

Лекция 18. Будущее SQL.

Язык SQL и основанные на нем реляционные базы данных находятся среди важнейших технологий современной информационной индустрии. Со времени своего первого появления, около тридцати лет назад, SQL прошел большой путь и стал стандартным языком для работы с базами данных. В первое десятилетие своей истории, благодаря поддержке со стороны компании IBM, производителей СУБД и международных органов стандартизации, он стал стандартом управления данными в корпоративной среде. Во втором десятилетии влияние SQL расширилось на рынок персональных компьютеров, а также на новые сегменты рынка, такие как хранилища данных. Третье десятилетие SQL сделало его основой управления данными в интернет-вычислениях и создало новые многомиллиардные ниши рынка с его участием типа интеллектуальных ресурсов предприятия. Приведенные ниже факты ясно свидетельствуют о большом значении SQL.

- Компания Oracle, второй по величине производитель программного обеспечения, была основана на волне популярности реляционной модели управления данными, и SQL поддерживается во всех ее флагманских продуктах и приложениях уровня предприятия.

- Компания IBM, крупнейший производитель на компьютерном рынке, предлагает реляционную СУБД DB2 для всех линий своих продуктов, а также для конкурирующих платформ.

- Компания Microsoft, крупнейший производитель программного обеспечения, рассматривает свою СУБД SQL Server в качестве центрального звена в стратегии завоевания рынка, а также в качестве средства управления данными для всех своих служб и приложений.

- Компания Sun Microsystems увидела в SQL достаточную ценность для того, чтобы купить MySQL AG, тем самым приобретя наиболее популярную в мире РСУБД с открытым кодом MySQL, которая обслуживает огромное количество веб-сайтов в Интернете.

- Потенциальный конкурент SQL и реляционной модели, язык XML, был внедрен во многие SQL-продукты для управления данными, что только увеличило доминирование SQL на рынке.

- Большинство прикладных корпоративных систем (ПО планирования и учета, финансовые, маркетинговые и торговые приложения) основано на реляционных базах данных.

- SQL является стандартом для специализированных баз данных для мобильных устройств и ноутбуков, а также для встраиваемых приложений в области телекоммуникаций и производственных систем.

- Доступ к базам данных на основе SQL представляет собой неотъемлемую часть серверов приложений в Интернете, а SQL-базы данных используются всеми крупными сайтами электронной коммерции, от Amazon до eBay.

Насыщение рынка корпоративных баз данных

Реляционная технология составляет ядро корпоративных систем обработки данных, и реляционные базы данных используются практически во всех крупных компаниях. Поскольку корпоративные базы данных чрезвычайно важны и многие из них эксплуатируются уже в течение многих лет, большинство компаний выбрали какую-то одну СУБД в качестве собственного корпоративного стандарта. После того как такой стандарт формально установился в компании, переход на другую СУБД становится проблематичным. Хотя конкурирующие СУБД могут работать лучше в некоторых ситуациях или предлагать новые полезные возможности, достаточно заявления поставщика "стандартной" СУБД о том, что эти возможности будут включены в следующую версию продукта, и потери клиентов не произойдет.

Все это привело к усилению позиций ведущих поставщиков СУБД. Наличие служб прямой доставки и технической поддержки, а также многолетние соглашения о сотрудничестве теперь играют такую же, если не более важную, роль, что и технологические преимущества. В связи с этим лидеры рынка СУБД стремятся наращивать свой бизнес в рамках инсталлированной базы собственных продуктов, а не пытаться привлечь клиентов своих конкурентов.

Важным фактором является тенденция к консолидации рынка. В общих чертах этот процесс можно описать следующим образом. Первопроходцами новой технологии становятся, как правило, начинающие компании, которые растут за счет продажи своих продуктов приверженцам всего нового. Последние помогают сформировать технологию и определить ниши рынка, в которых она может приносить прибыль. Через несколько лет, когда преимущества новой технологии становятся очевидными, компании-новаторы поглощаются крупными поставщиками СУБД, которые распространяют технологию на инсталлированную базу своих продуктов. В первой половине 90-х годов описанную схему можно было наблюдать в отношении поставщиков СУБД и разработчиков различных утилит и инструментов для баз данных. Во второй половине 90-х годов поглощаться начали разработчики специализированных баз данных, таких как хранилища данных или объектно-реляционные базы данных. В середине 2000-х годов Oracle и IBM приобрели производителей баз данных в памяти, а Sun приобрела MySQL AG, крупнейшего производителя базы данных с открытым кодом. Следующей мишенью должны стать специализированные хранилища данных.

Сегментация рынка СУБД

Несмотря на насыщение отдельных сегментов рынка (особенно в сфере корпоративных систем), постоянно формируются его новые сегменты и ниши. Традиционно рынок делился на базы данных для мэйнфреймов (где доминировала IBM), баз данных для информационных центров (Oracle) • и баз данных для рабочих групп (Microsoft). Сегодня рынок более многолик и точнее сегментирован с учетом сферы применения СУБД и ее

специализированных возможностей. Новые и наиболее быстро развивающиеся сегменты рынка включают следующее:

- хранилища данных, ориентированные на управление терабайтами информации;
- интеллектуальные ресурсы предприятий и базы данных для оперативного анализа (OLAP), предназначенные для выполнения сложного анализа данных на предмет выявления скрытых тенденций развития;
- мобильные базы данных, которые предназначены для "мобильных" со-трудников, таких как продавцы на местах, консультанты и т.п. Такие мо-бильные базы данных могут связываться с центральным сервером для синхронизации с центральной базой данных;
- встраиваемые базы данных, работающие в закрытых системах, таких как производственное оборудование, системы управления автомобилями или сетевое оборудование. Такие базы данных нетребовательны к ресурсам и требуют минимального администрирования (или обходятся без него);
- базы данных в памяти и кеши баз данных, разработанные приложений со сверхвысокими требованиями ко времени отклика;
- базы данных для истории посещений и записи активности миллионов пользователей веб-сайтов, позволяющие оптимизировать веб-сайты на основе анализа реального поведения пользователей;
- кластеризованные базы данных, использующие преимущества параллельной работы мощных дешевых серверов для повышения масштабируемости и надежности;
- потоковые базы данных, работающие с передаваемыми по сети динамическими данными, такими как сетевой трафик, биржевые котировки или банковские транзакции.

Пакеты корпоративных приложений

В начале эры SQL большинство корпоративных приложений для повседневных нужд разрабатывалось собственными силами информационных служб предприятий. Сегодня большинство компаний отказались от стратегии "сделай" в пользу стратегии "купи" в отношении большинства корпоративных приложений, таких как системы планирования ресурсов предприятий (Enterprise Resource Planning — ERP), системы управления поставками (Supply Chain Management — SCM), системы управления человеческими ресурсами (Human Resource Management — HRM), системы автоматизации торговых операций (Sales Force Automation — SFA), системы обслуживания клиентов и др. Все эти системы в настоящее время поставляются в виде комплексных программных пакетов, обеспеченных обслуживанием, технической поддержкой и консультациями. И во всех них используются реляционные базы данных.

Появление комплексных корпоративных систем оказало существенное влияние на развитие рынка баз данных. Большинство подобных систем поддерживало СУБД лишь двух или трех ведущих поставщиков. Это привело к еще большему усилению позиций ведущих СУБД и усложнило жизнь

начинающим компаниям. Другим следствием стало снижение цен на СУБД, которые теперь стали чаще рассматриваться как составная часть более крупного программного пакета, а не как отдельный компонент. Например, если клиент планирует установить в своей компании некоторое программное обеспечение, скажем, для выполнения структурного анализа, то он вынужден приобретать и базу данных, причем она должна быть из ограниченного списка тех баз данных, которые поддерживаются приобретаемым программным обеспечением. В результате доминирующие позиции главных производителей баз данных на рынке только укрепляются, а возможность выхода на рынок для новичков становится все более призрачной. В результате складывается тенденция лидирования на рынке больших корпоративных баз данных IBM и Oracle и сдерживания принятия SQL Server, Sybase и MySQL корпоративными центрами данных. Сами СУБД все больше рассматриваются не как отдельное стратегическое решение, а как компонент приложения, приобретаемого для решения тех или иных корпоративных задач.

В ответ на эти тенденции и насыщение рынка Oracle стремится занять позиции лидера не только в области разработчика баз данных, но и производителя корпоративных приложений. Сначала компания Oracle разрабатывала собственные корпоративные приложения, которые имели на рынке ограниченный успех. Изменив стратегию, в последнее десятилетие Oracle строит свой бизнес на затратах десятков миллиардов долларов на приобретение компаний по разработке программного обеспечения (и их клиентов). IBM проявляет такую же активность, но сторонится многих важных категорий программного обеспечения.

Взаимосвязи между корпоративными приложениями и корпоративными базами данных все еще присутствуют на рынке. Oracle рекламирует преимущества принципа "все в одном" для всех видов программного обеспечения, постоянно обещая (но далеко не всегда выполняя обещанное) более тесную интеграцию между корпоративными приложениями в широком диапазоне. Программное обеспечение, приобретаемое Oracle, имеет тенденцию со временем поддерживать работу только с базами данных Oracle, тем самым уменьшая возможность выбора клиентами средств управления данными. IBM же приняла консультационно-сервисный нудход, в первую очередь предлагая клиентам помощь и консультации в выборе лучших из доступных продуктов. Злые языки, впрочем, утверждают, что таковыми с точки зрения IBM всегда оказываются именно ее аппаратные и программные продукты (включая базу данных DB2). SAP занимает доминирующие позиции по количеству установленных корпоративных приложений, заполняя свой портфель предложений небольшими точно выверенными покупками. Однако у SAP нет собственной базы данных уровня предприятия, так что приложения этой компании приносят Oracle и IBM сотни миллионов долларов прибыли. Будущее рынка корпоративных приложений пока что остается неясным, но уже понятно, что он будет разделен основными производителями баз данных.

Программное обеспечение в виде служб

На фоне распространения пакетов корпоративных приложений возникла еще одна тенденция — распространение корпоративных приложений через Интернет с доступом к ним при помощи веб-браузера. При использовании такой модели — программного обеспечения в виде службы (Software-as-a-Service, SaaS) — корпоративный отдел информационных технологий не устанавливает корпоративные приложения и не работает с ними в собственном центре данных. Вместо этого корпоративные приложения работают на серверах их разработчика или на серверах, управляемых им. Потенциальная выгода предприятия от такого решения заключается в сокращении штата отдела информационных технологий, снижении стоимости, быстром доступе к новым, усовершенствованным, версиям программного обеспечения, которое прозрачно для пользователей устанавливается через Интернет. Пионером в этой области стала компания Salesforce.com, которая начинала с приложений для автоматизации процесса продаж и чей годовой доход вырос до миллиарда долларов. Со временем и другие производители воспользовались этой моделью ДЛЯ распространения офисного программного обеспечения, а также приложений ERP, CRM, HR.

Модель SaaS потенциально в состоянии серьезно повлиять на рынок корпоративных баз данных. При работе с такого рода приложениями конкретная база данных в его основе оказывается невидимой для клиента. Это может быть Oracle,

IBM DB2, SQL Server, MySQL или какая-то иная база данных — до тех пор, пока все корректно работает, пользователя этот вопрос не должен интересовать. Так что распространение модели SaaS грозит тем, что покупка баз данных перестанет быть задачей информационных отделов корпораций и сконцентрируется в руках про-изводителей SaaS-приложений.

На момент написания книги SaaS оставалась небольшой частью корпоративных информационных технологий. Она пока что не распространилась далее определенных видов приложений типа приложений для автоматизации процесса продаж, но уже доказала свою популярность среди мелкого и среднего бизнеса, которому не по карману собственные информационные службы. Экономические выгоды модели SaaS неоспоримы, и компания Salesforce.com храбро заявила, что ее стратегия состоит в том, чтобы покончить с корпоративными приложениями. Если SaaS продолжит свое распространение и начнет проникать в финансовые системы и ERP, производителям основных баз данных придется прикладывать усилия для защиты уже установленных баз и своего места под рыночным солнцем.

Повышение производительности аппаратного обеспечения

Одной из главных причин роста популярности SQL было резкое увеличение производительности реляционных СУБД. Частично этот рост произошел благодаря улучшению технологии разработки баз данных и оптимизации запросов. Однако, в основном, увеличение производительности СУБД явилось результатом общего повышения быстродействия аппаратных средств и изменений в программном обеспечении, которые смогли в полной мере использовать аппаратные новшества. Повышение производительности

мэйнфреймов сопровождалось аналогичными процессами на рынке серверов Unix и Windows, где вычислительная мощность удваивается каждый год.

Самые разительные перемены происходят с системами симметричной много-процессорной обработки (Symmetric Multiprocessing — SMP), в которых рабочая нагрузка распределяется между двумя, четырьмя, восемью и более процессорами, работающими параллельно. Многопроцессорная архитектура особенно подходит для OLTP-приложений, где нагрузкой является большое число маленьких, одновременно выполняемых транзакций в базе данных. Традиционные поставщики OLTP-систем, например компания Tandem, всегда применяли многопроцессорную архитектуру, а в крупнейших системах на базе мэйнфреймов она используется уже больше десяти лет. В начале 90-х годов SMP-системы активно вторглись на рынок серверов Unix, а в текущем десятилетии — стали нормой для серверов на базе персональных компьютеров.

Сегодня даже настольные системы на базе процессоров Intel и AMD оснащены двух- или четырехъядерными процессорами, мощность которых близка к мощности двух или четырех отдельных процессоров. Блейд-серверы развили это направление, сделав высокоэкономичными серверы с десятками процессоров. Принятие на вооружение 64-битовой аппаратной архитектуры и соответствующих операционных систем позволяет поднять объем памяти серверов до десятков и даже сотен 1 гигабайт. Серверы, конкурирующие по мощности с мэйнфреймами, в настоящее время СТОЯТ ДО 100000 долл.

Многоядерные и многопроцессорные системы дают преимущества и для систем поддержки принятия решений и приложений для анализа данных. Производители СУБД ведут интенсивные исследования в области распараллеливания обработки запросов, когда один сложный SQL-Запрос разбивается на параллельно выполняемые фрагменты кода. При использовании таких технологий запрос, который требовал двух часов работы в однопроцессорной системе, может быть выполнен буквально за несколько минут. Однако еще более часто увеличение мощности процессоров используется не просто для ускорения вычислений, а выполнения более сложного и интеллектуального анализа, который был недоступен ранее.

Сегодня пока что нет никаких признаков остановки роста производительности баз данных, достигаемого, в первую очередь, за счет оптимизации СУБД для работы в многопроцессорных и многоядерных системах. В прошлом такой рост достигался, в первую очередь, за счет повышения производительности аппаратного обеспечения. В будущем, вероятно, необходимость повышения производительности баз данных будет оказывать еще большее влияние на развитие нового аппаратного обеспечения, процессоров и серверов.

Специализированные серверы баз данных

История SQL и реляционных баз данных — это еще и история серверов баз данных. Для построения высокоэффективных систем производители комбинировали высокопроизводительные процессоры, быстрые диски, предустановленное программное обеспечение, чтобы в результате получить

сервер, который достаточно просто включить в сеть для работы. Производители серверов баз данных обычно доказывают, что при помощи специально спроектированной системы можно достичь существенно более высокой производительности базы данных, чем при применении компьютерной системы общего назначения. В некоторых случаях такие системы включают специализированные интегральные микросхемы (application-specific integrated circuits, ASIC), которые реализуют некоторую логику СУБД на аппаратном уровне, достигая максимально возможной производительности. Специализированные системы от таких компаний, как Teradata, Sharebase (бывш. Britton-Lee) и Netezza, нашли применение в приложениях, включающих сложные запросы к очень большим базам данных (в настоящее время на рынке осталась только компания Teradata).

Идея специализированного сервера баз данных была вновь рассмотрена в конце 1990-х годов компанией Oracle Corporation и ее исполнительным директором Ларри Эллисоном (Larry Ellison). Эллисон утверждал, что эра Интернета доказала успех продуктов "все в одном", таких как сетевое оборудование или кеширующие веб-серверы. Oracle анонсировала сотрудничество с рядом производителей серверного аппаратного обеспечения по созданию специализированного сервера для баз данных Oracle. Однако все их усилия практически не отразились на состоянии рынка, так что постепенно эта работа сошла на нет.

Не так давно к этой идее вновь вернулись некоторые вновь созданные компании — в форме кеширующих серверов баз данных, которые располагаются в сети между ■ 1 приложением и корпоративной базой данных. В такой конфигурации критична абсолютная прозрачность кеша, и появление исключительно популярной

базы данных с открытым кодом MySQL помогло обеспечить эту прозрачность. Многие серверы баз данных работают под управлением MySQL, так что приложение, обращающееся к базе данных, не в состоянии выяснить, работает ли оно со специализированным сервером или с MySQL, запущенной на обычном компьютере. Oracle также возобновила свою работу над специализированным сервером, анонсировав высокопроизводительную базу данных Oracle на аппаратном обеспечении от Hewlett-Packard.

Стандартизация SQL

Принятие официального стандарта ANSI/ISO для языка SQL было одним из главных факторов, благодаря которым SQL в 1980-е годы стал стандартным языком реляционных баз данных. Соответствие стандарту ANSI/ISO стало отправной точкой для оценки реляционных СУБД, поэтому каждый поставщик утверждал, что его программный продукт совместим со стандартом ANSI/ISO. В течение последующих 20 лет все популярные СУБД в действительности стали поддерживать стандарт SQL, по крайней мере, широко употребляемые его части. Другие части, такие как язык модулей, чаще игнорировались, чем реализовывались. Так постепенно популярные СУБД пришли к поддержке базовых возможностей SQL, так что ядро языка SQL в разных СУБД стало практически одинаковым.

Как уже говорилось в главе 3, "Перспективы SQL", исходный стандарт SQL был довольно ограниченным, в нем имелось много пробелов, а также немало областей, оставленных на усмотрение разработчиков СУБД. Несколько лет комитет по стандартизации работал над стандартом SQL2, который устранял эти недостатки и расширял язык SQL. В отличие от первого стандарта, в котором описывались элементы языка, уже имевшиеся в большинстве СУБД, стандарт SQL2 на момент своей публикации в 1992 году был попыткой опередить разработчиков СУБД, а не следовать за ними. В этом стандарте описывались функции и особенности языка, которые еще не использовались широко в современных СУБД. Некоторые из этих возможностей, такие как расширенные возможности соединения и широкое использование подзапросов, сегодня широко реализованы; другие же так и остаются не принятыми большинством производителей.

То же самое произошло и при принятии последующих версий стандарта SQL как в 1999, 2003 годах, так и в последние годы. Стандарт существенно вырос в размерах (более чем в три раза со времени SQL2) и был разделен на десяток частей. В новых разделах, таких как встраивание XML, продолжается состязание между разработками производителей СУБД, которые надеются получить таким образом конкурентные преимущества, и стандартом, обеспечивающим переносимость.

Вероятное будущее развитие стандарта SQL представляется экстраполяцией пути, по которому он следовал последние годы. Ядро языка останется высоко стандартизированным. Все больше возможностей постепенно станут частью этого ядра и будут определены как дополнения к языку (либо для них будут разработаны собственные стандарты). Производители СУБД продолжают добавлять к своим реализациям SQL новые возможности в попытках выделить свои разработки и ответить на новые требования, выдвигаемые рынком; со временем те новые возможности, которые станут наиболее популярны, будут внесены в стандарт.

SQL в следующем Аесятилетии

Вряд ли сегодня кто-нибудь возьмется предсказать развитие рынка баз данных и SQL в следующие пять-десять лет. Каждая новая технология за последние три десятилетия оказывала существенное влияние на управление данными и SQL. Одним из примеров может быть появление персональных компьютеров и вызванная ими эра "клиент/сервер". Позже появление Интернета и браузерной архитектуры привело к новым методам управления данными — веб-службам. Если заглянуть в будущее, то можно предположить, что Интернет станет поистине вездесущим, связывающим буквально все электронные устройства. Пожалуй, грядущее развитие Интернета окажет на архитектуры управления данными большее влияние, чем само его появление. Тем не менее вполне можно сделать несколько предсказаний о грядущих тенденциях в эволюции управления базами данных (они будут рассмотрены в последнем разделе данной главы).

Распределенные базы данных

В наши дни все большее распространение получают корпоративные приложения и даже еще более масштабные системы. Но одной централизованной базе данных трудно поддерживать десятки крупных приложений и тысячи параллельно работающих пользователей. Поэтому массивные корпоративные базы данных становятся все более распределенными; они разделяются на меньшие базы данных, обслуживающие отдельные приложения и информационные потребности корпорации. Чтобы соответствовать растущим требованиям корпоративных и интернет-приложений, данные должны быть распределенными, но в то же время СУБД должна обеспечивать их целостность и координирование, чтобы гарантировать надежность и эффективность деловых операций.

Еще одной тенденцией, требующей отказа от централизованной архитектуры баз данных, является продолжающееся распространение портативных персональных компьютеров и других мобильных информационных устройств. устройства по

своей природе предназначены работы в распределенных сетях, причем в режиме непостоянного подключения — иногда они работают автономно, иногда подключенными к сети, и это подключение может выполняться посредством как проводной, так и беспроводной связи. Базы данных, составляющие ядро мобильных приложений, должны быть способны эффективно работать в таком непостоянном режиме.

Все эти тенденции требуют распределения, интеграции и синхронизации баз данных, а также разработки эффективных технологий управления распределенными данными. Модель "все в одном" больше не подходит для распределенных сред, работающих по принципу "всегда и везде". Вмсто этого одни транзакции требуют абсолютной синхронизации с центральной управляющей базой данных, тогда как другим необходима поддержка долго выполняемых операций, когда синхронизация может произойти по прошествии часов и даже дней. Без эффективных специализированных технологий и программных продуктов построение таких распределенных систем становится кошмаром администраторов баз данных. Разработка подобных технологий и будет одной из важнейших задач производителей СУБД в следующем десятилетии и одним из главных источников их будущих доходов.

Массивные хранилища данных для оптимизации бизнеса

Последние несколько лет показали, что компании, интенсивно использующие технологии баз данных, получают огромные преимущества по сравнению со своими более инертными конкурентами. Например, компания Wal-Mart обязана своим успехом именно использованию информационных технологий для каждодневного учета складских и коммерческих операций. Это позволило компании минимизировать объемы складских запасов и эффективнее организовать взаимодействие с клиентами. Технологии накопления и последующего анализа данных позволяют компаниям выявлять скрытые тенденции и взаимосвязи — стоит только вспомнить легендарное открытие одного розничного торговца, обнаружившего, что продажа детских

пеленок поздней ночью напрямую зависит от количества проданного накануне пива.

Поэтому очевидно, что компании будут продолжать накапливать всевозможную информацию о клиентах, продажах, складских операциях, ценах и других бизнес-факторах. Интернет создает новые возможности для сбора этого вида информации. Действия клиента на веб-сайте компании предоставляют информацию о его желаниях, потребностях и поведении. Информация о взаимодействии клиентов с активно посещаемым сайтом компании ежедневно дает гигабайты данных. Базы данных для управления таким огромным количеством данных должны быть способны быстро импортировать новые данные в огромном количестве и быстро выделять их подмножества для анализа. Они должны быть достаточно масштабируемыми, чтобы справиться с увеличением потока данных на порядки в течение года. Еще одна новая тенденция заключается в хранении таких баз данных в Интернете с использованием хранилищ и вычислительных мощностей компаний наподобие Amazon или Google. Чтобы справляться с огромными объемами данных за разумную цену, хранилища данных должны основываться на массовом применении дешевого широко распространенного аппаратного обеспечения.

Сверхпроизводительные базы данных

Внедрение интернет-ориентированных архитектур предъявляет ко всей инфра-структуре обработки данных предприятия невиданно высокие требования, ведь рабочая нагрузка на СУБД, генерируемая интернет-приложениями, просто не сравнима с той обычной нагрузкой, которую создавали корпоративные приложения еще несколько лет назад. Когда СУБД поддерживала приложения внутреннего пользования, с которыми одновременно работало несколько десятков сотрудников, проблемы производительности могли, конечно, вызывать раздражение у этих сотрудников, но это никак не сказывалось на клиентах компании. С появлением компьютерных центров технической поддержки качество обслуживания клиентов стало зависеть от эффективного управления данными, но приложения по-прежнему могли поддерживать максимум несколько сотен одновременно работающих пользователей (сотрудников, отвечающих на телефонные звонки клиентов).

С появлением Интернета связь между клиентом и базой данных компании становится и вовсе непосредственной. Недостаточная производительность системы проявляется в увеличении времени ответа на запрос клиента, а недоступность ба

зы данных — в потере заказов. Более того, базы данных больше не защищены от резких повышений интенсивности транзакций. Если финансовая компания предлагает услуги по интерактивному заключению сделок и выполнению операций с ценными бумагами, она должна быть готова к пиковым нагрузкам в дни сильных колебаний на фондовой бирже, которые могут в десятки и сотни раз превышать обычную каждодневную нагрузку системы. Подобным же образом компания, занимающаяся электронной

коммерцией, должна быть готова к сезонным изменениям активности покупателей, и под Новый год ее система должна так же хорошо справляться с нагрузкой, как и в середине марта.

Благодаря электронной коммерции и возможности доступа к информации через Интернет в реальном масштабе времени большинство популярных интернет-служб уже сейчас работает на порядок производительнее, чем самые быстрые реляционные СУБД. Чтобы справиться со столь высокими требованиями к производительности, компании активно внедряют распределенные архитектуры и технологии репликации баз данных. Они переносят активные данные поближе к пользователю, интенсивно используя кеширование всего, что только может помочь ускорить взаимодействие. Для обеспечения высокоскоростного доступа к базам данных их содержимое помещают прямо в оперативную память. В первую очередь все это относится к большим веб-сайтам с огромным количеством информации, но по мере того, как кеширование веб-страниц становится все более распространенным и превращается в основу для высокопроизводительной работы, точно так же и кеширование активных данных становится господствующей архитектурой управления растущими объемами данных в Интернете.

Одной из технологий, которая потенциально может справиться с требованиями к производительности, является применение недорогих твердотельных запоминающих устройств, способных заменить дисковые устройства, которые в настоящее время являются основным средством хранения информации. Хотя диски из года в год становятся все дешевле, их производительность растет не так быстро, поскольку зависит от электромеханических факторов. Твердотельная память дешевеет точно так же, как и дисковая, и широко применяется во множестве устройств, таких как МР3-плееры, смартфоны и др. Мы вступаем в эпоху твердотельных устройств для баз данных, в которых не будет ни одной движущейся части, — и эпоха эта начнется уже в ближайшие три-пять лет.

Интеграция Интернета и сетевых служб

В эпоху Интернета управление данными все в большей степени превращается в сетевую службу, которая должна быть тесно интегрирована с другими службами, такими как служба сообщений, транзакций или управление сетью. В некоторых из этих областей давно имеются стандарты, такие, например, как стандарт для управления распределенными транзакциями. В других областях стандарты находятся на завершающей стадии разработки, такие как стандарт SOAP для пересылки XML-данных посредством интернет-протокола HTTP и стандарт UDDI (Universal Description, Discovery, and Integration — универсальная система предметного описания и интеграции) для поиска служб в распределенном сетевом окружении. Пока что остается открытым вопрос создания эффективных стандартов

управления распределенными сетями служб и определения и поддержки уровней служб в этих сетях.

Многоуровневая архитектура также ставит новые вопросы о ролях, которые должны играть менеджер базы данных и другие компоненты информационных систем. Например, при рассмотрении сетевых транзакций с точки зрения распределенных баз данных решением может явля протокол двухфазного принятия транзакции, реализованный производителем СУБД. Когда сетевые транзакции включают комбинацию устаревших приложений, обновлений реляционных баз данных и коммуникации между приложениями, проблема управления транзакциями выходит за рамки базы данных и требует применения некоторых внешних механизмов.

Подобные проблемы сопровождают и появление серверов приложений на базе Java в качестве платформы промежуточного уровня для бизнес-логики. До эпохи Интернета бизнес-логика встраивалась в базы данных при помощи хранимых процедур. Позже языком хранимых процедур (вместо ранних фирменных языков различных производителей) стал язык программирования Java. В настоящее время серверы приложений образуют альтернативную платформу для бизнес-логики на языке программирования Java — также внешнюю по отношению к базе данных.

Похоже, что бизнес-логика и далее будет располагаться в базах данных или на серверах приложений, в зависимости от организационных, а не технических соображений. Группа управления данными для обеспечения высокоструктурированного обращения к данным будет применять хранимые процедуры и логику в базе данных. Группа интеграции приложений, для того чтобы объединить информацию из базы данных с данными, поступающими из других производственных систем, будет использовать код Java, работающий на сервере приложений. С приобретением Oracle компании BEA все основные производители баз данных теперь создают собственные серверы приложений, так что со временем базы данных и серверы приложений могут оказаться очень тесно интегрированы. Еще одной тенденцией является рост применения веб-платформы с открытым кодом LAMP (Linux, Apache, MySQL и PHP/Python/Perl).

Встраиваемые базы данных

Реляционные технологии управления данными распространились на многие сферы компьютерной индустрии, от маленьких наладочных устройств до больших мэйнфреймов. Практически все корпоративные приложения основаны на базах данных, служащих для хранения информации и управления ею. Еще большее количество приложений пользуется малыми базами данных. Например, специализированной формой баз данных являются службы каталогов, лежащие в основе нового поколения сетевых коммуникационных служб. Высокопроизводительные базы данных малого размера стали составной частью телекоммуникационных сетей, находя применение в сотовой связи, сложных схемах оплаты, интеллектуальных службах сообщений и др.

Встраиваемые базы данных традиционно реализовывались с помощью специально написанного программного кода, интегрированного в приложение. Благодаря максимальной специализации кода этот подход

обеспечивал самую высокую производительность, но достигалось это за счет создания мало приспособленной

для модификации и с трудом управляемой архитектуры. Со снижением цен на оперативную память и высокопроизводительные процессоры стало возможным встраивание в такие приложения малых реляционных СУБД.

Преимущества поддержки стандарта встраиваемыми базами данных весьма существенны. Без серьезного ухудшения производительности приложения можно сделать более модульными, прозрачно вносить изменения в структуру базы данных и надстраивать новые модули поверх существующей базы данных. Благодаря этим преимуществам приложения со встраиваемыми базами данных представляют собой новую потенциально растущую область применения SQL и технологий реляционных баз данных. Как и во многих других областях информатики, окончательным триумфом SQL-баз данных может быть их растворение в структуре других программных продуктов и служб — при их отсутствии в качестве автономных приложений они все равно будут жизненно необходимы для программ и служб, их содержащих.

Интеграция с объектно-ориентированными технологиями

В будущем развитии реляционных баз данных труднее всего спрогнозировать то, как будет вестись их интеграция с объектно-ориентированными технологиями. Центр тяжести технологий разработки приложений со всей очевидностью сместился в сторону объектно-ориентированных инструментальных средств и методик. Растет популярность языков C++ и Java, которые применяются не только в клиентских приложениях, но и для реализации деловой логики на сервере. В вебразработке доминируют объектно-ориентированные языки сценариев, такие как PHP и Perl. Однако базовые принципы табличной организации данных в реляционной модели уходят корнями в эру языка COBOL с его записями и полями и не имеют ничего общего с объектами.

Производители объектно-ориентированных СУБД решили проблему несоответствия между объектной и реляционными архитектурами радикально: они вообще отказались от реляционной модели в пользу строгой объектной организации баз данных. Однако отсутствие стандартов, сложность освоения, отсутствие простых средств выполнения запросов и другие недостатки таких СУБД пока не позволили им добиться значительного успеха на рынке. Производители реляционных СУБД ответили на вызов тем, что интегрировали в реляционную модель объектные технологии, в частности, поддержку XML. Уже выполнена большая работа в этом направлении, но можно держать пари, что реляционная и объектная технологии будут интегрированы еще теснее. На этом пути¹ можно отметить следующие тенденции.

- Продолжение роста популярности интерфейсов реляционных СУБД на основе языка Java, таких как JDBC и встраиваемый SQL для Java, а также интерфейсов объектно-ориентированных языков сценариев.

- Java становится стандартным языком хранимых процедур реализации

бизнес-логики в РСУБД. Постепенно в новых приложениях Java может вытеснить процедурные языки производителей, такие как Oracle PL/SQL.

- Растет поддержка разными СУБД сложных абстрактных типов данных, реализующих такие объектно-ориентированные возможности, как ин- 920 Часть VI. SQL сегодня и завтра

капсуляция и наследование. XML обеспечивает хранение структурированных нереляционных данных, которое будет дополнено возможностью хранения потоковых данных, таких как музыка и видео.

- Растет важность интерфейсов, ориентированных на обработку сообщений, включая триггеры, генерирующие внешние сообщения для взаимодействия с другими приложениями.

- Размываются границы между системами управления содержимым, используемыми для работы с документами, и системами управления реляционными базами данных — так же как XML стирает отличие между "документом" и "записью структурированных данных".

Эволюция РСУБД в последние пару десятилетий следует единому шаблону: на сцене появляется новая технология (такая, как объекты, XML или хранилища данных), которая вызывает массу энтузиазма и волну новых компаний. Проходит несколько лет, в течение которых новые игроки держатся на гребне волны и снимают сливки. В это же время производители РСУБД находят пути интеграции новой технологии в свои продукты — пусть вначале и на очень слабом уровне. Постепенно результат интеграции становится достаточно хорош для того, чтобы все основные производители баз данных приняли на вооружение новую технологию. Нет оснований считать, что в ближайшем будущем этот шаблон претерпит какие-то резкие изменения.

Горизонтально масштабируемые базы данных

С развитием Интернета многие стандартные элементы его архитектуры сталкиваются с необходимостью обработки очень большого количества "транзакций". Веб-серверы должны уметь обрабатывать миллионы запросов в час. Серверы приложений должны выполнять бизнес-логику для десятков тысяч транзакций. Примерно такого же масштаба задачи стоят и перед сетевым оборудованием — роутерами, коммутаторами и т.п.

В каждом из описанных случаев масштабирование горизонтально — один вебсервер превращается в два, три, десяток, а потенциально — в сотни серверов, разделяющих нагрузку на уровне неб-сервера. Эти серверы используют или одно общее, или реплицированные хранилища данных, причем каждый сервер в состоянии выдать любую запрошенную страницу. Чтобы такая схема работала, сервер должен представлять собой систему без сохранения состояния. Насколько это возможно, любая информация, которая должна храниться между отдельными запросами, должна передаваться обратно клиенту, а он должен передавать ее серверу при следующем запросе. Любая информация о состоянии, которая не может быть обработана указанным образом, передается "в обратном направлении" — на сервер базы данных, лежащий в основе описанной архитектуры.

Для многих крупных веб-сайтов или неб-служб такая база данных становится узким местом. К сожалению, горизонтальное масштабирование, работающее для неб-серверов и серверов приложений, в случае серверов баз данных приводит к проблемам. В силу самой их природы базы данных не могут быть без запоминания состояния. Все отдельные элементы транзакции должны быть одновременно

либо приняты, либо отвергнуты, а параллельно выполняющиеся транзакции не должны влиять друг на друга. Для них нет "обратного направления", куда можно передать информацию о состоянии — эта информация должна поддерживаться самой базой данных.

Так мы пришли к новой важной задаче архитектуры баз данных — разработка системы, которую можно эффективно горизонтально масштабировать. За последние годы были достигнуты значительные успехи в создании баз данных, обладающих возможностью кластеризации или применения грид-технологии. Эти новые архитектуры должны позволять распределить обработку данных среди десятков (или большего количества) серверов, но информация при этом должна оставаться согласованной во всех отношениях (см. главу 12, "Обработка транзакций"). Частично решение достигается путем интеллектуального разделения различных типов данных. Ссылки и неизменяемые данные (или изменяемые редко, такие как контактная информация клиентов) могут быть горизонтально реплицированы, так как поддержка их согласованности на десятках серверов не требует особых усилий. Изменяющиеся данные хранятся в одной системе кластера, и все запросы к таким данным автоматически перенаправляются в эту систему.