользователей на другой сервер, например с помощью концепции витрин данных. Квалификация пользователей системы также крайне важна, т.к. за счет построения оптимальных запросов к системе нагрузка на нее остается в норме.

4. Архитектурные способы повышения производительности.

Ключевое значение для производительности и пропускной способности всей системы имеет правильная архитектура, которая состоит из элементов системы. При правильно построенной архитектуре данные максимально быстро попадают из систем, в которых они создаются, к конечным пользователям. Оптимизация хранилища данных с точки зрения архитектуры может включать перенос нагрузки из узких мест в более производительные.

Выводы

Системы хранилищ данных на данный момент являются неотъемлемой частью крупных организаций. Реляционная технология построения хранилищ существует уже более 30 лет без изменений. При экспоненциальном росте объемов данных число администраторов БД и иных технических специалистов остается практически неизменным, а потребности в вычислительных мощностях серверов крайне велики. Существует множество способов оптимизации производительности системы. Одной из перспективных является технология хранения данных не по записям, а по колонкам, т.к. она показывает очень хорошие результаты в применении к хранилищам данных. Технология должна быть совершенно прозрачна для конечных пользователей и для приложений. Можно утверждать, что хранение данных по строкам выгодно, когда запрос использует большую часть информации строки, а колоночное хранение предпочтительнее, когда данных много, но выборка производится не по всем столбцам.

Библиографический список

1. *Daniel J. Abadi*, ColumnStores vs. RowStores:How Different Are They Really. – Yale University, 2008.

2. *Хоббс Л., Холсон С., Лоуенд Ш.*, Oracle 9iR2 Разработка и эксплуатация хранилищ баз данных. – М.: КУДИЦ-ОБРАЗ, 2004.

3. Интернет – ресурс: www.oracle.ru

4. Интернет – ресурс: www.sql.ru

5. Интернет – ресурс: as-ti.narod.ru

6. *W.H. Inmon*, Building the Data Warehouse, Fourth Edition. – Wiley Publishing, Inc. 2005.

7. *V. Rainardi*, Building a Data Warehouse, With Examples in SQL Server. – APRESS, 2008.

8. *D.T. LAROSE*, DATA MININGMETHODS ANDMODEL, – New Jersey, John Wiley & Sons, Inc., Hoboken, 2006.

9. *J. Han, M. Kamber*, Data Mining: Concepts and Techniques.- Morgan Kaufmann Publishers, 2000.

10. *S. Urman*, Программирование на языке PL/SQL. – М:. Издательство «ЛОРИ», 2002.

Московский авиационный институт
(государственный технический университет)