

Модернизация хранилища данных для использования самой передовой аналитики

Обновление хранилища, предназначенного для обработки и анализа больших объемов данных, не должно нарушать функционирования вашей информационной среды. Благодаря низкой стоимости, высокой скорости и масштабируемости массивно-параллельной архитектуры колоночная база данных, в частности Vertica, способна стать важнейшим элементом гибридной архитектуры Больших данных.

Знакомство с Vertica. Углубленное изучение данных

Ваше хранилище данных уже не справляется с растущими рабочими нагрузками?

Всегда есть соблазн отказаться от поддержки устаревшего хранилища, сославшись на то, что это «вещь из прошлого», которой нет места в архитектуре Больших данных. И все-таки не делайте этого: вы рискуете совершить не просто ошибку, а очень дорогостоящую ошибку.

Хранилище данных является лучшей в своем роде платформой для работы со структурированными данными. Его архитектура обеспечивает более высокую производительность и обходится гораздо дешевле любых других альтернатив, в том числе таких популярных платформ, как Hadoop и Apache Spark, которые лишь дополняют и расширяют хранилище данных. Они не подходят для аналитической обработки запросов, что как раз и является основной функцией хранилища данных.

Многое изменилось с тех пор, как 25 лет назад была впервые представлена архитектура хранилища данных. На смену строчным базам данных с непараллельной обработкой, которые лежали в основе традиционных решений, пришли гораздо более быстрые и эффективные колоночные платформы, где хранение данных организовано по столбцам и используется массивно-параллельная обработка (MPP).

Новшеством является не сама MPP-обработка, а ее доступность. Десять лет назад такие аналитические базы данных, как Vertica, сделали MPP привлекательной с экономической точки зрения, дав возможность выполнять масштабирование с разумными затратами. Для подавляющего числа запросов хранилище данных MPP обеспечивает очень высокую производительность — несопоставимо больше, чем реляционные СУБД без параллельной обработки. При выполнении аналитических

запросов колоночное хранилище данных с массивно-параллельной обработкой работает еще быстрее.

Устаревшее хранилище данных обходится слишком дорого и слишком медленно реагирует на запросы? Скорее всего, оно функционирует на основе строчной базы данных без параллельной обработки, поэтому ваши ожидания относительно его возможностей явно завышены. Аналитическое хранилище данных, такое как Vertica, позволит значительно увеличить производительность приложений, создать новые средства анализа данных и — главное — разработать логичную, предсказуемую и не слишком затратную стратегию масштабирования.

Все это становится возможным благодаря тому, что многие хранилища данных MPP масштабируются линейно: при удвоении количества серверов в кластере хранилища с массивно-параллельной обработкой производительность возрастает в два раза. Удобное потерабайтное ценообразование в отношении Vertica, аналитической СУБД с MPP, позволяет масштабировать производительность хранилища с учетом потребностей бизнеса. Это очень важное отличие от традиционных систем, которые лицензируются по числу процессоров, количеству пользователей, подключенных систем или по всем этим параметрам сразу.

«Имеющиеся в организациях системы порой обходятся слишком дорого, так как обязательные лицензионные сборы рассчитываются исходя из количества пользователей, подключенных систем и т. д. К тому же они работают на базе устаревшей технологии, которая не способна справиться с Большими данными, — поясняет Стив Сарсфилд (Steve Sarsfield), руководитель отдела маркетинга Vertica. — Платформа Vertica лишена этих недостатков. Благодаря непревзойденным возможностям масштабирования она может справиться с самыми сложными задачами, а прозрачная модель оплаты по терабайтам позволяет избежать лишних расходов. Все эти достоинства обеспечивают простое и удобное масштабирование. Клиенты могут без труда модернизировать свою ИТ-среду и существенно снизить затраты при внедрении и использовании самой передовой аналитики».

Основа колоночного хранилища

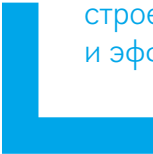
Архитектура традиционных хранилищ данных разрабатывалась на основе строчных реляционных СУБД, рассчитанных на обработку транзакций в реальном времени (OLTP), таких как Oracle, DB2 и Sybase.

OLTP-системы работают на платформе, состоящей из мэйнфреймов (в случае с Oracle и DB2) или систем с архитектурой «клиент-сервер» — NetWare, OS/2, Unix и (позже) Windows NT. Именно поэтому базы данных OLTP, как правило, дешевле и доступнее предшествующих платформ, таких как IMS и ADABAS, которые работали на мэйнфреймах и из-за сложных схем оплаты крупных и мощных компьютеров были чересчур дорогими.

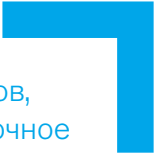
Другими словами, выбор строчных СУБД в качестве основы для хранилищ данных объясняется тем, что в то время это было лучшее — то есть самое удобное, экономичное и мобильное — решение. Но развитие колоночных баз данных изменило ситуацию.

В отличие от задач OLTP, в аналитические запросы включается лишь ряд атрибутов, а это означает, что для основной части аналитики колоночная архитектура обеспечит максимальную производительность и эффективность. Однако когда разрабатывались и реализовывались первые хранилища данных, качественных колоночных СУБД еще не было. Некоторые платформы, в частности IMS, предлагали колоночную обработку, но не были основаны на реляционной модели. (Более того, IMS предлагалась только для мэйнфреймов.)

Vertica стала одной из первых так называемых «аналитических» (или колоночных с массивнопараллельной обработкой) систем хранилищ данных. Ее придумал гур в области проектирования баз данных Майкл Стоунбрейкер (Michael Stonebraker), в свое время участвовавший в создании революционной РСУБД Ingres и ее преемника Postgres, а также VoltDB, Tamr и множества других замечательных СУБД и хранилищ данных. Стоунбрейкер полагал, что, поскольку строчные базы данных устарели, колоночная платформа значительно ускорит анализ данных, обеспечив и ряд других важнейших преимуществ.



В аналитические запросы включается лишь ряд атрибутов, а это означает, что для основной части аналитики колоночное строение обеспечит максимальную производительность и эффективность



«Секрет колоночной СУБД — в ее непревзойденной скорости работы. В традиционных реляционных СУБД информация хранится в строках, поэтому, даже если вашему запросу нужны данные из одного столбца, СУБД сканирует содержимое всех строк — каждого столбца в каждой строке. Таким образом, с диска каждый раз считывается вся строка», — поясняет Сарсфилд. Он отмечает, что для большинства типов аналитики необходима информация только из одного столбца. «Применяемый сегодня подход создает большую нагрузку на системы ввода-вывода и увеличивает время поиска данных, ведь для ответа на один простой запрос требуется обработать очень много данных».

Второе главное преимущество колоночного хранилища заключается в том, что его архитектура позволяет сократить объем операций ввода-вывода данных — а это самый существенный параметр для аналитической обработки. Третье преимущество — эффективное сжатие данных, обеспечивающее в 4–5 раз более высокую производительность, чем традиционные СУБД.

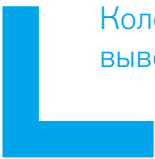
«Колоночные СУБД эффективно сжимают данные. В частности, Vertica, анализируя данные, определяет, какой алгоритм сжатия из имеющихся в ее распоряжении можно использовать», — поясняет Сарсфилд. Кроме того, он отмечает, что колоночная архитектура Vertica позволяет выполнять позднюю материализацию — осуществлять операции со сжатыми столбцами в оперативной памяти.

«Сжатие, которого нам удалось достичь, позволяет хранить данные на сравнительно небольшом объеме дискового пространства. К тому же оно обеспечивает гораздо более высокую производительность операций ввода-вывода, — уточняет Сарсфилд. — Например, при работе с полями, хранящими даты, Vertica автоматически подбирает лучший алгоритм сжатия, благодаря чему для их хранения требуется значительно меньшее дисковое пространство».


Строчные реляционные СУБД не способны эффективно сжимать данные. Мало того, практика показывает, что строчные СУБД, объединенные в ходе создания сервиса хранилища данных, могут использоваться вместе лишь при наличии программного средства поддержки принятия решений, которое будет обеспечивать совместную работу с ними. При таком подходе настройка и балансировка становятся более трудоемкими процедурами, чем при использовании колоночных СУБД, причем это становится особенно заметно по мере дальнейшего увеличения объемов данных. Кроме того, для обеспечения нужной производительности администраторам СУБД приходится реализовывать временные решения: материализованные представления, вертикальное партиционирование и индексацию каждого столбца.

В результате возрастают расходы на эксплуатацию и обслуживание хранилища данных. И чем больше объемы данных, многообразнее и сложнее запросы к базе данных, тем ниже эффективность их выполнения.

«Можно себе представить, насколько медленно будет работать традиционная СУБД, если в базе данных имеются сотни или тысячи столбцов — все их нужно считывать. Чтобы ускорить обработку данных, приходится реализовывать одно или несколько обходных решений, таких как материализованные представления или индексы, — добавляет Сарсфилд. — Эти обходные решения имеют ограничения. Материализованные представления необходимо обновлять и поддерживать, индексы предварительно строить и т. д. В конечном счете именно от этого и зависит их эффективность».



Колоночная архитектура сокращает объем операций ввода-вывода, обычно весьма значительный при анализе данных



Колоночное хранилище — это, конечно, не волшебная палочка, ведь нет ни одной технологии, которая чудесным образом решила бы все проблемы. Но недостатки колоночного хранилища превращаются в достоинства, когда речь идет не о транзакционной нагрузке, характерной для учетных систем, а об аналитической нагрузке. В частности, одна из особенностей колоночной платформы заключается в том, что колоночная СУБД хранит каждый столбец в отдельном множестве файлов. (Колоночная СУБД, как и ее реляционный аналог, оптимизируется для физического хранения данных на диске или в оперативной памяти. Это важное различие между СУБД, например колоночной или РСУБД OLTP, и хранилищем данных, например Hadoop. Мы поговорим об этом подробнее в следующем разделе.)

Развертывание колоночной СУБД в ее базовой конфигурации может привести к замедлению операций добавления и обновления данных, а также задержать — или усложнить — их загрузку. Чтобы избежать подобных проблем, в колоночных СУБД, в частности в Vertica, реализуются оптимизированные технологии массивно-параллельной загрузки и используются реляционные подходы OLAP (ROLAP) или комплексные методы OLAP (MOLAP). Кроме того, как отмечает Сарсфилд, колоночное проектирование позволяет применять и другие технологии оптимизации, повышающие производительность, например блочную итерацию. Так, несколько значений из столбца могут быть переданы от одного оператора другому в виде блока, что оказывается более эффективным, чем использование итераторов в строчной схеме.

«Мы сотрудничаем с очень крупными компаниями, например с Facebook, которая ежедневно загружает в Vertica более 35 ТБайт информации. Пока им этого достаточно, но при необходимости они смогут увеличить загрузку до 60 ТБайт. Мы можем выделить часть узлов для решения подобных задач загрузки, а на остальные возложить только аналитическую обработку, — поясняет он. — Кроме того, теперь Vertica предлагает и совершенно новые функции ROLAP/MOLAP, поскольку некоторые наши клиенты попросили их добавить».

Сейчас некоторые поставщики «встраивают» в строчные базы данные возможности колоночного хранилища. Например, база данных Microsoft SQL Server 2016, которая должна вскоре выйти, будет поддерживать «колоночное индексирование», при этом ее модуль расширения PowerPivot Excel позволит на обычных ПК использовать функции, аналогичные колоночному хранению. Но эти

продукты не только не реализуют колоночную архитектуру СУБД, но и не обеспечивают массивно-параллельную обработку.

Если колоночная архитектура оптимальна для выполнения анализа данных, то массивно-параллельная архитектура — для масштабирования аналитической обработки. По словам Сарсфилда, клиенты Vertica подтверждают неоспоримое преимущество колоночной MPP с точки зрения производительности.

«Около года назад мы попросили наших клиентов сравнить их старые хранилища данных и новые хранилища Vertica по скорости выполнения запросов. Около 10 % ответивших заявили, что смогли увеличить производительность на 1000 %, а 64 % добились увеличения на 100 %, — рассказывает он. — Увеличение производительности на 1000 % означает, что операции, которые занимали один час, теперь выполняются всего за шесть минут, а на то, что длилось в течение восьми часов, требуется всего 48 минут».

Если колоночная архитектура оптимальна для выполнения анализа данных, то массивно-параллельная архитектура — для масштабирования аналитической обработки.

Работа с Hadoop

Hadoop также является средой с массивно-параллельной обработкой данных, однако, в отличие от реляционных СУБД, Hadoop не является СУБД, она реализует распределенную файловую систему Hadoop (HDFS), которая обеспечивает хранение и управление данными на множестве серверов (узлов), объединенных в кластер.

Реляционные СУБД с MPP, например Vertica, тоже распределяют данные по кластеру. Основное различие между реляционной СУБД и файловой системой (которую все-таки можно считать в каком-то смысле базой данных) — это уровень абстракции. В большинстве случаев СУБД реализует оптимизацию размещения и физической структуры, логику и схему работы с данными «поверх» файловой системы. HDFS не обладает необходимыми для этого функциями. Как и реляционная СУБД, Hadoop может эффективно хранить и извлекать данные, которые записывает в блоки одинакового размера, но в Hadoop нет встроенных возможностей для сопоставления данных, хранящихся в этих блоках.


Если Hadoop использовать как платформу для выполнения SQL-запросов, то, чтобы воспользоваться функциями, присущими реляционной СУБД, придется внедрить технологии Hive, Impala, Drill или Spark (со Spark SQL). Однако эти инструменты не предлагают возможностей, необходимых для управления данными. В отличие от Vertica, которая совместима с новейшей — SQL 2011 — версией стандарта ANSI SQL, платформы Hive, Impala и Spark SQL не полностью совместимы с ANSI SQL, к тому же не поддерживают высокий уровень параллелизма.

«С помощью Hadoop многие стремятся решить проблемы с уже имеющимися хранилищами данных, не способных справиться со многими аналитическими запросами, которые генерируются пользователями. В то же время, если сравнивать производительность Hadoop и Vertica при выполнении произвольных запросов и SQL-анализа, отрыв окажется огромным: Vertica на порядок быстрее обрабатывает большинство запросов, — уточняет Сарсфилд. — Чтобы работать с Hadoop напрямую, исторически необходимо заниматься обучением имеющихся специалистов, искать новых сотрудников и приобретать дополнительное ПО, а для этого потребуется немало денег и времени. Перейдя на Vertica, вы сможете отказаться от прежних дорогостоящих хранилищ данных. Мы предлагаем способ модернизировать их без больших потрясений».

Сарсфилд подчеркивает, что стратегия Vertica никоим образом не направлена против Hadoop. Более того, он отмечает, что Hadoop позволяет дополнить, расширить и даже превзойти некоторые весьма важные возможности хранилищ данных.

Во-первых, Hadoop функционирует на основе распределенной файловой системы, поэтому может эффективно обрабатывать буквально все типы данных. Реляционная СУБД с массивно-параллельной обработкой, напротив, обрабатывает далеко не всё. (При этом данные за некоторыми исключениями должны соответствовать заранее определенной схеме.) Кроме того, реляционная СУБД с MPP проигрывает в обработке разнородных данных: РСУБД оптимизированы для выполнения запросов, в том числе операций со структурированными данными, представленными в виде таблицы (с помощью языка SQL). Как мы уже отметили, Hadoop позволяет без лишних затрат размещать, хранить и обрабатывать структурированные, полуструктурированные и мультиструктурированные данные. С помощью реляционной базы данных это сделать невозможно.

По словам Сарсфилда, в Vertica реализовано лучшее из этих двух вариантов. С одной стороны, Hadoop может стать экономичной платформой для длительного хранения данных и управления ими. С другой — с помощью Vertica клиенты смогут повысить скорость выполнения запросов традиционного хранилища данных и систем аналитики.



Платформа Vertica разработана таким образом, что может выполнять запросы к данным, хранящимся в Hadoop

«Если вы хотите хранить информацию «на всякий случай» (например, материалы из блогов или данные Интернета вещей), Vertica прекрасно для этого подойдет. Если нужен быстрый анализ огромных объемов данных вне зависимости от их источника, Vertica — идеальный вариант», — говорит Сарсфилд, ссылаясь на пример онлайн-агрегатора купонов RetailMeNot, который с помощью Vertica оперативно предоставляет купоны покупателям как на сайте, так и в обычных магазинах.

«Сведения о покупателях, переходящих по ссылке и использующих купоны, сопоставляются с информацией о тех, кто игнорирует эти предложения, — поясняет он. — Все данные собираются в Hadoop и анализируются в Vertica».


Гибридная архитектура Больших данных

По мнению Сарсфилда, невозможно дать точный ответ на вопрос, что лучше — хранилище данных или Hadoop. Нужно учиться использовать их совместно. Facebook и RetailMeNot применяют в рамках гибридной архитектуры, предназначенной для работы с Большими данными, и хранилище данных Vertica, и Hadoop.


Hadoop является недорогой и хорошо масштабируемой основой для хранения данных. По словам известного специалиста в этой отрасли Томаса Дэвенпорта (Thomas Davenport), старшего научного сотрудника Центра цифрового бизнеса Массачусетского технологического института, хранилище на основе Hadoop обходится в 0,23 доллара за гигабайт, в то время как при использовании

традиционного хранилища данных один гигабайт стоит 19 долларов. И хотя стоимость гигабайта хранилища в такой колоночной платформе MPP, как Vertica, гораздо ниже, чем в традиционном хранилище данных MPP или не-MPP, Hadoop является более гибким вариантом для хранения данных.

В Vertica реализовано лучшее из двух миров: Hadoop, для экономичного длительного хранения и управления данными, и расширенная аналитика для повышения эффективности выполнения запросов в традиционном хранилище.



С помощью таких колоночных платформ, как Vertica, свои среды хранения данных модернизируют Facebook, RetailMeNot и другие компании



Однако Facebook, RetailMeNot и другие компании не используют Hadoop вместо прежних систем хранения данных; они обращаются к таким колоночным платформам MPP, как Vertica, чтобы дополнить свою среду хранения данных. Vertica предлагает экономичный хорошо масштабируемый механизм и традиционной, и расширенной аналитики, а кроме того, — масштабируемую платформу управления данными, помогающую по максимуму использовать преимущества Hadoop.

Сарсфилд полагает, что секрет успеха заключается в том, чтобы применять каждую платформу там, где это нужно. Вот почему платформа Vertica разработана таким образом, чтобы можно было выполнять запросы к данным, хранящимся в Hadoop.

«Многие наши клиенты используют узлы Hadoop. В кластерах Hadoop они создают «песочницы» и размещают в них данные, но при попытке выполнить анализ этих данных выясняется, что в Hadoop нет необходимых для этого инструментов. Клиенты не могут воспользоваться всеми возможностями аналитики, которые предлагает Vertica. Hadoop не имеет служб каталогов метаданных и не поддерживает высокие уровни параллелизма, — рассказывает Сарсфилд, отмечая, что ядро Vertica способно работать с аналогом колоночного хранилища, а именно с файлами Parquet и Optimized Row Columnar (ORC), которые используются с Impala и Hive. — Мы усовершенствовали нашу систему обработки SQL-запросов на базе Vertica таким образом, что теперь с ее помощью можно получить прямой доступ к данным в Hadoop (например, файлам Parquet или ORC) и проанализировать их».

Универсального решения нет

Ни одна платформа не может идеально справиться абсолютно со всеми аналитическими задачами. Vertica — это непревзойденная платформа для обработки запросов и расширенной аналитики структурированных или преимущественно структурированных данных. Однако она не подходит для анализа неструктурированных данных — изображений или видеофайлов.

«В отличие от Hadoop, платформа Vertica была создана именно для выполнения аналитических запросов, касающихся структурированных данных. Hadoop же создавался для распределенного хранения файлов и разнесения вычислений MapReduce между узлами кластеров, — говорит Сарсфилд. — Расширенная аналитика в Vertica отличается от той, что выполняется в Hadoop. Hadoop идеально подходит для анализа некоторых видов неструктурированных данных. Обе платформы замечательны, просто их следует использовать для разных целей».

Дополнительную контактную информацию
и адреса представительств см. по адресу
www.microfocus.com

www.microfocus.com