Хранилища данных в AWS

Март 2016 г.

Уведомления

Этот документ предоставляется исключительно в информационных целях. В документе представлены текущие предложения продуктов и практики AWS, актуальные на дату публикации, которые могут меняться без предварительного уведомления. Клиенты несут ответственность за независимую оценку представленной в документе информации и использования продуктов и услуг AWS любым способом. Указанные информация, продукты и услуги предоставляются «как есть», без какой-либо явной или подразумеваемой гарантии. Данный документ не создает никаких гарантий, представлений, контрактных обязательств, условий или заверений от AWS, ее аффилированных лиц, поставщиков или лицензиатов. Обязанности и финансовые обязательства AWS в отношении клиентов компании регулируются соглашениями AWS, частью которых данных документ не является. Кроме того, данный документ не вносит изменения ни в одно из заключенных AWS и клиентами компании соглашений.

Содержание

Резюме

Во всем мире специалисты по хранению данных, аналитики и разработчики стремятся перенести хранилища данных в облако, чтобы повысить эффективность работы и снизить затраты. В этом техническом описании рассматривается современный подход к аналитике и архитектуре хранилищ данных, перечисляются сервисы, доступные в Amazon Web Services (AWS) для реализации этой архитектуры, и предоставляются стандартные шаблоны проектирования, позволяющие создавать решения для хранилищ данных с использованием этих сервисов.

Введение

В современном мире данные и аналитика являются неотъемлемой частью любого бизнеса. Практически на всех крупных предприятиях созданы хранилища данных, которые используются для отчетности и анализа, а данные в них собираются из различных источников, включая собственные системы обработки транзакций и другие базы данных.

Тем не менее создание и обслуживание хранилища данных (центрального репозитория информации, поступающей из одного или нескольких источников данных) всегда были сопряжены со значительными сложностями и затратами. Большинство систем хранилищ данных сложно создать и подготовить к работе, при этом требуются предварительные расходы на программное и аппаратное обеспечение в размере несколько миллионов долларов, а процедуры планирования, приобретения, внедрения и развертывания могут занять долгие месяцы. Сделав первоначальные инвестиции и подготовив хранилище данных к работе, необходимо нанять команду администраторов баз данных, которые обеспечивали бы быструю обработку запросов и предотвращали потери данных.

Кроме того, традиционные хранилища данных достаточно сложно масштабировать. Если объемы данных растут или требуется сделать аналитику и отчеты доступными большему числу пользователей, приходится выбирать между низкой производительностью обработки запросов и необходимостью вкладывать время, усилия и немалые средства в обновление системы. Некоторые ИТ-отделы не рекомендуют увеличивать объем данных или добавлять запросы, чтобы обеспечить соблюдение существующих соглашений об уровне обслуживания. Многие корпоративные организации прилагают массу усилий для сохранения нормальных взаимоотношений с традиционными поставщиками баз данных. Им часто приходится либо принудительно обновлять аппаратное обеспечение для используемой системы или вести длительные переговоры о лицензиях с истекшим сроком действия. По достижении лимита масштабирования одного ядра хранилища данных им приходится переходить на другое ядро того же поставщика с отличающейся семантикой SQL.

Amazon Redshift меняет взгляд корпоративных клиентов на хранилища данных, существенно снижая затраты и усилия на развертывание систем для хранения данных без ущерба для функциональности и производительности. Amazon Redshift – это быстрое, полностью управляемое решение для хранилищ данных объемом до нескольких петабайтов. Amazon Redshift позволяет просто и без лишних затрат анализировать большие объемы данных с использованием существующих средств бизнес-аналитики. Amazon Redshift использует производительность ядер хранилищ столбчатых данных, снижая расходы в десять раз благодаря использованию массово-параллельной обработки (MPP). Можно начать с малого (0,25 долларов США за час без каких-либо обязательств) и постепенно перейти к обработке петабайтов данных по цене 1000 долларов США за терабайт в год.

Выпущенный в феврале 2013 года сервис Amazon Redshift стал одним из наиболее быстро растущих сервисов AWS. Сегодня его услугами пользуются тысячи клиентов из разных отраслей и компаний разных масштабов. Многие корпоративные клиенты, такие как NTT DOCOMO, FINRA, Johnson & Johnson, Hearst, Amgen и NASDAQ, перешли на Amazon Redshift. Не удивительно, поэтому, что сервис Amazon Redshift был назван лидером в отчете *Forrester Wave: корпоративные хранилища данных, IV квартал 2015 года*.[1]

В этом техническом описании представлена информация, которая позволит вам реализовать преимущества стратегического сдвига, который сейчас наблюдается в сфере хранилищ данных, а именно перехода от локальных систем к облачным.

- Архитектура современной аналитики
- Доступные в этой архитектуре варианты технологий для хранилищ данных
- Подробное рассмотрение сервиса Amazon Redshift и его отличительных преимуществ
- Проект создания полноценной системы для хранилищ данных на базе AWS с использованием Amazon Redshift и других сервисов
- Практические рекомендации по переходу с других решений для хранилищ данных и использованию ресурсов нашей партнерской экосистемы

Современная аналитика и архитектура хранилищ данных

Повторимся: *хранилище данных* – это центральный репозиторий информации, поступающей из одного или нескольких источников данных. Как правило, данные (структурированные, полуструктурированные и неструктурированные) поступают в хранилище данных из транзакционных систем и других реляционных баз данных. Данные обрабатываются, преобразуются и поглощаются хранилищем с регулярными промежутками. Пользователи (включая специалистов по работе с данными, бизнес-аналитиков и лиц, ответственных за принятие решений) осуществляют доступ к данным с помощью средств бизнес-аналитики, SQL-клиентов и электронных таблиц.

Зачем вообще создавать хранилище данных? Почему бы не выполнять аналитические запросы непосредственно в базе данных OLTP, где фиксируются транзакции? Чтобы ответить на этот вопрос, проанализируем различия хранилищ данных и баз данных OLTP. Хранилища данных оптимизированы для пакетных операций записи и чтения больших объемов данных, в то время как базы данных OLTP оптимизированы для непрерывных операций записи и больших объемов мелких операций чтения. Как правило, в хранилищах данных используются ненормализованные схемы (например, схема звезды или схема снежинки), поскольку требования к пропускной способности данных достаточно высоки, в то время как в базах данных OLTP используются в высшей степени нормализованные схемы, которые более соответствуют высоким требованиям к пропускной способности транзакций. Схема звезды состоит из нескольких крупных таблиц фактов, которые ссылаются на несколько таблиц измерений. Схема снежинки, которая представляет собой расширение схемы звезды, состоит из таблиц измерений, которые еще более нормализованы.

Рекомендуется создать эффективный конвейер данных, чтобы реализовать преимущества использования хранилища данных, управляемого как отдельное хранилище данных и использующего в качестве исходной системы OLTP или любую другую систему. Такой конвейер извлекает данные из исходной системы, преобразует их в схему, подходящую для хранилища данных, а затем загружает их в хранилище. В следующем разделе мы рассмотрим составляющие аналитического конвейера и различные сервисы AWS, которые можно использовать для проектирования такого конвейера.

Архитектура аналитики

Конвейеры аналитики предназначены для обработки больших объемов входящих потоков данных из разных источников, включая базы данных, приложения и устройства.

Стандартный конвейер аналитики состоит из нескольких этапов.

1. Сбор данных.
2. Хранение данных.

3. Обработка данных.
4. Анализ и визуализация данных.

См. пример на рис. 1 далее.

Рис. 1. Конвейер аналитики

Сбор данных

На этапе сбора данных нужно исходить из того, что работать придется с данными разных типов: транзакционными, потоковыми, данными журналов и «Интернета вещей» (IoT). AWS предоставляет решения, позволяющие создавать хранилища для каждого из этих типов данных.

Транзакционные данные

Транзакционные данные, такие как данные о транзакциях покупки в системе интернет-коммерции или финансовых транзакциях, как правило, сохраняются в системах управления реляционными базами данных (RDBMS) или системах баз данных NoSQL. Выбор решения для создания базы данных зависит от варианта использования и характеристик приложения. База данных NoSQL подходит, если данные недостаточно хорошо структурированы, чтобы использовать определенную схему, или схема слишком часто меняется. Решение RDBMS (система управления реляционными базами данных), с другой стороны, подходит для сценариев, когда транзакции выполняются в многих строках таблицы, а запросы требуют сложных объединений. Amazon DynamoDB – это полностью управляемый сервис баз данных NoSQL, который можно использовать в качестве хранилища OLTP для своих приложений. Amazon RDS позволяет использовать для выполнения ваших задач решение для реляционных баз данных на базе SQL.

Данные журналов

Надежная фиксация создаваемых системой журналов поможет в диагностике и устранении неполадок, проведении аудитов и выполнении аналитических операций с использованием хранимой в таких журналах информации. Amazon Simple Storage Service (Amazon S3) – это популярное решение для хранения нетранзакционных данных (таких как данные журнала), которое используется в аналитике. Кроме того, высочайший показатель бесперебойной работы, выражаемый 11 девятками (то есть 99,999999999), делает Amazon S3 популярным решением для архивации.

Потоковые данные

Веб-приложения, мобильные устройства и многочисленные программные приложения и сервисы могут генерировать немыслимые объемы потоковых данных (до нескольких терабайт в час), и эти данные нужно собирать, хранить и непрерывно обрабатывать.[2] Сервисы Amazon Kinesis позволяют решать эти задачи легко и с минимальными затратами.

Данные IoT

Устройства и датчики в разных точках планеты непрерывно отправляют сообщения. Сегодня корпоративные организации испытывают растущую потребность фиксировать эти данные и извлекать из них ценные аналитические сведения. Благодаря AWS IoT подключенные устройства легко и безопасно взаимодействуют с облаком AWS. AWS IoT упрощает использование многих сервисов AWS, таких как AWS Lambda, Amazon Kinesis, Amazon S3, Amazon Machine Learning и Amazon DynamoDB, и позволяет создавать приложения для сбора, обработки, анализа данных IoT и принятия решений на основе этих данных. Необходимость самостоятельно управлять инфраструктурой при этом отсутствует.

Обработка данных

В процессе сбора можно получить данные, которые содержат потенциально ценную информацию. Анализ извлеченной информации позволяет извлечь ценные аналитические сведения для развития и роста вашего бизнеса. Так, из этих сведений можно узнать о поведении пользователей и относительной популярности ваших товаров. Оптимальным подходом для извлечения ценных аналитических сведений считается загрузка необработанных данных в хранилище данных с целю выполнения дальнейшего анализа.

Для этого существует два типа обработки данных: пакетная и в реальном времени. Эти разновидности используются в двух наиболее распространенных формах обработки данных: оперативная аналитическая обработка (OLAP) и оперативная обработка транзакций (OLTP). Как правило, оперативная аналитическая обработка (OLAP) выполняется с использованием пакетов. Напротив, в системах OLTP выполняется обработка данных в реальном времени, поэтому для пакетной обработки такие системы не подходят. Если отделить обработку данных от системы OLTP, обработка данных не будет влиять на вашу рабочую нагрузку OLTP.

Для начала проанализируем, что входит в пакетную обработку.

Извлечение, преобразование, загрузка (ETL)

ETL – это процесс извлечения данных из многочисленных источников с целью последующей загрузки в системы хранения данных. Как правило, ETL представляет собой непрерывный процесс, который выполняется по четко определенным принципам. В начале этого процесса данные извлекаются из одного или нескольких источников. Затем извлеченные данные очищаются, обогащаются, преобразуются и загружаются в хранилище данных. Инструменты платформы Hadoop, такие как Apache Pig и Apache Hive, традиционно используются в конвейере ETL для преобразования больших объемов данных.

Извлечение, загрузка, преобразование (ELT)

ELT – это разновидность ETL, когда извлеченные данные сначала загружаются в целевую систему. Преобразования выполняются после загрузки данных в хранилище. ELT отлично подходит для сценариев, когда целевая система достаточно мощная для выполнения преобразований. Amazon Redshift часто используется в конвейерах ELT, поскольку обеспечивает высокую эффективность преобразования данных.

Оперативная аналитическая обработка (OLAP)

Системы OLAP хранят агрегированные данные за прошлые периоды в многомерных схемах. Системы OLAP, которые широко используются в интеллектуальном анализе данных, позволяют извлекать данные и выявлять тенденции в нескольких измерениях. Поскольку сервис Amazon Redshift оптимизирован для быстрого создания объединений данных, он часто используется для создания систем OLAP.

А сейчас давайте проанализируем, что подразумевается под обработкой данных в реальном времени.

Обработка данных в реальном времени

Ранее мы говорили о потоковых данных и упомянули о том, что решение Amazon Kinesis предназначено для фиксации и хранения потоковых данных. Эти данные можно обрабатывать последовательно и поступательно, работая с отдельными записями или скользящими временными окнами. Обработанные данные можно использовать для решения разнообразных аналитических задач, включая создание корреляций, агрегаций и выборок, а также фильтрацию данных. Такой тип обработки называется обработкой в реальном времени. Информация, полученная в результате обработки данных в реальном времени, обеспечивает обзорность многих аспектов бизнеса и взаимодействия с клиентами, включая использование сервисов (для измерения и выставления счетов), серверную активность, число переходов на веб-сайты, определение географического расположения устройств, людей и физических товаров, и позволяет своевременно реагировать на изменение ситуации. Для обработки данных в реальном времени требуется слой обработки с высоким уровнем параллелизма и масштабирования.

Для обработки потоковых данных в реальном времени можно воспользоваться сервисом AWS Lambda. Lambda позволяет обрабатывать данные, поступающие непосредственно из AWS IoT или Amazon Kinesis Streams. Для выполнения кода в сервисе Lambda подготавливать серверы и управлять ими не требуется.

Amazon Kinesis Client Library (KCL) – еще одно средство для обработки данных из системы Amazon Kinesis Streams. KCL отличается большей гибкостью по сравнению с AWS Lambda и позволяет объединять входящие данные в пакеты для дальнейшей обработки. KCL обеспечивает широкие возможности преобразования и настройки логики обработки данных.

Использование Amazon Kinesis Firehose – это простейший способ загрузки потоковых данных в AWS. Сервис позволяет фиксировать потоковые данные и автоматически загружать их в систему Amazon Redshift. Таким образом, вы сможете использовать имеющиеся средства бизнес-аналитики и панели управления для выполнения аналитических операций практически в режиме реального времени. С помощью Firehose можно установить правила формирования пакетов, после чего приложение будет надежно формировать пакеты данных и отправлять их в Amazon Redshift.

Хранение данных

Данные можно хранить в хранилищах или киосках данных, и далее мы рассмотрим эти сценарии.

Хранилище данных

Как уже упоминалось, *хранилище данных* – это центральный репозиторий информации, поступающей из одного или нескольких источников данных. В хранилищах данных можно быстро выполнять аналитические операции с большими объемами данных и выявлять скрытые в данных закономерности с помощью средств бизнес-аналитики. Специалисты по работе с данными отправляют запросы в хранилища данных для выполнения аналитических операций в автономном режиме и обнаружения тенденций. Пользователи организации принимают критически важные для бизнеса решения, выполняя запросы SQL, составляя периодические отчеты и работая с панелями управления.

Киоск данных

Киоск данных – это простейшая форма хранилища данных, ориентированная на конкретную функциональную или предметную область. Так, можно создать отдельные киоски данных для каждого подразделения организации или сегментировать киоски данных по регионам. Создавать киоски данных можно в крупном хранилище данных или операционном хранилище. Кроме того, два этих подхода можно сочетать. Киоски данных просты в проектировании, создании и администрировании. Поскольку киоски данных ориентированы на конкретные функциональные области, обработка запросов в этих функциональных областях может осложняться распределенной структурой таких киосков.

Amazon Redshift можно использовать для создания киосков данных в дополнение к хранилищам.

Анализ и визуализация

После того как данные обработаны и доступны для дальнейшего анализа, вам потребуются подходящие инструменты для анализа и визуализации обработанных данных.

Во многих случаях для анализа данных можно использовать те же инструменты, что и для обработки данных. Для анализа данных в Amazon Redshift с ANSI SQL можно использовать такие инструменты, как SQL Workbench. Сервис Amazon Redshift также совместим с распространенными на рынке сторонними решениями бизнес-аналитики.

Amazon QuickSight – это сервис бизнес-аналитики на базе облачных технологий, который отличается быстродействием и позволяет легко создавать визуализации, выполнять специальный анализ и быстро извлекать коммерчески ценные сведения из имеющихся данных. Сервис Amazon QuickSight интегрирован с Amazon Redshift и в настоящее время доступен в ознакомительной версии. Коммерческий запуск сервиса будет осуществлен позже в 2016 году.

Если Amazon S3 используется в качестве основного хранилища данных, для анализа и визуализации данных можно использовать блокноты Apache Spark на базе сервиса Amazon Elastic MapReduce (Amazon EMR). Такой подход позволяет выполнять запросы SQL и выполнять пользовательский код, написанный на таких языках, как Python и Scala.

Возможен и другой сценарий визуализации данных: Apache Zeppelin – это решение бизнес-аналитики с открытым исходным кодом, которое можно запускать в среде Amazon EMR для визуализации данных в сервисе Amazon S3 с использованием Spark SQL. Кроме того, с помощью Apache Zeppelin можно визуализировать данные в сервисе Amazon Redshift.

Конвейер аналитики и сервисы AWS

AWS предлагает широкий ассортимент сервисов для реализации комплексной платформы аналитики. На рисунке 2 показаны сервисы, о которых говорилось выше, и их место в конвейере аналитики.

Рис. 2: Конвейер аналитики и сервисы AWS

Хранилище данных: варианты технологий

В этом разделе обсуждаются варианты создания хранилища данных: строковые и столбчатые базы данных, а также архитектуры с массово-параллельной обработкой данных.

Строковые базы данных

Как правило, в строковых базах данных целые строки данных хранятся в физических блоках. Высокая производительность операций чтения достигается использованием дополнительных индексов. Базы данных Oracle Database Server, Microsoft SQL Server, MySQL и PostgreSQL относятся к строковым. Эти системы традиционно используются для хранения данных, однако они больше подходят для обработки транзакций (OLTP), чем для аналитики.

В целях повышения производительности строковой системы, используемой как хранилище данных, разработчики используют разнообразные техники, такие как создание материализованных представлений, предварительно агрегированных сводных таблиц и индексов для всех возможных комбинаций предикатов, использование разделов данных, которые могут при необходимости отсекаться оптимизатором запросов, и объединений на основе индексов.

Традиционные строковые хранилища данных ограничены ресурсами, доступными на определенном компьютере. Благодаря функциональному сегментированию в киосках данных эта проблема стоит не столь остро. Хранилище данных можно разделить на несколько киосков данных, предназначенных для определенных функциональных областей. Тем не менее с увеличением размера киоска данных обработка данных замедляется.

В строковом хранилище данных при обработке каждого запроса необходимо прочитать все столбцы для всех строк в блоках, соответствующих предикату запроса, включая столбцы, которые вы не выбирали. Это способствует образованию значительных «узких мест» с точки зрения производительности в хранилищах данных, таблицы которых содержат много столбцов, а в запросах используется лишь небольшое их количество.

Столбчатые базы данных

В столбчатых базах данных целые строки не упаковываются в блоки – вместо этого каждый столбец представляет собой отдельный набор физических блоков. Эта функциональная возможность позволяет эффективнее обрабатывать операции ввода-вывода при работе с доступными только для чтения запросами, потому что в этом случае достаточно прочитать столбцы, доступ к которым запрос осуществляет с диска (или из памяти). Поэтому столбчатые базы данных подходят для хранилищ данных больше, чем строковые.

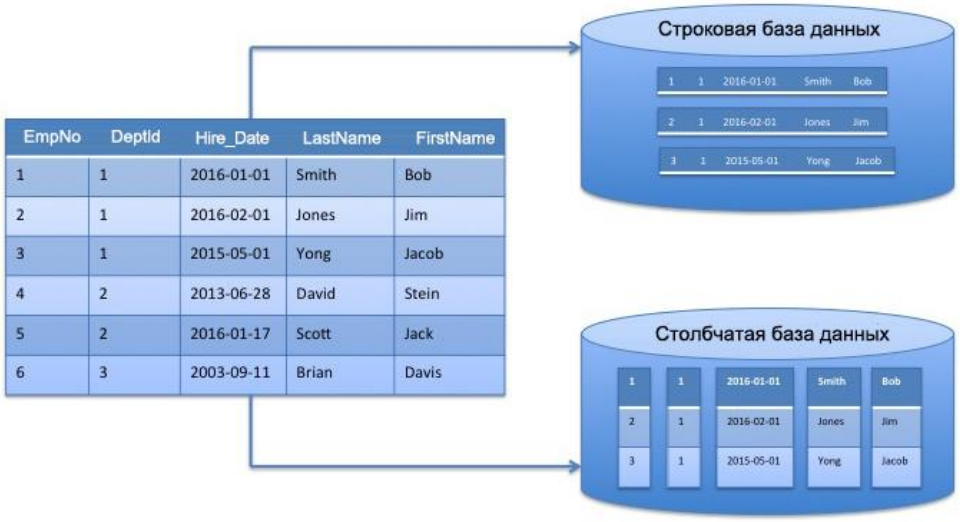

Рис. 3. Строковые и столбчатые базы данных

На рисунке 3 выше проиллюстрировано основное отличие строковых баз данных от столбчатых. В строковой базе данных строки упакованы в собственные блоки, а в столбчатой базе данных в собственные блоки упакованы столбцы.

Помимо ускорения обработки ввода-вывода нельзя не отметить еще одно важное преимущество столбчатых баз данных: усовершенствованное сжатие. Поскольку каждый столбец упакован в собственный набор блоков, каждый физический блок содержит данные одного типа. Если все данные относятся к одному типу, в базе данных можно реализовать чрезвычайно эффективные алгоритмы сжатия. В результате для хранения данных требуется меньше ресурсов по сравнению со строковой базой данных. Кроме того, в этом сценарии существенно уменьшается число операций ввода-вывода, потому что для хранения однотипных данных требуется меньше блоков.

Среди столбчатых баз данных, которые используются для хранилищ данных, можно назвать Amazon Redshift, Vertica, Teradata Aster и Druid.

Архитектуры с массово-параллельной обработкой данных

Архитектура с массово-параллельной обработкой данных позволяет использовать все доступные в кластере ресурсы для обработки данных, что в разы повышает производительность хранилищ, объем данных в которых исчисляется петабайтами. Для повышения производительности хранилищ данных с массово-параллельной обработкой данных достаточно добавить в кластер дополнительные узлы. Amazon Redshift, Druid, Vertica, GreenPlum и Teradata Aster относятся к хранилищам с массово-параллельной обработкой данных. Платформы с открытым исходным кодом, такие как Hadoop и Spark, также поддерживают массово-параллельную обработку данных.

Подробный обзор Amazon Redshift

Amazon Redshift – столбчатая база данных с массово-параллельной обработкой данных – предоставляет все необходимое для создания высокопроизводительных и экономичных хранилищ данных, включая эффективное сжатие, уменьшение операций ввода-вывода и снижение требований к ресурсам хранения. В основе этой базы данных – технология ANSI SQL, поэтому выполнять существующие запросы можно с нулевыми или минимальными модификациями. Вот почему сегодня Amazon Redshift так часто выбирают для корпоративных хранилищ и киосков данных. В этом разделе мы рассмотрим Amazon Redshift и ее возможности более подробно.

Amazon Redshift позволяет быстро обрабатывать запросы и операции ввода-вывода при работе с данными практически любого размера. Для этого используются столбчатые хранилища, параллелизация и распределение запросов по нескольким узлам. Система автоматизирует большинство распространенных административных задач, связанных с подготовкой, настройкой, мониторингом, архивацией и обеспечением безопасности хранилища данных, управлять которым в результате становится просто и экономично. Благодаря вышеперечисленным технологиям автоматизации создать хранилище данных, размер которого исчисляется петабайтами, можно за несколько минут (а не недель или месяцев, как в случае с традиционными локальными системами).

Производительность

В Amazon Redshift для уменьшения числа операций ввода-вывода, необходимых для обработки запросов, используются столбчатое хранилище, сжатие данных и карты зон. Сортировка с чередованием обеспечивает высокую производительность, но тратить ресурсы на обслуживание индексов или проекций при этом не требуется.

Архитектура Amazon Redshift с массово-параллельной обработкой данных позволяет использовать все доступные ресурсы благодаря параллельному выполнению и распределению операций SQL. Оборудование в основе системы предназначено для высокопроизводительной обработки данных, в то время как использование локального подключенного хранилища увеличивает до максимума пропускную способность между ЦП и дисками, а узловая сеть 10 GigE – пропускную способность между узлами. Параметры производительности можно настроить с учетом потребностей вашего хранилища данных: AWS предлагает узлы типа Dense Compute (DC) с дисками SSD и узлы типа Dense Storage (DS) Благодаря непрерывному развертыванию обновлений программного обеспечения производительность системы постоянно растет без вмешательства пользователей.

Надежность и доступность

Чтобы обеспечить максимальную надежность и доступность данных, Amazon Redshift автоматически выявляет и заменяет любой неисправный узел в кластере хранилища данных. В результате замененный узел становится доступным практически немедленно; первыми загружаются наиболее часто используемые данные, поэтому вы можете в кратчайшие сроки возобновить запросы данных. Поскольку Amazon Redshift зеркально отражает ваши данные в кластере, для перестройки неисправного узла используются данные с другого узла. Кластер доступен только для чтения до тех пор, пока заменяющий узел не будет подготовлен к работе и добавлен в кластер. Как правило, на это уходит всего несколько минут.

Кластеры Amazon Redshift расположены в одной зоне доступности.[3] Тем не менее при необходимости использования в Amazon Redshift структуры с несколькими зонами доступности можно создать зеркало, после чего система будет самостоятельно управлять репликацией и обработкой отказа.

Для создания надежной среды аварийного восстановления в системе Amazon Redshift достаточно нажать всего несколько кнопок на консоли управления Amazon Redshift. Копии резервных копий можно хранить в нескольких регионах AWS. Если в работе сервиса в одном из регионов AWS возникнут сбои, можно восстановить кластер из резервной копии в другом регионе AWS. Получить доступ к кластеру с возможностью чтения и записи можно через несколько минут после запуска операции восстановления.

Масштабируемость и эластичность

С помощью всего нескольких щелчков на консоли или API-вызова можно легко изменить число и тип узлов в хранилище данных, если ваши требования к производительности или потребности в ресурсах изменятся.[4] Amazon Redshift позволяет приступить к работе с узлом размером всего 160 ГБ и масштабировать его до многоузловой архитектуры, объем сжатых пользовательских данных в которой исчисляется петабайтами. Дополнительные сведения см. в разделе О кластерах и узлах в *Руководстве по управлению кластерами Amazon Redshift*.[5]

При изменении размера архитектуры Amazon Redshift помещает существующий кластер в режим «только чтение», подготавливает к работе

новый кластер нужного размера, а затем параллельно копирует данные из старого кластера в новый. В это время вы платите только за активный кластер Amazon Redshift. Во время подготовки нового кластера вы можете продолжать отправлять запросы в старый кластер. После того как данные копированы в новый кластер, Amazon Redshift автоматически перенаправляет запросы в новый кластер и удаляет старый.

Можно использовать действия в API Amazon Redshift для программного запуска и масштабирования кластеров, создания резервных копий, восстановления данных из резервных копий и т. д. Такой подход позволяет интегрировать выполняемые в APIШ действия в существующий стек автоматизации или создать пользовательскую систему автоматизации, соответствующую вашим потребностям.

Интерфейсы

В Amazon Redshift используются специальные драйверы Java Database Connectivity (JDBC) и Open Database Connectivity (ODBC), которые можно загрузить на вкладке **Клиент подключений** консоли. Это означает, что можно использовать широкий ассортимент знакомых клиентов SQL. Кроме того, можно использовать стандартные драйверы JDBC и ODBC PostgreSQL. Дополнительные сведения о драйверах Amazon Redshift см. в разделе Amazon Redshift и PostgreSQL *Руководства для разработчиков баз данных Amazon Redshift*. [6]

Кроме того, можно найти многочисленные примеры проверенных интеграций с продуктами многих популярных поставщиков BI- и ETL-систем.[7] В этих интеграциях загрузка и разгрузка данных выполняются параллельно на каждом вычислительном узле, благодаря чему скорость получения данных из разных источников, включая Amazon S3, Amazon EMR и Amazon DynamoDB, значительно увеличивается (равно как и скорость экспорта данных в эти системы). Средство Amazon Kinesis Firehose позволяет легко загружать потоковые данные в систему Amazon Redshift, то есть решать аналитические задачи с использованием существующих средств бизнес-аналитики и панелей управления практически в реальном времени. Показатели использования вычислительных ресурсов, памяти, ресурсов хранения и трафика чтения и записи для своего кластера в хранилище данных Amazon Redshift можно изучить на консоли или с помощью API Amazon CloudWatch.

Безопасность

В целях обеспечения безопасности данных можно запускать хранилище Amazon Redshift внутри виртуального частного облака с помощью сервиса Amazon Virtual Private Cloud (Amazon VPC). Определенную программным способом сетевую модель VPC можно использовать для определения правил брандмауэра, ограничивающих трафик в соответствии с настроенными вами правилами.[8] Amazon Redshift поддерживает подключения на базе SSL между вашим клиентским приложением и кластером хранилища данных Amazon Redshift, что позволяет шифровать передаваемые данные.

Ваши данные хранятся в вычислительных узлах Amazon Redshift, однако доступ к этим данным осуществляется только из ведущего узла кластера. Подобная изоляция обеспечивает дополнительный уровень безопасности. Amazon Redshift интегрируется с AWS CloudTrail, обеспечивая возможность аудита всех вызовов API Amazon Redshift.[9] Для обеспечения безопасности хранимых данных Amazon Redshift шифрует каждый блок, используя шифрование AES-256 с аппаратным ускорением во время записи каждого блока на диск. Данные шифруются на низком уровне подсистемы ввода-вывода; подсистема ввода-вывода шифрует все, что записывается на диск, включая промежуточные результаты запросов. Резервное копирование блоков выполняется «как есть», что означает, что резервные копии также шифруются. По умолчанию Amazon Redshift самостоятельно управляет ключами, однако можно настроить управление ключами с использованием собственных аппаратных модулей безопасности (HSM) или службы AWS Key Management Service.[10],[11]

Модель затрат

Использование Amazon Redshift не сопряжено с долгосрочными вложениями или авансовыми платежами. Подобный подход к оплате освобождает вас от капитальных издержек и сложности планирования и приобретения избыточных ресурсов хранилища данных. Сумма оплаты зависит от размера и числа узлов в кластере.

Дополнительная плата за хранение резервных копий, которые занимают до 100 % предоставленного хранилища, не взимается. Например, если у вас есть активный кластер с двумя узлами XL и общим объемом хранилища 4 ТБ, AWS предоставит вам до 4 ТБ резервного хранилища в Amazon S3 без дополнительной платы. Плата за резервное хранилище, превышающее размер предоставленного хранилища, и хранение резервных копий по окончании использования кластера взимается по стандартным ставкам Amazon S3.[12] Плата за передачу данных в рамках обмена данными между системами Amazon S3 и Amazon Redshift не взимается. Дополнительные сведения см. в разделе Цены на Amazon Redshift.[13]

Идеальные сценарии использования

Amazon Redshift идеально подходит для оперативной аналитической обработки (OLAP) с использованием существующих средств бизнес-аналитики. Организации используют Amazon Redshift для решения следующих задач:

- Функционирование корпоративной системы бизнес-аналитики и составление отчетности
- Анализ глобальных данных о сбыте нескольких товаров
- Хранение данных о торговле акциями за прошлые периоды
- Анализ рекламных показов и переходов
- Агрегирование игровых данных
- Анализ тенденций в социальных сетях
- Измерение качества клинических услуг, эффективности работы и финансовых показателей учреждений здравоохранения

Нерекомендуемые сценарии

Amazon Redshift не рекомендуется использовать в следующих сценариях:

- **Небольшие наборы данных** – хранилище Amazon Redshift предназначено для параллельной обработки запросов в кластере. Если размер вашего набора данных менее 100 гигабайт, вы вряд ли сможете оценить все преимущества Amazon Redshift. В этом случае рекомендуется выбрать Amazon RDS.
- **OLTP** – хранилище Amazon Redshift предназначено для рабочих нагрузок хранилища данных, требующих чрезвычайного быстродействия и бюджетных функциональных возможностей в сфере аналитики. Если вам нужна система для быстрой обработки транзакций, имеет смысл выбрать традиционную систему реляционных баз данных на основе Amazon RDS или базы данных NoSQL, например Amazon DynamoDB.
- **Неструктурированные данные** – данные в Amazon Redshift должны быть структурированы по определенной схеме. Amazon Redshift не поддерживает произвольную структуру схемы для каждой строки. Если вы работаете с неструктурированными данными, необходимо выполнить процесс извлечения, преобразования и загрузки данных (ETL) с помощью Amazon EMR, чтобы подготовить данные к загрузке в Amazon Redshift. Что касается данных JSON, можно хранить пары «ключ-значение» и использовать в своих запросах собственные функции JSON.[14]
- **Данные больших двоичных объектов (BLOB)** – если вы планируете хранить BLOB-файлы (например, цифровое видео, изображения или музыку), разумно хранить данные в Amazon S3 и использовать в Amazon Redshift ссылки на их местоположение. В этом сценарии Amazon Redshift отслеживает метаданные (имя элемента, размер, дата создания, владелец, расположение и т. д.) о больших двоичных объектах, однако сами такие объекты хранятся в Amazon S3.

Переход на Amazon Redshift

Если вы приняли решение перейти от существующего хранилища данных к Amazon Redshift, выбор стратегии миграции зависит от нескольких факторов:

- Размер базы данных и ее таблиц
- Пропускная способность сети между исходным сервером и AWS
- Планируется осуществить переход и переключение на AWS сразу (одним действием) или постепенно (последовательностью действий)
- Темпы изменения данных в исходной системе
- Преобразования во время миграции
- Партнерский инструмент, который планируется использовать для перехода и операций ETL

Миграция в один шаг

Это хороший вариант для небольших баз данных, непрерывное функционирование которых не является обязательным. Клиенты могут извлечь существующие базы данных в качестве CSV-файлов, а затем с помощью таких сервисов, как AWS Import/Export Snowball, перенести наборы данных в Amazon S3 для последующей загрузки в Amazon Redshift. Затем клиент проверяет единообразие данных в источнике и конечной базе данных Amazon Redshift. После успешного прохождения всех проверок можно переключать базу данных на AWS.

Миграция в два шага

Миграция в два шага может использоваться для работы с базами данных любого размера.

1. **Перенос первоначальных данных:** данные извлекаются из исходной базы данных (предпочтительно в период неактивного использования, чтобы нарушение работы было минимальным); затем данные переносятся в Amazon Redshift по вышеописанному сценарию миграции в один шаг.

2. **Перенос измененных данных:** данные, измененные в исходной базе данных после переноса первоначальных данных, перед переключением передаются в конечное расположение; этот шаг позволяет синхронизировать исходную и конечную базы данных. После переноса всех измененных данных можно проверить данные в конечной базе данных, провести необходимые тесты и в случае успешного их прохождения переключиться на хранилище данных Amazon Redshift.

Инструменты для переноса баз данных

Доступно несколько инструментов и технологий для переноса данных. Некоторые из них можно использовать поочередно; кроме того, можно использовать и другие представленные на рынке сторонние инструменты и инструменты с открытым исходным кодом.

1. AWS Database Migration Service поддерживает оба вышеописанных сценария миграции: в один или два шага.[15] При выполнении сценария миграции в два шага необходимо включить дополнительное ведение журнала, чтобы зафиксировать изменения в исходной системе. Включить дополнительное ведение журнала можно на уровне таблиц или баз данных.
2. Доступны следующие дополнительные партнерские инструменты для интеграции данных:
   - Attunity
   - Informatica
   - SnapLogic
   - Talend
   - Bryte

Дополнительные сведения об интеграции данных и партнерах, предоставляющих консультации, см. в разделе Партнеры Amazon Redshift.[16]

Проектирование бизнес-процессов в хранилище данных

В предыдущих разделах мы обсуждали характеристики сервиса Amazon Redshift, благодаря которым он идеально подходит для хранения данных. Чтобы понять принципы проектирования бизнес-процессов хранилища данных с помощью Amazon Redshift, давайте рассмотрим наиболее распространенный шаблон проектирования и пример использования.

Допустим, международная компания по производству одежды имеет более тысячи розничных магазинов, продает определенные линейки одежды в универмагах и дисконт-центрах, а часть продукции реализует через интернет-магазин. С технической точки зрения в настоящее время эти три канала сбыта функционируют независимо друг от друга. У них разные схемы управления, системы обслуживания торговых точек и бухгалтерские отделы. Ни одна из используемых систем не позволяет объединить все связанные наборы данных и не обеспечивает обзорность происходящего в бизнесе в целом для исполнительного директора компании.

Допустим, исполнительному директору нужна информация о функционировании этих каналов сбыта в масштабах компании и возможность получить ответы на следующие вопросы с помощью специальной аналитики:

- Какие тенденции наблюдаются в разных каналах сбыта?
- В каких регионах у того или иного канала сбыта наилучшие показатели?
- Насколько эффективны рекламные мероприятия и специальные акции компании?
- Какие тенденции наблюдаются в разных линейках одежды?
- Какие внешние факторы влияют на объем продаж компании (например, уровень безработицы и погодные условия)?
- Как характеристики магазинов влияют на продажи, например срок пребывания в должности сотрудников и руководства, формат магазина (торговые ряды или торговый центр в отдельном здании), расположение товаров в магазине, специальные акции, расположение товаров на торцевых полках в магазине, инструкции и директивы по продажам и оформление витрин?

Корпоративное хранилище данных позволяет решить эту проблему. Оно собирает данные из разных информационных систем всех трех каналов сбыта, а также данные, находящиеся в общем доступе (прогнозы погоды, экономические отчеты и так далее). Каждый источник данных ежедневно отправляет данные в хранилище. Поскольку источники данных могут иметь разную структуру, для переформатирования данных и унификации их структуры используется процедура извлечения, преобразования и загрузки данных (ETL). После этого можно выполнять аналитические операции с данными из всех источников. Для этого используется следующая архитектура информационного потока.

Рис. 4. Организация бизнес-процесса в корпоративном хранилище данных

1. Первым шагом этой процедуры является поставка данных из разных источников в Amazon S3. Amazon S3 представляет собой экономичную и масштабируемую платформу для хранения данных с высокой степенью надежности, которая позволяет параллельно записывать данные из большого числа разных источников по минимальной цене.
2. Amazon EMR используется для очистки данных и преобразования из исходного в целевой формат. Amazon EMR имеет встроенные средства интеграции с Amazon S3, что позволяет направлять параллельные потоки данных с каждого узла кластера Amazon EMR в систему Amazon S3 и обратно.

Как правило, хранилище данных получает новые данные каждую ночь. Поскольку решать аналитические задачи в ночное время не требуется, единственным требованием к этому процессу преобразования является завершение до начала рабочего дня, когда руководству и другим бизнес-пользователям потребуется доступ к отчетам и панелям управления. Следовательно, использование спотового рынка Amazon EC2 позволяет дополнительно снизить себестоимость процесса ETL в таких системах.[17] Грамотная спотовая стратегия заключается в том, чтобы начинать торги по очень низким ценам в полночь и постепенно увеличивать цену до тех пор, пока не будет предоставлен ресурс. По мере приближения к сроку, если спотовые ставки не дали нужного результата, можно воспользоваться резервным вариантом – расценками по требованию – и выполнить требования к времени выполнения. Данные из каждого источника могут преобразовываться в сервисе Amazon EMR по-разному, однако, работая по модели оплаты AWS по факту использования, можно создать отдельные кластеры Amazon EMR для каждого преобразования и настроить ресурсы кластера в точном соответствии с объемом заданий по преобразованию данных, избегая конкуренции ресурсов, предназначенных для разных заданий.

3. В результате каждого задания преобразования данных форматированные, очищенные данные загружаются в Amazon S3. Amazon S3 снова используется здесь, потому что Amazon Redshift может загружать данные из Amazon S3 параллельно, используя несколько потоков из каждого узла кластера. Amazon S3 также предоставляет записи за прошлые периоды и служит источником форматированных достоверных сведений для нескольких систем. Хранимые в Amazon S3 данные могут использоваться и другими аналитическими инструментами, если в будущем возникнут дополнительные требования к работе с данными.

4. Amazon Redshift загружает, сортирует, распределяет и сжимает данные в таблицы, поэтому аналитические запросы могут эффективно обрабатываться в параллельном режиме. По мере увеличения объема данных и расширения бизнеса можно легко увеличить доступные ресурсы системы, добавив новые узлы.

5. Для визуализации аналитических данных можно использовать Amazon QuickSight или одну из многих партнерских платформ визуализации, подключающихся к Amazon Redshift с использованием ODBC или JDBC. После визуализации данных исполнительный директор и ее подчиненные могут просматривать отчеты, панели управления и диаграммы. Теперь руководство компании, используя эти данные, может принимать боле взвешенные решения о ресурсах компании, что в конечном итоге приведет к росту прибыли компании и доходов акционеров.

Подобную гибкую архитектуру легко расширить по мере роста бизнеса, появления новых каналов сбыта, запуска дополнительных мобильных приложений для работы с клиентами и увеличения числа источников данных. Достаточно нескольких щелчков на панели управления Amazon Redshift или нескольких API-вызовов.

Заключение

В настоящее время корпоративные организации переносят свои аналитические базы данных и решения из локальных решений в облако, чтобы воспользоваться основными преимуществами облачных технологий (простотой, высокой производительностью и экономичностью), в связи с чем наблюдается стратегический сдвиг в сфере хранения данных. В этом техническом описании дается комплексный обзор актуального состояния хранилищ данных на базе AWS. AWS предоставляет широкий ассортимент сервисов и имеет мощную партнерскую экосистему, что позволяет легко создавать и обслуживать корпоративные хранилища данных в облаке. В результате клиенты получают высокопроизводительную и экономичную аналитическую архитектуру, которую можно масштабировать в соответствии с динамикой развития вашего бизнеса, используя глобальную инфраструктуру AWS.

Авторский коллектив

Данный документ был подготовлен при участии следующих лиц и организаций.

- Бабу Элумалай (Babu Elumalai), архитектор решений, Amazon Web Services
- Грег Хайралла (Greg Khairallah), главный менеджер по развитию бизнеса, Amazon Web Services
- Паван Потукучи (Pavan Pothukuchi), главный менеджер по продуктам, Amazon Web Services
- Джим Гутенкауф (Jim Gutenkauf), старший разработчик технической документации, Amazon Web Services
- Мелани Генри (Melanie Henry), старший редактор технической документации, Amazon Web Services
- Шандер Матрубутам (Chander Matrubhutam), специалист по маркетингу продуктов, Amazon Web Services

Дополнительная литература

См. дополнительные справочные сведения в следующих источниках:

- Библиотека программного обеспечения Apache Hadoop[18]
- Рекомендации по работе с Amazon Redshift[19]
- Лямбда-архитектура[20]

Комментарии

1 https://www.forrester.com/report/The+Forrester+Wave+Enterprise+Data+Warehouse+Q4+2015/-/E-RES124041

2 http://aws.amazon.com/streaming-data/

3 http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html

4 http://docs.aws.amazon.com/redshift/latest/APIReference/Welcome.html

5 http://docs.aws.amazon.com/redshift/latest/mgmt/working-with-clusters.html#rs-about-clusters-and-nodes

6 http://docs.aws.amazon.com/redshift/latest/dg/c_redshift-and-postgres-sql.html

7 http://aws.amazon.com/redshift/partners/

8 https://aws.amazon.com/vpc/

9 https://aws.amazon.com/cloudtrail/

10 http://docs.aws.amazon.com/redshift/latest/mgmt/working-with-HSM.html

11 https://aws.amazon.com/kms/

12 http://aws.amazon.com/s3/pricing/

13 http://aws.amazon.com/redshift/pricing/

14 http://docs.aws.amazon.com/redshift/latest/dg/json-functions.html

15 https://aws.amazon.com/dms/

16 https://aws.amazon.com/redshift/partners/

17 http://aws.amazon.com/ec2/spot/

18 https://hadoop.apache.org/

19 http://docs.aws.amazon.com/redshift/latest/dg/best-practices.html

20 https://en.wikipedia.org/wiki/Lambda_architecture