ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

УПРАВЛЕНИЕ ЦИФРОВЫХ ОБРАЗОВАТЕЛЬНЫХ ТЕХНОЛОГИЙ

Кафедра «Математика и информатика»

Курс лекций

по дисциплине

«Анализ и поиск в больших базах данных»

Автор

Галабурдин А.В.

Ростов-на-Дону, 2024

Аннотация

«Лекции» предназначены для студентов заочной формы обучения направления 09.04.02 «Информационные системы и технологии», профиль: Искусственный интеллект, математическое моделирование и суперкомпьютерные технологии в разработке информационных систем.

Авторы

доцент, кандидат физико-математических наук,
доцент кафедры «Математика и информатика»
Галабурдин А.В.

Оглавление

Тема №1 Понятие "Большие данные"

1.Информация и особенности ее хранения и обработки

Человек постоянно получает информацию из окружающего мира, анализирует ее, выявляет существенные закономерности и таким образом познает мир. В процессе понимания информации, ее анализа и применения на практике у человека формируются знания. Одна и та же информация может приводить к появлению разных знаний у разных людей. Сформированные знания человек использует в своей деятельности. Информация для человека – это сведения, которые уменьшают существующую до их получения неопределенность знания.

Информация обладает определенными свойствами, такими как достоверность (отражает истинное положение дел), объективность (не зависит от чьего-либо мнения или суждения), полнота (достаточна для принятия решений), актуальность (необходима в данный момент), понятность (выражена на языке, понятном для человека) и доступность (имеется возможность ее получения).

Совокупность последовательных действий с информацией образует информационный процесс.

Информационные процессы могут быть как целенаправленными (объяснение нового материала на занятии), так и случайными (например, непроизвольное запоминание текста и мелодии рекламного ролика). Естественные информационные процессы протекают в биологических системах (в живой природе) и социальных системах (в обществе).

В природе получение, преобразование, хранение и использование информации являются условиями жизнедеятельности любого живого организма. Человек преобразует информацию с помощью головного мозга и центральной нервной системы, принимает на основе этой информации решения и выполняет определенные действия. В социуме люди постоянно общаются друг с другом, обмениваются информацией.

Искусственные информационные процессы искусственно порождаются людьми с помощью разнообразных технических

устройств для осуществления различных действий с информацией и происходят в социотехнических системах (например, пилоты управляют самолетом на основе информации с бортовых приборов) и технических системах (примером может служить мобильный телефон).

Выделяют следующие виды информационных процессов: получение, обработка, передача, хранение, поиск, кодирование и защита информации.

1. Получение информации осуществляется человеком с помощью органов чувств, различных приборов (термометра, барометра, весов, микроскопа и др.).

2. Обработка информации выполняется человеком, как в уме, так и с помощью вспомогательных средств (например, калькулятора). В результате обработки человек делает выводы, получает информацию, новую по форме представления или содержанию.

3. Передача информации осуществляется при помощи речи, жестов, мимики, условных сигналов (дыма костра, взмаха флажка, сигнала автомобиля), специальных средств (телеграфа, телефона, радио, телевидения, компьютерных сетей).

4. Хранение полученной информации необходимо для ее неоднократного использования. Человек сохраняет информацию как в собственной (внутренней), так и во внешней памяти, делая, например, записи в телефонной книжке или дневнике или тетради, мобильном телефоне или облачном хранилище и т. п.

5. Поиск информации можно провести оперативно, если информация упорядочена (номер телефона в телефонной книжке можно быстро найти по фамилии человека).

6. Кодирование информации осуществляется с помощью фонем (звуков), символов (письменная речь). С помощью специальных кодов люди стремятся представить информацию компактно (стенографическая запись, идеограммы), в форме, удобной для передачи или хранения (азбука Морзе), а также защитить информацию с помощью криптографических секретных кодов (шифров).

7. Защита информации необходима для предотвращения ее потери, искажения, умышленного уничтожения и незаконного использования (организация технической защиты каналов связи, дублирование блоков информации или защита информационных хранилищ от несанкционированного доступа с помощью пароля).

Все информационные процессы взаимосвязаны между собой: защита информации может осуществляться с помощью процесса кодирования, а кодирование информации невозможно без процесса обработки; обработка, в свою очередь, является важной частью процесса поиска, а поиск информации подразумевает процесс передачи; передача невозможна без хранения, а хранение информации дает возможность ее получения.

Автоматизируя информационные процессы, человек имеет возможность автоматизировать и свою информационную деятельность. Автоматизация этих процессов с помощью современных технических средств позволяет избавить человека от рутинных и однотипных действий с информацией, увеличить объем хранимой информации, повысить скорость ее обработки и передачи.

В современном мире информация – это один из важнейших ресурсов и в то же время одна из движущих сил развития человеческого общества. Результат фиксации, отображения информации на каком-либо материальном носителе, т. е. зарегистрированное на носителе представление сведений независимо от того, дошли ли эти сведения до какого-нибудь приемника и интересуют ли они его, называют данными.

Данные – это представление фактов и идей в формализованном виде, пригодном для передачи и обработки в некотором информационном процессе. Данные сами могут выступать как источник информации. Информация, извлекаемая из данных, может подвергаться обработке, и результаты обработки фиксируются в виде новых данных

Именованную совокупность данных, отражающую состояние объектов и их отношений в рассматриваемой предметной

области, называют базой данных (БД). БД используют для организации управления и автоматизации,

Для управления БД разработаны языковые и программные средства, предназначенные для создания, наполнения, обновления и удаления БД

Накопление достаточного количества данных привело к необходимости начать серьезный анализ имеющейся информации, возникла проблема подсчета получаемых данных.

Скорость анализа была не единственным вопросом, над которым размышляли ученые в середине XX в. Проблемой стали объемы хранилищ (в частности библиотек) в связи с ростом выпускаемых печатных трудов. Начиная с 1950-х гг. в результате решения вопросов хранения информации и ее быстрого анализа появляются центры обработки данных (ЦОД).

По мере увеличения объемов информации, большая доля которой представляла собой неструктурированные данные, вопросы корректной интерпретации информационных потоков становились все более актуальными и сложными.

Данные, которые сейчас называют большими данными (Big Data), имеют свою историю, хотя и не такую давнюю. Концепция больших данных возникла во времена мэйнфреймов (70-е гг. XX в.) и компьютерных вычислений. Наукоемкие вычисления всегда отличались сложностью и требовали обработки больших объемов информации. Ключевое событие в этой сфере произошло в 1970 г., когда британский ученый Эдгар Кодд описал реляционную модель данных (представляет собой набор двумерных таблиц), которая совершила переворот в способе хранения данных, их индексировании и извлечении из баз.

Реляционная модель позволила извлекать данные из базы путем простых запросов, которые определяли, что нужно пользователю, не требуя от него знаний о внутренней структуре данных или о том, где они физически хранятся.

Реляционные базы хранят данные в таблицах со структурой из одной строки на объект и одного столбца на атрибут. Такое отображение идеально подходит для хранения данных

с четкой структурой, которую можно разложить на базовые атрибуты.

В 1990-х гг. для анализа данных компаниям потребовалась технология, которая могла бы объединять и согласовывать данные из разнородных баз и облегчать проведение более сложных аналитических операций. Решение этой бизнес-задачи привело к появлению хранилищ данных и технологий создания БД (OLAP, NoSQL).

В 2004 г. корпорация Google предложила действенный подход к обработке огромного количества данных (MapReduce). Большие данные привели к появлению новых платформ для их обработки (Hadoop). В целом можно сказать, что Google создал то, что все сейчас называют Big Data. Сам термин Big Data впервые появился в прессе 3 сентября 2008 г., когда редактор журнала Nature Клиффорд Линч опубликовал статью на тему развития будущего науки с помощью технологий работы с большим количеством данных.
Сущность понятия Big Data

Появление больших данных связано с расширением источников информации в современном мире. Сейчас в качестве таковых могут выступать: непрерывно поступающие данные с измерительных устройств, события от радиочастотных идентификаторов, с устройств аудио- и видеорегистрации, потоки сообщений из социальных сетей, метеорологические данные, данные дистанционного зондирования земли, потоки данных о местонахождении абонентов сетей сотовой связи, GPS-сигналы от автомобилей для транспортной компании, информация о транзакциях всех клиентов банка, всех покупках в крупной ретейл сети и многое другое. Термин «большие данные» относится к наборам данных, размер которых превосходит возможности типичных БД по занесению, хранению, управлению и анализу информации

Большие данные – это наборы данных, размеры которых выходят за пределы возможностей по сбору, хранению, управлению и анализу, присущих обычному программному обеспечению БД.

Слово «большие» – это не только возросший объем, но и возросшая скорость передачи и разнообразие источников данных. Таким образом, приходится иметь дело не просто с большим количеством данных, а с тем, что они поступают очень быстро, в сложных формах и из разнообразных источников.

Большим данным свойственны следующие особенности.

– Они часто автоматически генерируются машиной без участия человека (так, встроенный в двигатель датчик генерирует данные, даже если никто его об этом не просит), в то время как традиционные источники данных всегда предполагают присутствие человека, совершающего какие-либо действия (например, выставление счетов на оплату, телефонные звонки и др.).

– Большие данные обычно соотносятся с совершенно новыми источниками данных.

– Данные могут быть структурированными, неструктурированными, полуструктурированными или даже мультиструктурированными. Большие данные часто описываются как неструктурированные, а традиционные данные – как структурированные, т. е. представляемые в четко предопределенном, неизменном формате, что облегчает работу с ними. Источники неструктурированных данных невозможно контролировать. Значительная часть данных относится к категории полуструктурированных. Они подразумевают логическую схему и формат, который может быть понятным, но «недружественным» к пользователю. Полуструктурирован-ные данные иногда называют мультиструктурированными. В потоке таких данных кроме ценных фрагментов информации может присутствовать множество ненужных и бесполезных данных. Чтобы прочитать полуструктурированные данные, необходимо использовать сложные правила, которые определяют, что следует делать после чтения каждого фрагмента информации.

– Некоторые источники больших данных могут не учитывать правила грамматики, синтаксиса или лексические

нормы. Работать с такими данными бывает очень трудно, а иногда и не совсем приятно.

– Потоки больших данных не всегда представляют собой особую ценность, могут быть бесполезными. Требуется сортировка информации и извлечение ее ценных и релевантных (соответствующих) фрагментов. Традиционные же источники данных с самого начала разрабатывались так, чтобы содержать на 100 % релевантные данные. Это было связано с ограничениями масштабируемости: включение в поток данных чего-то неважного слишком дорого обходилось.
Сейчас люди не ограничены объемом носителей информации. Поэтому большие данные по умолчанию включают всю возможную информацию, в которой приходится разбираться. В этом случае ничего не будет упущено, но усложняется процесс анализа данных.

– С большими данными связаны определенные риски. Так, например, организация может оказаться настолько перегруженной большими данными, что не будет способна на какой-либо прогресс; расходы по сбору больших данных могут расти быстрее, чем возможности организации по их использованию и др.

Большие данные интересны для организаций не только тем, что они «большие», не только с точки зрения важности учета перечисленных выше особенностей, но и тем, что их использование на благо организации требует внедрения новых инновационных средств анализа. Без них использование больших данных станет невозможным.

Big Data – это серия подходов, инструментов и методов обработки структурированных и неструктурированных данных огромных объемов и значительного многообразия для получения воспринимаемых человеком результатов, эффективных в условиях непрерывного прироста данных, распределения по многочисленным узлам вычислительной сети, альтернативных традиционным системам управления базами данных и решениям класса Business Intelligence.

Толкование термина Big Data предполагает нечто большее, чем просто анализ больших данных. Проблема не в том,

что организации создают огромные объемы данных, а в том, что бóльшая их часть представлена в формате, плохо соответствующем традиционному структурированному формату больших данных. Данные хранятся во множестве разнообразных хранилищ, иногда даже за пределами организации. В результате фирмы могут иметь доступ к огромному объему своих данных, но не иметь необходимых инструментов, чтобы установить взаимосвязи между этими данными и сделать на их основе соответствующие выводы.

Итак, под Big Data будем понимать технологии работы с информацией огромного объема и разнообразного состава, часто обновляемой и находящейся в разных источниках («большие данные») в целях увеличения эффективности работы, создания новых продуктов и повышения конкурентоспособности предприятия.

Тема №2 Основы управления большими данными

Подходы к управлению Big Data

Использование больших данных потребовало решения вопросов, связанных с хранением и обработкой информации. В результате были выявлены три направления, на которых стоит сосредоточиться для решения вопросов управления данными: Volume (объем), Velocity (скорость) и Variety (разнообразие). Позже они легли в основу описательной модели больших данных под названием 3V (VVV).

1. Volume – объем. Big Data – это целый набор методик и технологий получения, хранения и обработки информации, так как информация постоянно меняется: имеющаяся обновляется, к ней добавляется новая. При работе с большими массивами информации необходимо быть готовым к оперативному горизонтальному масштабированию из-за потенциального роста входящих данных.
2. Velocity – скорость. При постоянном росте количества данных важна их обработка с той скоростью, которую требуют цели проекта. Например, огромное количество датчиков фиксируют сейсмические изменения на территории конкретной

страны или в мире, данные с них поступают в ЦОД, где выполняются обработка и анализ полученной информации.

Если поступившие данные в силу тех или иных причин будут обрабатываться несколько часов вместо, например, нескольких минут, то в случае получения информации о землетрясении после обработки данных будет невозможно вовремя принять превентивные меры и последствия катастрофы будут ужасными. Не случайно к уже имеющимся параметрам V со временем был добавлен еще один – Value, или ценность информации. В приведенном примере эта ценность была равна нулю, так как потеряла свою актуальность раньше, чем ею смогли воспользоваться.

Validity – полезного действия этой информации.

Ценность информации заключается и в ее достоверности (Veracity).

Скорость обработки поступающих данных очень важна, иначе можно потерять их ценность и передать на дальнейший анализ уже неактуальные данные или в качестве результата предоставить неактуальную информацию. И скорость обработки данных, и хранилища должны легко наращиваться при необходимости, что также заложено в технологии Big Data

3. Variety – разнообразие. Как уже отмечалось выше, наряду со структурированными данными есть информация, которая поступает в неструктурированном виде. И именно она преобладает в общем потоке информации

Одна из задач, которая ставится перед использованием Big Data (в большей степени, чем хранение информации), – оперативно выстроить между полученными данными связи и на выходе выдать данные, доступные для структурированного или полуструктурированного анализа. Уметь находить связи между любыми данными вне зависимости от уровня их структурированности и уметь получать результат, который можно однозначно анализировать для решения той или иной задачи, является для Big Data весьма важным. Кроме того, система должна быть хорошо масштабируемой, иначе будут

получены недостоверные данные ввиду потери одного из параметров V.

Направления работы по управлению Big Data должны основываться на определенных принципах.

1. Горизонтальная масштабируемость. Поскольку данных может быть очень много, то любая система, которая подразумевает обработку больших данных, должна быть расширяемой.

2. Отказоустойчивость. Методы работы с большими данными должны учитывать возможность выхода из строя машин (а их может быть много – до нескольких тысяч) и способность преодолевать эти проблемы без каких-либо значимых последствий.

3. Локальность данных. В больших распределенных системах данные рассредоточены по большому количеству машин. Принцип локальности данных заключается в том, чтобы по возможности обрабатывать данные на той же машине, где они хранятся.

Содержание и задачи процесса управления большими данными

Управление большими данными строится с учетом так называемого «жизненного пути» данных (или, по-другому, истории данных) внутри организации. Существует несколько моделей «пути». Одной из них является модель Малькольма Чисхолма. Она состоит из семи активных фаз взаимодействия с данными. Каждая фаза содержит в себе задачи по управлению данными.

1-я фаза. Data Capture – создание или сбор значений данных, которые еще не существуют и никогда не существовали в компании. Сюда относят: а) Data Acquisition – покупку данных, предложенных внешними компаниями; б) Data Entry – генерацию данных ручным вводом при помощи мобильных устройств или программного обеспечения; в) Signal Reception – получение данных с помощью телеметрии (Интернет вещей).

2-я фаза. Data Maintenance – передача данных в точки, где происходит синтез данных и их использование в форме, наиболее подходящей для этих целей. Фаза часто включает в

себя такие задачи, как перемещение, интеграция, очистка, обогащение, изменение данных, а также процессы экстракции (извлечения), преобразования и загрузки. Смешение и интеграция данных нужны, если есть несколько разных источников данных, и нужно анализировать эти данные в комплексе. Например, магазин ведет торговлю офлайн и через Интернет (в том числе через маркетплейсы). Чтобы получить полную информацию о продажах и спросе, надо собрать множество данных: кассовые чеки, товарные остатки на складе, интернет-заказы, заказы через маркетплейс и т. д. Все эти данные поступают из разных мест и обычно имеют разный формат. Чтобы работать с ними, их нужно привести к единому виду.

Традиционные методы интеграции данных основаны на процессе EIL – извлечения, преобразования и загрузки. Данные получают из разных источников, очищают и загружают в хранилище. Специальные инструменты экосистемы больших данных от Hadoop до баз NoSQL также имеют собственный подход для извлечения, преобразования и загрузки данных. После интеграции большие данные подвергаются дальнейшим манипуляциям: анализу, обработке и т. д.

3-я фаза. Data Synthesis – создание ценности из данных через индуктивную логику (занимается логическими процессами умозаключений от частного к общему – индукцией), использование других данных в качестве входных данных.

4-я фаза. Data Usage – применение данных как информации для задач, которые должно ставить и выполнять предприятие. Использование данных имеет специальные задачи управления данными. Одна из них заключается в выяснении того, является ли законным использование данных в том виде, в котором хочет бизнес. Речь идет о так называемом «разрешенном использовании данных», поскольку могут существовать регулирующие или контрактные ограничения на фактическое использование данных, а их необходимо соблюдать.

5-я фаза. Data Publication – отправка данных в место за пределами предприятия, например отсылка ежемесячных отчетов клиентам, после чего эти данные де-факто невозможно

отозвать. Неверные значения данных не могут быть исправлены, поскольку они уже недоступны для предприятия. Управление данными может потребоваться, чтобы решить, как будут обрабатываться неверные данные, которые были отправлены клиентам.
6-я фаза. Data Archival – копирование данных в среду, где они хранятся, до тех пор, пока не они понадобятся снова, для их активного использования и удаления из всех активных производственных сред.
7-я фаза. Data Purge – удаление каждой копии элемента данных с предприятия. В идеале это необходимо делать из архива. Задача управления данными на этой фазе – определить, что очистка действительно была выполнена должным образом.

Данные не обязательно должны проходить все семь фаз; фазы взаимодействия не обязательно выстраиваются в конкретную последовательность; в реальности фазы могут проявляться в хаотичном порядке.

Тема №3 Технологии работы с большими данными

1. *Становление технологии работы с большими данными*

Еще в конце XX в. многие организации столкнулись с тем, что существующих IT-решений уже не хватало, чтобы справиться с увеличивающимися потоками данных, которые выходили далеко за пределы оперативной памяти. Потребовались новые технологии хранения и анализа информации

При работе с большим объемом данных, когда заканчиваются ресурсы, есть два возможных решения: вертикальное или горизонтальное масштабирование.

При использовании вертикального масштабирования добавляется больше вычислительной мощности в машину (например, в центральный процессор, оперативное записыва-

ющее устройство). В горизонтальном масштабировании добавляется больше машин одинаковой емкости для распределения рабочей нагрузки.

Вертикальное масштабирование проще в управлении и контроле, чем горизонтальное, и доказано, что оно эффективно при работе с проблемами сравнительно небольшого размера. Однако горизонтальное масштабирование обычно дешевле и быстрее вертикального масштабирования при работе с большой задачей.

Ко времени появления больших объемов информации вертикальное масштабирование больше не обеспечивало нужды бизнеса. Компаниям требовались отказоустойчивые и хорошо горизонтально масштабируемые технологии.

Алгоритм MapReduce, созданный в свое время корпорацией Google и основанный на горизонтальном масштабировании, стал ответом на запрос бизнеса.
MapReduce представляет собой технологию разбиения процесса обработки на две простые функции: Map и Reduce.

Единую задачу разбивают на бесконечно большое количество малых подзадач, которые будут выполняться параллельно друг с другом, а потом полученный результат просто складывают. Таким образом, в MapReduce входные данные делятся на множество частей, каждая их которых затем отправляется на другой компьютер для обработки и последующего агрегирования в соответствии с заданной функцией группировки. Каждую часть одной большой задачи можно отдать на обработку одному из узлов единого кластера.

Кластер – это группа серверов (именуемых нодами), которые работают вместе, выполняют общие задачи, и клиенты видят их как одну систему. Серверов в кластере может быть много.

Благодаря специальному оборудованию и программному обеспечению реализуется такой уровень защиты от сбоев, который невозможен при использовании одного сервера. В случае выхода из строя одного из серверов задачи, которые он выполнял, берет на себя другой сервер, и работоспособность системы восстанавливается. При этом пользователи замечают

лишь временную потерю работоспособности, а иногда и вовсе ничего не замечают (кроме небольшой паузы). При увеличении объемов информации кластер нужно расширять до заданных задачей размеров.

Таким образом, алгоритм MapReduce представляет собой модель для распределенных вычислений, а кластер компьютеров используется для распараллеливания больших данных, что упрощает их обработку. Происходит распределение входных данных на рабочие узлы (individual nodes) распределенной файловой системы для предварительной обработки, а затем свертка (объединение) уже предварительно обработанных данных.

Для получения итоговой суммы алгоритм будет параллельно вычислять промежуточные суммы в каждом из узлов распределенной файловой системы и затем суммировать эти промежуточные значения.

В алгоритме MapReduce обработка данных происходит в три стадии

1.Стадия Map. Работа на этой стадии заключается в предобработке и фильтрации данных в функциональных языках программирования. Функция Map, примененная к одной входной записи, выдает множество пар «ключ – значение» (может выдать только одну запись, может не выдать ничего, а может выдать несколько пар «ключ – значение»). Что будет находиться в ключе и в значении – решать пользователю, но ключ – очень важная вещь, так как данные с одним ключом в будущем попадут в один экземпляр функции Reduce.

2. Стадия Shuffle. Проходит незаметно для пользователя. В этой стадии вывод функции Map разбирается по «корзинам»: каждая «корзина» соответствует одному ключу вывода стадии Map. В дальнейшем эти «корзины» послужат входом для Reduce.

3.Стадия Reduce. Каждая «корзина» со значениями, сформированная на стадии Shuffle, попадает на вход функции Reduce.

Функция Reduce задается пользователем и вычисляет финальный результат для отдельной «корзины». Множество

всех значений, возвращенных функцией Reduce, является финальным результатом MapReduce-задачи

Особенности алгоритма MapReduce.

1. Все запуски функции Map и Reduce работают независимо и могут работать параллельно, в том числе на разных машинах кластера.
2. Shuffle внутри себя представляет параллельную сортировку, поэтому также может работать на разных машинах кластера.

Пункты 1 – 2 обеспечивают принцип горизонтальной масштабируемости.

3. Функция Map, как правило, применяется на той же машине, где хранятся данные, это позволяет снизить передачу данных по сети (принцип локальности данных).
4. MapReduce – это всегда полное сканирование данных, что означает, что алгоритм плохо применим, когда ответ требуется очень быстро.

Предложенный алгоритм MapReduce стал отправной точкой для создания систем, работающих с большими данными (социальные сети, Интернет вещей, банковский сектор, научно-исследовательская сфера и др.), и помог компании Google повысить эффективность своего поискового ресурса.

Классический алгоритм MapReduce имеет одну особенность: вся цепочка результатов работы алгоритма сохраняется в дисковую подсистему. А в ней операций чтения и записи очень много, что влияет на время работы алгоритма. Заложив основы работы с большими данными, алгоритм MapReduce инициировал появление новых, более совершенных инструментов управления ими.

2. ***Современные технологии управления большими данными***

Проблемы использования алгоритма MapReduce попытались решить с помощью создания новых инструментов, переводящих бóльшую часть вычислений в оперативную память. Так появились такие инструменты, как Hadoop, Spark, Pig,

Hive, Cassandra и Kafka, каждый из которых имеет свои преимущества и недостатки. Остановимся на некоторых из них: Hadoop и Spark. Их появление относится к началу 2000-х гг.

1. Платформа Hadoop – это набор программ с открытым исходным кодом, написанных на Java, которые можно использовать для выполнения операций с большим объемом данных. Hadoop – это масштабируемая, распределенная и отказоустойчивая экосистема.

Платформа включает несколько десятков проектов, которые работают самостоятельно или в комплексе с другими для создания систем, решающих конкретные задачи. В состав Hadoop входят инструменты, покрывающие все аспекты работы с большими данными: файловые системы (HDFS, MapR-FS); фреймворки для выполнения распределенных вычислений (MapReduce, Spark); NoSQL-базы и SQL движки (HBase, Hive, Spark SQL); инструменты для захвата данных из внешних источников и интеграции с реляционными системами управления БД (СУБД) – Flume, Kafka, Sqoop; инструменты для построения потоков обработки и загрузки данных, в том числе непрерывно поступающих (Spark Streaming, Storm, Flink, NiFi) и др.

Основные компоненты Hadoop:

– Hadoop MapReduce – используется для загрузки данных из БД, их форматирования и проведения количественного анализа;

– Hadoop YARN – планирует ресурсы системы и управляет ими, разделяя рабочую нагрузку на кластер машин;

– распределенная файловая система Hadoop (HDFS) – кластерная система хранения файлов любого типа в любом возможном формате, разработанная для обеспечения отказоустойчивости, высокой пропускной способности данных.

К преимуществам платформы Hadoop относят:

– сокращение времени на обработку данных;

– снижение стоимости оборудования;

– повышение отказоустойчивости;

– линейную масштабируемость;

– работу с неструктурированными данными

2. Платформа Apache Spark. Она отличается скоростью работы, которая примерно в сто раз выше, чем у MapReduce (промежуточные результаты не сохраняются и все выполняется в памяти).

Ее обычно используют для чтения хранимых данных и данных в реальном времени, предварительной обработки большого количества данных (SQL), анализа данных с помощью машинного обучения и графовых сетей.

Apache Spark можно использовать с такими языками программирования, как Python, R и Scala. Для запуска Spark обычно используются облачные приложения, такие как Amazon Web Services, Microsoft Azure и Databricks.

При использовании Spark большие данные распараллеливаются с использованием эластичных распределенных наборов данных (RDDs). Они являются отказоустойчивыми и могут восстанавливать потерянные данные в случае сбоя любого из узлов.

RDDs можно использовать для выполнения двух типов операций в Spark: преобразования и действия. Преобразования создают новые наборы данных из RDDs (Resilient Distributed Dataset) и возвращают их в результате RDDs (например, отображают, фильтруют и сокращают по ключевым операциям). Все преобразования выполняются только один раз, когда вызывается действие (они помещаются в карту выполнения, а затем выполняются, когда вызывается действие).

Обе платформы позволяют успешно работать с большими данными. Hadoop была первой системой, которая сделала MapReduce доступной в большом масштабе, однако в настоящее время многие компании отдают предпочтение Apache Spark.

3.Общие черты и различия платформ Hadoop и Spark.

Hadoop и Spark, являясь средами больших данных, не выполняют одни и те же задачи, они не являются взаимоисключающими, поскольку могут работать вместе. Распределенное

хранилище является основополагающим для многих современных проектов больших данных, поскольку позволяет хранить огромные наборы данных на почти бесконечном количестве жестких дисков компьютера.

Однако Spark не имеет своей собственной системы для организации файлов распределенным способом (файловой системы), поэтому для нее требуется система, предоставленная третьей стороной. По этой причине многие проекты больших данных включают установку Spark поверх Hadoop, где современные аналитические приложения Spark могут использовать данные, хранящиеся с использованием распределенной файловой системы Hadoop (HDFS).

Преимущество Spark над Hadoop заключается в скорости. Spark выполняет большинство своих операций «в памяти», копируя их из распределенного физического хранилища в гораздо более быструю логическую оперативную память. Это сокращает время записи и чтения по сравнению с Hadoop MapReduce.

Функциональность Spark для решения сложных задач обработки данных, таких как обработка потоков, в реальном времени и машинное обучение, намного превосходит возможности, которые предоставляются Hadoop. Наряду с приростом скорости это является реальной причиной роста популярности Hadoop. Обработка в режиме реального времени означает, что данные могут быть переданы в аналитическое приложение в тот момент, когда они были получены, и немедленно передаются пользователю через панель мониторинга, чтобы можно было предпринять какое-либо действие. Этот вид обработки все чаще используется во всех видах приложений для работы с большими данными.

Алгоритмы создания машинного обучения являются областью аналитики, которая хорошо подходит платформе Spark благодаря ее скорости и способности обрабатывать потоковые данные. Этот вид технологии используется в новейших передовых производственных системах, которые могут предсказать, например, когда детали машин и станков на предприятии выйдут из строя и когда нужно сделать заказ на их

замену; они лежат в основе работы автомобилей и кораблей без водителя.

Spark поддерживает многие технологии кластерных вычислений и имеет несколько библиотек-надстроек для решения распространенных аналитических задач, включая Spark SQL (SQL-подобные запросы к данным), MLlib (алгоритмы машинного обучения), GraphX (анализ графов) и Spark Streaming (обработка потоковых данных).

Отличия платформ Hadoop и Spark по ряду критериев

Критерий - Функционал

Hadoop-Формирует инфраструктуру распределенных данных: большие коллекции данных распределены между множеством узлов, образующих кластер из стандартных серверов, что не требует покупки специализированного оборудования, индексирует и отслеживает состояние данных, что делает их обработку и анализ эффективнее

Apache Spark-Позволяет выполнять различные операции над распределенными коллекциями данных, но не обеспечивает их распределенного хранения

Критерий - Использование Hadoop - В состав Hadoop входит и компонент хранения Hadoop Distributed File System, и компонент обработки MapReduce, поэтому обработку можно осуществлять без Spark

Apache Spark- Может использоваться без Hadoop, но не имеет собственной системы управления файлами, поэтому необходима интеграция либо с HDFS, либо с какой-то другой облачной платформой хранения данных

Критерий - Скорость работы

Hadoop- Работает медленнее из-за пошагового режима обработки в MapReduce

Apache Spark- Работает быстрее, так как оперирует всем набором данных как единым целым

Критерий - Устойчивость к сбоям

Hadoop- Устойчива к системным сбоям, поскольку после выполнения каждой операции данные записываются на диск

Apache Spark- Восстановление после сбоев осуществляется благодаря тому, что объекты данных хранятся в распределенных в пределах кластера наборах

Критерий- Аналитические возможности

Hadoop- Не имеет библиотеки машинного обучения, должна связываться со сторонней библиотекой, например, с Apache Mahout

Apache Spark- Включает собственные библиотеки машинного обучения – MLib

Иногда платформы Hadoop и Spark считают конкурентами, стремящимися к доминированию, но на самом деле это не так. У них существует некоторое пересечение функций, обе платформы являются некоммерческими продуктами. Все зависит от потребностей конкретной компании. Если в компании большие данные состоят только из огромного количества очень структурированных данных (например, имен и адресов клиентов), то расширенная функциональность потоковой аналитики и машинного обучения, предоставляемая Spark, вообще не требуется. Выбирается платформа Hadoop.

Любую из двух технологий (Hadoop и Spark) можно использовать отдельно, не обращаясь к другой. Вместе с тем технология Spark проектировалась для Hadoop, поэтому многие считают, что лучше все же использовать их вместе. Неслучайно точка зрения о том, что они дополняют друг друга, является все-таки преобладающей.

Тема №4 Хранилища данных

1. *Большие данные и хранилища данных*

В начале 80-х гг. XX в., в период бурного развития регистрирующих информационных систем, появилось осознание ограниченности их применения для анализа данных и построения систем поддержки принятия решений. Менеджерам и аналитикам требовались системы, которые бы позволяли: анализировать информацию во временном аспекте, формиро-

вать произвольные запросы к системе, обрабатывать большие объемы данных, интегрировать данные из различных регистрирующих систем (данные обычно хранились в многочисленных разрозненных базах в рамках одной организации).

Используемые регистрирующие системы не удовлетворяли ни одному из этих требований: информация в такой системе актуальна только на момент обращения к БД, а в следующий момент времени по тому же запросу можно получить совершенно другой результат. Интерфейс регистрирующих систем рассчитан на проведение жестко определенных операций и возможности получения результатов на нерегламентированный (ad-hoc) запрос сильно ограничены.

Возможности обработки больших массивов данных также были невелики из-за настройки СУБД на выполнение коротких транзакций. Базы были оптимизированы для хранения и извлечения информации путем простых операций, таких как select, insert, update и delete. Для анализа данных компаниям требовалась технология, которая могла бы объединять и согласовывать данные из разнородных баз и облегчать проведение более сложных аналитических операций. Решение этой бизнес-задачи привело к появлению хранилищ данных.

Хранилище данных – это система, в которой собраны данные из различных источников внутри компании, которые используются для поддержки принятия управленческих решений.

Основная задача организации хранилищ данных – создание хорошо спроектированного централизованного банка данных. Основное преимущество хранилища данных – это сокращение времени выполнения проекта. Ключевой компонент любого процесса обработки данных – это сами данные, поэтому неудивительно, что во многих проектах бóльшая часть времени и усилий направляется на поиск, сбор и очистку данных перед анализом. Если в компании есть хранилище данных, то усилия и время, затрачиваемые на подготовку данных, значительно сокращаются.

Для описания стандартных процессов и инструментов для сопоставления, объединения и перемещения данных между базами используется термин ETL (Extract, Transform, Load) – извлечение, преобразование, загрузка.

Типичные операции, выполняемые в хранилище данных, отличаются от операций в стандартной реляционной БД. Для их описания используется термин «интерактивная аналитическая обработка» (OLAP). Операции OLAP, как правило, направлены на создание сводок исторических данных и включают в себя сбор данных из нескольких источников.

По сути, операции OLAP позволяют пользователям распределять, фрагментировать и переворачивать данные в хранилище, а также получать их различные отображения. Операции OLAP работают с отображением данных, называемым кубом данных, который построен поверх хранилища. Куб данных имеет фиксированный, заранее определенный набор измерений, где каждое измерение отображает одну характеристику данных.

Основное преимущество использования куба данных с фиксированным набором измерений состоит в том, что он ускоряет время отклика операций OLAP. Кроме того, поскольку набор измерений куба данных предварительно запрограммирован в систему OLAP, эти системы могут быть отображены дружественным пользовательским интерфейсом (GUI – graphical user interface – графический интерфейс пользователя) для формулирования запросов OLAP. Однако отображение куба данных ограничивает анализ набором запросов, которые могут быть сгенерированы только с использованием заранее определенных измерений.

Интерфейс запросов SQL сравнительно более гибок. Кроме того, хотя системы OLAP полезны для исследования данных и составления отчетов, они не позволяют моделировать данные или автоматически выявлять в них закономерности.

Появление больших данных привело к разработке новых технологий создания БД. БД нового поколения часто называют базами NoSQL. Они имеют более простую модель, чем

привычные реляционные БД, и хранят данные в виде объектов с атрибутами, используя такой язык представления объектов, как JavaScript Object Notation (JSON).

Преимущество использования объектного представления данных (по сравнению с моделью на основе реляционной таблицы) состоит в том, что набор атрибутов для каждого объекта заключен в самом объекте, а это дает возможность гибко отображать данные.

Например, один из объектов в БД может иметь сокращенный набор атрибутов по сравнению с другими объектами. В структуре реляционной БД, напротив, все значения в таблице должны иметь одинаковый набор атрибутов (столбцов). Эта гибкость важна в тех случаях, когда данные (из-за их разнообразия или типа) не раскладываются естественным образом в набор структурированных атрибутов. Однако, хотя эта гибкость представления позволяет собирать и хранить данные в различных форматах, для последующего анализа их все равно приходится структурировать.

2.Подходы к архитектуре и принципам проектирования хранилищ данных

В основе концепции хранилища данных лежат две основные идеи: интеграция разъединенных детализированных данных в едином хранилище и разделение наборов данных и приложений, используемых для обработки и анализа.

Выделяют следующие подходы к проектированию хранилищ.

1.Подход «снизу-вверх» Кимбалла. Он основывается на важности витрин данных, которые являются хранилищами данных, принадлежащих конкретным направлениям бизнеса. По мнению автора подхода, хранилище данных – это просто сочетание различных витрин данных, которые облегчают отчетность и анализ данных.

2.Нисходящий подход Инмона. Он основывается на том, что хранилище данных является централизованным хранилищем всех корпоративных данных. При таком подходе организация сначала создает нормализованную модель хранилища

данных, а затем создаются витрины размерных данных на основе модели хранилища

При проектировании системы по методологии Р. Кимбалла фронтендом БД должна выступать витрина данных – Data Mart, которая использует Analysis Services для куба в качестве источника данных. Методология Б. Инмона сложнее Data Mart'а, включает в себя не только БД, но и систему поддержки принятия решений и клиент-серверную архитектуру, тогда как Data Mart по сути является БД, созданной с учетом требований будущих кубов

Соответственно, исходя из предложенных подходов, можно выделить разную архитектуру хранилища данных. В первом случае используется двухуровневая архитектура. Она предполагает построение витрин данных (Data Mart) без создания центрального хранилища, информация поступает из регистрирующих систем (OLTP) и ограничена конкретной предметной областью.

При построении витрин используются основные принципы построения хранилищ данных, поэтому их можно считать хранилищами данных в миниатюре.

Такая архитектура имеет свои плюсы: простота и малая стоимость реализации; высокая производительность за счет физического разделения регистрирующих и аналитических систем, выделения загрузки и трансформации данных в отдельный процесс, оптимизированный под анализ структуры хранения данных; поддержка истории; возможность добавления метаданных.

Во втором случае построение полноценного корпоративного хранилища данных (Data Warehouse) выполняется в трехуровневой архитектуре.

На первом (нижнем) уровне расположены разнообразные источники данных: внутренние регистрирующие системы, справочные системы, внешние источники (данные информационных агентств, макроэкономические показатели и др.).

Здесь находится сервер БД, используемый для извлечения данных из множества различных источников.

Второй (средний) уровень содержит центральное хранилище, куда стекается информация от всех источников с первого уровня, и, возможно, оперативный склад данных, который не содержит исторических данных и выполняет две основные функции. Во-первых, он является источником аналитической информации для оперативного управления и, во-вторых, здесь подготавливаются данные (осуществляется их преобразование и проводятся определенные проверки) для последующей загрузки в центральное хранилище. Наличие оперативного склада данных просто необходимо при различных регламентах поступления информации из источников.

Третий (верхний) уровень представляет собой набор предметноориентированных витрин данных, источником информации для которых является центральное хранилище данных. Именно с витринами данных и работает большинство конечных пользователей. Он содержит инструменты, используемые для высокоуровневого анализа данных, создания отчетов клиентами.

В традиционной архитектуре хранилищ выделяют несколько моделей: виртуальное хранилище, витрину данных и корпоративное хранилище данных.

1. Виртуальное хранилище данных – это набор отдельных БД, которые можно использовать совместно, чтобы можно было эффективно получать доступ ко всем данным, как если бы они хранились в одном хранилище данных.

2. Модель витрины данных используется для отчетности и анализа конкретных бизнес-процессов. В этой модели хранилища находятся агрегированные данные из ряда исходных систем, относящихся к конкретной бизнес-сфере, например, продажам, клиентам или финансам.

3. Модель корпоративного хранилища данных предполагает хранение агрегированных данных, охватывающих всю организацию. Эта модель рассматривает хранилище данных как сердце информационной системы предприятия с интегриро-

ванными данными всех бизнес-единиц. Различают два архитектурных направления построения хранилищ: нормализованные хранилища данных и хранилища с измерениями

В нормализованных хранилищах данные находятся в предметно ориентированных таблицах третьей нормальной формы (нормальная форма – это требование, предъявляемое к структуре таблиц в теории реляционных БД для устранения из них избыточных функциональных зависимостей между атрибутами (полями таблиц)). Они считаются простыми в создании и управлении.

К недостаткам нормализованных хранилищ можно отнести большое количество таблиц вследствие нормализации, из-за чего для получения какой-либо информации нужно делать выборку из многих таблиц одновременно, что приводит к ухудшению производительности системы.

Хранилища с измерениями используют разные типы схем хранения, такие как «звезда» и «снежинка». Одно измерение куба может содержаться как в одной таблице (в том числе и при наличии нескольких уровней иерархии), так и в нескольких связанных таблицах, соответствующих различным уровням иерархии в измерении. Если каждое измерение содержится в одной таблице, такая схема хранилища данных носит название «звезда» (star schema).

В центре схемы «звезда» находятся данные (таблица фактов), а измерения образуют ее лучи. Таблица фактов содержит агрегированные данные, которые будут использоваться для составления отчетов, а таблица измерений описывает хранимые данные. Достаточно простая конструкция звездообразной схемы значительно упрощает написание сложных запросов.

Если же хотя бы одно измерение содержится в нескольких связанных таблицах, такая схема хранилища данных носит название «снежинка» (snowflake schema). Дополнительные таблицы измерений в такой схеме, обычно соответствующие верхним уровням иерархии измерения и находящиеся в соот-

ношении «один ко многим» в главной таблице измерений, соответствующей нижнему уровню иерархии, иногда называют консольными таблицами (outrigger table).

Схема разбивает таблицу фактов на ряд денормализованных таблиц измерений. Денормализованные проекты менее сложны, потому что данные сгруппированы. Таблица фактов использует только одну ссылку для присоединения к каждой таблице измерений. Схема типа «снежинка» отличается тем, что использует нормализованные данные. Нормализация означает эффективную организацию данных, чтобы все зависимости данных были определены, и каждая таблица содержала минимум избыточности. Таким образом, таблицы измерений разветвляются на отдельные таблицы измерений.

Эта схема использует меньше дискового пространства и лучше сохраняет целостность данных. Основной ее недостаток – сложность запросов, необходимых для доступа к данным: каждый запрос должен пройти несколько соединений таблиц, чтобы получить соответствующие данные.

Даже при наличии иерархических измерений с целью повышения скорости выполнения запросов к хранилищу данных нередко предпочтение отдается схеме «звезда». Однако не все хранилища данных проектируются по этим двум схемам. Так, довольно часто вместо ключевого поля для измерения, содержащего данные типа «дата», и соответствующей таблицы измерений сама таблица фактов может содержать ключевое поле типа «дата». В этом случае соответствующая таблица измерений просто отсутствует.

Основное достоинство хранилищ с измерениями – их простота и понятность для разработчиков и пользователей. Кроме того, благодаря более эффективному хранению данных и формализованным измерениям облегчается и ускоряется доступ к данным, особенно при сложных анализах. Основной недостаток – более сложные процедуры подготовки и загрузки данных, а также управление и изменение измерений данных.

Хранилища данных отличаются разными способами загрузки данных. Выделяют:

– ETL – сначала извлекают данные из пула источников данных. Данные хранятся во временной промежуточной БД. Затем выполняются операции преобразования, чтобы структурировать и преобразовать данные в подходящую форму для целевой системы хранилища данных. Затем структурированные данные загружаются в хранилище и после этого становятся готовы к анализу;
– ELT (Extract, Load, Transform) – данные сразу же загружаются после извлечения из исходных пулов данных. Промежуточная БД отсутствует, что означает, что данные немедленно загружаются в единый централизованный репозиторий. Данные преобразуются в системе хранилища данных для их использования с инструментами бизнес-аналитики и аналитики. Хранилище данных организации имеет следующую структуру. Базовая структура позволяет конечным пользователям хранилища напрямую приобретать доступ к сводным данным, полученным из исходных систем, создавать отчеты и анализировать эти данные. Эта структура используется в случаях, когда источники данных происходят из одних и тех же типов систем БД.

Хранилище с промежуточной областью является следующим логическим шагом в организации с разнородными источниками данных с множеством различных типов и форматов данных. Промежуточная область преобразует данные в обобщенный структурированный формат, который проще запрашивать с помощью инструментов анализа и отчетности.

Одной из разновидностей промежуточной структуры является добавление витрин данных в хранилище данных. В витринах данных хранятся сводные данные по конкретной сфере деятельности, что делает эти данные легкодоступными для конкретных форм анализа.

Например, добавление витрин данных может позволить финансовому аналитику легче выполнять подробные запросы к данным о продажах, прогнозировать поведение клиентов. Витрины данных облегчают анализ, адаптируя данные специально для удовлетворения потребностей конечного пользователя.

В последние годы хранилища данных переходят в облако. Новые облачные хранилища данных не придерживаются традиционной архитектуры, и каждое из них предлагает свою уникальную архитектуру (например, Amazon Redshift, Google BigQuery)

3.Системы хранения данных

Выделяют несколько подходов к хранению данных.

1.Традиционный подход основан на использовании системы SAN (Storage Area Network) для структурированных данных.

Первичные данные хранятся в виде блоков в дата-центре. Функции блочного хранения используются на низких уровнях в виде блоков фиксированного размера, которые легко индексируются и находятся в системе хранения.

Такой метод подходит при относительно небольших объемах хранения. При росте дискового хранилища возникают проблемы с файловой системой, таблицы становятся непомерно огромными. Это сильно замедляет поиск нужного блока и увеличивает возможность ошибок. Поэтому пользователи вынуждены разбивать свои наборы данных на многочисленные логические узлы LUN (Logical Unit Number), чтобы как-то поддержать скорость на приемлемом уровне. При этом значительно увеличивается сложность администрирования и поддержки IT-системы и, соответственно, растут затраты, а также возможны потери данных и простои системы.

2.Для решения проблем, связанных с увеличением объемов данных, стали использоваться так называемые горизонтально-масштабируемые (Scale-out) файловые системы, такие как HDFS (Hadoop Distributed File System).

Файловая система хранения часто организуется в иерархии файлов и папок, которые существуют в системах хранения NAS. В устройствах SAN используются протоколы iSCSI и Fibre Channel, а в файловых системах NAS – протоколы SMB или NFS. Хранилища этих типов обычно располагаются поблизости от вычислительных ресурсов. Однако по мере того как объемы данных продолжают расти, их приходится все больше

располагать в удаленных дата-центрах. В большинстве случаев это так называемые холодные данные, которые нечасто используются при вычислениях, но их все равно нужно хранить. Поэтому должны быть варианты для эффективного, надежного и экономичного хранения этих данных.

Файловые системы решают проблему масштабирования, однако поддержка таких систем трудоемка. Они конструктивно сложны и требуют постоянного обслуживания. К тому же в них чаще всего используется механизм репликации данных, т. е. хранения копий одних и тех же данных в разных местах системы. Стандартно сохраняются три копии каждого файла. Это увеличивает требуемый дисковый объем на целых 200 %.

Для минимизации затрат многие компании стали прибегать к использованию облачных хранилищ. Экономия на оплате по мере потребления (pay-as-you-go) возможна, если речь идет об относительно небольших объемах данных и их нечастом использовании. При постоянном масштабировании объемов данных, интенсивной работе с ними этот подход также становится затратным и обойдется не дешевле HDFS. Дело в том, что многие облачные провайдеры берут плату не только за объем хранимых данных, но и за трафик извлекаемых/записываемых данных, а также за число обращений к хранилищу. Поэтому, когда приходится иметь дело с анализом больших данных, передачей массивных объемов данных, то хранилище в публичном облаке становится дорогостоящим вариантом. Кроме того, могут возникнуть проблемы конфиденциальности данных и производительности системы, если много других пользователей также будут интенсивно использовать ресурсы данного облака.

3.По мнению специалистов, самым приемлемым выходом может быть объектная система хранения (object storage), в которой используются примерно те же технологии, что и в публичном облаке (HyperText Transfer Protocol (HTTP) – протокол передачи данных, предназначенный для передачи гипертекстовых документов, которые могут содержать ссылки, позволяющие организовать переход к другим документам;

Application Programing Interface (API) – интерфейс прикладного программирования). Объектные хранилища можно легко масштабировать до объемов петабайта в одном домене без какого-либо снижения производительности. Кроме того, объектные хранилища обладают функционалом управления данными, чего нет в традиционных системах: управление версиями, кастомизация метаданных и встроенная аналитика.

Такие характеристики достигаются за счет абстрагирования уровней системы – общего подхода, который сейчас используется практически во всех ИКТ-системах, не только в системах хранения. Каждый диск на нижележащем уровне форматируется простой локальной файловой системой, такой как EXT4. На верхнем уровне, абстрагированном от нижнего, размещаются средства управления, что позволяет интегрировать все элементы в единый унифицированный том. Файлы различного вида хранятся как «объекты», а не как файлы в файловой системе. Поскольку низкоуровневое управление блоками передано локальной файловой системе, объектное хранилище ведает только функциями управления высокого уровня, которые управляют нижележащим уровнем через стандартный API.

Принцип объектного хранения можно сравнить с услугой парковки, когда водитель просто оставляет машину (объект) для ее размещения на парковочном пространстве и получает карточку, по которой может забрать машину. В карточку могут быть внесены «метаданные»: имя водителя, номер и марка машины. Где именно припаркуют машину, водителю неважно (абстрагирование), и ему не нужно кружить по парковке в поисках свободного места.

Такой подход позволяет сохранять таблицы просмотра файловой системы каждого узла нижележащего уровня в пределах легкоуправляемого размера. Это позволяет масштабировать систему до сотен петабайт без заметного снижения производительности. Объектное хранилище предназначено в основном для работы с неструктурированными данными.

Понятие «неструктурированные данные» весьма относительно. Все файлы с данными имеют ту или иную структуру,

тип. Неструктурированные данные просто не хранятся в единой базе и содержат разные типы данных. Это набор разнородных файлов, созданных в различных приложениях и полученных из разных источников. Это примерно то же, что папка «Мои документы» на компьютере.

Объекты неструктурированных данных можно пометить метаданными, которые описывают их содержимое и помогают быстро извлечь из хранилища нужный объект. В этом случае сами метаданные будут структурированы, т. е. будут иметь стандартную форму, определенную в API. Это позволяет отслеживать и индексировать объекты без необходимости применения внешних программ или БД. Использование метаданных открывает новые возможности для аналитики. Файлы (объекты) можно индексировать и искать в объектном хранилище, не зная структуру их содержимого или того, в какой программе они были созданы.

Репликация данных для надежного хранения в объектной системе нужна (как и в других подходах), но при этом не требуется утраивать объем дискового пространства. Для максимизации доступного дискового пространства и защиты данных используется технология Erasure Coding (EC) – следующее поколение метода защиты данных RAID, при котором необходимо двойное или тройное резервирование.

В методе EC файлы объектов разделяются на фрагменты (shards). Для некоторых из них создаются копии избыточности в формате N+M., Например, если для шести из десяти фрагментов создаются копии, это будет формат 10+6. Если для данных нужно, например, N дисков, копии избыточности распределяются по N+M дискам (в данном случае 16). При потере любых шести дисков, оставшихся десяти достаточно для восстановления исходных данных. Таким образом, объем хранения получается не такой большой, как в RAID, и риск потери данных в случае отказа дисков незначителен. Тома EC могут выдерживать больше отказов дисков, чем дисковые массивы RAID. При этом петабайтное масштабирование системы не будет приводить к столь большим затратам на закупку дисков, как в файловых системах.

Особенности объектных систем хранения:
– данные хранятся как объекты, а не в виде традиционных блоков или файлов, состоящих из блоков;
– объекты могут включать в себя самые разные форматы: резервные копии, архивы, видео, изображения, лог-журналы, файлы HTML и т. д.;
– объекты неструктурированы по своей природе, потому что нет единого формата для способа хранения таких данных;
– в отличие от структуры каталогов, которая имеется в традиционных файловых системах хранения, в объектных системах хранения используется простой список объектов, хранящихся в «пакетах» (buckets);
– объекты хранятся с использованием уникальных идентификаторов, а не имён файлов, что резко снижает «накладные расходы», необходимые для хранения данных;
– объекты хранятся вместе с определенными пользователем метаданными, что облегчает поиск объектов при масштабировании данных;
– объекты могут иметь как терабайтные объемы, так и быть размером в несколько килобайт, а один «пакет» может содержать миллиарды объектов;
– разработчики приложений могут легко получить доступ к объектам, используя простые команды через интерфейсы API с помощью запросов GET и PUT без сложных структур каталогов.

Объектные системы хранения целесообразно использовать в следующих случаях:
– при долгосрочном хранении статичных данных, например, различной нормативной документации (WORM);
– резервном копировании – дампы БД, файлы журналов, резервные копии существующего программного обеспечения;
– разработке среды DevOps – одно глобальное пространство имен, которое легкодоступно с использованием простых запросов для управления различными объектами: большими БД, таблицами, изображениями, звуко- и видеофайлами и

пр.; – работе с неструктурированными данными – мультимедийные файлы, документы, изображения, звуко- и видеофайлы;
– в отраслях с большими объемами хранения данных, таких как здравоохранение, электронная почта, мессенджеры, оцифрованные архивы музеев, конструкторских бюро и т. п.

Таким образом, объектные системы хранения хорошо подходят для хранения массивных разнородных (неструктурированных) данных и отвечают запросам быстрого роста объемов данных, которые нужно хранить, обрабатывать и анализировать в различных отраслях. Именно поэтому объемы объектных систем хранения растут значительно быстрее объемов файловых систем.

Тема №5 Современные технологии хранения данных

1.*Архитектура корпоративной системы хранилища – DWH*

Системы складирования данных ориентируются на анализ накопленных данных, т. е. на BI (business intelligence) – процесс анализа данных и получения информации, помогающей компаниям принимать решения. Значит, структуризация данных в хранилище должна быть выполнена таким образом, чтобы данные эффективно использовались в аналитических приложениях. В корпоративных хранилищах в удобном для анализа виде хранятся данные из разных источников. Эти данные предварительно обрабатываются и загружаются в хранилище с помощью технологии ETL. Поэтому главная особенность концепции складирования данных – это структуризация, систематизация, классификация, фильтрация и так далее больших массивов информации в виде, удобном для анализа, визуализации результатов анализа и производства корпоративной отчетности.

Системы, построенные на основе информационной технологии складирования данных, обладают рядом особенностей, которые выделяют их как новый класс информационных си-

стем. К таким особенностям относятся: предметная ориентация системы, интегрированность хранимых в ней данных, собираемых из различных источников, инвариантность этих данных во времени, относительно высокая стабильность данных, необходимость поиска компромисса при избыточности данных. Большое разнообразие видов данных затрудняет получение консолидированной отчетности, когда нужна целостная картина из всех прикладных систем. В результате в 90-х гг. XX в. в компании IBM зародилась информационная технология складирования данных – Data Warehousing (DWH).

DWH – предметно-ориентированные БД для консолидированной подготовки отчетов, интегрированного бизнес-анализа и оптимального принятия управленческих решений на основе полной информационной картины. Решения DWH, по сути, представляют собой ту же систему для хранения и работы с корпоративной информацией, что и ETL.

Архитектура DWH – многоуровневая, слоеная, называется LSA (Layered Scalable Architecture). Она реализует логическое деление структур с данными на несколько функциональных уровней. Данные копируются с уровня на уровень и трансформируются при этом, чтобы в итоге предстать в виде информации, пригодной для анализа.

Классически LSA реализуется в виде следующих уровней:
– операционный слой первичных данных (Primary Data Layer, или стейджинг), на котором выполняется загрузка информации из систем-источников в исходном качестве и с сохранением полной истории изменений. Здесь происходит абстрагирование следующих слоев хранилища от физического устройства источников данных, способов их сбора и методов выделения изменений;
– ядро хранилища (Core Data Layer) – центральный компонент, который выполняет консолидацию данных из разных источников, приводя их к единым структурам и ключам. Именно здесь происходят основная работа с качеством данных и общие трансформации, чтобы абстрагировать потребителей от особенностей логического устройства источников данных и необходимости их взаимного сопоставления. Так

решается задача обеспечения целостности и качества данных;

– аналитические витрины (Data Mart Layer), где данные преобразуются в структуры, удобные для анализа и использования в системах-потребителях. Витрины могут брать данные из ядра (регулярные витрины), операционного слоя (операционные витрины), могут использоваться для представления результатов сложных расчетов и нетипичных трансформаций (вторичные витрины). Таким образом, витрины обеспечивают разные представления единых данных под конкретную бизнес-специфику;

– сервисный слой (Service Layer) обеспечивает управление всеми вышеописанными уровнями. Он не содержит бизнес-данных, но оперирует метаданными и другими структурами для работы с качеством данных, позволяя выполнять сквозной аудит данных (data lineage). Также здесь доступны средства мониторинга и диагностики ошибок, что ускоряет решение проблем.

Все слои, кроме сервисного, состоят из области постоянного хранения данных и модуля загрузки и трансформации. Области хранения содержат технические (буферные) таблицы для трансформации данных и целевые таблицы, к которым обращается потребитель. Для обеспечения процессов загрузки и аудита ETL-процессов данные в целевых таблицах стейджинга, ядра и в витринах маркируются техническими полями (метаатрибутами). Еще выделяют слой виртуальных провайдеров данных и пользовательских отчетов для виртуального объединения (без хранения) данных из различных объектов. Каждый уровень может быть реализован с помощью разных технологий хранения и преобразования данных или универсальных продуктов, например SAP NetWeaver Business Warehouse (SAP BW).

2.Хранилище данных и озеро данных

Озеро данных – это не синоним хранилищ данных или витрин данных. Конечно, это хранилище данных, но оно принципи-

ально отличается от остальных. Идея озера данных заключается в том, чтобы хранить необработанные данные в их оригинальном формате до тех пор, пока они не понадобятся.

Озеро данных принимает любые файлы всех форматов. Источник данных тоже не имеет никакого значения: это могут быть данные из CRM- или ERP-систем, продуктовых каталогов, банковских программ, датчиков или умных устройств – любых систем, которые использует бизнес. Потом, когда данные будут сохранены, с ними можно работать: извлекать по определенному шаблону из классических БД или анализировать и обрабатывать прямо внутри озера.

Озеро данных состоит из разных типов данных, которые стекаются из многочисленных источников. Заполнение озера только структурированными данными означало бы, что оно теряет хотя бы часть своей структуры и значения.

DWH и озеро данных имеют разное целевое назначение. DWH используется менеджерами, аналитиками и другими конечными бизнес-пользователями, тогда как озеро данных в основном экспертами по аналитическим данным, которые обладают техническими навыками для решения сложных задач, а также любопытством, которое помогает эти задачи ставить (их называют Data Scientist'ами). Неструктурированная, «сырая» информация, которая хранится в озере данных (ви-деозаписи с беспилотников и камер наружного наблюдения, транспортная телеметрия, графические изображения, логи пользовательского поведения, метрики сайтов и информационных систем, а также прочие данные с разными форматами хранения (схемами представления)), пока непригодна для ежедневной аналитики в BI-системах, но может использоваться Data Scientist'ами для быстрой отработки новых бизнес-гипотез с помощью алгоритмов машинного обучения.

DWH и озеро данных отличаются разными подходами к проектированию. Дизайн DWH основан на реляционной логике работы с данными (третья нормальная форма для нормализованных хранилищ, схемы «звезда» или «снежинка» для хранилищ с измерениями). При проектировании озера

данных архитекторы Big Data и Data Engineer больше внимания уделяют ETL-процессам с учетом многообразия источников и приемников разноформатной информации. А вопрос непосредственного хранения данных решается достаточно просто – требуется лишь масштабируемая, отказоустойчивая и относительно дешевая файловая система.

Озеро данных отличается гибкостью и доступностью данных: может предоставлять пользователям и последующим приложениям данные без схемы, т. е. данные в «естественном» формате независимо от их происхождения. Здесь ничего не нужно определять заранее, как в случае использования корпоративных хранилищ, когда еще на старте нужно выявить актуальные для нее типы данных и структуру, а в случае появления данных новых форматов базу придется перестраивать.

В озерах данных хранятся, в том числе, и бесполезные данные, которые могут пригодиться в будущем или не понадобиться никогда. В них удобно хранить архивы неочищенной информации, создавать большую базу для масштабной аналитики.

Озеро данных обычно строится на базе бюджетных серверов с Apache Hadoop, без дорогостоящих лицензий и мощного оборудования, в отличие от больших затрат на проектирование и покупку специализированных платформ класса Data Warehouse, таких как SAP, Oracle, Teradata и пр.

При доступе к озерам данных пользователи должны знать конкретные типы данных и источники, в которых они нуждаются; сколько данных им нужно; когда им это нужно; методы аналитики, которые будут применяться к этим данным. Такое невозможно в хранилище данных. Поэтому схема озера данных определяется не «по записи», а «по чтению».

Для озера данных все еще требуется схема, но она не предопределена. Это ad hoc107. Данные используются по плану или схеме, когда пользователи извлекают их, а не когда загружают. Озера данных сохраняют данные в неизменном (естественном) состоянии; требования не определя-

ются до тех пор, пока пользователи не запросят данные. Таким образом, в случае с озером данных информацию структурируют на выходе, когда надо извлечь данные или проанализировать их. При этом процесс анализа не влияет на сами данные в озере: они так и остаются неструктурированными, чтобы их было также удобно хранить и использовать для других целей.

При правильном использовании озеро данных предоставляет бизнес-пользователям и техническим пользователям возможность запрашивать меньшие, более актуальные и более гибкие наборы данных. В результате время запросов может сократиться до работы как в витрине данных, хранилище данных или реляционной БД.

Озера данных можно использовать в любом бизнесе, который собирает данные: маркетинг, ритейл, IT, производство, логистика и др. Озеро данных позволяет накапливать данные «про запас», а не под конкретный запрос бизнеса. За счет того, что данные всегда «под рукой», компания может быстро проверить любую гипотезу или использовать данные для своих целей.

Таким образом, озера данных нужны для гибкого анализа данных и построения гипотез. Они позволяют собрать как можно больше данных, чтобы потом с помощью инструментов машинного обучения и аналитики сопоставлять разные факты, делать невероятные прогнозы, анализировать информацию с разных сторон и извлекать из данных все больше пользы.

Вместе с преимуществами у озер данных есть одна серьезная проблема. Любые данные попадают туда практически бесконтрольно. Это значит, что определить их качество невозможно. Если у компании нет четкой модели данных, т. е. понимания типов структур данных и методов их обработки, то управление озером плохо организовано, в нем быстро накапливаются огромные объемы неконтролируемых данных, чаще всего бесполезных. В итоге озеро превращается в «бо-

лото» данных – бесполезное, поглощающее ресурсы компании и не приносящее пользы. В таком случае его нужно полностью стереть и начать собирать данные заново.

Чтобы озеро не стало «болотом», нужно наладить в компании процесс управления данными (Data governance). Главная составляющая этого процесса – определение достоверности и качества данных еще до загрузки в озеро.

Для этого необходимо:

– отсекать источники с заведомо недостоверными данными;

– ограничить доступ на загрузку для сотрудников, у которых нет на это прав;

– проверять некоторые параметры файлов, например, не пропускать в озеро картинки, которые весят десятки гигабайт.

Настроить такую фильтрацию проще, чем каждый раз структурировать данные для загрузки в БД. Если процесс налажен, в озеро попадут только актуальные данные, а значит, и сама база будет достоверной. При проектировании любого озера данных надо заранее определиться, для каких целей его строить. Есть мнение, что озеро данных – это не только важное, но и обязательное условие для компаний, которые управляют данными, поскольку хранилища данных не создавались для обработки огромных потоков неструктурированных данных.

Тема №6 Аналитика больших данных и ее инструментарий

Данные, собранные в хранилища, нужны не сами по себе, а для их анализа и принятия управленческих решений. Традиционный пакетный анализ или Аналитика 1.0, сформировался в 80-е гг. XX в. Изначально Аналитика 1.0 задумывалась как средство хранения и загрузки информации для составления отчетов и предназначалась для руководителей высшего звена.

Аналитика 1.0 в бóльшей степени опиралась на описательную статистику и отчетность с редкими вкраплениями

прогностической аналитики. Данные поставлялись почти исключительно из внутренних источников и были хорошо структурированы. Они собирались, хранились IT-отделом и предоставлялись по запросу.

Чтобы сделать данные доступными для анализа, IT-специалистам требовалось довольно много времени. После получения данных аналитики выполняли массу дополнительной подготовительной работы: разного рода преобразований, агрегирования и комбинирования данных из различных источников. Все это затягивало процесс получения результатов. Получалось, что время в основном тратилось на сбор и обработку данных, а не на собственно анализ.

В начале 2000-х гг. началось становление эры больших данных. Произошла замена технологий на более дешевые и быстрые версии прежних аналогов, которые отвечали требованиям времени. Это этап становления Аналитики 2.0; этап прогностической аналитики.

Новый этап развития технологий Big Data в условиях современного цифрового бизнеса называют Аналитикой 3.0, или операционной аналитикой.

Речь идет о переходе к совершенно новой для бизнеса ситуации, когда аналитические решения внутри компании не просто помогают видеть результаты прошлого и тестировать сценарии будущего. Правильно настроенная аналитическая машина способна на основании доступных ей данных самостоятельно принимать решения операционного уровня, делая это тысячи или миллионы раз за день. Очень многие управленческие решения могут приниматься роботизированными алгоритмами без вмешательства человека. Таким образом, операционная аналитика интегрирует аналитику в бизнес-процессы и автоматизирует принятие решений без участия человека.

Таким образом, операционная аналитика интегрирует аналитику в бизнес-процессы и автоматизирует принятие решений без участия человека. Такая аналитика транзакционного

уровня – это новый шаг по сравнению с традиционным пониманием бизнес-анализа как базы для принятия решений на стратегическом уровне.

В отличие от неторопливой пакетной аналитики операционная аналитика выполняется намного быстрее и непрерывно. При этом она интегрируется с существующими бизнес-процессами и системами. Переход к операционной аналитике не устраняет ни одного из шагов, которые традиционно требовались для создания аналитического процесса. При этом процесс развивается дальше. Операционная аналитика придает аналитике промышленный масштаб.

У современного бизнеса есть все возможности для применения операционной аналитики. И она уже работает и оказывает влияние на многие стороны жизни человека. Например, в случае задержки рейса авиакомпании автоматически перенаправляют пассажиров на другой маршрут. При этом аналитические программы принимают во внимание множество факторов, в том числе касающихся конкретного клиента, других пассажиров и статуса альтернативных рейсов.

Пример. Приходя в магазин, люди могут на месте получить кредит на основе оценки их текущей кредитоспособности, которая определяется с помощью анализа широкого диапазона данных о кредитной истории клиента.

Еще пример. Операционная аналитика, основанная на показателях датчиков двигателя автомобиля, выдается почти сразу. Она выполняется параллельно с работой двигателя, а поступающая с датчиков информация анализируется в режиме реального времени. Если выявляется некая проблема, то принимаются меры по ее предотвращению. Например, водитель за рулем автомобиля получает упреждающее уведомление о том, что с двигателем начинает твориться что-то неладное.

Переход к операционной аналитике может сводиться к модернизации существующего аналитического процесса, но чаще операционная аналитика включает в себя разные типы аналитики.

Операционная аналитика используется для поддержки не стратегических, а повседневных тактических решений. Она не только рекомендует те или иные действия, а непосредственно их реализует. Причем эти действия осуществляются незамедлительно, человек не участвует ни в принятии решения, ни в осуществлении действия. Получается, что операционная аналитика выходит за пределы описаний или прогнозов. Она предписывает. Это значит, что операционная аналитика встраивается в бизнес-процесс, чтобы самостоятельно принимать решения и выполнять действия на основе заложенных в нее алгоритмов.

На протяжении последнего десятилетия много внимания уделялось переходу аналитики от описательной к прогностической. Если в традиционной бизнес-аналитике внимание сосредоточивалось на анализе произошедшего с описательной точки зрения (например, определение объема продаж товара по каждому региону, доли поставок или других важных показателей), то целью прогностической аналитики, наоборот, является предсказание того, что произойдет в будущем (например, как увеличить долю своевременных поставок товара, какие клиенты с наибольшей вероятностью откликнутся на новое маркетинговое предложение).

Операционная аналитика идет еще дальше и делает аналитику предписывающей. Операционно-аналитический процесс начинается с определения того, какие действия повлияют на время поставки или повысят уровень откликов, а затем автоматически вынуждает эти действия произойти.

Виды аналитики и их сущность

Описательная аналитика: анализирует и описывает события, произошедшие

Прогностическая аналитика: прогнозирует будущие события

Предписывающая аналитика: определяет действия, необходимые для достижения целей

Одно из важных отличий операционной аналитики состоит в том, что анализ выполняется в автоматическом и интегри-

рованном режимах в пределах так называемого времени принятия решения, т. е. анализ выполняется со скоростью, позволяющей быстро принять решение. В некоторых случаях принятие решений происходит в режиме реального времени (или очень близко к тому). В других случаях период ожидания может составлять несколько минут, часов или даже дней. Знать время принятия решения крайне важно для достижения успеха, поскольку аналитический процесс должен быть доступен и выполняться в пределах этого интервала.

Следовательно, к основным отличиям операционной аналитики от традиционной можно отнести следующие:
1) операционная аналитика автоматизирована: операционно-аналитический процесс выполняется внутри операционных систем в автоматическом режиме;
2) операционная аналитика принимает решения, а затем выполняет действия, которые из них вытекают, в то время как в традиционной аналитике анализ производит рекомендации, а человек решает, принять их или отклонить;
3) операционная аналитика осуществляется в пределах «времени принятия решения». Во многих случаях оно соответствует реальному времени. В некоторых случаях аналитика применяется ко входящему потоку, а не к хранилищу данных.

Операционная аналитика не может позволить себе ждать до следующего сеанса пакетной обработки: она должна осуществляться немедленно, чтобы принять решение и исполнить его.

Таким образом, операционная аналитика – это интегрированные автоматические процессы принятия решений, предписывающие и реализующие действия в пределах «времени принятия решения». Как только операционно-аналитический процесс получает одобрение и запускается, он начинает автоматически принимать многочисленные решения.

Но в любом случае центральная роль остается за человеком: кто-то должен разрабатывать, выстраивать, конфигурировать и контролировать операционно-аналитические процессы. Компьютеры сами по себе не смогут принимать решения.

Аналитика 3.0 способна считывать информацию и реагировать на события, которые влияют на пользователя, машины и девайсы в режиме реального времени. Преимущество Аналитики 3.0 заключается в том, что есть возможность синтезировать и соотносить друг с другом разрозненные источники информации, чтобы принимать автоматические решения на основе полученных данных.

2. Инструментарий для анализа больших данных (реляционные и нереляционные СУБД)

Хотя бизнес-аналитика и большие данные имеют одинаковую цель (поиск ответов на вопрос), они отличаются друг от друга:

1) технологии Big Data предназначены для обработки:

– сразу всего массива разных типов данных по сравнению с инструментами бизнес-аналитики, что дает возможность фокусироваться не только на структурированных хранилищах;

– получаемых в реальном времени и меняющихся сведений, что означает глубокое исследование и интерактивность. В некоторых случаях результаты формируются быстрее, чем загружается веб-страница. Тем самым скорость обработки больших данных позволяет сделать анализ предсказательным, способным давать бизнесу рекомендации на будущее; – неструктурированных данных в их исходном виде, алгоритмы и способы, использования которых находятся в процессе становления.

2) подход к работе с большими данными отличается от подхода к проведению бизнес-анализа. В отличие от простого сложения известных значений в традиционной аналитике при работе с большими данными результат получается в процессе их очистки путем последовательного моделирования: сначала выдвигается гипотеза, строится статистическая, или визуальная, или семантическая модель, на ее основании проверяется верность выдвинутой гипотезы и затем выдвигается следующая. Этот процесс требует от исследователя либо интерпретации визуальных значений или составления интерактивных

запросов на основе знаний, либо разработки адаптивных алгоритмов машинного обучения.

Для анализа данных используются разные инструменты. Одним из самых известных инструментов анализа является Hadoop – программное обеспечение, позволяющее обрабатывать большие объемы данных различных типов и структур. С его помощью собранные данные можно распределить и структурировать, настроить аналитику для построения моделей и проверки предположений, использовать машинное обучение.

К аналитическим движкам для работы с большими данными можно отнести Apache Chukwa, Apache Hadoop, Apache Hive, Apache Pig!, Jaspersoft, LexisNexis Risk Solutions HPCC Systems, MapReduce, Revolution Analytics (на базе языка R для матстатистики).

Аналитика больших данных развивалась постепенно по мере развития двухуровневой модели обработки. Первый уровень представляет собой традиционную аналитику Big Data, когда большие массивы данных подвергаются анализу не в режиме реального времени. Второй уровень обеспечивает возможность анализа относительно больших объемов данных в реальном времени в основном за счет технологий аналитики в памяти (in-memory).

Аналитика в памяти предполагает наличие поддерживающих технологий, чтобы обеспечить достаточные объемы памяти для размещения действительно масштабных наборов данных, для эффективного перемещения данных между большими объектными хранилищами и системами, ведущими анализ в памяти. Важную роль в этом играют решения с открытым кодом.

Наиболее популярными в мировом IT-сегменте продуктами для решения проблем Big Data считаются аналитические платформы NoSQL и In-memory.

Изначально основным способом работы с БД был SQL (БД – structured query language) – язык структурированных запросов, появившийся в 1974 г., применяемый для создания, мо-

дификации и управления данными в реляционной БД. Он позволял выполнять следующие операции: создание в БД новой таблицы; добавление в таблицу новых записей; изменение записей; удаление записей; выборка записей из одной или нескольких таблиц (в соответствии с заданным условием); изменение структур таблиц.

Со временем SQL усложнился: обогатился новыми конструкциями, обеспечил возможность описания новых хранимых объектов (например, индексов, представлений, триггеров и хранимых процедур) и управления ими и стал приобретать черты, свойственные языкам программирования.

Во второй половине 2000-х гг. ради горизонтальной масштабируемости появилась система NoSQL (в названии No значит отрицание SQL). В ранних NoSQL-системах поддержка SQL отсутствовала, со временем некоторые из СУБД обзавелись специфическими SQL-подобными языками запросов. В 2010-е гг. ряд СУБД отнесли себя к категории NewSQL, в них при сохранении свойств масштабируемости NoSQL-систем реализована и поддержка SQL, в разных системах – в разной степени совместимости со стандартами. Кроме того, поддержка SQL в 2010-е гг. появилась не только в СУБД, но и для экосистемы Hadoop (Spark SQL, Phoenix, Impala), а также в связующем программном обеспечении (брокер сообщений Kafka, система потоковой обработки Flink). Таким образом, язык постепенно становится фактическим стандартом доступа к любым обрабатываемым данным, не только реляционной природы.

Реляционные БД (SQL) хранят данные в формате таблиц, они строго структурированы и связаны друг с другом. В таблице есть строки и столбцы, каждая строка представляет отдельную запись, а столбец – поле с назначенным ему типом данных. В каждой ячейке информация записана по шаблону. Эти базы отличают надежность и неизменяемость данных, низкий риск потери информации, при обновлении данных целостность гарантируется, они заменяются в одной таблице. Реляционные БД, в отличие от нереляционных, соответствуют следующим требованиям к транзакционным системам

(ACID). Соответствие им гарантирует сохранность данных и предсказуемость работы БД.

1. Atomicity, или атомарность, – ни одна транзакция не будет зафиксирована в системе частично.
2. Consistency, или непротиворечивость, – фиксируются только допустимые результаты транзакций.
3. Isolation, или изолированность, – на результат транзакции не влияют транзакции, проходящие параллельно ей.
4. Durability, или долговечность, – изменения в БД сохраняются, несмотря на сбои или действия пользователей.

Реляционные БД идеальны для работы со структурированными данными, структура которых не подвержена частым изменениям. При поступлении большого объема данных рано или поздно наступит предел их вертикального масштабирования, и увеличивать производительность сервера будет невозможно. Это не значит, что СУБД на SQL не подходят для больших проектов, но тогда потребуется настройка системы либо использование БД в облаке.

Что касается нереляционных БД (NoSQL), то они хранят данные без четких связей друг с другом и четкой структуры. В отличие от реляционных БД NoSQL-базы не поддерживают запросы SQL, в них схема данных является динамической и может меняться в любой момент времени, к данным сложнее получить доступ (с таблицей это просто, достаточно знать координаты ячейки).

Зато такие СУБД отличаются высокой производительностью и высокой скоростью. Физические объекты в NoSQL обычно можно хранить прямо в том виде, в котором с ними потом работает приложение. БД NoSQL хороши также для быстрой разработки и тестирования гипотез. В них можно хранить данные любого типа и добавлять новые в процессе работы.

NoSQL-базы имеют распределенную архитектуру, поэтому хорошо масштабируются горизонтально и отличаются высокой производительностью. Технологии NoSQL могут автоматически распределять данные по разным серверам. Это повышает скорость чтения данных в распределенной среде.

Существуют следующие виды нереляционных БД.

1.Документо-ориентированные БД (например, MongoDB). В таких базах данные хранятся в коллекциях документов, обычно с использованием форматов JSON, XML или BSON. Одна запись может содержать столько данных, сколько нужно, в любом типе данных (или типах) – ограничений нет. У каждого документа есть внутренняя структура, однако она может отличаться от структуры других документов. Также документы можно вкладывать друг в друга. Вместо столбцов и строк все данные описываются в одном документе. Если было бы нужно добавить новые данные в таблицу реляционной БД, пришлось бы изменять схему данных. В случае с документами нужно только добавить в них дополнительные пары ключ – значение.

2. БД «ключ – значение» (например, Redis). Здесь каждая запись имеет ключ и значение. Разработчики в основном используют такие базы данных, когда данные не слишком сложные, а важна скорость. Сохраненным данным не назначается никакой схемы, а сама БД намного легче по сравнению с реляционной.

3. Графовые БД (например, Neo4j, OrientDB) состоят из узлов и связей между ними. Узлы обозначают элементы в БД, а связи между ними определяют их отношения между собой. Из всех типов БД они считаются лучшим вариантом в случаях, когда приоритетными являются различные взаимосвязи между данными. Недостатком графовых БД является то, что для доступа к данным нельзя использовать ни SQL, никакой-либо другой общепринятый подход. Отсутствие стандартизации означает, что большинство языков запросов могут использоваться только в одном или нескольких типах графовых БД. Графовые БД хранят сами данные и взаимосвязи между ними.

4. Колоночные СУБД (например, Cassandra) – хороший вариант для обработки больших данных, отличаются высокой производительностью, эффективным сжатием данных и отличной масштабируемостью. В таких системах данные хра-

нятся в виде разреженной матрицы, строки и столбцы которой используются как ключи. Подобно таблице семейство столбцов содержит столбцы и строки. Вместе с тем есть четкое различие: столбец не охватывает все строки. Вместо этого он содержится в строке, что также означает, что разные строки могут иметь разные столбцы.

Помимо столбцов каждая строка имеет идентификатор, называемый ключом, а каждый столбец содержит имя, значение и метку времени. Таким образом, в колоночной БД данные тоже хранятся в таблице, только она состоит из совокупности колонок, каждая из которых, по сути, является отдельной таблицей. Это позволяет быстрее получать данные из базы для анализа.

Самыми популярными нереляционными БД являются следующие.

1.MongoDB, которая может работать как со структурированными, так и со неструктурированными данными. Подходит для проектов, работающих с разнородными данными, с трудом поддающимися классификации, или если в будущем ожидается значительное изменение структуры данных, в том числе для OLAP-сценариев. Эта БД хорошо масштабируется горизонтально без потери скорости, проста в применении, производительна, подходит для больших объемов данных, ее легко установить, она имеет много настроек.

Однако она не использует в качестве языка запросов SQL, у нее есть инструменты для перевода SQL-запросов, но они требуют настройки. Также отсутствует связность данных. MongoDB сложна в сопровождении, потому что требует опыта работы с NoSQL. MongoDB удобно использовать в облаке, так как у нее меньше проблем с настройками и управлением. Это решение для кеширования данных, хранения документов, контента и других неструктурированных данных, для работы с большими данными и машинным обучением, очередями сообщений.

2.Redis можно использовать как самостоятельную СУБД для быстрой работы с небольшими объемами данных либо как кэширующий слой для работы с другой СУБД, т. е. как замена

memcached. Помогает ускорить работу медленной БД, увеличивает скорость обработки запросов. Например, можно использовать в качестве основной базы MySQL, а для кеша – Redis. Может работать с разными типами данных, оперативно обрабатывать их в памяти, сохранять на диске, отличается простой репликацией данных. При работе с большими данными их объем не должен превышать объем свободного ОЗУ сервера, иначе работа замедлится. Есть риск несохранения данных, сложности с настройкой кластера и шардингом. Все эти проблемы решаются при запуске СУБД Redis в облаке, где заботу о поддержке, хостинге и бэкапах данных берет на себя провайдер.

Хотя реляционные и нереляционные БД отличаются, между ними нет противоречия, их часто используют совместно для решения разных задач:

1) реляционные SQL-базы подходят для хранения структурированных данных, особенно в тех случаях, когда крайне важна их целостность. Также эту модель лучше выбрать, если на проекте нужна технология, основанная на стандартах, при использовании которой можно рассчитывать на большое количество дополнений и большой опыт разработчиков;

2) нереляционные NoSQL-базы используют, если требования к данным нечеткие, неопределенные, могут меняться с ростом и развитием проекта и когда одно из основных требований к БД – высокая скорость работы.

3. Технологии аналитики в памяти (in-memory)

Технологии обработки данных in-memory до недавнего времени использовались мало из-за их высокой стоимости. Сейчас память становится все более дешевой и емкой, и поэтому растет популярность систем класса In-Memory Data Grid. Они содержат в себе только уровень обработки данных в памяти, а все остальные элементы наподобие Hadoop и HDFS используются как постоянное хранилище.

Сейчас компании перестраивают архитектуру своих информационных систем, чтобы использовать преимущества быстрой транзакционной обработки данных, предлагаемых

этими решениями. Вследствие падения стоимости оперативной памяти (RAM) становится возможным хранение всего набора операционных данных в памяти, при этом скорость их обработки увеличивается более чем в тысячу раз. Продукты In-Memory Compute Grid и In-Memory Data Grid предоставляют необходимые инструменты для построения таких решений.

Задача In-Memory Data Grid (IMDG) – обеспечить сверхвысокую доступность данных посредством хранения их в оперативной памяти в распределенном состоянии. Современные IMDG способны удовлетворить большинство требований к обработке больших массивов данных.

IMDG – это распределенное хранилище объектов, схожее по интерфейсу с обычной многопоточной хэш-таблицей. Объекты хранятся по ключам. Однако в отличие от традиционных систем, в которых ключи и значения ограничены типами данных «массив байт» и «строка», в IMDG можно использовать любой объект бизнес-модели в качестве ключа или значения. Это значительно повышает гибкость, позволяя хранить в Data Grid в точности тот объект, с которым работает бизнес-логика, без дополнительной сериализации/десер-иализации, которую требуют альтернативные технологии. Это также упрощает использование Data Grid, поскольку в большинстве случаев можно работать с распределенным хранилищем данных как с обычной хеш-таблицей.

Возможность работать с объектами из бизнес-модели напрямую – одно из основных отличий IMDG от In-Memory-баз (IMDB). В последнем случае пользователи все еще вынуждены осуществлять объектно-реляционное отображение (Object-To-Relational Mapping), которое, как правило, приводит к значительному снижению производительности.

IMDG отличается от других продуктов, таких как IMDB, NoSql или NewSql-базы. Одно из отличий – по-настоящему масштабируемое секционирование данных (Data Partitioning) в кластере. IMDG, по сути, распределенная хэш-таблица, где каждый ключ хранится на строго определенном сервере в кластере. Чем больше кластер, тем больше данных можно в нем хранить.

Принципиально важным в этой архитектуре является то, что обработку данных следует производить на том же сервере, где они расположены (локально), исключая (или сводя к минимуму) их перемещение по кластеру. Фактически при использовании хорошо спроектированного IMDG перемещения данных не будет за исключением случаев, когда в кластер добавляются новые серверы или удаляются существующие, меняя тем самым топологию кластера и распределение данных в нем. Внешняя БД не является обязательной. Если она присутствует, IMDG, как правило, будет автоматически читать данные из базы или записывать их в нее.

Еще одна отличительная особенность IMDG – поддержка транзакционности, удовлетворяющей требованиям ACID. Как правило, чтобы гарантировать целостность данных в кластере, используют двухфазную фиксацию (2-phase-commit, или 2PC). Разные IMDG могут иметь разные механизмы блокировок, но наиболее продвинутые реализации обычно используют параллельные блокировки (например, MVCC – multi-version concurrency control, управление конкурентным доступом с помощью многоверсионности), сводя тем самым сетевой обмен к минимуму и гарантируя транзакционную целостность ACID с сохранением высокой производительности.

Целостность данных является одним из главных отличий IMDG от NoSQL-баз. NoSQL-базы в большинстве случаев спроектированы с использованием подхода, называемого «целостность в конечном итоге» (Eventual Consistency, EC), при котором данные могут некоторое время находиться в несогласованном состоянии, но обязательно станут согласованными «со временем». В целом операции записи в EC системах происходят достаточно быстро по сравнению с более медленными операциями чтения. Последние IMDG с «оптимизированным» 2PC как минимум соответствуют EC системам по скорости записи (если не опережают их) и значительно превосходят их по скорости чтения.

Разные продукты могут предлагать разные 2PC оптимизации, но в целом задачами всех оптимизаций являются увели-

чение параллелизма (concurrency), минимизация сетевого обмена и снижение числа блокировок, требуемых для совершения транзакции.

Даже несмотря на то, что у разных IMDG обычно много общих базовых функциональных возможностей, существует множество дополнительных возможностей и деталей их реализации, которые отличаются в зависимости от производителя.

Хранение данных в IMDG – это лишь половина функционала, требуемого для in-memory архитектуры. Данные, хранимые в IMDG, также должны обрабатываться параллельно и с высокой скоростью. Типичная in-memory архитектура секционирует данные в кластере с помощью IMDG, и затем исполняемый код отправляется именно на те серверы, где находятся требуемые ему данные.

Поскольку исполняемый код (вычислительная задача) обычно является частью вычислительных кластеров (Compute Grids) и должен быть правильно развернут, сбалансирован по нагрузке, обладать отказоустойчивостью, а также иметь возможность запуска по расписанию (scheduling), интеграция между Compute Grid и IMDG очень важна. Наибольший эффект можно получить, если IMDG и Compute Grid являются частями одного и того же продукта и используют одни и те же API. Это снимает с разработчика бремя интеграции и обычно позволяет достигнуть наибольшей производительности и надежности in-memory решения.

IMDG (вместе с Compute Grid) находят свое применение во многих областях, таких как анализ рисков, торговые системы, системы реального времени для борьбы с мошенничеством, биометрика, электронная коммерция, онлайн-игры. По сути, любой продукт, перед которым стоят проблемы масштабируемости и производительности, может выиграть от использования In-Memory Processing и IMDG-архитектур.

Тема №7 Аналитика больших данных: техника обработки и анализа

1.Методы анализа и обработки больших данных

В настоящее время существует множество разнообразных методик анализа массивов данных, в основе которых лежит инструментарий, заимствованный из статистики и информатики. При этом исследователи продолжают работать над созданием новых методик и совершенствованием существующих. К основным техникам и методам анализа и обработки данных можно отнести следующие.

1.Методы класса, или глубинный анализ (Data Mining).

Data Mining (добыча данных, интеллектуальный анализ данных, глубинный анализ данных) – собирательное название, используемое для обозначения совокупности методов обнаружения в данных ранее неизвестных, нетривиальных, практически полезных и доступных интерпретации знаний, необходимых для принятия решений в различных сферах человеческой деятельности.

Компания применяет Data Mining, когда у нее уже есть некий массив данных, который ранее был как-то обработан, а теперь он обрабатывается вновь, возможно, как-то иначе, чем прежде, для получения неких полезных выводов.

Data Mining решает следующие задачи: работа с данными (агрегация, анализ, описание), выявление взаимосвязей и построение трендов (возможно, с конечной целью предсказания).

2.Краудсорсинг.

Позволяет получать данные одновременно из нескольких источников, причем количество последних практически не ограничено. Краудсорсинг предлагает способ систематизации данных путем избавления от лишнего материала; категоризации и оценки нужной информации. Это позволяет получать доступ к ценной информации, содержащейся в недрах груды сырых данных. Систематизация большого объема данных методом краудсорсинга позволяет избежать значительных дополнительных расходов.

3. A/B-тестирование.

Данный метод предполагает выбор из всего объема данных контрольной совокупности элементов, которую поочередно сравнивают с другими подобными совокупностями, где был изменен один из элементов. Проведение подобных тестов помогает определить, колебания какого из параметров оказывают наибольшее влияние на контрольную совокупность. Благодаря объемам Big Data можно проводить огромное число итераций, с каждой из них приближаясь к максимально достоверному результату.

4. Прогнозная аналитика.

Прогнозная аналитика задействует множество методов из статистики, интеллектуального анализа данных, анализирует как текущие данные, так и данные за прошлые периоды, на основе которых и составляет прогнозы о будущих событиях. Модели прогнозирования выявляют связи среди многих факторов, чтобы сделать возможной оценку рисков или потенциала, связанного с конкретным набором условий. Итог использования прогнозной аналитики – принятие верных (максимально эффективных для бизнеса) решений.

С помощью моделей прогнозирования можно предсказать поведение потенциальных клиентов, выявить наиболее популярные продукты и услуги, понять, что движет клиентами, почему они уходят, и предотвратить это и т. д. Использование инструментов прогнозной аналитики помогает создать модель поведения клиентов, а значит, и увеличить прибыль компании.

5.Машинное обучение (Machine Learning).

Предполагает эмпирический анализ информации и последующее построение алгоритмов самообучения систем.

Машинное обучение – это метод анализа данных, основанный на построении автоматизированной аналитической модели. Используя математические алгоритмы анализа данных, машинное обучение позволяет находить скрытые факторы и зависимости, не будучи заранее запрограммированным на определенное место поиска.

Машинное обучение способно адаптироваться и переобучаться на основе вновь поступивших данных для получения надежных и репрезентативных результатов.
Итог машинного обучения – ценные предсказания, которые помогают принять лучшее решение и осуществить правильные действия в реальном времени без вмешательства человека.
Машинное обучение продолжает развиваться и модернизироваться. В число наиболее важных разработок входят ансамблевые методы, в которых прогнозирование осуществляется на основе набора моделей, где каждая модель участвует в каждом из запросов, а также дальнейшее развитие нейронных сетей глубокого обучения, имеющих более трех слоев нейронов. Такие глубокие слои в сети способны обнаруживать и анализировать отображения сложных атрибутов (состоящие из нескольких взаимодействующих входных значений, обработанных более ранними слоями), которые позволяют сети изучать закономерности и обобщать их для всех входных данных. Благодаря своей способности исследовать сложные атрибуты сети глубокого обучения лучше других подходят для многомерных данных – именно они произвели переворот в таких областях, как машинное зрение и обработка естественного языка.
6.Сетевой анализ.
Сетевой анализ – наиболее распространенный метод для исследования социальных сетей: после получения статистических данных анализируются созданные в сетке узлы, т. е. взаимодействия между отдельными пользователями и их сообществами.
7. Использование Dark Data.
Так называемые Dark Data (темные данные) – это вся неоцифрованная информация о компании, которая не играет ключевой роли при ее непосредственном использовании. Аналитики причисляют к темным данным информацию, которую сотрудники используют в работе только один раз. После

этого она теряется на просторах неорганизованного контента. На практике это означает, что около 80 % документов в компании не используется повторно.

В управлении темными данными используют метаданные, или «данные о данных», для идентификации, связывания отдельных файлов, организации информации, отсылок на другие материалы. В совокупности это позволяет разблокировать темные данные и использовать их в работе.

С их помощью обеспечивается классификация данных по проекту, клиенту, рабочему процессу, статусу и другим факторам. Благодаря метаданным нетрудно проверить, правильно ли функционируют рабочие процессы, потоки документов и маршруты задач. Метаданные выступают как микросигналы, которые добавляются к документам, и корпоративная информация, включая темные данные, становится доступной и экспортируемой. Такой подход к управлению информацией позволяет включать темные данные в результаты поиска и распределять их по запросам пользователей. Организация получает полный обзор корпоративного контента.

8.Искусственный интеллект.

Искусственный интеллект (ИИ) как нельзя лучше подходит для обработки большого объема постоянно меняющейся информации. Машина делает все то же самое, что должен был бы сделать человек, но при этом вероятность ошибки значительно снижается. ИИ стимулирует идеи и ускоряет принятие решений. Более того, он сокращает трудоемкие ручные процедуры и ускоряет обслуживание внутренней инфраструктуры организации.

9.Blockchain.

Blockchain – это технология распределенного реестра, которая позволяет ускорить и упростить многочисленные интернет-транзакции, в том числе международные. Благодаря этой технологии снижаются затраты на проведение транзакций. Интеграция Blockchain с Big Data несет в себе синергетический эффект и открывает бизнесу широкий спектр новых возможностей, в том числе позволяя:

– получать доступ к детализированной информации о потребительских предпочтениях, на основе которых можно выстраивать подробные аналитические профили для конкретных поставщиков, товаров и компонентов продукта;

– интегрировать подробные данные о транзакциях и статистике потребления определенных групп товаров различными категориями пользователей;

– получать подробные аналитические данные о цепях поставок и потребления, контролировать потери продукции при транспортировке (например, потери веса вследствие усыхания и испарения некоторых видов товаров);

– противодействовать фальсификации продукции, отмыванию денег, мошенничеству и т. д.

Доступ к подробным данным об использовании и потреблении товаров в значительной мере раскрывает потенциал технологии Big Data для оптимизации ключевых бизнес-процессов, снижения рисков, появления новых возможностей создания продукции, отвечающей актуальным потребительским предпочтениям.

Технология распределенного реестра обеспечивает целостность информации, а также надежное и прозрачное хранение всей истории транзакций. Big Data, в свою очередь, предоставляет новые инструменты для эффективного анализа, прогнозирования, экономического моделирования и, соответственно, открывает новые возможности для принятия более взвешенных управленческих решений.

10. Облачные хранилища.

Хранение и обработка данных становятся более быстрыми и экономичными по сравнению с расходами на содержание собственного дата-центра и возможное увеличение персонала. Аренда облака представляется гораздо более дешевой альтернативой содержанию собственного дата-центра.

Объектные хранилища (например, Amazon S3, Google Cloud Storage, Microsoft Blobs Storage) являются высоконадежными и предназначены для хранения большого количества файлов и сотен петабайт данных. Именно их используют многие сервисы синхронизации и обмена файлами.

Файлы в объектном хранилище сопровождаются метаданными, которые позволяют обрабатывать эти файлы как объекты: документы, видеозаписи, проекты, фотографии и т. п. Для взаимодействия с облачным объектным хранилищем используется программный интерфейс (API).

2.Анализ больших данных в облаках

Обрабатывать большие данные можно в дата-центре компании, на физических серверах. Для хранения, обработки и анализа больших данных нужны соответствующие возможности IT-инфраструктуры. Кроме того, потребуются расходы для содержания собственного дата-центра с десятками и сотнями серверов, обеспечения информационной и физической безопасности и бесперебойности работы. Поэтому часто компании для анализа больших данных переходят к облакам.

У облаков для аналитики больших данных есть определенные преимущества.

1. Экономичность. Анализ данных в публичных облаках может быть экономически выгоднее и дешевле, если компания сталкивается с непредсказуемой нагрузкой, быстро растет или часто тестирует гипотезы.
2. Масштабируемость. Облако позволяет использовать для анализа и хранения больших данных столько ресурсов, сколько нужно и гибко подстраивается под бизнес-процессы.
3. Вместимость. У облаков выше вместимость, практически не ограничен объем хранилища больших данных.
4. Эффективность. Облако позволяет исключить рутину администрирования средств обработки Big Data и сфокусировать команду на более творческих задачах анализа, тестирования бизнес-гипотез и получения ключевой для бизнеса информации.
5. Безопасность. В облаках риск потери данных ниже, а бесперебойность предсказуема и защищена SLA (соглашением о качестве услуг) с провайдером

Компания может организовать собственное, частное облако на базе физической инфраструктуры, арендовать облачные мощности у провайдера или совмещать эти модели.

Частное облако может быть расположено в локальном дата-центре компании или у стороннего поставщика, но инфраструктура всегда размещена в частной сети, аппаратное и программное обеспечение предназначено для одной компании. Как правило, такие облака разворачиваются крупными организациями, которых закон обязывает хранить данные у себя: госорганами, финансовыми и медицинскими учреждениями.

У частного облака есть плюсы: IT-ресурсы проще настроить под потребности компании, их использует только одна компания, она полностью контролирует всю инфраструктуру. Но есть и минусы: стоимость развертывания частного облака достаточно высока: нужно организовать собственный ЦОД, на котором будет развернута облачная платформа, нужно обслуживать оборудование, оплачивать услуги персонала, администрирующего систему. Кроме того, собственное оборудование компании постоянно устаревает, а приложения требуют обновления. При аренде облака все это берет на себя провайдер.

Если данные компании будут храниться в одном месте (одном ЦОДе), то есть риск их потери, например, из-за стихийного бедствия или пожара. Избежать этого можно с помощью распределенного ЦОДа: когда инфраструктура дублируется в других дата-центрах. Однако такой вариант IT-инфраструктуры еще дороже. Кроме того, хранение данных в частном облаке и полный контроль компании над инфраструктурой не исключают злоупотреблений со стороны сотрудников: данные могут быть похищены или утрачены из-за непредумышленных и умышленных действий персонала.

Для хранения и обработки данных можно использовать публичное облако. Оно управляется провайдером услуг, у которого компания арендует готовую платформу для анализа Big Data, такую форму аренды называют облачная платформа как услуга (PaaS). При этом облаком пользуются совместно

несколько компаний, однако каждая получает доступ только к своим данным.

Уровень сервиса, гарантии защиты и конфиденциальности прописывают в SLA, NDA (соглашении о неразглашении) и других соглашениях. Поставщик несет юридическую и финансовую ответственность за работу приложений, размещенных в облаке, и сохранность информации бизнеса.

В общедоступных облаках ниже риск потери данных и доступа к сервисам, так как хранение данных и выполнение приложений на многих серверах параллельно обеспечивают защиту от сбоев. Кроме того, публичные облака обладают почти неограниченной емкостью и «резиновым» масштабированием – провайдер может выдать компании столько мощностей, сколько нужно для обработки данных, почти мгновенно, даже если их количество неожиданно вырастет в десятки раз.

Есть два основных варианта предоставления услуг анализа больших данных в облаке.

1. Подход IaaS (Infrastructure as a Service, инфраструктура как услуга) – провайдер предоставляет клиенту виртуальные машины, хранилище и необходимые подключения. Клиент отвечает за донастройку операционной системы, установку приложений, их интеграцию и администрирование. Этот подход дает компании максимальную гибкость в выборе платформы анализа больших данных и контроле над ее тонкими конфигурациями, но требует усилий по ее администрированию.

2. Подход PaaS (Platform as a Service, платформа как услуга) – провайдер развертывает и настраивает для пользователя все сервисы у себя в облаке, пользователю нужно только указать количество необходимых ресурсов. Ему не придется заниматься установкой, настройкой программного обеспечения и его поддержанием.

Сервис для анализа больших данных PaaS обычно состоит из предварительно настроенного кластера на основе платформ анализа данных с открытым кодом, например: Hadoop, Spark, Kafka, с некоторыми предварительно загруженными и настроенными инструментами. Из нескольких таких инструментов в облаке можно составлять «конвейеры» обработки

больших данных. Провайдеры таких PaaS обеспечивают легкую интеграцию с другими облачными сервисами хранения и машинной обработки.

Есть возможность использовать и гибридное облако.

Гибридное облако – это комбинация частного и публичного облака. Такой вариант подходит для компаний, у которых уже есть своя инфраструктура, но нужно снизить нагрузку на нее или протестировать новые сервисы без первоначальных капитальных затрат. Общедоступное облако можно использовать для систем с большим объемом данных, у которых отсутствуют требования к хранению данных «у себя», а частное облако – для ситуаций, когда такие требования есть: например, для определенных типов персональных и финансовых данных.

Характеристики частного и публичного облака

Экономичность

Частное облако -Требуются затраты на оборудование, персонал, инфраструктуру, как и с традиционной IT-инфраструктурой.

Публичное облако -Аренда предполагает оплату по модели pay-as-you-go – компания платит только за используемые мощности, нет первоначальных вложений и затрат на обслуживание. Выгодно малому и среднему бизнесу, подходит для новых проектов в крупных компаниях без ЦОД

Масштабируемость

Частное облако - Возможности масштабирования ограничены мощностью физического оборудования, скоростью закупки и ввода в эксплуатацию новых мощностей

Публичное облако - Может подстроиться под изменения и выделить больше мощностей для хранения и обработки данных за несколько минут. Если ресурсы для анализа Big Data стали не нужны, мощности облачной IT-инфраструктуры не тратятся, компания за них не платит

Эффективность

Частное облако - Компания сама обслуживает и администрирует облако

Публичное облако - Команда компании меньше занимается обслуживанием системы обработки данных, сосредоточивается на создании и тестировании идей, что повышает эффективность аналитики

Быстрый запуск проекта (time-tomarket)

Частное облако - Может замедлить выпуск IT-продуктов на рынок, так как требуются огромные инфраструктурные мощности с высокими капитальными затратами на запуск

Публичное облако - Позволяет запустить IT-инфраструктуру без больших первоначальных инвестиций, создать и настроить инфраструктуру для анализа данных за считанные часы; конкретный PaaS-сервис подключается за минуты

Отказоустойчивость

Частное облако - Можно обеспечить средствами Disaster Recovery, но потребуются серьезные капитальные вложения, расходы на введение в эксплуатацию и поддержку этих средств

Публичное облако - Провайдер обеспечивают бесперебойную работу, что сводит простои инфраструктуры к минимуму.

Обеспечение требований законодательства

Частное облако - Сама компания следит за выполнением требований законодательства и регуляторов

Публичное облако- Ответственность за соблюдение законодательства, требований и стандартов, сертификацию ЦОД лежит на провайдере

Тема№8 Визуализация данных и результатов анализа

Визуализация - наглядное представление результатов анализа. Визуализация очень важна в современном мире, особенно с большими объемами данных. Все потому, что во всей получаемой информации существует множество связей и зависимостей, просто так их очень сложно заметить. Кроме того, она дает ответы на многие вопросы гораздо быстрее. К примеру, по колонке цифр не все так хорошо ясно, как по графику.

Типы визуализации. Выделим условно три типа визуализации.

- Научная визуализация. При моделировании различных объектов или процессов появляются большие объемы данных.
- Информационная визуализация. Описание/представление некой абстрактной информации, полученной при сборе и обработке многокатегориальных данных, для анализа которых необходимо применять различные количественные и качественные меры оценки.
- Визуализация работы программного обеспечения.

Визуализация BigData имеет определенные задачи: 1. визуализация потоков данных; 2. визуальный интеллектуальный анализ данных; 3. визуальный поиск и рекомендации; 4. описание ситуаций на основе больших данных с использованием визуализации; 5. масштабируемые методы параллельной визуализации; 6. современные аппаратные средства и архитектуры для анализа и визуализации данных; 7 человеко-компьютерный интерфейс и визуализация больших данных; 8 приложения визуализации больших данных.

Традиционные виды визуализации

Графики и диаграммы

Используется как для презентации данных, так и для анализа. Существуют порядка 15 общеизвестных типов диаграмм, а всего их более 60, при этом их количество увеличивается с каждым днём – люди придумывают новые типы для визуализации сложных и необычных данных.

Инфографика

Инфографика стала очень популярна в последние годы, хотя существуют уже давно. Инфографика относится к журналистике данных, где графики и схемы объясняют какие-либо факты по выбранной теме. Обычно инфографика статична и представляет собой длинную «простыню» с картинками и текстом. Отличительной особенностью инфографики является то, что в ней приводятся уже готовые выводы, то есть читателя проводят за руку по выбранной теме и при этом приправляют это все цифрами и картинками. Часто использу-

ется рисованный или анимационный стиль. Часто используется не к месту или «для красоты», хотя, конечно же, есть замечательные и интересные примеры.

Презентация и анализ данных

Один самых привычных способов использования визуализации данных - презентация информации в виде диаграмм или инфографики. И если с этим все понятно, то использование визуализации для анализа информации в основном используется только бизнес-аналитиками и учеными.

При анализе данных с помощью визуализации используют так называемое быстрое прототипирование – то есть создание большого количества различных визуальных представлений одних и тех же данных. Делается это для возможности нахождения скрытых, на первый взгляд, взаимосвязей и зависимостей, а также первичной оценки набора данных для возможности применения в дальнейшем более сложных инструментов анализа. Этот подход называется Exploratory data analysis (EDA), что на русский можно перевести как разведывательный анализ данных. Основное отличие от презентации данных - визуализация здесь может быть «черновой», но выполняется быстро и одним человеком или небольшой рабочей группой.

Интерактивный сторителлинг

Сторителлинг – это преподнесение какой-либо полезной информации в форме интересного рассказа. Интерактивный сторителлинг – рассказ, с которым слушатель может взаимодействовать. Пользователь может управлять отображением информации и находить те зависимости, которые не нашёл автор. В этом смысле он близок к разведывательному анализу данных, но отличается тем, что данные заранее обработаны и представлены в удобном для анализа виде, а также имеются подсказки или заранее прописанные сценарии использования.

Поэтому, чаще всего интерактивный сторителлинг называют интерактивной инфографикой, но для того чтобы ей

стать недостаточно просто к статичной инфографике добавить всплывающие окошки.

Дашборды и бизнес аналитика

Для визуализации больших данных активно используют дашборды – дисплеи, на которых выведены все необходимые показатели в одном месте в виде графиков, диаграмм и таблиц. Проектирование эффективных дашбордов – сложная и неординарная задача. Зачастую их перегружают ненужной информацией или стараются использовать все возможные типы шаблонных графиков. Часто для того, чтобы спроектировать хороший дашборд, необходимо создание новых типов визуализации информации. Тематика активно развивается за счет все большего применения аналитики в бизнесе.

Визуализация в медицине и науке

Специфический вид визуализации Его целью обычно является выделение закономерностей или аномалий. От обычной визуализации данных отличается тем, что часто бывает трёхмерной и требует специальной подготовки для интерпретации.

Карты и картограммы

Карты – одни из древнейших способов визуализации, отображающих окружающую реальность. Картограмма – карта с нанесенной на неё информацией в виде цвета или других способов. Картограммы могут быть использованы для отображения любой информации – от плотности населения, до частоты использования мобильных телефонов в каждом районе страны.

Облако тегов

Каждому элементу в облаке тегов присваивается определенный весовой коэффициент, который коррелирует с размером шрифта. В случае анализа текста величина весового коэффициента напрямую зависит от частоты употребления (цитирования) определенного слова или словосочетания. Позволяет читателю в сжатые сроки получить представление о ключевых моментах сколько угодно большого текста или набора текстов.

Кластерграмма

Метод визуализации, использующийся при кластерном анализе. Показывает, как отдельные элементы множества данных соотносятся с кластерами по мере изменения их количества. Выбор оптимального количества кластеров – важная составляющая кластерного анализа.
Исторический поток
Помогает следить за эволюцией документа, над созданием которого работает одновременно большое количество авторов. В частности, это типичная ситуация для сервисов wiki в том числе. По горизонтальной оси откладывается время. По вертикальной - вклад каждого из соавторов, т.е. объем введенного текста. Каждому уникальному автору присваивается определенный цвет на диаграмме

Тема № 9 Алгоритм нечеткого поиска в базах данных

Постановка задачи и определение области применения.

Задача анализа алгоритмов нечёткого поиска на сегодняшний момент актуальна, так как область применения данных алгоритмов невероятно велика и разнообразна. Начнём с распознавания рукописных символов, которое с массовым распространением устройств с сенсорным экраном активно используется для обеспечения удобства ввода. Введённый символ преобразуется в комбинацию цифр в зависимости от последовательности произведённых жестов, и полученная комбинация сравнивается со значениями, заранее известными для всех символов используемого алфавита, записанными в таблицу. Символ, для которого совпадение будет самым полным, и считается распознанным. Именно для определения полноты совпадения и используются алгоритмы нечёткого поиска, поскольку распознанные значения могут отличаться от заложенных в таблице для некого конкретного символа.

Следующая область, где данные алгоритмы успешно применяются, это формы заполнения информации на сайтах и полноценные поисковые системы вроде Google или Yandex.

Также данные алгоритмы активно используются в биоинформатике для сравнения генов, белков, хромосом, для работы с базами данных в системах мониторинга лесопожарной обстановки, обработки массивов данных в интересах кредитных организаций и многих других областях.

Анализ существующих алгоритмов.

Рассмотрим существующие алгоритмы нечёткого поиска и отберём самые актуальные на данный момент для дальнейшего анализа. Начнём с разработанного Робертом Расселом и Маргарет Кинг Оделл алгоритма Soundex. Это один из алгоритмов сравнения двух строк по их звучанию. Он устанавливает одинаковый индекс для строк, имеющих схожее звучание в языке согласно заданной таблице схожих по звучанию символов и их сочетаний.

Следующий из рассматриваемых алгоритмов — алгоритм расширения выборки. Он основан на сведении задачи о нечетком поиске к задаче о точном поиске. Данный метод подразумевает построение наиболее вероятных «неправильных» вариантов поискового шаблона. Т.е. строится множество всевозможных «ошибочных» слов, например, получающихся из исходного в результате одной операции редактирования, после чего построенные термины сравниваются на точное соответствие.

Ещё один из алгоритмов для нечёткого поиска — это алгоритм, использующий код Хэмминга. Он давно и успешно применяется при кодировании и декодировании, позволяя восстановить утерянную при передаче информацию. Следует отметить, что, несмотря на большую эффективность кодов Хемминга, они не лишены определенных недостатков. Линейные коды, как правило, хорошо справляются с редкими и большими опечатками. Однако, их эффективность при сравнении слов с частыми, но небольшими ошибками, менее высока. Также стоит обратить внимание на то, что в данном алгоритме присутствуют дополнительные затраты на кодирование информации.

Следующий из рассматриваемых алгоритмов это алгоритм, использующий триангуляционные деревья, которые позволяют индексировать множества произвольной структуры при условии, что на них задана метрика. Существует довольно много различных модификаций данного метода, но все они не слишком эффективны в случае текстового поиска и чаще используются в базе данных изображений или других сложных объектов.

Алгоритм Вагнера-Фишера, который позволяет для двух строк найти расстояние Левенштейна — минимальное количество операций вставки одного символа, удаления одного символа и замены одного символа на другой, необходимых для превращения одной строки в другую. Данный алгоритм имеет ряд значительных преимуществ, а именно: относительно невысокую сложность реализации, возможность качественного сравнения схожести более чем двух строк, несколько вариантов реализации, которые можно использовать в зависимости от конфигурации системы, универсальность для всевозможных алфавитов.

К недостаткам же можно отнести, что при перестановке местами слов или их частей получаются сравнительно большие расстояния. Значения между совершенно разными короткими словами оказываются маленькими, а между похожими и длинными строками — значительными.

Алгоритм Вагнера-Фишера.

Рассмотрим формулу, по которой можно найти расстояние Левенштейна. Элементы строк нумеруются с первого символа.

Пусть S_1 и S_2 — две строки (длиной M и N соответственно) над некоторым алфавитом, тогда редакционное расстояние (расстояние Левенштейна) $d(S_1, S_2)$ можно посчитать по следующей рекуррентной формуле (1)

$$d(S_1, S_2) = D(M, N) \tag{1}$$

где:

$$D(i,j) = \begin{cases} 0; i = 0, j = 0 \\ i; j = 0, i > 0 \\ j; i = 0, j > 0 \\ \min \begin{pmatrix} \mathrm{D(i, j-1)+1}, \\ \mathrm{D(i-1, j)+1}, \\ \mathrm{D(i-1, j-1)+m(S_1[i], S_2[j])} \end{pmatrix}; \mathrm{j>0, i>0} \end{cases} \tag{2}$$

Где $m(a,b)$ равна нулю, если $a = b$ и единице в противном случае; $\min(a,b,c)$ возвращает наименьший из аргументов.

Здесь шаг по i символизирует удаление (D) из первой строки, по j — вставку (I) в первую строку, а шаг по обоим индексам символизирует замену символа (R) или отсутствие изменений (M).

Справедливы следующие утверждения (3):

$$\begin{aligned} & d(S_1, S_2) \geq \big||S_1| - |S_2|\big| \\ & d(S_1, S_2) \leq \max(|S_1|, |S_2|) \\ & d(S_1, S_2) = 0 \Leftrightarrow S_1 = S_2 \end{aligned} \tag{3}$$

Рассмотрим формулу (2) более подробно. Очевидно, что редакционное расстояние между двумя пустыми строками равно нулю. Также очевидно то, что для получения пустой строки из строки длиной i, нужно совершить i операций удаления, а для получения строки длиной j из пустой, нужно произвести j операций вставки.

Осталось рассмотреть нетривиальный случай, когда обе строки непустые.

Для начала заметим, что в оптимальной последовательности операций их можно произвольно менять местами.

В самом деле, рассмотрим две последовательные операции:

• Две замены одного и того же символа — неоптимально (если заменить x на y, потом y на z, то выгоднее было сразу поменять x на z).

• Две замены разных символов можно менять местами. • Два стирания или две вставки можно менять местами.

• Вставка символа с его последующим стиранием — неоптимально (можно их обе отменить).

• Стирание и вставку разных символов можно менять местами.

• Вставка символа с его последующей заменой — неоптимально (лишняя замена).

• Вставка символа и замена другого символа меняются местами.

• Замена символа с его последующим стиранием — неоптимально (лишняя замена).

• Стирание символа и замена другого символа меняются местами.

Символ «a», на который кончается S_1, в какой-то момент был стёрт. Сделаем это стирание первой операцией. Итак, стёрли символ «a», после чего превратили первые i-1 символов S_1 в S_2 (на что потребовалось $D(i-1, j)$ операций), значит, всего потребовалось $D(i-1, j)+1$ операция.

Символ «b», на который кончается S_2, в какой-то момент был добавлен. Сделаем это добавление последней операцией. Превратили S_1 в первые j-1 символов S_2, после чего добавили «b». Аналогично предыдущему случаю, потребовалось $D(i-1, j)+1$ операций.

Оба предыдущих утверждения неверны. Если символы добавлялись справа от финального «a», то, чтобы сделать последним символом «b», нужно было или в какой-то момент добавить его (но тогда утверждение 2 было бы верно), либо заменить на него один из этих добавленных символов (что тоже невозможно, потому что добавление символа с его последующей заменой не оптимально). Значит, справа от финального «a» символы не добавлялись. Стирание самого финального «a» не проводилось, поскольку утверждение 1 неверно. Получается, что единственный способ изменения последнего символа — его замена. Заменять символ 2 или больше раз не оптимально. Значит,

1. Если $a = b$, то последний символ не изменялся. Поскольку он также не стирался и справа от него не приписывалось, он не влиял на наши действия, и, значит, было выполнено $D(i-1, j-1)$ операций.
2. Если $a \neq b$, то последний символ изменялся один раз. Сделаем эту замену первой. В дальнейшем, аналогично предыдущему случаю, нужно выполнить $D(i-1, j-1)$ операций, значит, всего потребуется $D(i-1, j-1)+1$ операций.

Было реализовано и исследовано 3 варианта этого алгоритма:

- Матричная реализация: строим полноценную матрицу D согласно формуле (2).
- Рекурсивная реализация: начинаем с последнего элемента матрицы D, в котором и содержится искомое расстояние, и рекурсивно находим все недостающие для расчётов элементы по формуле (2).
- Стековая реализация: будем сохранять следующие значение i, j и промежуточное результирующее значение в стеке. Тем самым можно отказаться от рекурсии и заменить её циклом, выполняемым, пока стек не окажется пустым.

Модифицированный алгоритм Вагнера-Фишера.

Рассмотрим алгоритм поиска расстояния Дамерау-Левенштейна. Для его вычисления необходимо внести в алгоритм Вагнера-Фишера несколько изменений, а именно: независимо от выбранной реализации (ранее их предлагалось 3 варианта, далее будет рассмотрен матричный способ представления данных) работать теперь нужно не с двумя предыдущими строками матрицы, а с тремя. Появляется необходимость выполнять дополнительную проверку на использование транспозиции. Если она применялась, то при расчете расстояния необходимо учесть её стоимость. Данная модификация избавляет алгоритм поиска расстояния Левенштейна от основного недостатка, и расстояния между похожими длинными словами не оказываются теперь столь значительными

Тема №10 Цифровая экономика

Начиная со второй половины XX в., информационные технологии приобретают все более значимую роль в экономическом развитии многих стран мира. Единое информационное экономическое пространство, формирование которого стало возможным благодаря научно-техническому прогрессу, способствует экономическому росту и повышению производительности труда, созданию инновационных рабочих мест и цифровых активов, расширению возможностей и прав граждан, улучшению доступа к глобальным рынкам и повышению конкурентоспособности предприятий, повышению качества государственных услуг и др.

Цифровизация бизнеса, начавшись с локальных внутрифирменных и корпоративных проектов, постепенно приобретает глобальные масштабы, а крупные игроки цифрового бизнеса вышли на первые позиции в мире.

По имеющимся оценкам, доля цифровой экономики в ВПП развитых стран мира за период 2010–2016 гг. выросла с 4,3% до 5,5%, в развивающихся странах этот показатель изменился с 3,6% до 4,9%. В России эта доля на 2010 г. составляла 1,9%, а на 2016 г. — уже 2,8%. Несмотря на сравнительное отставание, наша страна демонстрирует довольно

высокую динамику: можно отметить значительное увеличение доли цифровой экономики в ВВП РФ.

Подобно тому, как неравномерно развивается экономика и общество в целом, так же неравномерно происходит и их цифровая трансформация. Политика, правовые нормы, традиции и культура, достигнутый уровень экономического развития, развитость образования и собственной технологической базы, а также многие другие факторы играют существенную роль в формировании цифровой экономики той или иной страны.

Цифровая экономика по своей сути интер- и транснациональна. Поэтому, несмотря на стремление к защите национального цифрового пространства, которое демонстрируют правительства многих стран, одновременно наблюдается противоположная тенденция, связанная с унификацией технических стандартов и правил регулирования в этой сфере. Так, в Европейском союзе насчитывается свыше 400 млн. интернет-пользователей, но его рынок все еще фрагментирован. Лидеры стран ЕС, в этой связи, активно работают над созданием единого цифрового рынка этого интеграционного объединения. Подобные проблемы могут возникать и на уровне отдельных, достаточно крупных, стран. Например, в Индии имеется свыше 460 млн интернет-пользователей. Но индийская цифровая экономика мультиязычна (финансовые операции в ней осуществляются на нескольких языках), что негативно сказывается на функционировании цифрового рынка.

Взрывной рост социальных сетей, увеличение количества смартфонов, облегчение широкополосного доступа к интернету, распространение технологий машинного обучения и искусственного интеллекта изменяют современный мир. Цифровая трансформация организаций, как коммерческих, так и некоммерческих (в том числе государственных) — это реакция на развитие и активное распространение по всему миру новых информационных цифровых технологий. При этом, исходя из господствующей в науке со времени Ренессанса парадигмы прогресса, мы полагаем, что главная цель развития цифровой экономики — улучшить жизнь населения, повысив

качество товаров и услуг, произведенных с использованием современных цифровых технологий, а также их доступность.

Эффективное развитие рынков в цифровой экономике возможно только при наличии развитых технологий, поэтому меры по ее стимулированию должны быть сфокусированы на двух направлениях. Первое — институты; требуется их перестройка и модернизация для создания условия развития цифровой экономики (нормативное регулирование цифровых рынков и цифрового производства, подготовка кадров с цифровыми компетенциями и т. д.). Второе — техническая инфраструктура (сети передачи данных, центры обработки данных, программные сервисы и др.), создание которой требует не только значительных усилий, но и инвестиций.

Несмотря на имеющиеся препятствия и сложности, цифровая экономика в целом в мире продолжает бурно развиваться. Так, например, согласно имеющимся оценкам, «в 2015 г. объем рынка [интернет-торговли — прим. авт.] составил 1,8 трлн. долл. (+17,7% к 2014 г.). При этом доля интернет-торговли в совокупном объеме розничной торговли в мире постепенно растет, она увеличилась практически в 1,5 раза с 6,5% в 2012 г. до 8,6% в 2015 г.». «Объем онлайн-продаж к 2019 г. вырастет до уровня 3,5 трлн. долл. Кроме того, доля интернет-торговли в мировом ритейле увеличится до 12%». Растет и производственный сектор цифровой экономики. Автоматизация производства, большие данные и искусственный интеллект, использование которых стало возможным благодаря цифровым технологиям, трансформируют производственные процессы и модели производственно-технологической кооперации, ускоряют и удешевляют выпуск различной продукции, выполнение работ и оказание услуг. Это позволяет открыть новые пути использования человеческого потенциала, но, одновременно с этим, может порождать социальные проблемы, связанные с исчезновением (прежде всего — в развитых странах) ряда массовых, «традиционных» профессий.

Развитие цифровой экономики в РФ

В целях развития цифровой экономики в России 9 мая 2017 г. был издан указ «О Стратегии развития информационного общества в Российской Федерации на 2017–2030 годы», который определил программу мероприятий по развитию экономики в России на среднесрочную перспективу с учетом возможностей ее информатизации и цифровизации. В развитие этого документа, 28 июля 2017 г. было выпущено распоряжение Правительства России, утвердившее программу «Цифровая экономика Российской Федерации».

Основными целями программы являются: создание условий для развития высокотехнологичных отраслей и недопущение создания ограничений в традиционных отраслях экономики; повышение конкурентоспособности отраслей национальной экономики и ее усиление на мировом рынке. Утвержденная программа состоит из пяти базовых направлений: нормативное регулирование; образование и трудовые ресурсы; формирование исследовательских компетенций; ИТ-инфраструктура; кибербезопасность.

Согласно документу, в ближайшее время должны выйти на рынок не меньше десяти национальных компаний-лидеров (среди операторов экосистем), которые смогли бы конкурировать на мировых рынках. Отмечается, что к этому же сроку в стране должны функционировать 10 цифровых платформ для базовых областей экономики: в цифровом образовании, цифровом здравоохранении, для создания «умного города». Кроме того, в сфере цифровых услуг успешное функционирование должны осуществлять не меньше 500 малых и средних бизнесов в сфере создания цифровых технологий и оказания цифровых услуг. В программе также указано, что количество студентов высших учебных заведений, обучающихся по специальностям, связанным с информационными технологиями, через 8 лет будет составлять 120 тыс. в год. А количество выпускников, обладающих профессиональными знаниями на среднем уровне, должно составлять 800 тыс. в год.

Акцент в программе ставится на построении инфраструктуры, которая необходима при создании и функционировании цифровой экономики. Прежде всего, это центры обработки

данных, сети связи и доступ к интернету. По сути, данная программа представляет список нормативно закрепленных целей развития цифровой экономики в России, а конкретные мероприятия, инструменты их реализации и источники финансирования будут утверждаться регулярно пересматриваемым трехлетним планом правительства.

Россия имеет неплохие стартовые позиции для развития цифровой экономики. Так, например, согласно данным Росстата, аудитория интернета в России в конце 2017 г. достигла 89 млн. человек (73% населения в возрасте от 12 до 64 лет), что на 3% больше, чем в 2016 г. При этом порядка 60% населения РФ пользуются интернетом, в том числе через мобильные устройства, а 20% населения страны используют доступ в интернет только с мобильных устройств.

Сильными сторонами России также являются: доступность информационных и коммуникационных технологий (ИКТ); способность населения использовать ИКТ благодаря наличию базовых навыков в области образования, связанных с качеством образовательной системы, уровнем грамотности взрослых и уровнем охвата средним образованием; развитие инфраструктуры ИКТ (покрытие мобильной сети, пропускная способность интернета, доступность цифрового контента); проникновение и распространение ИКТ на индивидуальном уровне. Слабыми сторонами были признаны «Политическая среда и регулирование» и «Эффективность законодательных органов».

Цифровая экономика России получила значительный импульс развития за последние годы. Определенных успехов достигли частные компании, преобразуется рынок труда, при поддержке государства реализуются беспрецедентные инфраструктурные проекты, повышающие уровень доступности цифровых услуг для населения и бизнеса, широкое распространение получили интернет, мобильная и широкополосная связь. И это уже повлекло такие положительные изменения, как: повсеместное распространение интернета; развитие банковского сектора; расширение рынка электронных услуг; улучшение инфраструктуры городов; повышение доступности

учебных материалов; появление все более современной компьютерной техники и др.

Применение Больших данных в развитии Российской цифровой экономики

Большие данные являются основной темой в цифровизации российской экономики, тем не менее, очевидно, чтобы разблокировать потенциал больших объемов данных России необходимо разработать последовательную политику и практику для сбора, транспортировки, хранения и использования данных. Данная политика должна решать вопросы защиты частной жизни, открытый доступ к данным, инфраструктура и измерение. Также очевидно, что существует несоответствие между предложением и спросом на квалифицированные кадры в области управления данными и аналитикой (наука о данных).

Государственный сектор является не только важным пользователем данных, но и основным их источником. Более широкий доступ к эффективному использованию информации государственного сектора (PSI), как это предусмотрено в Рекомендации Совета Организация экономического сотрудничества и развития (ОЭСР) приняла ОЭСР по PSI, могут создавать выгоды во всей цифровой экономике.

Ведущие аналитики в сфере цифровой экономики называет 4 сферы, где раскрытие потенциала Больших данных позволит получить максимальный экономический эффект в современном развитии государства в целом: интернет-реклама, коммунальные услуги, логистика и транспорт, государственное управление.

Преимущества, которые Большие данные могут создавать в этих секторах, включают:

♦ развитие новых товаров и услуг на основе передачи данных;

♦ улучшение производства или доставки процессов;

♦ улучшение маркетинга (путем предоставления целевых рекламных объявлений и персонализированных рекомендаций);

♦ новые организационные и управленческие подходы или значительное совершенствование процесса принятия решений в рамках существующей практики;
♦ расширение научных исследований и развитие.

Три свойства — объем, скорость и разнообразие — рассматриваются как три основных характеристики больших объемов данных и обычно упоминаются как три V (volume, velocity, variety), тем не менее, это технические свойства, которые зависят от эволюции хранения данных и технологии их обработки.

Значение также имеет четвертое «V — value», которое связано с увеличением социально-экономического значения, которое будет получено с использованием больших объемов данных, это экономический потенциал и социальное значение, которое, в конечном счете, мотивирует накопление, обработку и использование данных. Авторы многих исследований считают, что готовность к применению технологии Больших данных в экономике складывается из четырех составляющих: накопленные данные, адаптация технологий, отлаженные процессы и персонал. Успех компании в области Больших данных в равной степени зависит от зрелости компании во всех этих областях, включая цифровизацию экономики.

Проблемы, вызванные ускорением технологических инноваций, развитием цифровой экономики, достигли нового уровня сложности и масштаба — сегодня ответственность за кибербезопасность в организациях больше не является обязанностью главного сотрудника по безопасности, а затрагивает всех. Сегодня все компании претерпевают трансформационную цифровизацию своих отраслей, что открывает новые рынки. Лидеры кибербезопасности должны взять на себя более сильную и стратегическую руководящую роль. Эта новая роль неотъемлемо связана с необходимостью выйти за рамки роли наблюдателей нормативных требований.

Как неотъемлемая часть на успех современного бизнеса, и цифровой экономики, кибербезопасность имеет прямое влияние на деловую репутацию, стоимость акций, выручка, отношения с клиентами и время выхода продукта на рынок.

Развитие экономических инноваций в компаниях так же идет за счет инвестиций в более широкий ряд нематериальных активов, от программного обеспечения и Больших данных до дизайна, фирм с сотрудниками, обладающими конкретными навыками, и новых организационных процессов, основанные на знаниях.

Слияние нескольких направлений, в том числе растущей социально-экономической деятельности в Интернете (через интернет-сервисы, такие как социальные сети, электронную коммерцию, электронное здравоохранение и электронное правительство) и снижение стоимости сбора, хранения и обработки данных, приводят к производству и использованию огромных объемов данных, которые, как было отмечено выше, называются «Большие данные» (Big Data).

Данные обрабатываются и передаются круглосуточно по всему миру, в сочетании с мощным анализом данных, Большие данные предлагают перспективу значительного создания стоимости, социальных благ и повышения производительности труда.

Ведущие эксперты в процессе развития цифровой экономики, выделяют следующие факторы роста данных:

1. слияние технологических разработок, в частности, увеличение повсеместного широкополосного доступа в Интернет, рост интеллектуальных устройств и смарт-ИКТ приложений, таких как «смарт-счетчиков» и «смарт-транспорта» на основе сенсорных сетей;

2. большое снижение стоимости доступа в Интернет в течение последних 20 лет; 3. снижение затрат на хранение данных приводит к тому, что данные могут храниться в течение длительного времени, а может и бесконечно;

4. инструменты для обработки становятся все более мощными, сложными, повсеместными и недорогими, что позволяет данные легко найти и связать.

Сегодня в условиях цифровизации экономики, это могут делать не только правительства и крупные корпорации, но и многие другие среднего сегмента компании. Облачные вычисления играют значительную роль в увеличении хранения

данных и мощности переработки при переходе в цифровую экономику.

Программное обеспечение с открытым исходным кодом (OSS), которое охватывает полный спектр решений, необходимых для больших объемов данных и Интернета вещей, в том числе хранения, обработки и аналитики, также внесли свой значительный вклад, чтобы сделать большие аналитические данные доступными для более широких слоев населения в процессе цифровизации экономики.

Потоки данных через публичные и частные сети могут улучшить качество статистики и могут создавать близкие к реальному времени доказательства для выработки политики в таких областях, как цены, занятость, экономическое производство и развитие, а также демография. Проект «Миллиард цен», например, собирает информацию о ценах через Интернет, чтобы вычислить ежедневный онлайн индекс цен и оценить годовую и месячную инфляцию. Более полумиллиона цен на товары (не услуги) собирается ежедневно с различных ресурсов. Кроме того, в отличие от официальных цифр, которые публикуются ежемесячно с отставанием на недели, онлайн индекс цен обновляется ежедневно с отставанием всего в три дня.

Использование данных может существенно отличаться в разных секторах экономики, в некоторых из них данные используются более интенсивно, чем в других. Уровень интенсивности использования Больших данных (измеряется как среднее количество данных в компании) является самым высоким в области финансовых услуг (в том числе в сфере ценных бумаг, инвестиционных услугах и банковских услугах), в СМИ, коммунальных услугах, в государственном секторе и дискретном производстве.

Цели использования Больших данных:

♦ для создания новых видов продукции (товаров и услуг);

♦ для оптимизации или автоматизации производства, процесса.

Такие данные могут занимать центральное место в новых организационных и управленческих подходах — для повышения научных исследований и разработок в цифровизации экономики.

Основной вопрос, который приходится сейчас решать на ИТ уровне, это интеграция различных приложений в рамках сквозных бизнес-процессов, именно поэтому сейчас актуальна технология RPA (Robotic process automation), которая позволяет «склеивать» внутренние и внешние приложения, а также микросервисная архитектура, которая позволяет уйти от монолитных приложений в ИТ-архитектуре.

Переход части ИТ-приложений в облако — процесс неизбежный, а интеграция с собственными внутренними ИТ-приложениями будет все сложнее при большом количестве изменений в бизнес-процессах. Поэтому, на дальнем горизонте планирования можно ожидать появления экосистем на корпоративных рынках ИТ-приложений, где между участниками экосистемы интеграция будет предварительно настроена.

Контрольные вопросы

1. Информация и особенности ее хранения и обработки
2. Сущность понятия Big Data
3. Подходы к управлению Big Data
4. Содержание и задачи процесса управления большими данными
5. Становление технологии работы с большими данными
6. Современные технологии управления большими данными
7. Большие данные и хранилища данных
8. Подходы к архитектуре и принципам проектирования хранилищ данных
9. Системы хранения данных
10. Архитектура корпоративной системы хранилища – DWH
11. Хранилище данных и озеро данных
12. Аналитика больших данных и ее инструментарий (Виды аналитики и их сущность)
13. Инструментарий для анализа больших данных (реляционные и нереляци-онные СУБД)
14. Технологии аналитики в памяти (in-memory)
15. Методы анализа и обработки больших данных
16. Анализ больших данных в облаках
17. Визуализация данных и результатов анализа
18. Цифровая экономика
19. Развитие цифровой экономики в РФ

Литература

1. Парфенов, Ю. П. Постреляционные хранилища данных : учеб. пособие / Ю. П. Парфенов. — Екатеринбург : Изд-во Урал. ун-та, 2016. — 120 с

2. В.А. Резниченко ЧТО ТАКОЕ BIG DATA https://doi.org/10.15407/pp2019.03.086

3. Кобзаренко Д.Н., Мустафаев А.Г. Учебное пособие дисциплины «Анализ больших данных» для направления подготовки 38.03.05«Бизнесинформатика», профиль «Электронный бизнес». – Махачкала: ДГУНХ, 2019 г. – 107 с.

4. И. Б. Тесленко, А. М. Губернаторов, О. Б. Дигилина, В. Е. Крылов. Big Data = Большие данные : Владим. гос. ун-т им. А. Г. и Н. Г. Столетовых. – Владимир : Изд-во ВлГУ, 2021. – 123 с.

5.А.В. Макшаков, А.Е. Журавлев, Л.Н. Тындыкврь Большие данные. Big Data. Учебник для вузов, Санкт-Петербург: Лана, 2023,188с.