Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»

BIG DATA
БОЛЬШИЕ ДАННЫЕ

Учебное пособие

Владимир 2021

УДК 005.572:004(075.8)
ББК 65.290-2я73
В56

Авторы:
И. Б. Тесленко, А. М. Губернаторов, О. Б. Дигилина, В. Е. Крылов

Рецензенты:
Доктор экономических наук, доцент
зав. кафедрой менеджмента и маркетинга
Владимирского государственного университета
имени Александра Григорьевича и Николая Григорьевича Столетовых
Н. Н. Ползунова

Генеральный директор СП ООО «ТехноСтройИнвест»
В. А. Вашурин

Издается по решению редакционно-издательского совета ВлГУ

Big Data = Большие данные : учеб. пособие / И. Б. Тесленко
В56 [и др.] ; Владим. гос. ун-т им. А. Г. и Н. Г. Столетовых. – Владимир : Изд-во ВлГУ, 2021. – 123 с.
ISBN 978-5-9984-1425-1

Изложены основные принципы, подходы и направления технологий Big Data. Представлены обзор экосистемы и областей применения больших данных, архитектура системы обработки больших данных, а также раскрывается тема систем управления технологиями Big Data.

Предназначено для студентов направления подготовки бакалавриата 38.03.05 «Бизнес-информатика» всех форм обучения, а также руководителей организаций и специалистов, занимающихся вопросами цифровизации предпринимательской и управленческой деятельности.

Рекомендовано для формирования профессиональных компетенций в соответствии с ФГОС ВО.

Табл. 4. Ил. 16. Библиогр.: 90 назв.

УДК 005.572:004(075.8)
ББК 65.290-2я73

ISBN 978-5-9984-1425-1

ВВЕДЕНИЕ

За последнее десятилетие объем данных, с которыми приходится иметь дело, многократно увеличился, и в то же время стоимость хранения данных снизилась. Частные компании и исследовательские учреждения обрабатывают терабайты информации о взаимодействиях своих пользователей, бизнесе, социальных сетях, а также собирают данные с датчиков таких устройств, как мобильные телефоны и автомобили. Задача современной эпохи состоит в том, чтобы разобраться в этом море данных. Именно здесь аналитика больших данных вступает в свои права.

Big Data Analytics в основном включает в себя сбор данных из разных источников, изменение их таким образом, чтобы они стали доступными для использования аналитиками, и, наконец, предоставление продуктов данных, полезных для бизнеса.

Учебное пособие признано формировать определенную систему знаний у студентов в области информационных технологий: об особенностях хранения и обработки информации; сущности понятия *Big Data* и его значении; принципах и подходах к управлению *Big Data*, а также современных технологиях хранения данных и инструментариях для их анализа.

При написании учебного пособия авторский коллектив руководствовался следующими важнейшими методологическими и методическими положениями.

1. Содержание пособия должно полностью соответствовать ФГОС ВО для подготовки бакалавров по направлению 38.03.05 «Бизнес-информатика».

Пособие может быть использовано студентами в качестве дополнительного материала для углубления знаний при подготовке докладов, рефератов, а также аспирантами и преподавателями.

2. Теоретической основой работы послужили современные концепции, категории и понятия, ведущие мировые практики, используемые в области *Big Data*.

3. Учебное пособие выступает как основа воспитания экономического мышления, понимания современных задач в области *Big Data* в контексте реализации информационных стратегий современных предприятий.

Издание подготовлено преподавателями кафедры «Бизнес-информатика и экономика» Владимирского государственного университета имени Александра Григорьевича и Николая Григорьевича Столетовых: доктором экономических наук, профессором заведующим кафедрой бизнес-информатики и экономики И. Б. Тесленко (главы 1, 2, 3); доктором экономических наук профессором А. М. Губернаторовым (введение, главы 4, 5, заключение); доктором экономических наук, профессором О. Б. Дигилиной (глава 6); кандидатом физико-математических наук, доцентом В. Е. Крыловым (глава 7).

Глава 1. ИНФОРМАЦИЯ, БОЛЬШИЕ ДАННЫЕ, BIG DATA

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[1].

Слово «**информация**» происходит от латинского слова *informatio*, что в переводе означает «сведение, разъяснение, ознакомление».

Понятие «информация» многозначно. Существуют различные *подходы* к пониманию информации.

Например, с *семантической* точки зрения информация – это сведения, обладающие новизной[2].

С *технической* точки зрения информация – это все сведения, которые представлены в определенной форме для хранения, передачи и обработки с помощью технических средств[3].

С точки зрения *традиционного* (*обыденного*) подхода информация – это сведения, знания, сообщения о положении дел, которые человек воспринимает из окружающего мира с помощью органов чувств (зрения, слуха, вкуса, обоняния, осязания).

С точки зрения *вероятностного* подхода информация – это сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состоянии, которые уменьшают имеющуюся о них степень неопределенности и неполноты знаний.

---

[1] Информация. Свойства информации [Электронный ресурс]. URL: https://www.sites.google.com/site/3kursmimi/1-informacia-svojstva-informacii (дата обращения: 15.06.2020).

[2] Информация [Электронный ресурс]. URL: https://investments.academic.ru/1012/ (дата обращения: 20.06.2020).

[3] Информация. Свойства информации.

По мнению основоположника кибернетики *Н. Винера*, информация – это обозначение содержания, полученного людьми из внешнего мира в процессе приспосабливания к нему самих людей и их чувств[4].

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).

Говоря об информации, люди всегда подразумевают информационные процессы (например, получение, хранение информации). 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)[5].

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).

---

[4] Информация.

[5] Информация. Свойства информации.

Выделяют следующие *виды* информационных процессов: получение, обработка, передача, хранение, поиск, кодирование и защита информации[6].

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 еженедельнике, 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 коди-

[6] Информация. Свойства информации.

рования, 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.

Пример информационного процесса представлен на рис. 1.

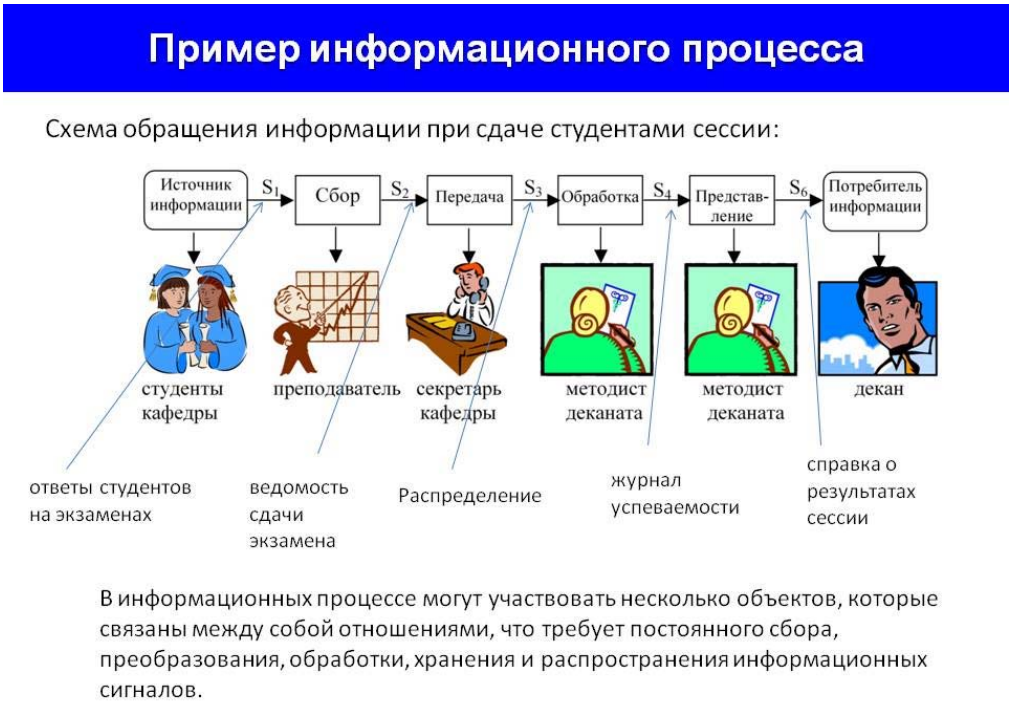

Рис. 1. Пример информационного процесса

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. Исследованием непосредственно информации занимаются две комплексные отрасли науки: **кибернетика** и **информатика**[7].

[7] Информация. Свойства информации.

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[8].

Именованную совокупность данных, отражающую состояние объектов и их отношений в рассматриваемой предметной области, называют **базой данных (БД)**. БД используют для организации управления и автоматизации, например, предприятия, вуза и т. д.

Для управления БД разработаны языковые и программные средства, предназначенные для создания, наполнения, обновления и удаления БД[9].

Люди испокон веков пытались сохранить имеющиеся у них данные (наскальные записи, надписи на костях и бересте, деревянных столбах) и использовать их. Однако с развитием письменности способность фиксировать опыт и события окружающего мира значительно увеличила объем собираемых данных.

Самая ранняя форма письма была разработана в Месопотамии около 3200 г. до н. э. и использовалась для коммерческого учета. Этот тип учета фиксирует так называемые *транзакционные данные* – информацию о событиях, таких как продажа товара, выставление счета, доставка, оплата кредитной картой, страховые требования и т. д.

Нетранзакционные данные, например демографические, также имеют долгую историю. Первые известные переписи населения прошли в Древнем Египте около 3000 г. до н. э. Причина, по которой древние правители вкладывали так много усилий и ресурсов в масштабные

[8] Данные [Электронный ресурс]. URL: https://dic.academic.ru/dic.nsf/ruwiki/71919 (дата обращения: 18.06.2020).

[9] Базы данных [Электронный ресурс]. URL: https://siblec.ru/informatika-i-vychislitelnaya-tekhnika/bazy-dannykh (дата обращения: 18.06.2020).

проекты по сбору данных, заключалась в том, что им нужно было повышать налоги и увеличивать армию[10].

С середины XVII в. накопление достаточного количества данных привело к необходимости начать серьезный анализ имеющейся информации.

В XIX в. возникла проблема подсчета получаемых данных. С ней столкнулись специалисты в США в 1880 г. при переписи населения: с имеющимися в то время подходами к работе с данными подсчеты производились в течение 8 – 10 лет, т. е. результаты были готовы как раз к следующей переписи.

В 1881 г. американский инженер и изобретатель Герман Холлерит создал устройство (табулятор), которое, оперируя перфокартами, сокращало 10-летний труд до трех месяцев. Им была создана компания *TMC*, специализирующаяся на создании табулирующих машин, которую позже купила компания *C-T-R*, переименованная в 1924 г. в *IBM* [11].

Во время Второй мировой войны британские ученые создали машину *Colossus*, данные с которой подавались через перфорированное колесо, поскольку личной памяти у нее не было. Эта машина ускоряла расшифровку сообщений неприятеля с нескольких недель до нескольких часов.

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).

С появлением Интернета возникают поисковые системы; одна из них – система *AltaVista*. Она использовала лингвистический алгоритм, разбивая поисковую фразу на слова и проводя поиск по существующим индексам для ранжирования результата. Использование этой системы позволило увеличить количество запросов с 300 тыс. до 80 млн в день[12].

---

[10] Келлехер Д., Тирни Б. Наука о данных. М. : Альпина Диджитал, 2020. 222 с.

[11] История больших данных (Big Data) – часть 1 [Электронный ресурс]. URL: https://www.computerra.ru/234239/istoriya-bolshih-dannyh-big-data-chast-1/ (дата обращения: 05.07.2020).

[12] Там же.

По мере увеличения объемов информации, большая доля которой представляла собой неструктурированные данные, вопросы корректной интерпретации информационных потоков становились все более актуальными и сложными. Крупные компании *IT*-рынка стали приобретать наиболее успешные узкоспециализированные фирмы (стартапы) и развивать инструменты для работы с большими объемами данных. Работа с такими данными велась и в научно-исследовательских центрах[13].

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[14].

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.

Документ Кодда послужил основой для современных БД и разработки *SQL* (языка структурированных запросов), международного стандарта формулировки запросов к БД. 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[15].

---

[13] Большие данные (Big Data) [Электронный ресурс]. URL: https://www.tadviser.ru/index.php/Статья:Большие_данные_(Big_Data) (дата обращения: 21.06.2020).

[14] Там же.

[15] Келлехер Д., Тирни Б. Указ. соч.

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* (Дж. Дин и С. Гемават) 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*[16].

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. До 2009 г. термин *Big Data* рассматривался только с точки зрения научного анализа, но после выхода еще нескольких статей пресса стала широко его использовать[17].

Итак, начало 1970-х гг. ознаменовало приход современной технологии с реляционной моделью данных Эдгара Кодда и последующий взрывной рост генерации данных и их хранения, который в 1990-х гг. привел к развитию хранилищ, а позднее – к возникновению феномена больших данных.

Что же такое большие данные и *Big Data*?

2. 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

[16] История больших данных (Big Data) – часть 1.

[17] Big Data – что такое системы больших данных? Развитие технологий Big Data [Электронный ресурс]. URL: https://promdevelop.ru/big-data/ (дата обращения: 15.06.2020).

связи, *GPS*-сигналы от автомобилей для транспортной компании, информация о транзакциях всех клиентов банка, всех покупках в крупной ретейл сети и многое другое[18].

Под термином «большие данные» понимают и наличие данных больше, чем 100 Гб (например, 500 Гб, 1 ТБ), и такие данные, которые невозможно обрабатывать в *Excel*, т. е. традиционным способом, и данные, которые невозможно обработать на одном компьютере[19].

Некоторые специалисты предлагают следующую *классификацию* объемов данных:

– *большие наборы данных:* от 1 тыс. мегабайт (1 гигабайт) до сотен гигабайт;

– *огромные наборы данных:* от 1 тыс. гигабайт (1 терабайт) до нескольких терабайт;

– *Big Data:* от нескольких терабайт до сотен терабайт;

– *Extremely Big Data:* от 1 тыс. до 10 тыс. терабайт, т. е. от 1 до 10 петабайт (1 тыс. петабайт, или 1 тыс. × 1 тыс. терабайт, – это 1 эксабайт)[20].

Согласно отчету *McKinsey Institute* «Большие данные: новый рубеж для инноваций, конкуренции и производительности» термин «большие данные» относится к наборам данных, размер которых превосходит возможности типичных БД по занесению, хранению, управлению и анализу информации[21].

Таким образом, понятие «большие данные» отличается от традиционного подхода к терминам «большое количество данных» или «большой массив данных».

Большие данные, по определению, предложенному в 2011 г. Мервом Адрианом из компании *Gartner*, – это данные, сбор, управление и обработку которых невозможно осуществить с помощью наиболее часто используемых аппаратных сред и программных инструментов в течение допустимого для пользователя времени.

---

[18] Большие данные (Big Data).

[19] Big Data от А до Я. Часть 1: Принципы работы с большими данными, парадигма MapReduce [Электронный ресурс]. URL: https://habr.com/ru/company/dca/blog/267361/ (дата обращения: 05.07.2020).

[20] Революция Big Data : Как извлечь необходимую информацию из «Больших Данных»? [Электронный ресурс]. URL: http://statsoft.ru/products/Enterprise/big-data.php (дата обращения: 18.06.2020).

[21] Большие данные (Big Data).

В докладе *McKinsey Global Institute* от 2011 г. дается такое определение: большие данные – это наборы данных, размеры которых выходят за пределы возможностей по сбору, хранению, управлению и анализу, присущих обычному программному обеспечению БД[22].

Словосочетание «большие данные» подразумевает не только их объем. Согласно *Gartner Group* 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.

Не случайно большие данные сравнивают с приливной волной, а ее приручение – это настоящее испытание.

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[23].

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. Полу-

[22] Фрэнкс Б. Укрощение больших данных. Как извлекать знания из массивов информации с помощью глубокой аналитики. М. : Манн, Иванов и Фербер, 2014. 352 с.

[23] Там же.

структурированные 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[24].

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.

Большие данные – это очередная волна новых данных, которая раздвигает существующие пределы. В отличие от традиционных данных поток больших данных – это большой объем, скорость передачи, разнообразие и сложность. «Укрощение» больших данных похоже не на закачку воды в бассейн, а скорее на питье воды из шланга: человек отхлебывает только то, что ему нужно, а остальному позволяет течь мимо.

Главное в процессе «укрощения» больших данных – определить, какие фрагменты имеют долгосрочное стратегическое значение, какие пригодны только для немедленного и тактического использования, а какие вообще бесполезны.

– С большими данными связаны определенные риски. Так, например, организация может оказаться настолько перегруженной

[24] Фрэнкс Б. Указ. соч.

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[25].

Неслучайно в научной литературе есть расширенный подход к пониманию больших данных, который обозначают термином *Big Data*.

Термин *Big Data* шире простого понятия «большие данные». Под ***Big Data*** понимают не только какой-то конкретный большой объем данных (хотя это тоже важно), но и методы их обработки и использования, методы поиска необходимой информации в больших массивах, хотя это и не означает, что методы работы с большими данными нельзя применять к небольшим массивам данных.

По мнению исследователей, *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*[26].

Становится ясным, что 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

[25] Фрэнкс Б. Указ. соч.

[26] Big Data от А до Я.

огромному объему своих данных, но не иметь необходимых инструментов, чтобы установить взаимосвязи между этими данными и сделать на их основе соответствующие выводы.

К тому же данные сейчас обновляются все чаще и чаще, и традиционные методы анализа информации не могут угнаться за огромными объемами постоянно обновляемых данных, что в итоге и вызывает необходимость разработки технологий обработки больших данных, приводя к появлению *Big Data*.

В отечественной литературе *Big Data* рассматривается как социально-экономический феномен, который связан с появлением новых технологических возможностей для анализа огромного количества данных. Фактически *Big Data* – это решение проблем управления информацией и альтернатива традиционным системам управления данными[27].

Big Data – это естественное развитие приемов математической статистики и усовершенствованный метод обработки информации[28].

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[29].

3. Значение *Big Data*

В настоящее время нет такой сферы, где бы использование технологий *Big Data* не давало положительного результата. Так, в коммерции эти технологии выполняют роль создателя портрета потенциального клиента с целью персонализирования, таргетирования рекламы.

---

[27] Что такое Big data: собрали все самое важное о больших данных [Электронный ресурс]. URL: https://rb.ru/howto/chto-takoe-big-data/ (дата обращения: 19.06.2020).

[28] Анализ Big Data в медицине поможет страховщикам понимать потребности клиентов в здравоохранении и страховании [Электронный ресурс]. URL: https:// forinsurer.com/news/17/01/18/34782 (дата обращения: 05.07.2020).

[29] Большие данные (Big Data).

Благодаря *Big Data* не является секретом, какие продукты покупает конкретный клиент, как часто ходит в магазин, какой у него средний чек, где он отдыхает, каков его примерный уровень дохода и др. Информация, попадая в систему *Big Data*, связывается воедино и создает приближенный к действительности портрет покупателя.

Пример. Разработчики игры *World of Tanks* на основе *Big Data* провели исследование информации обо всех игроках, что помогло спрогнозировать их возможный будущий отток, организовать более эффективное взаимодействие с пользователями.

Благодаря технологиям *Big Data* компании могут проводить репутационный анализ, обрабатывая комментарии пользователей в социальных сетях, на торговых площадках, форумах и других ресурсах.

Банковский сектор активно использует аналитику больших данных в своих процессах для привлечения клиентов, анализа их активности, решения проблем информационной безопасности.

Интернет вещей – еще одна сфера, которую изменили *Big Data*. Это анализ тысячи параметров на производстве, в дорожном движении, сельском хозяйстве, здравоохранении и других отраслях.

Перспективы *Big Data* огромны. Например, только в медицине можно проводить бесчисленное множество экспериментов по борьбе с неизлечимыми болезнями, опасными вирусами и анализировать их результаты.

Интерес к инструментам сбора, обработки, управления и анализа больших данных проявили многие ведущие *IT*-компании, поскольку большие данные открывают отличные возможности для освоения новых ниш рынка и привлечения новых заказчиков. Это такие компании, как *Amazon, Dell, eBay, Facebook, Fujitsu, Google, IBM, Microsoft, Oracle, SAP, SAS, Teradata, Yahoo, HSBC, Nasdaq, Coca-Cola, Starbucks, AT&T,* государственные органы и др.

К примеру, *IBM* применяет методы обработки больших данных к проводимым денежным транзакциям. С их помощью было выявлено на 15 % больше мошеннических транзакций, что позволило увеличить сумму защищенных средств на 60 %. Также были решены проблемы с ложными срабатываниями системы – их число сократилось более чем наполовину.

Министерство труда Германии сумело сократить расходы на 10 млрд евро, внедрив технологию *Big Data* в процесс по выдаче пособий по безработице. Было выявлено, что пятая часть граждан данные пособия получала безосновательно[30].

Исследование *Accenture*, в котором приняло участие более 1 тыс. руководителей компаний из 19 стран мира, показало, что главными преимуществами *Big Data* являются: поиск новых источников дохода, улучшение опыта клиентов, новые продукты и услуги, приток новых клиентов и сохранение лояльности старых.

Вместе с тем при внедрении технологий *Big Data* многие компании столкнулись с традиционными проблемами. Для почти половины это проблемы безопасности, финансов, нехватки необходимых кадров, для других – сложности при интеграции с существующей системой (их устаревшая *IT*-инфраструктура неспособна обеспечить необходимую емкость систем хранения, процессы обмена данных, утилиты и приложения, необходимые для обработки и анализа больших массивов неструктурированных данных для извлечения из них ценной информации).

По мнению *Gartner*, в конкурентной борьбе победят именно те, кто научится обращаться с самыми разными источниками информации[31].

Специалисты *TmaxSoft* считают, что если компания станет успешно применять *Big Data*, она сможет предложить продукты и сервисы лучше, чем у конкурентов[32].

На рис. 2 дана сравнительная характеристика показателей деятельности компаний-лидеров по использованию *Big Data* и их конкурентов. Практически во всех указанных сферах и доход *(revenue)*, и общий объем прибыли до вычета расходов по выплате процентов, налогов и начисления амортизации (*Ebitda*) у компаний, применяющих технологии *Big Data*, выше, чем у тех, которые такие технологии не используют.

---

[30] Большие данные (Big Data).

[31] Там же.

[32] Там же.

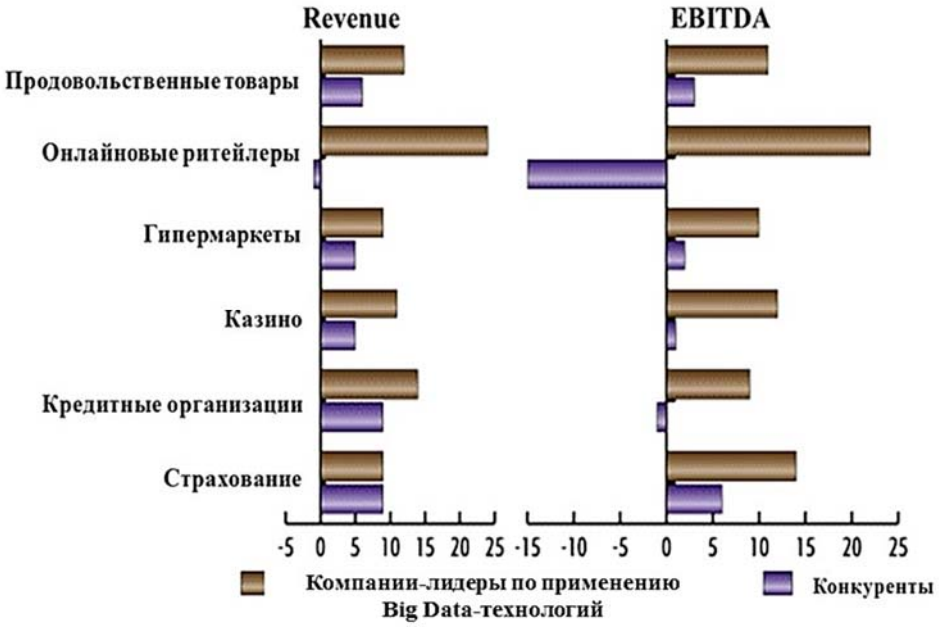

Рис. 2. Сравнительный анализ показателей деятельности компаний – лидеров в использовании Big Data и их конкурентов

По мнению Б. Фрэнка, поскольку *Big Data* появляются везде, их игнорирование опасно для организации. Чтобы оставаться конкурентоспособными, крайне важно, чтобы организации активно анализировали эти новые источники данных и пользовались содержащимися в них ценными сведениями[33].

Познакомимся с основами управления большими данными.

[33] Фрэнкс Б. Указ. соч.

Глава 2. ОСНОВЫ УПРАВЛЕНИЯ БОЛЬШИМИ ДАННЫМИ

1. Подходы к управлению *Big Data*

Использование больших данных потребовало решения вопросов, связанных с хранением и обработкой информации. На это еще в 2001 г. обратил внимание Дуг Лейни из *Meta Group* (входит в состав *Gartner*).

В результате были выявлены три *направления*, на которых стоит сосредоточиться для решения вопросов управления данными: *Volume, Velocity* и *Variety*. Позже они легли в основу описательной модели больших данных под названием 3*V* (*VVV*). Остановимся на них подробнее.

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* компания *IDC* добавила еще один – ***Value***, или ценность информации. В приведенном примере эта ценность была равна нулю, так как потеряла свою актуальность раньше, чем ею смогли воспользоваться за период ***Validity*** – полезного действия этой информации. Ценность информации заключается и в ее достоверности (***Veracity***).

Становится понятно, что 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*[34].

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 (рис. 3).

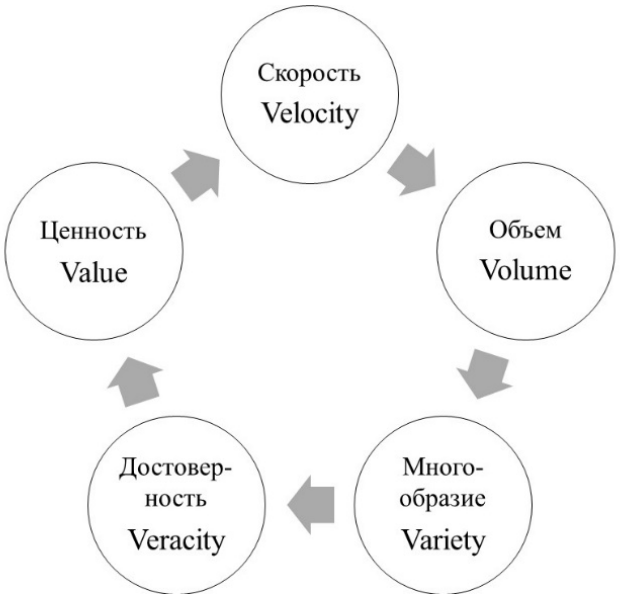

Рис. 3. Описательная модель Big Data

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.

Действительно, 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.

[34] История больших данных (Big Data) – часть 1.

Кроме того, система должна быть хорошо масштабируемой, иначе будут получены недостоверные данные ввиду потери одного из параметров *V*[35].

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.

Для того чтобы следовать этим принципам, необходимы технологии (включают разные методы и способы) средств обработки данных[36].

С учетом принципов и направлений решения вопросов управления большими данными можно так определить *Big Data*: это горизонтально масштабируемая система, основанная на определенных принципах, использующая набор методик и технологий, позволяющих обрабатывать структурированную и неструктурированную информацию и строить связи, необходимые для получения однозначно интерпретируемых человеком данных, не успевших потерять актуальность, и несущая ценность для достижения поставленных целей[37].

2. 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.

---

[35] История больших данных (Big Data) – часть 1.

[36] Big Data от А до Я.

[37] История больших данных (Big Data) – часть 1.

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[38].

---

[38] Шпрингер Е. Технологии big data: как анализируют большие данные, чтобы получить максимум прибыли [Электронный ресурс]. URL: https:// mcs.mail.ru/blog/ tekhnologii-big-data-kak-analiziruyut-bolshie-dannye (дата обращения: 18.06.2020).

После интеграции большие данные подвергаются дальнейшим манипуляциям: анализу, обработке и т. д. Описанный процесс представлен на рис. 4.

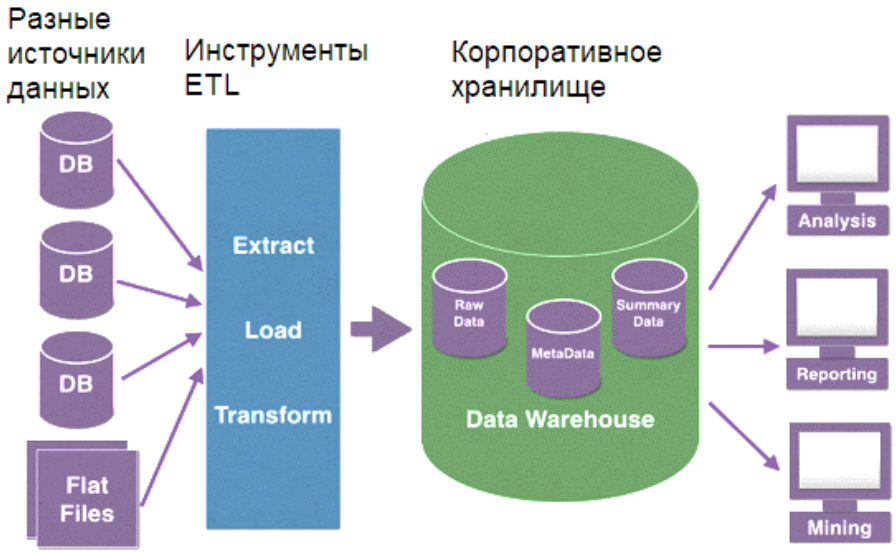

Рис. 4. Схема реализации второй фазы

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[39].

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.

[39] Благирев А., Хапаева Н. Big data простым языком. М : АСТ, 2019. 256 с.

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.

Следует отметить, что данные не обязательно должны проходить все семь фаз; фазы взаимодействия не обязательно выстраиваются в конкретную последовательность; в реальности фазы могут проявляться в хаотичном порядке.

В любом случае стратегия управления данными в организации имеет большую самостоятельную ценность. Современный вид системы управления данными – это результат развития организационных основ работы с *Big Data*[40].

3. Эволюция организационных основ работы с *Big Data*

В первое десятилетие XXI в. термин *Big Data* воспринимался как инфраструктурный – под ним понимался специальный класс БД, которые позволяли быстро обрабатывать большие объемы информации. Название *Big Data* применялось к категории серверов («железа»), которые умели выполнять определенные вычисления.

Они были нужны, потому что обычное «железо» не было приспособлено для работы с большим количеством данных. Ему не хватало памяти и скорости.

Аудиофайлы, изображения, сложные и слабоструктурированные данные мало обрабатывались. Для них требовалось специальное программное обеспечение.

Постепенно стало понятным, что проблема заключается не только в «железе», а в том числе в программном обеспечении («софте»), которое работает на самых обычных компьютерах, объединенных в узлы. Такие конструкции могут работать параллельно над

[40] Благирев А., Хапаева Н. Указ. соч.

конкретной задачей по обработке данных. Их назвали программными комплексами, или кластерами. После осознания этого технологии работы с большими данными стали быстро развиваться[41].

Уже с 2010 г. стали осуществляться первые попытки решения нарастающих проблем больших данных и были выпущены программные продукты, направленные на минимизацию рисков при использовании огромных информационных массивов[42].

В настоящее время к инструментам управления *Big Data*, встроенным в корпоративную архитектуру многих крупных компаний, относят:

– *программно-аппаратные комплексы и ускорители для БД*. Специализированные устройства объединяют хранение, обработку, подключение и быстрое выполнение запросов на выделенной программно-аппаратной платформе, оптимизированной для работы и управления БД. Ускорители БД используют самые последние достижения в хранении данных и оптимизации запросов для снижения размеров БД и повышения скорости выполнения сложных запросов. Если простое обновление аппаратной платформы может повысить производительность традиционной реляционной БД в два раза, то применение специализированных устройств и ускорителей способно улучшить показатель «цена/производительность» почти в сто раз. Важно и то, что эти технологии упрощают управление и администрирование, устраняя потребность в квалифицированной настройке и конфигурировании данных;

– *NOSQL – хранилища данных.* Технология *Not-Only-SQL*, появившаяся в среде Интернета, с самого начала была спроектирована для управления огромными распределенными наборами данных, запрос к которым должен был выполняться за миллисекунды. Вместо нормализации данных по реляционным таблицам, которые затем должны объединяться для ответов на запросы, сверхбольшие массивы данных распределяются по сотням или тысячам процессоров, организованных так, чтобы связанные данные располагались рядом. Запросы выполняются параллельно на всех процессорах; каждый возвращает ответы, основываясь на своих локальных данных. Этот простой и масштабируемый

[41] Благирев А., Хапаева Н. Указ. соч.

[42] Big Data – что такое системы больших данных?

подход оказался очень эффективным и гибким, и он позволяет совместно хранить данные самых разных типов, а также выполнять сложные запросы;

– *автоматизированную аналитику.* Чтобы получить преимущество от обработки сверхбольших массивов данных, требуются хорошая аналитика и группа высококвалифицированных специалистов. Выборка, очистка и обработка терабайтов данных часто выполняется аналитиками вручную, так как некоторые операции не могут быть автоматизированы.

Тем не менее в последнее десятилетие наметился прогресс в самообучающихся алгоритмах, генетических алгоритмах и автоматизированном тестировании, что привело к появлению программ, способных распознавать образцы, делать заключения и улучшаться с течением времени, т. е. самообучаться. Эти системы не всегда работают лучше людей-аналитиков, но их автоматизированные процессы могут оказаться единственным способом масштабирования в соответствии с требованиями сверхбольших массивов[43].

Вышеперечисленные инструменты управления *Big Data* позволили понять важность правильного использования больших данных и привели к трансформации концепции управления всем предприятием.

Если раньше данными занимались только айтишники, работая со специальными операционными хранилищами (они называются *ODS*), куда данные загружались все вместе из разных источников, то теперь стало понятным, что данные крайне важны и на них можно строить успешный бизнес.

Теперь многие понимают термин *Big Data* не просто как технологии управления большими данными, а как новую модель зрелости бизнеса, общества и государства. Пользователь понимает, как с помощью *Big Data* быстро и легально обработать информацию и как ее структурировать таким образом, чтобы результаты этой работы были понятны окружающим.

Big Data привели к централизации управленческих решений и постепенно распространились на все ключевые бизнес-процессы. Это, в

[43] Управление «большими данными» – что должен знать каждый ИТ-директор [Электронный ресурс]. URL: https://www.itweek.ru/idea/article/detail.php?ID =136441 (дата обращения: 05.07.2020).

свою очередь, привело к возникновению новой формы внутренней работы организаций, превратив ее в *data-driven*-организацию[44].

В 2011 г. крупные компании (*Microsoft, Oracle, EMC, IBM*) стали первыми использовать наработки *Big Data* в своих стратегиях развития. *Google, Facebook* и другие крупные компании провозгласили себя *data-driven*-организациями.

***Data-driven*-организации** – это такие компании, в которых все внутренние процессы и большинство решений вокруг них строятся исключительно на основании данных.

В 2007 г. эксперт в области веб-аналитики Авинаш Кошик выделил семь *ключевых шагов*, которые позволяют трансформировать культуру работы организации и перейти к управляемой данными (или дата-управляемой) организации[45].

1. ***Go to the Outcomes*** – **переходите к результатам.**

Основа коллаборации (совместной деятельности, взаимодействия) между людьми с использованием данных лежит, прежде всего, в понимании того, что важно для каждого из участников: от чего зависят их бонусы или выплаты, на что обращают внимание люди, которые принимают решения. Для этого нужно понимать, какими объектами оперирует компания, и это понимание перенести на уровень данных. Традиционная ошибка – начать собирать все данные компании, считать на их основе все возможные метрики и отправлять всем заинтересованным людям отчеты с этими показателями.

2. ***Reporting is not Analysis*** **– отчетность – это еще не аналитика.**

Ключевой ошибкой всегда и везде была простая демонстрация данных в надежде, что решение с использованием этих данных найдется само собой.

3. ***Depersonalise Decisions making*** **– деперсонифицируйте принимаемые решения.**

Переход к фокусировке на тех данных, которые действительно нужны организации, ведет к созданию новой формы культуры, где данным отводят центральное место, а все решения деперсонализированы, потому что важно не мнение людей в офисе, а данные, на которых оно строится.

---

[44] Благирев А., Хапаева Н. Указ. соч.

[45] Big Data – что такое системы больших данных?

Данные сами покажут, что идет не в соответствии с ожиданиями, и это никак не связано с субъективной оценкой.

4. ***Proactive insights rather than reactive*** **– проактивный инсайт (прогноз) важнее реактивной аналитики.**

В тот момент, когда данные получены и началась подготовка инсайта, они уже устарели. Поэтому вместо того чтобы выполнять и готовить отчетность, специалистам нужно выполнить анализ, который еще никто не просит. Он необходим, так как данные быстро устаревают и ряд ключевых аспектов может быть не освещен во время процесса принятия решения.

5. ***Empower your Analyst*** **– расширяйте полномочия аналитиков.**

Для того чтобы аналитик мог потратить свое рабочее время на анализ неструктурированных или слабоструктурированных данных вместо подготовки регулярной отчетности, у него должны быть достаточные для этого полномочия. *Data-driven*-организация вряд ли будет существовать в условиях регулярного процесса выпуска отчетности, на который тратится более 80 % времени работы команды. Аналитики должны тратить минимум времени на отчетность, используя остальное время, чтобы улучшить иные процессы организации по работе с данными и их продуктом – ежемесячной отчетностью.

6. ***Solve the Trinity*** **– треугольник ценности.**

Внутри треугольника находятся метрики и инсайты, которые приводят к действию по созданию ценности. На вершинах треугольника обозначены ключевые направления создания ценности с использованием данных:

– *поведение (behaviour)* – необходимо думать широко при анализе поведения клиентов;

– *результаты (outcomes)* – надо научиться связывать поведение клиентов с ключевыми показателями или критическими факторами успеха организации;

– *опыт (experience)* – инсайты должны приходить через эксперименты, исследования, тестирование клиентов или поиск закономерностей в их поведении. Всем этим необходимо постоянно заниматься[46].

[46] Big Data – что такое системы больших данных?

7. ***Got Process*** **– создайте вокруг процесс.**

Data-driven-организация – это не пункт назначения, а процесс или путь, по которому идет организация. Этот процесс позволяет пользователям и сотрудникам применять тот или иной фреймворк работы с данными. Он не должен быть сложным и запутанным, а должен показывать, кто и на каком конкретном шаге участвует в создании ценности с использованием данных.

Ответственным за данные, аналитику и поиск инсайтов в организации должно быть обособленное бизнес-подразделение (а не *IT*).

Исследования компании *Nucleous Research* в 2014 г. показали, что за каждый вложенный доллар в решения и процессы по аналитике и работе с данными компания получала в среднем 13,01 доллара. Это еще раз подтверждает важность внедрения и правильной организации применения *Big Data*[47].

Вместе с тем в работе с большими данными есть свои сложности.

4. Проблемы использования *Big Data*

Самая большая проблема больших данных – *затраты* на их обработку. Сюда входят расходы и на дорогостоящее оборудование (его приходится регулярно обновлять для поддержания минимальной работоспособности при увеличении объема данных), и на заработную плату квалифицированным специалистам, обслуживающим огромные массивы информации.

Вторая проблема связана с *профессионализмом* самого аналитика, поскольку ему необходимо обрабатывать большое количество информации. Если, например, исследование выдает не два-три, а большое количество результатов, очень сложно остаться объективным и выделить из общего потока данных только те, которые окажут реальное влияние на состояние какого-либо явления или процесса.

Следующая проблема – *потеря информации*. Меры предосторожности требуют не ограничиваться простым однократным резер-

[47] Big Data – что такое системы больших данных?

вированием данных, а делать хотя бы две-три резервные копии хранилища. С увеличением объема растут сложности с резервированием, и *IT*-специалистам требуется найти оптимальное решение данной проблемы.

Не менее важной проблемой является проблема *конфиденциальности Big Data*. При переходе большинства сервисов по обслуживанию клиентов в онлайн очень легко стать мишенью для киберпреступников. Даже простое хранение личной информации без совершения каких-либо интернет-транзакций может быть чревато нежелательными для клиентов облачных хранилищ последствиями.

Общие проблемы с производительностью, обеспечением надежности, оптимизацией хранения, вызванные новым форматом работы с данными, требуют постоянного совершенствования технологии *Big Data*.

Глава 3. ТЕХНОЛОГИИ РАБОТЫ С БОЛЬШИМИ ДАННЫМИ

1. Становление технологии работы с большими данными

Еще в конце 90-х гг 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 (рис. 5).

Рис. 5. Предпосылки развития технологий работы с большими данными

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 цен-

тральный процессор, оперативное записывающее 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[48], 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 (*groupby*).

Каждую часть одной большой задачи можно отдать на обработку одному из узлов единого кластера[49].

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. Например, *Hadoop*-кластер *Yahoo* имеет более 42 тыс. машин.

---

[48] Анализ больших данных: Spark и Hadoop [Электронный ресурс]. URL: https://coincase.ru/blog/47715/ (дата обращения: 20.06.2020).

[49] История больших данных (Big Data) – часть 2 [Электронный ресурс]. URL: https://www.computerra.ru/234346/istoriya-bolshih-dannyh-big-data-chast-2/ (дата обращения: 03.07.2020).

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)[50].

При увеличении объемов информации кластер нужно расширять до заданных задачей размеров.

Итак, 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[51].

В алгоритме *MapReduce* обработка данных происходит в три стадии (рис. 7).

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*.

---

[50] Введение в кластеры [Электронный ресурс]. URL: https://onix.kiev.ua/news.aspx?id=172 (дата обращения: 18.06.2020).

[51] Революция Big Data.

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 (рис. 6)[52].

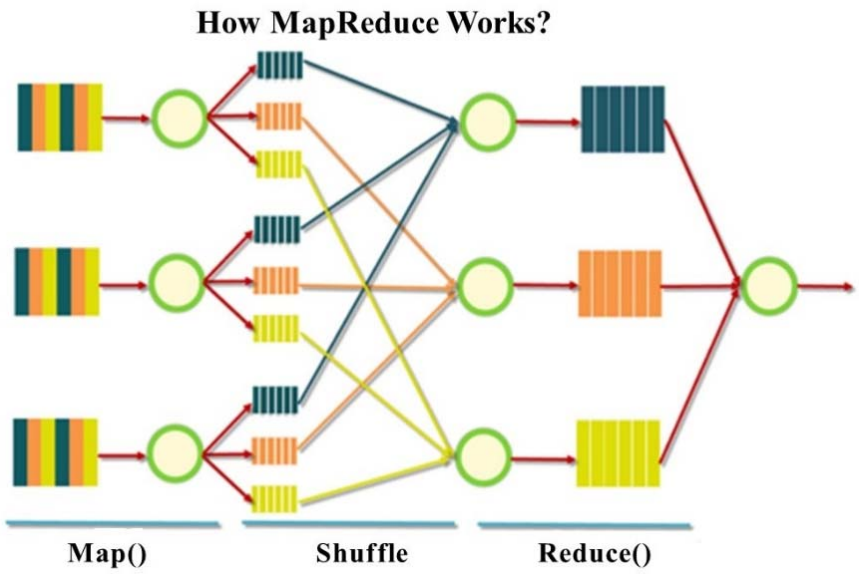

Рис. 6. Стадии алгоритма MapReduce

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.

[52] Big Data от А до Я.

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.

Обратимся к примеру работы алгоритма. Предположим, стоит задача посчитать все упоминания Ивана Иванова, Петра Петрова и Сидора Сидорова на всех страницах в Интернете. Потребуется проанализировать огромнейший объем информации, и для одного узла такая задача просто непосильна. Используя алгоритм *MapReduce*, можно разделить все страницы на части и распределить их анализ на разные ноды кластера.

Сначала данные со страниц будут отданы в функцию *Map*, которая при наличии совпадения выдаст пары «ключ – значение». Это будет (Иван Иванов, 1), (Петр Петров, 1), (Сидор Сидоров, 1). Таким образом, при каждом нахождении упоминания нужных людей функция *Map* будет выдавать ключ (имя и фамилию) и значение, которое свидетельствует об обнаружении упоминания. В итоге получится следующая картина:

* (Сидор Сидоров, 1)
* (Иван Иванов, 1)
* (Петр Петров, 1)
* (Сидор Сидоров, 1)
* (Иван Иванов, 1)

Затем информация будет саккумулирована путем передачи ее в функцию *Reduce*, которая также выдаст на выходе пары «ключ – значение», но уже в обработанном виде:

* (Сидор Сидоров, 2)
* (Иван Иванов, 2)
* (Петр Петров, 1)[53]

[53] Big Data от А до Я.

Поставленная задача выполнена, нужные данные получены. Схематично работа алгоритма представлена на рис 7.

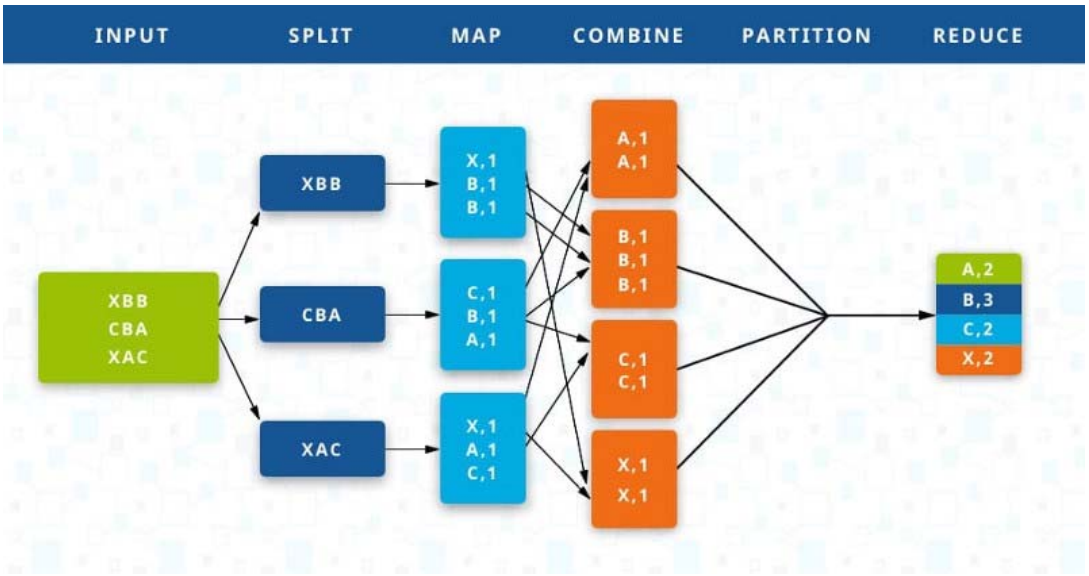

Рис. 7. Схема работы алгоритма MapReduce

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[54].

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.

[54] История больших данных (Big Data) – часть 2.

Так появились такие инструменты, как *Hadoop, Spark, Pig, Hive, Cassandra* и *Kafka*, каждый из которых имеет свои преимущества и недостатки (рис. 8).

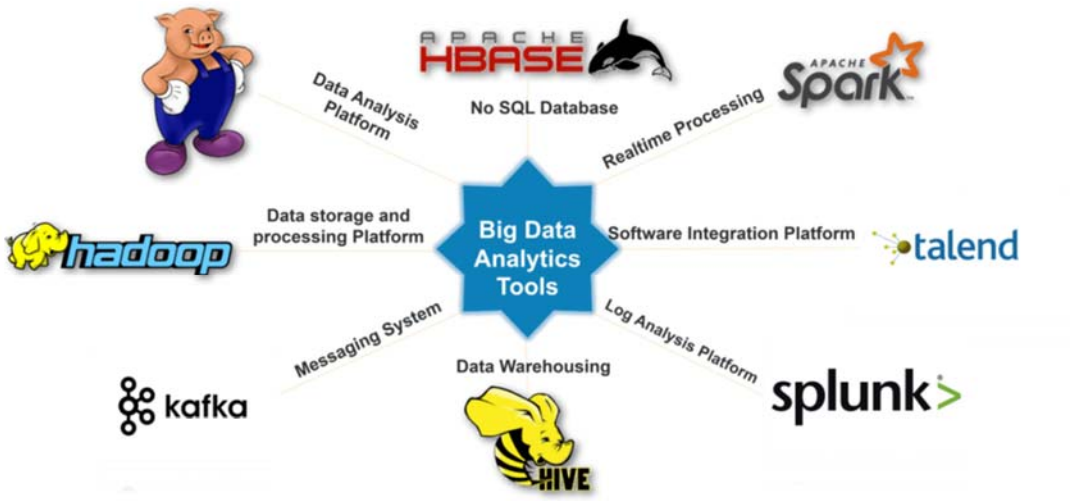

Рис. 8. Инструменты управления большими данными

Остановимся на некоторых из них: *Hadoop* и *Spark*. Их появление относится к началу 2000-х гг.

После публикации в 2003 г. исследователями из компании *Google* общих принципов построения *Google File System (GFS)* – файловой системы, в которой данные разбиваются на отдельные блоки, хранящиеся в нескольких копиях на разных компьютерах – и представления метода *MapReduce* для выполнения распределенных вычислений над данными в *GFS* группа инженеров и исследователей из компании *Yahoo!* приступила к их практической реализации в рамках проекта с открытым исходным кодом, который впоследствии (в 2006 г.) стал известен миру как *Hadoop*. Вариант *GFS*, созданный в рамках этой программной платформы, получил название *Hadoop Distributed Files System (HDFS)*.

Хотя платформа *Hadoop* позволила многим компаниям успешно применять алгоритм *MapReduce* для распределенных вычислений над огромными объемами данных, каждый раз при возникновении новой задачи требовалось написание нового кода для операций *Map* и *Reduce*, что было неудобно и трудоемко. Для решения этой проблемы в 2008 г. инженеры из *Facebook* создали *Hive* – систему управления БД на ос-

нове *Hadoop*. Главной особенностью *Hive* стала поддержка *SQL*-подобных запросов к данным, хранящимся в *HDFS* (этот новый диалект *SQL* получил название *Hive Query Language, HQL*).

В 2009 г. в Калифорнийском университете в Беркли был запущен исследовательский проект *Spark* с целью повышения эффективности распределенных вычислений методом *MapReduce* и создания универсальной платформы для таких вычислений. В 2010 г. *Spark* был опубликован как проект с открытым кодом, а в 2013 г. передан фонду *Apache Software Foundation*[55].

Остановимся подробнее на этих инструментах.

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.

Платформу *Hadoop* разработали Дуг Каттинг и Майк Кафарелл в 2006 г. Их проект был назван в честь игрушечного слоненка сына Д. Каттинга. Через два года *Hadoop* управлял распределенной поисковой системой, развернувшейся на 10 тыс. процессорных ядрах. Основа *Hadoop* – распределенная файловая система *HDFS* и алгоритм распределенных вычислений *Hadoop MapReduce*.

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.[56]

---

[55] Spark и sparklyr для работы с большими данными в R [Электронный ресурс]. URL: https://r-analytics.blogspot.com/2020/02/spark-intro.html (дата обращения: 06.07.2020).

[56] Бородаенко В., Ермаков А. Универсальная платформа обработки больших данных [Электронный ресурс]. URL: https://www.osp.ru/os/2017/03/13052699 (дата обращения: 05.07.2020).

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.

Система *Hadoop* применяется разными компаниями и организациями, например, *Yahoo* – при поиске данных; *Facebook* – при обработке журналов / хранилищ данных; *New York Times* – при анализе видео/изображений и др.[57]

К преимуществам платформы *Hadoop* относят:

– сокращение времени на обработку данных;

– снижение стоимости оборудования;

– повышение отказоустойчивости;

– линейную масштабируемость;

– работу с неструктурированными данными[58].

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*).

---

[57] Анализ больших данных: Spark и Hadoop.

[58] Назаренко Ю. Л. Обзор технологии «большие данные» (Big Data) и программно-аппаратных средств, применяемых для их анализа и обработки // European science, 2017. № 9 (31).

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)[59].

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 многопетабайтные 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

[59] Анализ больших данных: Spark и Hadoop.

распределенного физического хранилища в гораздо более быструю логическую оперативную память. Это сокращает время записи и чтения по сравнению с *Hadoop MapReduce*[60].

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[61].

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)[62]. Основные *отличия Hadoop* и *Apache Spark* представлены в табл. 1.

---

[60] Spark или Hadoop – Какая платформа для Big Data лучше? [Электронный ресурс]. URL: http://spbdev.biz/blog/spark-ili-hadoop-kakaya-platforma-dlya-big-data-luchshe (дата обращения: 18.06.2020).

[61] Там же.

[62] Spark и sparklyr для работы с большими данными в R.

Таблица 1

Отличия платформ *Hadoop* и *Spark* по ряду критериев[63]

Критерий	*Hadoop*	*Apache Spark*
Функционал	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	Позволяет выполнять различные операции над распределенными коллекциями данных, но не обеспечивает их распределенного хранения
Использование	В состав *Hadoop* входит и компонент хранения *Hadoop Distributed File System*, и компонент обработки *MapReduce*, поэтому обработку можно осуществлять без *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
Скорость работы	Работает медленнее из-за пошагового режима обработки в *MapReduce*	Работает быстрее, так как оперирует всем набором данных как единым целым
Устойчивость к сбоям	Устойчива к системным сбоям, поскольку после выполнения каждой операции данные записываются на диск	Восстановление после сбоев осуществляется благодаря тому, что объекты данных хранятся в распределенных в пределах кластера наборах
Аналитические возможности	Не имеет библиотеки машинного обучения, должна связываться со сторонней библиотекой, например с *Apache Mahout*	Включает собственные библиотеки машинного обучения – *MLib*

[63] Пять вещей, которые необходимо знать о Hadoop и Apache Spark [Электронный ресурс]. URL: https://www.osp.ru/news/articles/2015/49/13048137 (дата обращения: 21.06.2020).

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[64]. 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[65]. Неслучайно точка зрения о том, что они дополняют друг друга, является все-таки преобладающей.

В целом, благодаря развитию технологий *Big Data* стали возможными последние достижения в области новых технологий – искусственного интеллекта и глубокого обучения, что позволило машинам выполнять задачи, которые казались абсолютно невозможными всего несколько лет назад.

Систему больших данных невозможно построить без надежной системы хранения данных и сопутствующих технологий[66].

[64] Spark или Hadoop – Какая платформа для Big Data лучше?
[65] Пять вещей, которые необходимо знать о Hadoop и Apache Spark.
[66] Big Data – что такое системы больших данных?

Глава 4. ХРАНИЛИЩА ДАННЫХ: ЭВОЛЮЦИЯ И ОБЩИЕ ОСНОВЫ

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, формировать 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[67]. 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[68].

Хранилище данных – это система, в которой собраны данные из различных источников внутри компании, которые используются для поддержки принятия управленческих решений[69].

---

[67] Telecom & IT. Ликбез № 6. Объектные системы хранения (Object Storage) [Электронный ресурс]. URL: https://shalaginov.com/2019/08/06/6262/ (дата обращения: 15.06.2020).

[68] Келлехер Д., Тирни Б. Указ. соч.

[69] Telecom & IT.

Один из создателей архитектуры хранилищ данных У. Инмон так определяет хранилище данных: это предметно-ориентированная, интегрированная, содержащая исторические данные, неразрушаемая совокупность данных, предназначенная для поддержки принятия управленческих решений[70].

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.

Таблица 2

Сравнительные характеристики хранилищ данных и оперативных систем

Системы хранилищ данных	Оперативные системы
Используются руководством	Используются работниками «переднего края»
Стратегическое значение	Тактическое значение
Поддерживают стратегические направления развития бизнеса	Поддерживают повседневную деятельность
Используются для интерактивного анализа	Используются для обработки транзакций
Предметно-ориентированные	Ориентированы на приложения
Хранят исторические данные	Хранят только текущие данные
Непредсказуемые запросы	Предсказуемые запросы

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

[70] Особенности построения информационных хранилищ [Электронный ресурс]. URL: https://www.osp.ru/os/2003/04/182942 (дата обращения: 05.07.2020).

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.

Например, запрос *OLAP*, выраженный для удобства на естественном языке, может выглядеть так: «Отчет о продажах всех магазинов по регионам и кварталам и разница показателей по сравнению с отчетом за прошлый год». Этот пример показывает, что результат запроса *OLAP* часто напоминает стандартный бизнес-отчет. 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. Для приведенного выше примера запроса *OLAP* необходимы следующие измерения куба данных: продажи по магазинам, продажи по регионам и продажи по кварталам.

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[71].

Итак, в конце 1980-х – начале 1990-х гг. в связи с необходимостью анализа больших наборов данных началась разработка соответствующих хранилищ данных и технологии *OLAP*. Параллельно велись исследования и в других областях.

В 1989 г. математик Григорий Пятецкий-Шапиро поднял вопрос об обнаружении знаний в БД (*KDD*). По сути это тот же глубинный анализ данных. Таким БД обычно сопутствуют существенные знания предметной области, которые могут значительно облегчить обнаружение данных.

Доступ к большим БД недешевый, поэтому необходимо использовать разные статистические методы. Для обнаружения знаний в БД могут оказаться полезными существующие инструменты и методы из различных областей, таких как экспертные системы, машинное обучение, интеллектуальные БД, получение знаний и статистика[72].

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.

---

[71] Келлехер Д., Тирни Б. Указ. соч.

[72] Там же.

Существуют разные точки зрения на вопрос об архитектуре хранилищ данных: относительно подходов к проектированию, количеству уровней, направлений построения хранилищ, систем хранения данных.

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[73].

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'a*, 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[74].

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 (рис. 9). При построении витрин используются основные принципы построения хранилищ данных, поэтому их можно считать хранилищами данных в миниатюре.

---

[73] Что такое витрина данных? Определение, разновидности и примеры [Электронный ресурс]. URL: https://yandex.ru/turbo/s/fb.ru/article/402525/chto-takoe-vitrina-dannyih-opredelenie-raznovidnosti-i-primeryi (дата обращения: 05.07.2020).

[74] Data Mart vs Data Warehouse [Электронный ресурс]. URL: https://habr.com/ru/post/72389/ (дата обращения: 15.06.2020).

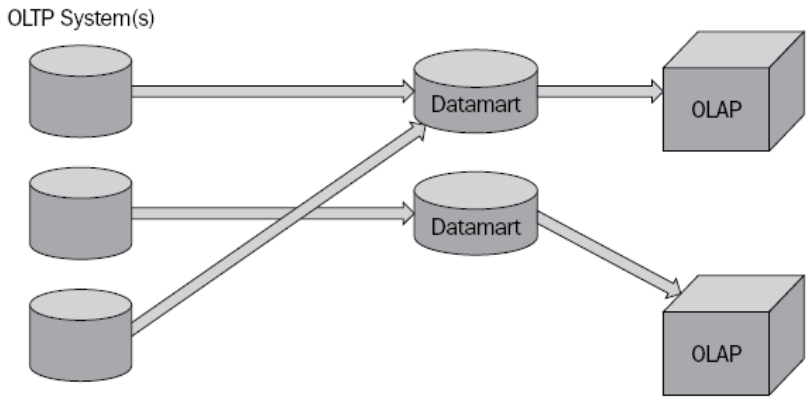

Рис. 9. Построение хранилища данных по Р. Кимболлу

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[75].

Во втором случае построение полноценного корпоративного хранилища данных (*Data Warehouse*) выполняется в *трехуровневой* архитектуре (рис. 10).

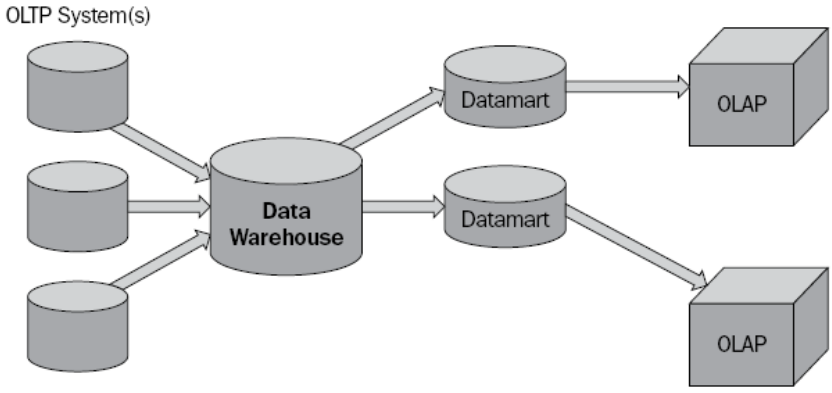

Рис. 10. Построение хранилища данных по Б. Инмону

На *первом (нижнем) уровне* расположены разнообразные источники данных: внутренние регистрирующие системы, справочные си-

[75] Особенности построения информационных хранилищ.

стемы, 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.

Второй уровень содержит сервер *OLAP*, который преобразует данные в структуру, наиболее подходящую для анализа и сложных запросов. Сервер *OLAP* может работать двумя способами: либо в качестве расширенной системы управления реляционными БД, которая отображает операции над многомерными данными в стандартные реляционные операции (*Relational OLAP*), либо с использованием многомерной модели *OLAP*, которая непосредственно реализует многомерные данные и операции[76].

Третий (верхний) уровень представляет собой набор предметно-ориентированных витрин данных, источником информации для которых является центральное хранилище данных. Именно с витринами данных и работает большинство конечных пользователей[77]. Он содержит инструменты, используемые для высокоуровневого анализа данных, создания отчетов клиентами[78].

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

---

[76] Архитектура хранилищ данных: традиционная и облачная [Электронный ресурс]. URL: https://habr.com/ru/post/441538/ (дата обращения: 15.06.2020).

[77] Особенности построения информационных хранилищ.

[78] Архитектура хранилищ данных: традиционная и облачная.

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. Идея создания витрин данных была предложена в 1991 г. международным аналитическим агентством *Forrester Research*. Авторы идеи представляли данное хранилище информации как определенное множество специфических БД, которые содержат в себе сведения, относящиеся к конкретным векторам деятельности корпорации[79].

3. *Модель корпоративного хранилища данных* предполагает хранение агрегированных данных, охватывающих всю организацию. Эта модель рассматривает хранилище данных как сердце информационной системы предприятия с интегрированными данными всех бизнес-единиц[80].

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

[79] Что такое витрина данных?

[80] Архитектура хранилищ данных: традиционная и облачная.

иерархии в измерении. Если каждое измерение содержится в одной таблице, такая схема хранилища данных носит название *«звезда» (star schema)*. Пример такой схемы приведен на рис. 11.

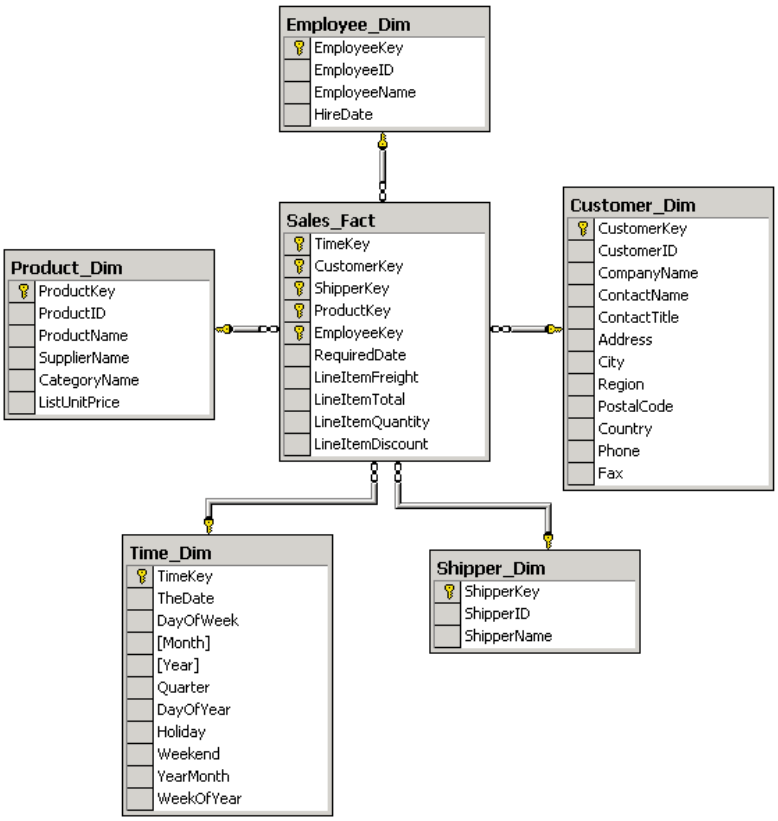

Рис. 11. Пример схемы «звезда»[81]

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 изме-

[81] Хранилища данных [Электронный ресурс]. URL: https://portal.tpu.ru/SHARED/p/PAN/ Wrk/Tab9/Lk.doc (дата обращения: 19.06.2020).

рения и находящиеся в соотношении «один ко многим» в главной таблице измерений, соответствующей нижнему уровню иерархии, иногда называют *консольными таблицами (outrigger table)*. Пример схемы «снежинка» приведен на рис. 12.

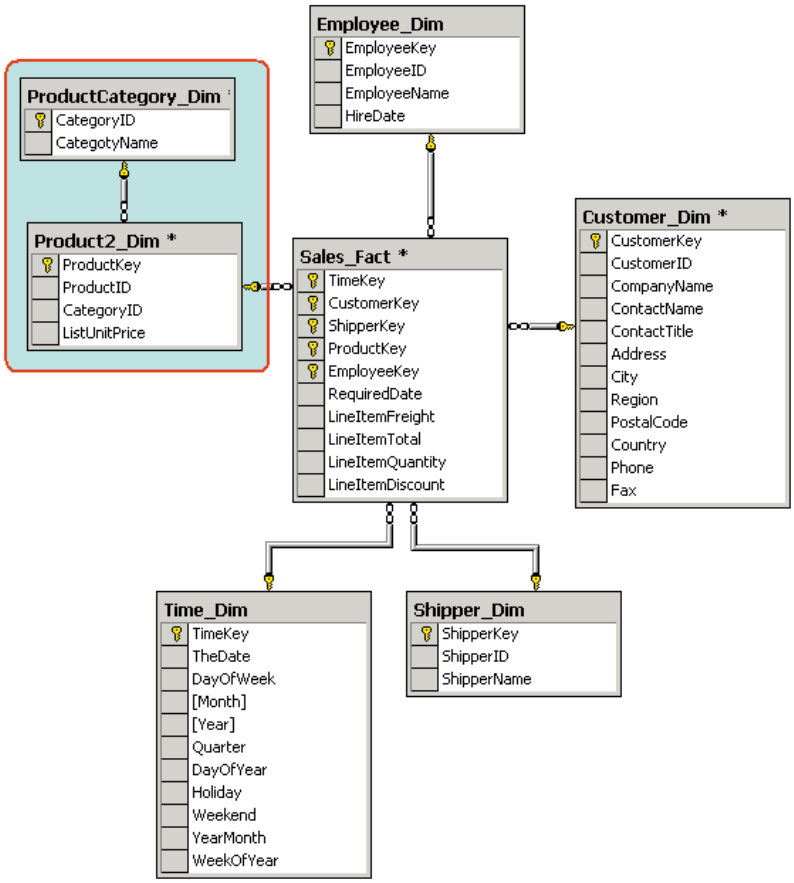

Рис. 12. Пример схемы «снежинка»[82]

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.

[82] Хранилища данных.

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[83].

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[84].

---

[83] Архитектура хранилищ данных: традиционная и облачная.

[84] Там же.

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*) [85].

Имеются различия и в *системах хранения данных*.

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[86].

---

[85] Архитектура хранилищ данных: традиционная и облачная.

[86] Telecom & IT.

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[87].

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 %[88].

---

[87] Telecom & IT.

[88] Объектные системы хранения – что, зачем и для чего [Электронный ресурс]. URL: https://itelon.ru/blog/obektnye-sistemy-khraneniya-chto-zachem-i-dlya-chego/ (дата обращения: 15.06.2020).

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[89].

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. Отличие традиционных хранилищ от объектных наглядно представлено на рис. 13.

[89] Объектные системы хранения – что, зачем и для чего.

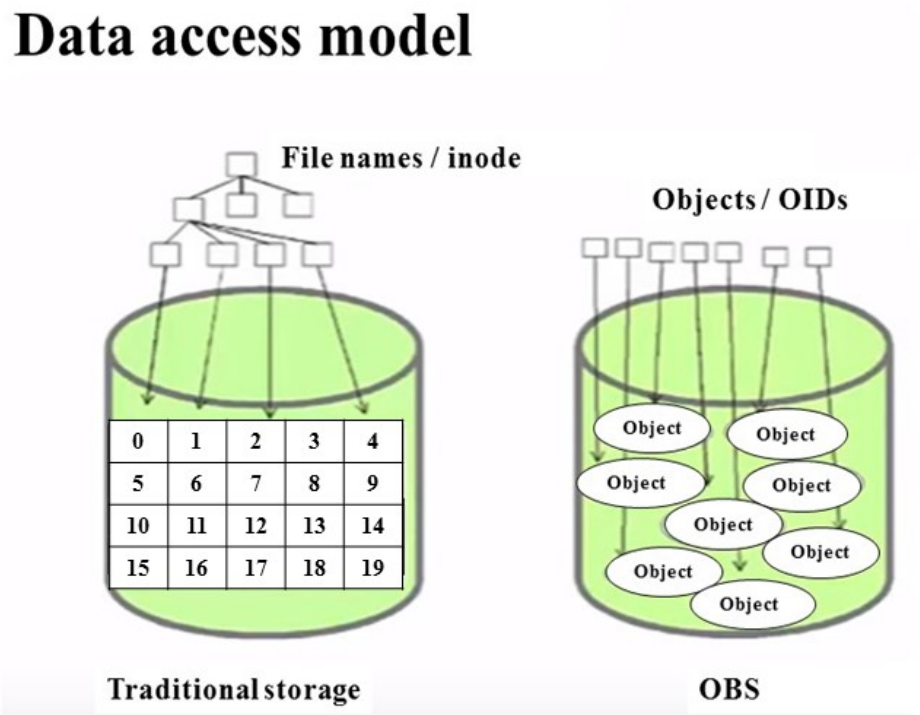

Рис.13. Традиционное и объектное хранилища

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[90].

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[91].

Объектное хранилище часто выбирается для данных *WORM*, которые пишутся один раз, но читаются много раз (*Write Once Read Many*). Этот тип хранилища подходит не для всех объемов данных и сценариев использования.

Объектные системы хранения целесообразно использовать в следующих случаях:

– *при долгосрочном хранении статичных данных*, например различной нормативной документации (*WORM*);

---

[90] Объектные системы хранения – что, зачем и для чего.

[91] Telecom & IT.

– *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.

Итак, 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[92].

Какие бы системы хранения ни применялись, цель у них одна – анализ имеющихся данных.

[92] Объектные системы хранения – что, зачем и для чего.

Глава 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 систем. 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[93].

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*), которая была сформулирована Б. Инмоном и Р. Кимбаллом.

DWH – предметно-ориентированные БД для консолидированной подготовки отчетов, интегрированного бизнес-анализа и оптимального принятия управленческих решений на основе полной информационной

[93] Хранилище данных [Электронный ресурс]. URL: https://www.intuit.ru/studies/courses/599/455/lecture/10155 (дата обращения: 07.07.2020).

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*[94]:

– **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.

---

[94] Где хранить корпоративные данные: краткий ликбез по Data Warehouse [Электронный ресурс]. – URL: https://www.bigdataschool.ru/bigdata/lsa-data-warehouse-architecture.html (дата обращения: 18.06.2020).

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*)[95].

Слоеная архитектура *DWH* представлена на рис. 14.

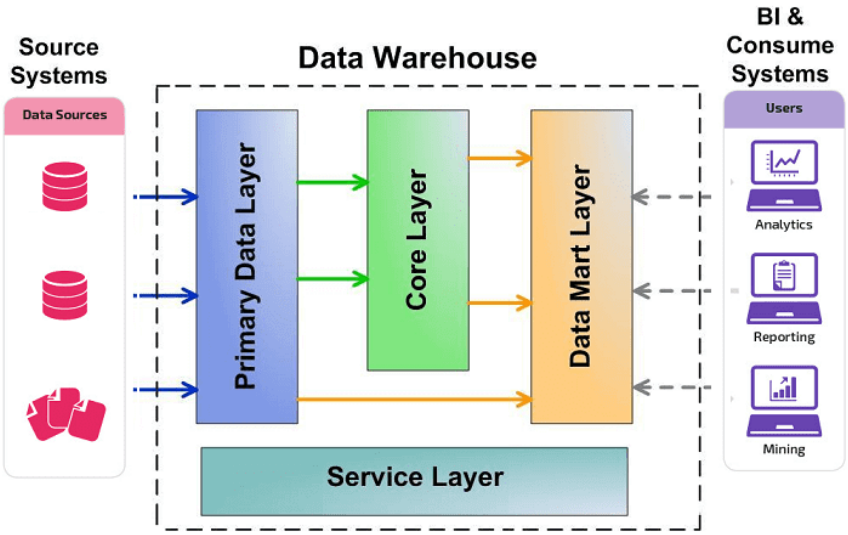

Рис. 14. Многоуровневая структура DWH

Значимость концепции *DWH* можно проиллюстрировать на примере. Допустим, в онлайн-магазине компании упала выручка. Менеджеры просят бизнес-аналитика разобраться в этом. Тот обращается к *DWH*, извлекает оттуда данные по продажам, выручке, количеству пользователей, расходам и собирает отчет, который точно и подробно сообщает причины падения финансовых показателей. Менеджеры на основе этой информации принимают решения по изменению ассорти-

[95] Где хранить корпоративные данные: краткий ликбез по Data Warehouse.

мента товаров и маркетинговой политики. Если бы такого аналитического отчета не было, управленцам пришлось бы искать проблему наугад.

Но аналитики ведь могут обращаться к БД разных систем и просто извлекать оттуда то, что им надо, не прибегая к *DWH*. Это возможно, но нецелесообразно по следующим *причинам*.

1. Если компания большая, то на получение данных из разных источников нужно собирать разрешения и доступы. У каждого подразделения, как правило, свои БД со своими паролями, которые надо будет запрашивать отдельно. В *DWH* все нужное уже находится под рукой в готовом виде. Там можно просто взять необходимую статистику.

2. Данные в *DWH* не теряются и хранятся в виде, удобном для принятия решений: есть исторические записи, есть агрегированные значения. В операционной БД такой информации может и не быть. Вряд ли на складском сервере будет храниться архив запасов за десять лет: БД склада в таком случае будет слишком тяжелой. А вот хранить агрегированные запасы на складе в *DWH* вполне реально.

3. *DWH* оптимизируется для работы аналитиков, для запросов очень больших объемов информации, что невозможно сделать с БД сервера, он не справится с этой задачей и создаст проблемы другим системам[96].

Итак, чтобы воспользоваться преимуществами больших данных, важно учитывать их инфраструктуру для их хранения и обработки.

Современная инфраструктура системы хранения больших данных должна, во-первых, обеспечить длительное хранение больших массивов данных с возможностью оперативного доступа к ним и, во-вторых, поддерживать функции их консолидации, обработки и структурирования[97].

Кроме того, современная архитектура хранения должна эффективно справляться с данными, которые поступают в реальном времени и на которые надо реагировать мгновенно, и с данными, которые накапливаются годами и имеют богатую историю.

---

[96] Что такое DWH и почему без них данные компании почти бесполезны [Электронный ресурс]. URL: https://mcs.mail.ru/blog/chto-takoe-dwh-i-pochemu-bez-nih-dannye-kompanii-bespolezny (дата обращения: 21.06.2020).

[97] Где хранить корпоративные данные: краткий ликбез по Data Warehouse.

Архитектура для обработки и хранения больших данных постоянно усложняется и может насчитывать десятки и сотни компонентов. Сейчас получил признание тип хранилища, который специалисты называют «озеро данных», куда можно записывать данные из множества источников в исходном виде и эффективно их обрабатывать. Наличие у него такого уровня обработки, как уровень памяти (*Speed Layer*), добавляет новые возможности обработки данных в потоке реального времени[98].

2. Хранилище данных и озеро данных

В 2010-х гг. с наступлением эпохи *Big Data* фокус внимания от традиционных *DWH* сместился к озерам данных (*Data Lake*) [99].

Озеро данных – это не синоним хранилищ данных или витрин данных. Конечно, это хранилище данных, но оно принципиально отличается от остальных. По мнению программиста Д. Лошина (*D. Loshin*), идея озера данных заключается в том, чтобы хранить необработанные данные в их оригинальном формате до тех пор, пока они не понадобятся[100].

Данные в *Data Lake* как рыба в озере, которая попала туда из реки, но точно не известно, какая именно это рыба и где она находится. А чтобы «приготовить» рыбу, т. е. обработать данные, ее нужно еще поймать.

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:

---

[98] BIG DATA 2017: Где хранить Большие Данные [Электронный ресурс]. URL: https://www.computerworld.ru/articles/BIG-DATA-2017-Gde-hranit-Bolshie-Dannye (дата обращения: 15.06.2020).

[99] Где хранить корпоративные данные: краткий ликбез по Data Warehouse.

[100] Саймон Ф. Озеро данных и хранилище данных – в чем разница? [Электронный ресурс]. URL: https://www.sas.com/ru_ru/insights/articles/data-management/data-lake-and-data-warehouse-know-the-difference.html (дата обращения: 19.06.2020).

извлекать по определенному шаблону из классических БД или анализировать и обрабатывать прямо внутри озера[101].

Озеро данных имеет свои особенности, поэтому не совсем корректно относить его к новому поколению корпоративных хранилищ данных по следующим *причинам*:

– в традиционных хранилищах люди и машины сначала собирали данные, «очищали», структурировали их и затем использовали. 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. Если компании необходимы только структурированные данные, корпоративное хранилище подходит ей лучше, чем озеро данных;

– *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 (видеозаписи 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[102].

Чтобы работать с озером данных, в компании должны быть технические специалисты: *Data Scientist*, *Data Engineer* (проектировщик надежной инфраструктуры для данных), бизнес-аналитик. Такие специалисты имеют доступ к данным в озере и могут их обрабатывать с

[101] Шпрингер Е. Что такое озера данных и почему в них дешевле хранить big data [Электронный ресурс]. URL: https://mcs.mail.ru/blog/chto-takoe-ozera-dannyh-i-zachem-tam-hranyat-big-data (дата обращения: 19.06.2020).

[102] Где хранить корпоративные данные: краткий ликбез по Data Warehouse.

помощью различных аналитических систем и подходов. В озере данные можно обрабатывать без извлечения – достаточно оборудовать системы для анализа прямо внутри него[103];

– *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 (см. в главе 4)). 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, например *HDFS* или *Amazon S3*;

– 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[104];

– в БД все данные полезны и актуальны для компании прямо сейчас. Данные, которые пока кажутся бесполезными, отсеиваются и теряются навсегда. БД идеальны для хранения важной информации, которая всегда должна быть под рукой, либо для основной аналитики. 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[105];

– большинство приложений озера данных н*е поддерживают частичную*, *или инкрементную*, *загрузку*. Организация не может загружать или перезагружать части своих данных в озеро данных (т. е. все или ничего);

---

[103] Шпрингер Е. Что такое озера данных и почему в них дешевле хранить big data.

[104] Там же.

[105] Там же.

– 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. [106];

– 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 hoc*[107]. Данные используются по плану или схеме, когда пользователи извлекают их, а не когда загружают. Озера данных сохраняют данные в неизмененном (естественном) состоянии; требования не определяются 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[108];

– 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[109].

Указанные выше особенности отличают озеро данных от корпоративных хранилищ. Вместе с тем архитектурный подход *LSA* может использоваться и при построении озера данных.

---

[106] Где хранить корпоративные данные: краткий ликбез по Data Warehouse.

[107] Латинская фраза, означающая «специально для этого», «по особому случаю»; обозначает способ решения специфической проблемы или задачи, который невозможно приспособить для решения других задач и который не вписывается в общую стратегию решений, составляет некоторое исключение.

[108] Шпрингер Е. Что такое озера данных и почему в них дешевле хранить big data.

[109] Саймон Ф. Указ. соч.

Например:

– на уровне *RAW* хранятся сырые данные различных форматов (*tsv*, *csv*, *xml*, *syslog*, *json* и т. д.);

– на операционном уровне (*ODD*, *Operational Data Definition*) сырые данные преобразуются в приближенный к реляционному формат;

– на уровне детализации (*DDS*, *Detail Data Store*) собирается консолидированная модель детальных данных;

– уровень *MART* (витрина данных, срез *Data Warehouse*) выполняет роль прикладных витрин данных для бизнес-пользователей и моделей машинного обучения[110].

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.

Например, на производстве, использующем Интернет вещей (*IoT*), на сложном оборудовании, которое часто ломается, устанавливаются датчики контроля, данные с которых можно собирать в озеро данных без фильтрации. Когда данных накопится достаточно, можно их проанализировать и понять, из-за чего случаются поломки и как их предотвратить.

В ритейле и *e-commerce* можно хранить в озере разрозненную информацию о клиентах: время, проведенное на сайте, активность в группе в соцсетях, тон голоса при звонках менеджеру и регулярность покупок. Потом эту информацию можно использовать для глобальной и масштабной аналитики и прогнозирования поведения клиентов.

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[111].

---

[110] Где хранить корпоративные данные: краткий ликбез по Data Warehouse.

[111] Шпрингер Е. Что такое озера данных и почему в них дешевле хранить big data.

Исследования показали, что компании, внедрившие озеро данных, на 9 % опережают своих конкурентов по выручке. Так что можно сказать, что озера данных нужны компаниям, которые хотят зарабатывать больше, используя для этого анализ собственных данных.

Компании-лидеры используют передовые подходы к аналитике данных, хранящихся в озере, например машинное обучение. С помощью такого подхода компания может получить полезные инсайты различной природы, вывести закономерности, прогнозировать сценарии будущего.

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

Проблемы в использовании озера данных представлены на рис. 15.

Рис. 15. Проблемы в использовании озера данных

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[112].

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[113]. Следовательно, организация эффективного управления озером данных – важная задача для компании.

[112] Шпрингер Е. Что такое озера данных и почему в них дешевле хранить big data.

[113] Саймон Ф. Указ. соч.

Глава 6. АНАЛИТИКА БОЛЬШИХ ДАННЫХ И ЕЕ ИНСТРУМЕНТАРИЙ

1. От традиционного анализа к операционному

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[114].

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[115].

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[116].

---

[114] Бочкарева Е. Как зарождалась эра Big Data [Электронный ресурс]. URL: https://rb.ru/story/era-big-data/ (дата обращения: 18.06.2020).

[115] Фрэнкс Б. Революция в аналитике. Как в эпоху Big Data улучшить ваш бизнес с помощью операционной аналитики [Электронный ресурс]. URL: https://lifeinbooks.net/read-online/revolyutsiya-v-analitike-kak-v-epohu-big-data-uluchshit-vash-biznes-s-pomoshhyu-operatsionnoy-analitiki-bill-frenks/ (дата обращения: 05.07.2020).

[116] Бочкарева Е. Указ. соч.

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.

Такая аналитика транзакционного уровня – это новый шаг по сравнению с традиционным пониманием бизнес-анализа как базы для принятия решений на стратегическом уровне. Например, правильно настроенный рекомендательный алгоритм на сайте интернет-магазина гораздо лучше любого человека-продавца умеет предлагать покупателю дополнительные сервисы и покупки.

В отличие от неторопливой пакетной аналитики операционная аналитика выполняется намного быстрее и непрерывно. При этом она интегрируется с существующими бизнес-процессами и системами.

Становление Аналитики 3.0 можно сравнить с историей развертывания промышленной революции, которая трансформировала ремесленничество в современные технологии производства, позволяющие производить качественные продукты в массовом масштабе. Что-то похожее происходит сейчас и в области аналитики. 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[117].

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.

По мере развития аналитики будет происходить смена существующих бизнес-моделей и конкурентной среды. Если раньше можно было довольствоваться данными недельной давности и аналитическими процессами, построенными на пакетной обработке, то сейчас этого уже недостаточно. Еще через пять – десять лет не останется практически ни одной бизнес-модели, которой не затронет данная тенденция.

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.

---

[117] Фрэнкс Б. Революция в аналитике.

Виды аналитики представлены на рис. 16.

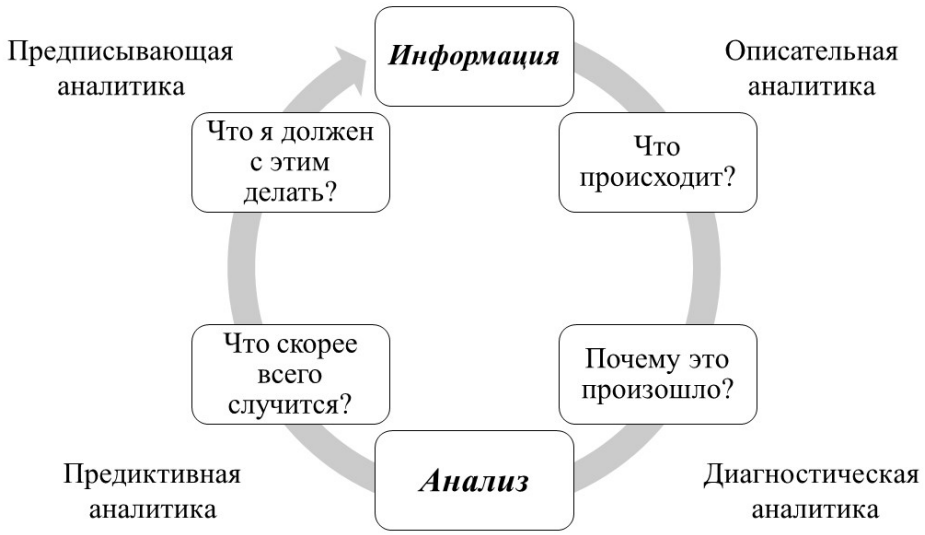

Рис. 16. Виды аналитики

Итак, 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. Сущность разных видов аналитики представлена в табл. 3[118].

[118] Фрэнкс Б. Революция в аналитике.

Таблица 3

Виды аналитики и их сущность

Вид аналитики	Особенность
Описательная аналитика	Анализирует и описывает события, произошедшие в прошлом
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 интегрированном 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.

Итак, 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 предписывает действия: она не просто рекомендует, какое наилучшее предложение следует сделать клиенту, когда он вернется, а действительно предписывает это сделать, отдав распоряжение соответствующей системе;

3) 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;

4) 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.

Чтобы организация смогла применять операционную аналитику, нужно иметь прочные аналитические основы, научиться успешно применять традиционную аналитику на основе пакетной обработки. Без этого операционная аналитика останется недостижимой мечтой.

Итак, **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.

Операционная аналитика представляет собой новую ступень эволюции аналитических технологий[119].

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. Сочетание искусственного интеллекта с Аналитикой 3.0 открывает вообще безграничные возможности[120].

Из всего вышесказанного следует, что анализ данных проводился в компаниях и до появления больших данных. Однако именно они подтолкнули дальнейшее развитие инструментария аналитики.

---

[119] Фрэнкс Б. Революция в аналитике.

[120] Бочкарева Е. Указ. соч.

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. А именно:

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.

Все это свидетельствует о больших перспективах технологий аналитики *Big Data* в отличие от традиционного анализа.

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.

Практически все современные средства анализа больших данных предоставляют средства интеграции с *Hadoop*. Их разработчиками выступают как стартапы, так и общеизвестные мировые компании.

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[121].

Наиболее популярными в мировом *IT*-сегменте продуктами для решения проблем *Big Data* считаются аналитические платформы *NoSQL* и *In-memory*[122].

Изначально основным способом работы с БД был *SQL* (БД – *structured query language*) – язык структурированных запросов, появившийся в 1974 г. (авторы – Д. Чемберлин и Р. Бойс), применяемый для создания, модификации 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.

---

[121] Шпрингер Е. Что такое озера данных и почему в них дешевле хранить big data.

[122] Революция Big Data : Как извлечь необходимую информацию из «Больших Данных»?

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 (*CQL, N1QL, AQL* и др.). 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[123].

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.

---

[123] SQL [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/SQL (дата обращения: 06.07.2020).

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[124].

Одна из самых популярных *open source* реляционных БД – ***MySQL***. Она подходит небольшим и средним проектам, поддерживает множество типов таблиц, имеет огромное количество плагинов и расширений, облегчающих работу с системой. Отличается простотой установки, может быть интегрирована с другими СУБД. Интеграция с *MySQL* есть в любой *CMS* (*Content Management System* – система, которая позволяет публиковать информацию на сайте и управлять его функционалом, движком сайта), фреймворке, языке программирования. Однако не все задачи в ней выполняются автоматически, нет встроенной поддержки *OLAP*.

MySQL доступна как облачный сервис. Эту базу выбирают на начальных этапах развития бизнеса, чтобы тестировать гипотезы с минимальными затратами, или для небольших проектов как транзакционную БД общего назначения.

Второй по популярности *open source SQL* СУБД является ***PostgreSQL***. У нее много встроенных функций и дополнений, в том числе для масштабирования в кластер и шардинга таблиц. Она подходит, если важна сохранность данных, предполагается их сложная структура; позволяет работать со структурированными данными, но поддерживает *JSON/BSON*, что дает некоторую гибкость в схеме данных; отличается стабильностью, ее практически невозможно вывести из строя или что-то сломать в таблицах, однако отличается сложностью

[124] Кушнир Е. Сравнение SQL и NoSQL: как выбрать систему хранения данных [Электронный ресурс]. URL: https://mcs.mail.ru/blog/sravnenie-sql-i-nosql-kak-vybrat-sistemu-hraneniya-dannyh (дата обращения: 18.06.2020).

конфигурации, скорость работы может падать во время проведения пакетных операций или при запросах на чтение.

PostgreSQL также можно развернуть в облаке. В отличие от *MySQL* она подходит для крупных и масштабных проектов. Ее выбирают, если недопустимы ошибки в данных или есть особые требования к БД, например поддержка геоданных. Различные расширения *PostgreSQL* позволяют реализовать многие специализированные запросы.

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 высокими производительностью и 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. **Документоориентированные 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*, ни какой-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 хранятся 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. Например, если нужно извлечь сумму среднего чека клиента из реляционной СУБД, придется искать это значение в каждой строке, а в колоночной СУБД можно сразу забрать информацию из нужной колонки.

В реляционных СУБД каждая запись должна иметь одинаковое число столбцов, а в колоночных – необязательно.

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.

Следует отметить, что реляционные БД не являются чем-то архаичным. Скорее всего, они будут использоваться по-прежнему активно, но все больше в симбиозе с *NoSQL*-базами. Некоторые специалисты говорят об эре *polyglot persistence*, когда для различных потребностей используются разные хранилища данных. Теперь нет монополизма реляционных БД, как безальтернативного источника данных. Все чаще архитекторы выбирают хранилище исходя из природы самих данных и того, как ими хотят манипулировать, какие объемы информации ожидаются[125].

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[126].

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

---

[125] Архитектура HDFS [Электронный ресурс]. URL: https://www.bigdata-school.ru/wiki/hdfs (дата обращения: 21.06.2020).

[126] BIG DATA 2017: Где хранить Большие Данные.

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/десериализации, 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[127].

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,

[127] Что такое In-Memory Data Grid [Электронный ресурс]. URL: https://habr.com/ru/post/160517/ (дата обращения: 05.07.2020).

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 2*PC*). 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» 2*PC* как минимум соответствуют *EC* системам по скорости записи (если не опережают их) и значительно превосходят их по скорости чтения. Таким образом, можно заметить, что индустрия сделала полный круг, двигаясь от когда-то медленных 2*PC* к *EC*, а теперь от *EC* к гораздо более быстрым «оптимизированным» 2*PC*.

Разные продукты могут предлагать разные 2*PC* оптимизации, но в целом задачами всех оптимизаций являются увеличение 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[128].

[128] Что такое In-Memory Data Grid.

Глава 7. АНАЛИТИКА БОЛЬШИХ ДАННЫХ: ТЕХНИКИ 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. Термин был введен математиком Григорием Пятецким-Шапиро в 1989 г.

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)[129].

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 значитель-

[129] Big Data vs Data Mining [Электронный ресурс]. URL: https://habr.com/ru/post/267827/ (дата обращения: 05.07.2020).

ных дополнительных расходов. Не случайно в список клиентов краудсорсинговых ресурсов входят компании *Yahoo!*, *IBM*, *eBay* и многие другие[130].

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[131].

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[132].

5. **Машинное обучение (*Machine Learning*).**

Предполагает эмпирический анализ информации и последующее построение алгоритмов самообучения систем.

---

[130] Почему краудсорсинг незаменим при обработке большого количества данных [Электронный ресурс]. URL: http://crowdsourcing.ru/article/why-crowdsourcing-is-the-perfect-solution-to-making-sense-of-big-data (дата обращения: 15.06.2020).

[131] Big Data – что такое системы больших данных?

[132] Прогнозная аналитика (Big Data) [Электронный ресурс]. URL: https://marketing-logic.ru/bigdata (дата обращения: 18.06.2020).

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.

По словам ученого в области аналитики Т. Дэвенпорта, в условиях быстро меняющихся, растущих объемов данных необходимо быстрое потоковое моделирование, чтобы не отставать, и это можно сделать с помощью машинного обучения. Люди могут создать одну или две хорошие модели в неделю, а *Machine Learning* – тысячи[133].

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[134].

---

[133] От Big Data к Machine Learning [Электронный ресурс]. URL: https://psm7.com/blogs/from-big-data-to-machine-learning.html (дата обращения: 15.06.2020).

[134] Келлехер Д., Тирни Б. Указ. соч.

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.

Компания понимает, что не так важно, где хранится информация. Гораздо важнее, что собой представляет информация и как ее лучше классифицировать, чтобы получить максимальную пользу в ежедневной работе[135].

---

[135] Что скрывается в тени больших данных? [Электронный ресурс]. URL: https://blog.fts-eu.com/ru/2016/01/26/ (дата обращения: 15.06.2020).

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 (отрасль ИИ появилась еще в 1956 г.).

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[136].

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.

---

[136] Big Data – что такое системы больших данных?

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 S*3, *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*)[137].

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.

[137] Храните данные в облаке [Электронный ресурс]. URL: https://habr.com/ru/company/bigdatahosting/blog/353168/ (дата обращения: 18.06.2020).

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[138].

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

[138] Кушнир Е. Анализ больших данных в облаке: как бизнесу стать data-ориентированным [Электронный ресурс]. URL: https://mcs.mail.ru/blog/analiz-bolshih-dannyh-v-oblake (дата обращения: 19.06.2020).

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.

Например, персональные данные можно хранить в компании в соответствии с законодательством, а в обезличенном виде обрабатывать в облаке, что также не противоречит закону.

Сравнительный анализ частного и публичного облака представлен в табл. 4.

Таблица 4

Характеристики частного и публичного облака

Характеристика	Частное облако	Публичное облако
Экономичность	Требуются затраты на оборудование, персонал, инфраструктуру как и с традиционной *IT*-инфраструктурой	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	Возможности масштабирования ограничены мощностью физического оборудования, скоростью закупки и ввода в эксплуатацию новых мощностей	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	Компания сама обслуживает и администрирует облако	Команда компании меньше занимается обслуживанием системы обработки данных, сосредоточивается на создании и тестировании идей, что повышает эффективность аналитики
Быстрый запуск проекта (*time-to-market*)	Может замедлить выпуск *IT*-продуктов на рынок, так как требуются огромные инфраструктурные мощности с высокими капитальными затратами на запуск	Позволяет запустить *IT*-инфраструктуру без больших первоначальных инвестиций, создать и настроить инфраструктуру для анализа данных за считанные часы; конкретный *PaaS*-сервис подключается за минуты

Окончание табл. 4

Характеристика	Частное облако	Публичное облако
Отказоустойчивость	Можно обеспечить средствами *Disaster Recovery*, но потребуются серьезные капитальные вложения, расходы на введение в эксплуатацию и поддержку этих средств	Провайдер обеспечивает бесперебойную работу, что сводит простои инфраструктуры к минимуму. Так, при *SLA* 99,95 % *IT*-инфраструктура простаивает всего около пяти часов в год
Обеспечение требований законодательства	Сама компания следит за выполнением требований законодательства и регуляторов	Ответственность за соблюдение законодательства, требований и стандартов, сертификацию ЦОД лежит на провайдере

По итогам отчета «Обзор тенденций и проблем больших данных 2018 года» 73 % компаний используют для обработки *Big Data* облачные сервисы (в 2017 г. таких компаний было 58 %).

Это говорит о том, что компании все больше осознают, что не всегда достаточно только собирать данные и делать какие-то отчеты. Результатом аналитики должны быть выводы, представляющие ценность для бизнеса, которые можно учитывать в процессе дальнейшей работы. Именно в этом и заключается суть *data-driven* (дата-ориентированного) подхода к принятию управленческих решений.

ТЕСТЫ ДЛЯ САМОКОНТРОЛЯ

1. Как не используют выборки из генеральной совокупности аналитики больших данных?

а) как метод формирования комплексного суждения о генеральной совокупности случайной величины;

б) как метод тестирования полученных моделей;

в) как метод верификации исходных данных.

2. Укажите лишний этап построения статистической модели:

а) сбор и верификация исходных данных;

б) выбор факторов;

в) построение модели;

г) получение оценок;

д) согласование полученных результатов с заинтересованными лицами;

е) проверка статистической значимости модели.

3. Глубокое обучение включает в себя:

а) регрессионные модели;

б) совокупность различных нейросетевых моделей;

в) методы классификации;

г) градиентный бустинг;

д) обучение с подкреплением.

4. Какой метод верификации исходных данных не применяется для верификации данных о стоимости активов?

а) семантические анализаторы;

б) матрицы граничных значений;

в) конверторы отраслевых классификаторов;

г) наборы решающих правил;

д) проверка данных с использованием колл-центра;

е) тестовые и валидационные выборки.

5. Какие нейронные сети лучше подходят для задач поиска аналога исследуемого объекта?

а) сети Кохонена;

б) сети встречного распространения;

в) *RBF*-сети на радиальных базисных функциях;

г) любые *MLP*-нейросети;

д) все вышеперечисленное.

6. Какая проблема решается путем логарифмического шкалирования исходных данных?

а) мультиколлинеарности;

б) робастности;

в) гетероскедастичности;

г) гомоскедастичности.

7. Какие требования к факторам предъявляют классические статистические модели?

а) значимость;

б) независимость;

в) внятная экономическая интерпретация;

г) все вышеперечисленное.

8. Какая технология машинного обучения реагирует на возникновение новых, не описанных ранее ситуаций, получая данные из внешней среды?

а) обучение с подкреплением;

б) обучение с противником;

в) вероятностное прогнозирование;

г) распознавание образов.

9. Как не используют выборки из генеральной совокупности аналитики больших данных?

а) как метод формирования комплексного суждения о генеральной совокупности случайной величины;

б) как метод тестирования полученных моделей;

в) как метод верификации исходных данных.

10. Метод главных компонент применяется для решения проблемы:

а) робастности;

б) мультиколлинеарности;

в) гомоскедастичности;

г) гетероскедастичности.

11. В искусственной нейронной сети типа *RBF* (сеть с радиальной базисной функцией) применяются следующие виды активационных функций:

а) радиальная базисная;

б) линейная;

в) пороговая;

г) сигмоидальная.

12. Понять, сколько нейронов не стали победителями ни для одного образца, можно с помощью следующего представления карты Кохонена:

а) матрица расстояний;

б) матрица плотности попадания;

в) проекция Саммона;

г) матрица кластеров 34.

13. В искусственной нейронной сети типа *MLP* (многослойный персептрон) присутствуют следующие связи:

а) каждый нейрон связан с каждым нейроном следующего слоя;

б) каждый нейрон связан с каждым нейроном своего слоя;

в) каждый нейрон связан с каждым нейроном входного слоя;

г) каждый нейрон связан с каждым нейроном выходного слоя.

14. В какой последовательности технология *MapReduce* использует в рабочем процессе задачи-распределители и задачи-редукторы?

а) последовательно, сначала одни, а затем другие;

б) параллельно или обе одновременно;

в) поочередно, одну за другой.

15. Какие функции в *MapReduce* запускает главный Мастер-контроллер (найдите неверный ответ)?

а) создание распределителей и редукторов;

б) назначение задачам рабочих процессов;

в) обработку отказа узла редуктора.

16. В чем состоит стратегия кластеризации?

а) в объединении близких точек многомерного пространства в один объект (кластер) с усредненными характеристиками;

б) разделении множества на части с помощью плоскостей;

с) разделении множества на внутренние, или «свои», точки и внешние, или «чужие», точки.

17. Как ускорить алгоритм иерархической кластеризации для евклидовой метрики?

а) следует матрицу расстояний между элементами рассчитать только один раз;

б) следует элементы, объединенные в кластер, вычеркивать из матрицы расстояний;

в) следует пересчитывать среднее значение для кластера без привлечения исходных координат.

18. В чем состоит алгоритм k-средних?

а) из исходного множества случайным образом выбираются k-центров кластеров;

б) рассчитывается диаметр множества, делится на k, и кругами, равными полученному значению, покрывается все множество; внутри каждого круга находится центр кластера;

в) множество делится на части с помощью k -плоскостей.

19. Как реализуется алгоритм кластеризации потока?

а) точки потока разбиваются на одинаковые интервалы, в которых хранится информация о кластере;

б) точки потока разбиваются на интервалы, размеры которых являются степенями двойки;

в) точки потока разбиваются на интервалы, размеры которых уменьшаются в два раза.

20. Как проводится кластеризация с помощью алгоритма *CURE*?

а) предварительно проводится кластеризация на части данных, проводится сдвиг данных относительно центра кластера, объединяется, если имеется близкая пара;

б) методом рекурсивной кластеризации;

в) методом кластеризации выборки, не принадлежащей нормальному распределению.

21. Перечислите четыре основные характеристики *Big Data*:

а) *Virtualization*, *Volume*, *Variability*, *Vehicle*;

б) *Variety*, *Velocity*, *Volume*, *Value*;

в) *Verification*, *Volume*, *Velocity*, *Visualization*;

г) *Video*, *Value*, *Variety*, *Volume*.

ЗАКЛЮЧЕНИЕ

Учебное пособие имеет своей целью формирование у студентов профессиональной компетенции в области разработки и использования систем обработки и анализа больших массивов данных, что соотносится с целью образовательной программы в части технологий разработки специализированных программных систем, отвечающих за обработку больших данных.

Пособие призвано помочь студенту в выполнении следующих профессиональных задач: анализ данных; предварительная обработка данных; визуализация данных; разработка, реализация и применение методов интеллектуального анализа данных к большим массивам данных; представление результатов работы.

ГЛОССАРИЙ

База данных (БД) – именованная совокупность данных, отражающая состояние объектов и их отношения в рассматриваемой предметной области.

Большие данные – наличие данных больше, чем 100 Гб; данные, которые невозможно обрабатывать в *Excel*, т. е. традиционным способом, и которые невозможно обработать на одном компьютере.

Дамп памяти (*memory dump*) – содержимое рабочей памяти одного процесса, ядра или всей операционной системы.

Данные – это представление фактов и идей в формализованном виде, пригодном для передачи и обработки в некотором информационном процессе.

Инкрементный – термин, означающий увеличение чего-то на фиксированную или переменную (измененную) величину. Например, инкрементная загрузка – это загрузка только тех данных, которые добавились (изменились) со времени полной загрузки.

Информация – это сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состоянии, которые уменьшают имеющуюся о них степень неопределенности и неполноты знаний.

Искусственный интеллект (ИИ; *artificial intelligence*, *AI*) – свойство интеллектуальных систем выполнять творческие функции, которые традиционно считаются прерогативой человека; наука и технология создания интеллектуальных машин, особенно интеллектуальных компьютерных программ.

Кластер – это группа серверов, именуемых нодами, которые работают вместе, выполняют общие задачи, и клиенты видят их как одну систему.

Коллаборация (сотрудничество) – процесс совместной деятельности в какой-либо сфере двух и более людей или организаций для достижения общих целей, при котором происходит обмен знаниями, обучение и достижение согласия (консенсуса).

Куб данных – многомерный массив данных, как правило, разреженный и долговременно хранимый, используемый в *OLAP*.

Облачное хранилище – это неограниченные вычислительные мощности с доступом через Интернет.

Озеро данных – это хранилище, в котором находятся необработанные данные в их оригинальном формате до тех пор, пока они не понадобятся.

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.

Открытый исходный код (*open source*) – используется для программного обеспечения, к которому можно свободно получить исходный код.

Релевантность – обозначение субъективной степени соответствия чего-либо в моменте времени.

Реляционная модель – совокупность данных, состоящая из набора двумерных таблиц. В теории множеств таблице соответствует термин отношение (*relation*), физическим представлением которого является таблица, отсюда и название модели – реляционная. Она отличается более высоким уровнем абстракции данных, является удобной и наиболее привычной формой представления данных, является фактическим стандартом, на который ориентируются практически все современные коммерческие СУБД. На реляционной модели данных строятся реляционные базы данных.

Транзакционные данные – это данные, каждая запись которых относится к фиксированному моменту времени и содержит сведения, фиксированные на данный момент времени, не изменяющиеся в будущем. Нетранзакционными являются все остальные данные.

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

Big Data **–** это 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.

***Data-driven*-организации** – это такие компании, в которых все внутренние процессы и большинство решений вокруг них строятся исключительно на основании данных.

DWH – предметно-ориентированные базы данных для консолидированной подготовки отчетов, интегрированного бизнес-анализа и оптимального принятия управленческих решений на основе полной информационной картины.

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.

IaaS (*Infrastructure as a Service*, инфраструктура как услуга) – вычислительная инфраструктура (серверы, хранилища данных, сети, операционные системы), которая предоставляется клиентам для разворачивания и запуска собственных программных решений.

KDD **–** процесс поиска полезных знаний в «сырых» данных, который включает: подготовку данных, выбор информативных признаков, очистку данных, применение методов *Data Mining* (*DM*), постобработку данных и интерпретацию полученных результатов. Именно *DM* позволяет обнаруживать знания: правила, описывающие связи между свойствами данных (деревья решений), часто встречающиеся шаблоны (ассоциативные правила), результаты классификации (нейронные сети) и кластеризации данных и т. д.

ODS – это открытый формат для электронных таблиц, выполненных в соответствии со стандартом *OpenDocument Format* (*ODF*).

OLAP (*online analytical processing*, интерактивная аналитическая обработка) – технология обработки данных (основоположник термина – Эдгар Кодд), заключающаяся в подготовке суммарной (агрегированной) информации на основе больших массивов данных, структурированных по многомерному принципу; является компонентом программных решений класса *Business Intelligence*.

PaaS (*Platform as a Service*, платформа как услуга) – набор инструментов и сервисов, облегчающих разработку и развертывание облачных приложений.

SaaS (*Software as a Service*, программное обеспечение как сервис) – приложения, работающие в облаке, доступ к которым конечные пользователи получают через веб.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. АльфаСтрахование внедрила сервис распознавания документов Smart Engines [Электронный ресурс]. – URL: https://www.alfastrah.ru/news/9219806/ (дата обращения: 17.06.2020).
2. Анализ больших данных: Spark и Hadoop [Электронный ресурс]. – URL: https://coincase.ru/blog/47715/ (дата обращения: 20.06.2020).
3. Анализ Big data в медицине поможет страховщикам понимать потребности клиентов в здравоохранении и страховании [Электронный ресурс]. – URL: https://forinsurer.com/news/17/1/18/34782 (дата обращения: 05.07.2020)
4. Анатомия больших данных в транспорте [Электронный ресурс]. – URL: https://iot.ru/monitoring/anatomiya-bolshikh-dannykh-v-transporte (дата обращения: 05.07.2020).
5. Архитектура HDFS [Электронный ресурс]. – URL: https://www.bigdataschool.ru/wiki/hdfs (дата обращения: 21.06.2020).
6. Архитектура хранилищ данных: традиционная и облачная [Электронный ресурс]. – URL: https://habr.com/ru/post/441538/ (дата обращения: 15.06.2020).
7. Базы данных [Электронный ресурс]. – URL: https://siblec.ru/informatika-i-vychislitelnaya-tekhnika/bazy-dannykh (дата обращения: 18.06.2020).
8. Библиотека видеоуроков школьной программы [Электронный ресурс]. – URL: https://interneturok.ru/ (дата обращения: 21.06.2020).
9. Благирев, А. Big data простым языком / А. Благирев, Н. Хапаева. – М. : АСТ, 2019. – 256 с. – ISBN 978-5-17-11129-7.
10. Большие данные (Big Data) [Электронный ресурс]. – URL: https://www.tadviser.ru/index.php/Статья:Большие_данные_(Big_Data) (дата обращения: 21.06.2020).
11. Большие данные [Электронный ресурс]. – URL: https://yandex.ru/profi/courses2019/big_data (дата обращения: 21.06.2020).
12. Большие данные в образовании [Электронный ресурс]. – URL: http://www.unkniga.ru/vishee/9614-bolshie-dannye-vobrazovanii.html (дата обращения: 19.06.2020).

13. Большие данные в управлении персоналом: что это такое и зачем это нужно [Электронный ресурс]. – URL: https://hr-portal.ru/blog/bolshie-dannye-v-upravlenii-personalom-chto-eto-takoe-i-zachem-eto-nuzhno (дата обращения: 07.07.2020).

14. Бородаенко, В. Универсальная платформа обработки больших данных [Электронный ресурс] / В. Бородаенко, А. Ермаков. – URL: https://www.osp.ru/os/2017/03/13052699 (дата обращения: 05.07.2020).

15. Бочкарева, Е. Как зарождалась эра Big Data [Электронный ресурс] / Е. Бочкарева. – URL: https://rb.ru/story/era-big-data/ (дата обращения: 18.06.2020).

16. В Vecam подсчитали потери предприятий из-за простоев и ограниченной доступности данных [Электронный ресурс]. – URL: https://www.osp.ru/news/2016/0306/13031909/ (дата обращения: 15.06.2020).

17. В приложение «Яндекс.Транспорт» добавили карту велопарковок Москвы [Электронный ресурс]. – URL: https://www.the-village.r/village/city/transport/265074-transport (дата обращения: 18.06.2020).

18. Введение в кластеры [Электронный ресурс]. – URL: https://onix.kiev.ua/news.aspx?id=172 (дата обращения: 18.06.2020).

19. Вичугова, А. Как и зачем HR использует Big Data: технологии больших данных в управлении человеческими ресурсами [Электронный ресурс] / А. Вичугова. – URL: https://www.bigdataschool.ru/big-data/big-data-hr (дата обращения: 15.06.2020).

20. Где откроются новые автобусные маршруты [Электронный ресурс]. – URL: https://www.mos.ru/news/item/39813073/ (дата обращения: 13.07.2020).

21. Где хранить корпоративные данные: краткий ликбез по Data Warehouse [Электронный ресурс]. – URL: https://www.bigdataschool.ru/bigdata/lsa-data-warehouse-architecture.html (дата обращения: 18.06.2020).

22. Готовы ли чиновники и врачи к внедрению Big Data в российское здравоохранение [Электронный ресурс]. – URL: https://vademec.ru/article/terabaytovoe_voysko/ (дата обращения: 03.08.2020).

23. Гула, Е. Большие данные Big Data для HR. Как увидеть личность за цифрой? [Электронный ресурс] / Е. Гула, И. Канардов. – URL: http:// hr-media.ru/bolshie-dannye-bigdata-dlya-hr-kak-uvidet-lichnost-za-tsifroj/ (дата обращения: 05.07.2020).

24. Данные [Электронный ресурс]. – URL: https://dic.academic.ru/dic.nsf/ruwiki/71919 (дата обращения: 18.06.2020).

25. Забиров, Р. Р. Управление персоналом в эпоху Больших данных / Р. Р. Забиров // Молодой ученый. – 2019. – № 31(269). – С. 52 – 53.

26. Зачем страховщикам телематика, фитнес-трекеры и умные зубные щетки [Электронный ресурс]. – URL: https://allinsurance.kz/articles/analytical/5813-kak-umnye-tekhnologii-iot-i-big-data-pomogut-strakhovym-kompaniyam-i-kak-oni-povliyayut-na-tarify-v-budushchem (дата обращения: 15.06.2020).

27. Инвестирование в рынок цифрового здравоохранения активно развивается [Электронный ресурс]. – URL: https://webiomed.ai/blog/investirovanie-v-rynok-tsifrovogo-zdravookhraneniia-aktivno-razvivaetsia/ (дата обращения: 03.02.2020).

28. Информация [Электронный ресурс]. – URL: https://investments.academic.ru/1012/ (дата обращения: 20.06.2020).

29. Информация. Свойства информации [Электронный ресурс]. – URL: https://www.sites.google.com/site/3kursmimi/1-informacia-svojstva-informacii (дата обращения: 15.06.2020).

30. История больших данных (Big Data) – часть 1 [Электронный ресурс]. – URL: https://www.computerra.ru/234239/istoriya-bolshih-dannyh-big-data-chast-1/ (дата обращения: 05.07.2020).

31. История больших данных (Big Data) – часть 2 [Электронный ресурс]. – URL: https://www.computerra.ru/234346/istoriya-bolshih-dannyh-big-data-chast-2/ (дата обращения: 03.07.2020).

32. Как можно применять «большие данные» в страховании: проекты университета ИТМО [Электронный ресурс]. – URL: https://habr.com/ru/company/spbifmo/blog/329762/ (дата обращения: 18.06.2020).

33. Как умные технологии IoT и Big data помогут страховым компаниям и как они повлияют на тарифы в будущем? [Электронный ресурс]. – URL: https://forinsurer.com/news/17/07/14/35332 (дата обращения: 13.06.2020).

34. Келлехер, Д. Наука о данных / Д. Келлехер, Б. Тирни. – М. : Альпина Диджитал, 2020. – 222 с. – ISBN 978-5-9614-3170-4.

35. Кесаев, У. С. Перспективы применения Big Data в управлении персоналом [Электронный ресурс] / У. С. Кесаев, В. В. Алехно. – URL: http://nauka-rastudent.ru/37/3942/ (дата обращения: 12.06.2020).

36. Кушнир, Е. Анализ больших данных в облаке: как бизнесу стать дата-ориентированным [Электронный ресурс] / Е. Кушнир. – URL: https://mcs.mail.ru/blog/analiz-bolshih-dannyh-v-oblake (дата обращения: 19.06.2020).

37. Кушнир, Е. Сравнение SQL и NoSQL: как выбрать систему хранения данных [Электронный ресурс] / Е. Кушнир. – URL: https://mcs.mail.ru/blog/sravnenie-sql-i-nosql-kak-vybrat-sistemu-hraneniya-dannyh (дата обращения: 18.06.2020).

38. Лаборатория Умного Вождения за год увеличила число клиентов в 10 раз [Электронный ресурс]. – URL: http://www.cnews.ru/news/line/2019-02-13_laboratoriya_umnogo_vozhdeniya_za_god_uvelichila (дата обращения: 13.06.2020).

39. Минкомсвязи предложило регулировать big data [Электронный ресурс]. – URL: http://www.tadviser.ru/index.php/ (дата обращения: 13.06.2020).

40. Навигационный рынок России: новые технологии, новые услуги, новые бизнес-модели [Электронный ресурс]. – URL: http://www.glonass-forum.ru/printer_news-item-255.html (дата обращения: 13.06.2020).

41. Назаренко, Ю. Л. Обзор технологии «большие данные» (Big Data) и программно-аппаратных средств, применяемых для их анализа и обработки / Ю. Л. Назаренко // European science. – 2017. – № 9 (31).

42. Наличие IoT-устройств сделает страхование жилья дешевле [Электронный ресурс]. – URL: https://hightech.fm/2017/01/20/iot-insurance (дата обращения: 08.07.2020).

43. Объектные системы хранения – что, зачем и для чего [Электронный ресурс]. – URL: https://itelon.ru/blog/obektnye-sistemy-khraneniya-chto-zachem-i-dlya-chego/ (дата обращения: 15.06.2020).

44. Особенности построения информационных хранилищ [Электронный ресурс]. – URL: https://www.osp.ru/os/2003/04/182942 (дата обращения: 05.07.2020).

45. От Big Data к Machine Learning [Электронный ресурс]. – URL: https://psm7.com/blogs/from-big-data-to-machine-learning.html (дата обращения: 15.06.2020).

46. Перспективы использования больших данных в современном образовании [Электронный ресурс]. – URL: https://cyberleninka.ru/article/n/perspektivy-ispolzovaniya-bolshih-dannyh-vsovremennom-obrazovanii (дата обращения: 12.06.2020).

47. Попазова, О. А. Управление персоналом на основе анализа больших данных: риски и возможности / О. А. Попазова, Н. Н. Шихова // Известия Санкт-Петербургского государственного экономического университета. – № 3 (117). – 2019. – С. 110 – 115.

48. Почему краудсорсинг незаменим при обработке большого количества данных [Электронный ресурс]. – URL: http://crowdsourcing.ru/article/why-crowdsourcing-is-the-perfect-solution-to-making-sense-of-big-data (дата обращения: 15.06.2020).

49. Применение Big Data в медицине [Электронный ресурс]. – URL: https://news.rambler.ru/other/39885536/?utm_content=news_media&utm_medium=read_more&utm_source=copylink (дата обращения: 07.07.2020).

50. Прогнозная аналитика (Big Data) [Электронный ресурс]. – URL: https://marketing-logic.ru/bigdata (дата обращения: 18.06.2020).

51. Прохоренко, Д. Ценный кадр: как предсказать увольнение сотрудников с помощью Big Data [Электронный ресурс] / Д. Прохоренко. – URL: https://www.forbes.ru/karera-i-svoy-biznes/362647-cennyy-kadr-kak-predskazat-uvolnenie-sotrudnikov-s-pomoshchyu-big-data (дата обращения: 13.06.2020).

52. Пять вещей, которые необходимо знать о Hadoop и Apache Spark [Электронный ресурс]. – URL: https://www.osp.ru/news/articles/2015/49/13048137 (дата обращения: 21.06.2020).

53. Революция Big Data : Как извлечь необходимую информацию из «Больших Данных»? [Электронный ресурс]. – URL: http://statsoft.ru/products/Enterprise/big-data.php (дата обращения: 18.06.2020).

54. РЖД обсуждают с «Яндексом» использование технологий big data [Электронный ресурс]. – URL: https://tass.ru/transport/3765979 (дата обращения: 13.06.2020).

55. Родители шокированы платной версией электронного дневника: реакция компании [Электронный ресурс]. – URL: https://www.ridus.ru/news/307873 (дата обращения: 07.07.2020).

56. Саймон, Ф. Озеро данных и хранилище данных – в чем разница? [Электронный ресурс] / Ф. Саймон. – URL: https://www.sas.com/ru_ru/insights/articles/data-management/data-lake-and-data-warehouse-know-the-difference.html (дата обращения: 19.06.2020).

57. Сколько стоит проект Big Data? [Электронный ресурс]. – URL: http://datareview.info/article/skolko-stoit-proekt-big-data/ (дата обращения: 18.06.2020).

58. Технологии Big data в страховании жизни [Электронный ресурс]. – URL: https://lifeinsurance.kz/ekspert/tehnologii-big-data-v-strahovanii-zhizni (дата обращения: 13.06.2020).

59. Управление «большими данными» – что должен знать каждый ИТ-директор [Электронный ресурс]. – URL: https://www.itweek.ru/idea/article/detail.php?ID=136441 (дата обращения: 05.07.2020).

60. Фрэнкс, Б. Революция в аналитике. Как в эпоху Big Data улучшить ваш бизнес с помощью операционной аналитики [Электронный ресурс] / Б. Фрэнкс. – URL: https://lifeinbooks.net/read-online/revolyutsiya-v-analitike-kak-v-epohu-big-data-uluchshit-vash-biznes-s-pomoshhyu-operatsionnoy-analitiki-bill-frenks/ (дата обращения: 05.07.2020).

61. Фрэнкс, Б. Укрощение больших данных. Как извлекать знания из массивов информации с помощью глубокой аналитики / Б. Фрэнкс. М. : Манн, Иванов и Фербер, 2014. – ISBN 978-5-00057-146-0.

62. Хранилища данных [Электронный ресурс]. – URL: https://portal.tpu.ru/SHARED/p/PAN/Wrk/Tab9/Lk.doc (дата обращения: 19.06.2020).

63. Хранилище данных [Электронный ресурс]. – URL: https://www.intuit.ru/studies/courses/599/455/lecture/10155 (дата обращения: 07.07.2020).

64. Храните данные в облаке [Электронный ресурс]. – URL: https://habr.com/ru/company/bigdatahosting/blog/353168/ (дата обращения: 18.06.2020).

65. Цифровая революция в здравоохранении: достижения и вызовы [Электронный ресурс]. – URL: https://tass.ru/pmef-2017/articles/4278264 (дата обращения: 08.07.2020).

66. Чопра, Х. Как развитие Big data улучшит страховые продукты для населения [Электронный ресурс] / Х. Чопра. – URL: https://www.rbc.ru/opinions/money/08/12/2016/584923d59a79476a92d8c7a6 (дата обращения: 07.08.2020).

67. Что скрывается в тени больших данных? [Электронный ресурс]. – URL: https://blog.fts-eu.com/ru/2016/01/26/ (дата обращения: 15.06.2020).

68. Что такое Big data: собрали все самое важное о больших данных [Электронный ресурс]. – URL: https://rb.ru/howto/chto-takoe-big-data/ (дата обращения: 19.06.2020).

69. Что такое DWH и почему без них данные компании почти бесполезны [Электронный ресурс]. – URL: https://mcs.mail.ru/blog/chto-takoe-dwh-i-pochemu-bez-nih-dannye-kompanii-bespolezny (дата обращения: 20.06.2020).

70. Что такое In-Memory Data Grid [Электронный ресурс]. – URL: https://habr.com/ru/post/160517/ (дата обращения: 05.07.2020).

71. Что такое витрина данных? Определение, разновидности и примеры [Электронный ресурс]. – URL: https://yandex.ru/turbo/s/fb.ru/article/402525/chto-takoe-vitrina-dannyih-opredelenie-raznovidnosti-i-primeryi (дата обращения: 05.07.2020).

72. Шпрингер, Е. Технологии big data: как анализируют большие данные, чтобы получить максимум прибыли [Электронный ресурс] /

Е. Шпрингер. – URL: https://mcs.mail.ru/blog/tekhnologii-big-data-kak-analiziruyut-bolshie-dannye (дата обращения: 18.06.2020).

73. Шпрингер, Е. Что такое озера данных и почему в них дешевле хранить big data [Электронный ресурс] / Е. Шпрингер. – URL: https://mcs.mail.ru/blog/chto-takoe-ozera-dannyh-i-zachem-tam-hranyat-big-data (дата обращения: 19.06.2020).

74. Эко-Big Data в большом городе: как технологии делают мегаполис чище [Электронный ресурс]. – URL: https://www.bigdataschool.ru/bigdata/iot-big-data-ml (дата обращения: 05.07.2020).

75. Big Data – что такое системы больших данных? Развитие технологий Big Data [Электронный ресурс]. – URL: https://promdevelop.ru/big-data/ (дата обращения: 15.06.2020).

76. BIG DATA 2017: Где хранить Большие Данные [Электронный ресурс]. – URL: https://www.computerworld.ru/articles/BIG-DATA-2017-Gde-hranit-Bolshie-Dannye (дата обращения: 15.06.2020).

77. Big Data vs Data Mining [Электронный ресурс]. – URL: https://habr.com/ru/post/267827/ (дата обращения: 05.07.2020).

78. Big Data в кадровом менеджменте [Электронный ресурс]. – URL: https://www.hr-director.ru/article/67163-big-data-v-menedjmente-18-m6 (дата обращения: 18.06.2020).

79. Big Data в медицине, возможности и примеры применения [Электронный ресурс]. – URL: https://zen.yandex.ru/media/id/5ad352821410c33175a78d99/big-data-v-medicine-vozmojnosti-i-primery-primeneniia-5ad483f6830905287b490bf0 (дата обращения: 06.07.2020).

80. Big data на страже здоровья: как и зачем медицинские организации собирают и хранят данные [Электронный ресурс]. – URL: https://hightech.fm/2018/09/21/bigdata-med (дата обращения: 06.07.2020).

81. Big Data от А до Я. Часть 1: Принципы работы с большими данными, парадигма MapReduce [Электронный ресурс]. – URL: https://habr.com/ru/company/dca/blog/267361/ (дата обращения: 05.07.2020).

82. Big data. Большие данные в медицине [Электронный ресурс]. – URL: https://medspecial.ru/news/1/28048/ (дата обращения: 13.06.2020).

83. Brandt, J. MPI to Release Results ot 2017 Internet of Things (IoT) Study [Электронный ресурс]. – URL: https://mpi-group.com/uncategorized/mpi-2017-iot-study/ (дата обращения: 08.07.2020).

84. Data Mart vs Data Warehouse [Электронный ресурс]. – URL: https://habr.com/ru/post/72389/ (дата обращения: 15.06.2020).

85. Spark и sparklyr для работы с большими данными в R [Электронный ресурс]. – URL: https://r-analytics.blogspot.com/2020/02/spark-intro.html (дата обращения: 06.07.2020).

86. Spark или Hadoop – Какая платформа для Big Data лучше? [Электронный ресурс]. – URL: http://spbdev.biz/blog/spark-ili-hadoop-kakaya-platforma-dlya-big-data-luchshe (дата обращения: 18.06.2020).

87. SQL [Электронный ресурс]. – URL: https://ru.wikipedia.org/wiki/SQL (дата обращения: 06.07.2020).

88. Telecom & IT. Ликбез № 6. Объектные системы хранения (Object Storage) [Электронный ресурс]. – URL: https://shalaginov.com/2019/08/06/6262/ (дата обращения: 15.06.2020).

89. 6 главных трендов цифровой медицины 2019 [Электронный ресурс]. – URL: https://firstlinesoftware.ru/news/6-glavnyh-trendov-cifrovoj-mediciny-2019/ (дата обращения: 21.06.2020).

90. 12 кейсов по биг дате: подтвержденные примеры из индустрии, когда биг дата приносит деньги [Электронный ресурс]. – URL: https://habr.com/ru/company/newprolab/blog/314926/ (дата обращения: 05.07.2020).

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ 3

Глава 1. ИНФОРМАЦИЯ, БОЛЬШИЕ ДАННЫЕ, BIG DATA 5

Глава 2. ОСНОВЫ УПРАВЛЕНИЯ БОЛЬШИМИ ДАННЫМИ 21

Глава 3. ТЕХНОЛОГИИ РАБОТЫ С БОЛЬШИМИ ДАННЫМИ 33

Глава 4. ХРАНИЛИЩА ДАННЫХ: ЭВОЛЮЦИЯ И ОБЩИЕ ОСНОВЫ 46

Глава 5. СОВРЕМЕННЫЕ ТЕХНОЛОГИИ ХРАНЕНИЯ ДАННЫХ 64

Глава 6. АНАЛИТИКА БОЛЬШИХ ДАННЫХ И ЕЕ ИНСТРУМЕНТАРИЙ 75

Глава 7. АНАЛИТИКА БОЛЬШИХ ДАННЫХ: ТЕХНИКИ ОБРАБОТКИ И АНАЛИЗА 92

ТЕСТЫ ДЛЯ САМОКОНТРОЛЯ 103

ЗАКЛЮЧЕНИЕ 108

ГЛОССАРИЙ 109

БИБЛИОГРАФИЧЕСКИЙ СПИСОК 113

Учебное издание

ТЕСЛЕНКО Ирина Борисовна
ГУБЕРНАТОРОВ Алексей Михайлович
ДИГИЛИНА Ольга Борисовна
и др.

BIG DATA

БОЛЬШИЕ ДАННЫЕ

Учебное пособие

Редактор Е. А. Платонова
Технический редактор Ш. В. Абдуллаев
Корректор О. В. Балашова
Компьютерная верстка Е. А. Кузьминой
Выпускающий редактор А. А. Амирсейидова

Подписано в печать 24.12.21.
Формат 60×84/16. Усл. печ. л. 7,21. Тираж 50 экз.
Заказ
Издательство
Владимирского государственного университета
имени Александра Григорьевича и Николая Григорьевича Столетовых.
600000, Владимир, ул. Горького, 87.