МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

УНИВЕРСИТЕТ ИТМО

И.А. Радченко, И.Н. Николаев

ТЕХНОЛОГИИ И ИНФРАСТРУКТУРА BIG DATA

РЕКОМЕНДОВАНО К ИСПОЛЬЗОВАНИЮ В УНИВЕРСИТЕТЕ ИТМО
по направлению подготовки (специальности) 27.04.05 Инноватика
в качестве учебного пособия для реализации основных профессиональных образовательных программ высшего образования магистратуры

Санкт-Петербург

2018

Радченко И.А, Николаев И.Н. Технологии и инфраструктура Big Data. – СПб: Университет ИТМО, 2018. – 52 с.

Рецензенты: Муромцев Д.И, кандидат технических наук, доцент

В учебном пособии в сжатой форме излагаются основные принципы, подходы и направления технологий и инфраструктуры Big Data. Авторы дают краткий обзор подходов и определений, предоставляют обзор экосистемы Больших данных и раскрывают тему систем управления Большими данными. В учебном подобии также представлен краткий обзор областей применения Больших данных и архитектура системы обработки Больших данных. Отдельно рассказывается о Hadoop/MapReduce и параллельных алгоритмах для работы с данными, а также об оборудовании для работы с Большими данными и центрах обработки данных.

В библиографическом списке приведены информационные источники, ссылки на которые встречаются в тексте данного учебного пособия. Представлен также список для самостоятельного изучения предметной области Больших данных.

Учебное пособие прежде всего ориентировано на студентов факультета технологического менеджмента и инноваций, и может быть полезным для студентов других факультетов и всех заинтересованных в технологиях Big Data лиц.

УНИВЕРСИТЕТ ИТМО

Университет ИТМО – ведущий вуз России в области информационных и фотонных технологий, один из немногих российских вузов, получивших в 2009 году статус национального исследовательского университета. С 2013 года Университет ИТМО – участник программы повышения конкурентоспособности российских университетов среди ведущих мировых научно-образовательных центров, известной как проект «5 в 100». Цель Университета ИТМО – становление исследовательского университета мирового уровня, предпринимательского по типу, ориентированного на интернационализацию всех направлений деятельности.

Содержание

Введение

Обычно в введении говорят о том, что будет в курсе. Однако мы для начала расскажем о том, чего в курсе не будет.

Понятие Больших данных (Big data) сейчас является очень модным. Термин Big data используется практически так же часто, как и термины: Blockchain, Digital Economy, ICO и др. Зачастую их путают с открытыми данными, придумывают разнообразные мистические смыслы, формы и содержания. В учебном пособии этого не будет.

В нем не будет также программирования, теорем и основательной математики.

А что же здесь будет?

Мы исходим из того, что существенными являются знания, причем в первую очередь фундаментальные знания, выражающиеся в понимании, откуда взялись Большие данные, что с ними имеет смысл делать, какие результаты можно получить, что же делать дальше и что всё это обозначает.

При этом студент, осознанно прошедший предлагаемый курс, должен уверенно отвечать на достаточно простые вопросы, предлагаемые в качестве проверочных. Эти вопросы рассматриваются как основа для беседы по предмету, выявляющей общее понимание по циклу жизни данных и алгоритмов для работы с ними.

Опишем кратко содержание учебного пособия.

В главе 1 вводится понятие данных, дается краткий обзор подходов и определений, рассказывается о жизненном цикле данных. Это важно для понимания того, что есть данные. Зачастую работа с Большими данными существенно упрощается за счет работы с их метаданными, поэтому в главе 2 приводятся краткие сведения о метаданных и о жизненном цикле метаданных. В главе 3 представлен обзор экосистемы Больших данных, раскрывается тема систем управления Большими данными. Глава 4 содержит краткий обзор архитектур систем обработки Больших данных. В главе 5 дается представление о Hadoop/MapReduce и о параллельных алгоритмах для работы с данными. В главе 6 в сжатой форме рассказывается о программных платформах и системах для Больших данных. В главе 7 излагается материал об оборудовании для обработки Больших данных. Глава 8 предлагает краткие сведения о центрах обработки Больших данных.

Все главы снабжены вопросами для проверки усвоенных знаний.

В библиографическом списке приведены информационные источники, ссылки на которые встречаются в тексте данного учебного пособия. Отдельно также приведен список информационных источников для самостоятельного изучения, позволяющий погрузиться в предметную область Больших данных и получить более основательное представление об этой области знаний.

1. Данные. Подходы и определения

1.1. Определения данных

Согласно ГОСТ Р 52653-2006[1], данные – представление информации в формализованном виде, пригодном для передачи, интерпретации и обработки.

Согласно ГОСТ 7.0-99[2], данные – информация, обработанная и представленная в формализованном виде для дальнейшей обработки.

В Кэмбриджском словаре приводятся следующие определения данных: данные – это информация, особенно факты и числа, собранные для последующего использования при принятии решений. Данные – это информация в электронной форме, пригодная для хранения и использования компьютером. [13]

На сегодняшний день также довольно широко распространена формулировка, заключающаяся в том, что данные являются нефтью цифровой экономики. [12]

1.2. Философский подход

Исходно понятие данных – философское, оно возникает в эпистемологии при рассмотрении основой проблемы гносеологии – познаваемости мира, поиска и осмысления истины. Процедуры верификации или фальсификации данных создают информацию, осмысление истины создает знание.

Философия рассматривает преобразование сведений в данные, данных в информацию, а информации – в знания. Истинность сведений субъективна. Сведения, выраженные в формальном представлении, являются данными. Обработка данных позволяет определить сколько в них содержится информации. При осмыслении информации экспертом создаются знания.

1.3. Юридический подход

В Конституции РФ следующие статьи регламентируют действия с информацией:

- часть 1 статьи 24 запрещает сбор, хранение, использование и распространение информации о частной жизни лица без его согласия;
- часть 4 статьи 29 предоставляет каждому право свободно искать, получать, передавать, производить и распространять информацию любым законным способом;
- статья 42 предоставляет каждому право на достоверную информацию об окружающей среде.

---

[1] ГОСТ Р 52653-2006. http://www.ifap.ru/library/gost/526532006.pdf

[2] ГОСТ 7.0-99. http://www.docload.ru/Basesdoc/33/33922/index.htm

При этом, согласно статьи 71 конституции в ведении РФ находится “информация и связь”.

На основе Конституции разрабатываются федеральные законы. В РФ принят ФЗ-149 "Об информации, информационных технологиях и о защите информации". Он регулирует отношения возникающие при осуществлении права часть 4 статьи 29 Конституции РФ, при применении информационных технологий и обеспечении защиты информации.

В статье 3 ФЗ-149 дается определение информации как таковой, а также связанные с информацией объекты и действия. [7]

1) информация – сведения (сообщения, данные) независимо от формы их представления;
2) информационные технологии – процессы, методы поиска, сбора, хранения, обработки, предоставления, распространения информации и способы осуществления таких процессов и методов;
3) информационная система – совокупность содержащейся в базах данных информации и обеспечивающих ее обработку информационных технологий и технических средств;
4) информационно-телекоммуникационная сеть – технологическая система, предназначенная для передачи по линиям связи информации, доступ к которой осуществляется с использованием средств вычислительной техники;
5) обладатель информации – лицо, самостоятельно создавшее информацию либо получившее на основании закона или договора право разрешать или ограничивать доступ к информации, определяемой по каким-либо признакам;
6) доступ к информации – возможность получения информации и ее использования;
7) конфиденциальность информации – обязательное для выполнения лицом, получившим доступ к определенной информации, требование не передавать такую информацию третьим лицам без согласия ее обладателя;
8) предоставление информации – действия, направленные на получение информации определенным кругом лиц или передачу информации определенному кругу лиц;
9) распространение информации – действия, направленные на получение информации неопределенным кругом лиц или передачу информации неопределенному кругу лиц;
10) электронное сообщение – информация, переданная или полученная пользователем информационно-телекоммуникационной сети;
11) документированная информация – зафиксированная на материальном носителе путем документирования информация с реквизитами, позволяющими определить такую информацию или в установленных

законодательством Российской Федерации случаях ее материальный носитель;

11.1) электронный документ – документированная информация, представленная в электронной форме, то есть в виде, пригодном для восприятия человеком с использованием электронных вычислительных машин, а также для передачи по информационно-телекоммуникационным сетям или обработки в информационных системах;

12) оператор информационной системы – гражданин или юридическое лицо, осуществляющие деятельность по эксплуатации информационной системы, в том числе по обработке информации, содержащейся в ее базах данных;

13) сайт в сети "Интернет" – совокупность программ для электронных вычислительных машин и иной информации, содержащейся в информационной системе, доступ к которой обеспечивается посредством информационно-телекоммуникационной сети "Интернет" (далее – сеть "Интернет") по доменным именам и (или) по сетевым адресам, позволяющим идентифицировать сайты в сети "Интернет";

14) страница сайта в сети "Интернет" (далее также – интернет-страница) – часть сайта в сети "Интернет", доступ к которой осуществляется по указателю, состоящему из доменного имени и символов, определенных владельцем сайта в сети "Интернет";

15) доменное имя – обозначение символами, предназначенное для адресации сайтов в сети "Интернет" в целях обеспечения доступа к информации, размещенной в сети "Интернет";

16) сетевой адрес – идентификатор в сети передачи данных, определяющий при оказании телематических услуг связи абонентский терминал или иные средства связи, входящие в информационную систему;

17) владелец сайта в сети "Интернет" – лицо, самостоятельно и по своему усмотрению определяющее порядок использования сайта в сети "Интернет", в том числе порядок размещения информации на таком сайте;

18) провайдер хостинга – лицо, оказывающее услуги по предоставлению вычислительной мощности для размещения информации в информационной системе, постоянно подключенной к сети "Интернет";

19) единая система идентификации и аутентификации – федеральная государственная информационная система, порядок использования которой устанавливается Правительством Российской Федерации, и которая обеспечивает в случаях, предусмотренных законодательством Российской Федерации, санкционированный доступ к информации, содержащейся в информационных системах;

20) поисковая система – информационная система, осуществляющая по запросу пользователя поиск в сети "Интернет" информации

определенного содержания и предоставляющая пользователю сведения об указателе страницы сайта в сети "Интернет" для доступа к запрашиваемой информации, расположенной на сайтах в сети "Интернет", принадлежащих иным лицам, за исключением информационных систем, используемых для осуществления государственных и муниципальных функций, оказания государственных и муниципальных услуг, а также для осуществления иных публичных полномочий, установленных федеральными законами.

К сожалению, формулировки ФЗ-149 недостаточно полные. Например, в законе не определено, что такое сеть “Интернет”. Некоторые определения не соответствуют принятым в IT-отрасли терминам. Например, в российском языке термин “электронный документ” обозначает информацию на конкретном материальном носителе.

Принятые в других странах термины могут отличаться, например, в США электронный документ обозначает любую информацию в цифровой форме, где “информация“ может включать в себя данные, текст, звуки, коды, компьютерные программы, программное обеспечение или базы данных. “Данные“ в этом контексте относятся к ограниченному набору элементов данных, каждый из которых состоит из содержимого или значения вместе с пониманием того, что означает контент или значение; где электронный документ содержит данные, это понимание того, что элемент данных или значение элемента данных должно быть явно включено в сам электронный документ или быть легко доступным для получателя электронного документа. [4]

1.4. Жизненный цикл данных

Жизненный цикл данных – это последовательность этапов, которую конкретная порция данных проходит от начального этапа создания или получения до момента архивации или удаления. [14]

Основные этапы жизненного цикла данных представлены на рис. 1.

Рассмотрим эти этапы подробнее.

1.4.1. Создание данных (Data Generation/Data Capture)

На этом этапе данные генерируются или захватываются. Этот этап обычно еще делят на три типа получения данных:

1. Приобретение данных (Data Acquisition)

   Получение организацией данных, уже сгенерированных вне предприятия.
2. Запись данных (Data Entry)

   Создание новых данных оператором или компьютером. Данные имеют ценность для предприятия.
3. Регистрация сигналов (Signal Reception)

Захват данных устройствами. Особенно важно в системах управления, но в последнее время особенно ценно при использовании такого подхода, как Интернет вещей.

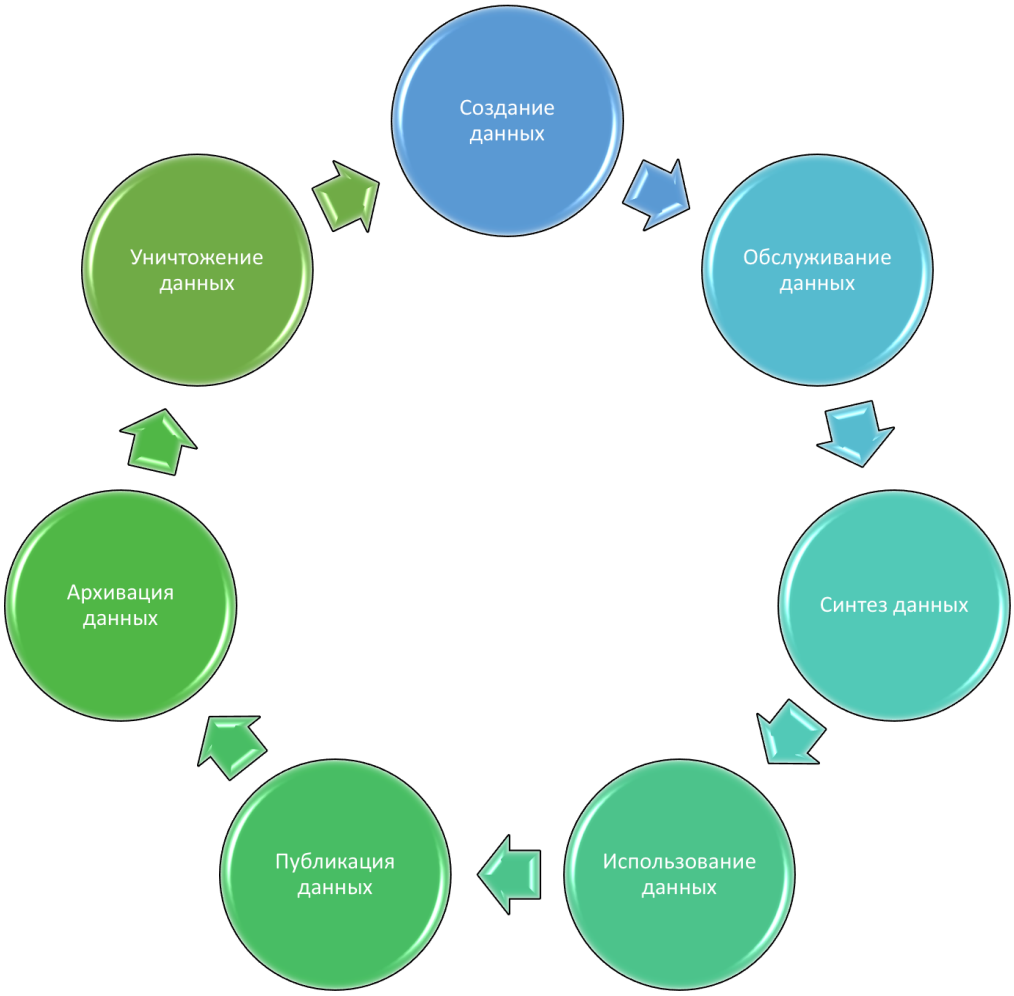

Рис. 1 – Жизненный цикл данных

Все три варианта получения данных очень важны в рамках рассмотрения процесса управления данными. [17]

1.4.2. Обслуживание данных (Data Maintenance)

После того, как данные были созданы, необходимо их хранить и обслуживать. Необходимо осуществлять доставку данных в точку, где их будут использовать или производить над ними манипуляции (например, операции синтеза).

Можно говорить о том, что обслуживание данных – это обработка данных без получения из извлечения из них полезной информации для предприятия.

Зачастую обслуживание данных включает в себя такие действия с данными, как перемещение, интеграция, очистка, обогащение и ETL-процессы (Extract, Transform, Load).

Обслуживание данных обычно подразумевает применение широкого спектра методов из области управления данными (Data Management). [17]

1.4.3. Синтез данных (Data Synthesis)

Это сравнительно недавно появившаяся стадия в жизненном цикле данных. Используется не во всех моделях жизненных циклах данных.

Синтез данных – это процесс получения дополнительной ценности из данных при помощи использования индуктивной логики и сторонних информационных источников.

Это стадия, на которой с данными работаю аналитики, причем они могут использовать в своей работы методы моделирования рисков, актуарного моделирования, моделирования для принятия инвестиционных решений и др.

На этой стадии используется индуктивная логика, не дедуктивная. Индуктивная логика требует использования экспертного мнения, т.к. именно компетенции экспертов необходимы для построения моделей скоринга и др. [17]

1.4.4. Использование данных (Data Usage)

До сих пор шла речь об использовании данных внутри одного предприятия, которые возможно были подвержены очистке и обогащению на стадии обслуживания данных и использовались совместно с дополнительным третьими источниками данных на стадии синтеза данных.

На стадии использования данных они применяются в качестве полезной информации для задач, которые должны выполняться и управляться на основе данных.

Эти задачи могут быть вне жизненного цикла данных. Тем не менее, данные становятся все более значимой частью бизнес-процессов предприятий. Данные сами могут быть продуктом или услугой (или быть частью продукта или услуги), предлагаемой предприятием.

Использование данных имеет специальные задачи в рамках управления данными (Data Governance). Одна из задач заключается в законном использовании данных в требуемом виде. Это называется “разрешенное использование данных” (permitted use of data). Могут существовать регулирующие или договорные ограничения на то, как фактически можно использовать данные, а часть роли управления данными (Data Governance) заключается в обеспечении соблюдения этих ограничений. [17]

1.4.5. Публикация данных (Data Publication)

При использовании данных возможна ситуация, когда данные отправляются за пределы предприятия. В этом случае говорят о публикации данных. Публикация данных – это вынос данных за пределы предприятия.

Примером этого процесса может быть маклер, рассылающий ежемесячные отчеты клиентам. Все данные, которые были разосланы, уже не могут быть отозваны. Если были разосланы данные с неверными значениям, то такие данные не могут быть исправлены, поскольку они уже становятся недоступны для предприятия. Управление данными (Data Governance) может потребоваться, чтобы помочь принять решение о том, как будут обрабатываться неверные данные, которые были отправлены из предприятия. [17]

1.4.6. Архивация данных (Data Archival)

Данные могут быть использованы как однократно, так и несколько раз. Но затем рано или поздно жизненный цикл данных начинает подходить к концу.

Первая стадия этого состояния заключается в архивации данных.

Архивация данных – это копирование данных в пассивную среду, в которой они хранится, для тех случаев, когда они понадобятся снова в активной производственной среде, и удаление этих данных из всех активных производственных сред.

Архив данных – это просто место, где хранятся данные, без их обслуживания, использования или публикации. В случае необходимости данные могут быть восстановлены из архива. [17]

1.4.7. Уничтожение данных (Data Purging)

Уничтожение данных – это последовательность операций для выполнения необратимого удаления данных, делающая невозможным как восстановление данных, так и получение остаточной информации (Data Remanence) о них. Это одна из самых сложно реализуемых процедур управления данными.

Даже с теоретической точки зрения существует команда записи значения в ячейку памяти, но команды стирания значения как таковой нет. Для уничтожения данных необходимо изготовить высокопроизводительный источник случайных чисел и перезаписать ими весь носитель информации (перезаписи области хранения недостаточно, так как сохраняется информация об исходном количестве данных).

Иначе говоря, при уничтожении данных необходимо не только сделать недоступными от восстановления на физическом уровне сами данные, но и связанную с ними информацию в других наборах данных. Уничтожение данных регламентируется в ГОСТ Р 50735[3], причём классы защищенности данных

[3] ГОСТ Р 50735. http://docs.cntd.ru/document/gost-r-50739-95

и протоколы работы устанавливаются на уровне руководящих документов гостехкомиссии (ФСТЭК, Федеральной службы по техническому и экспортному контролю).

В настоящее время в РФ предусмотрено семь классов защиты информации. Согласно действующему на настоящему моменту пункту 19.2 приказа ФСТЭК N17 от 11.02.2013 “При выводе из эксплуатации машинных носителей информации, на которых осуществлялись хранение и обработка информации, осуществляется физическое уничтожение этих машинных носителей информации”. [19]

Вопросы для проверки

1. Что такое данные?
2. Какие ГОСТы с определениями данных вам известны?
3. Какие определения даются в ФЗ-149?
4. Что такое жизненный цикл данных?
5. Перечислите этапы жизненного цикла данных.
6. Для каких целей нужен этап “Синтез данных” (один из этапов жизненного цикла данных)?
7. Для каких целей нужен этап “Использование данных” (один из этапов жизненного цикла данных)?
8. Для каких целей нужен этап “Публикация данных” (один из этапов жизненного цикла данных)?
9. Для каких целей нужен этап “Архивация данных” (один из этапов жизненного цикла данных)?

2. Метаданные

2.1. Понятие метаданных

При сборе данных возникают метаданные, содержащие какую-либо информацию о собранных данных. Например, время создания набора данных, авторство и первоисточник, размер и кодировка данных – все это метаданные.

В соответствии с ГОСТ Р 52438-2005[4] метаданные – это сведения о данных.

Структурированные метаданные называют онтологией или схемой метаданных.

В методическом пособии “Онтологический инжиниринг знаний в системе PROTÉGÉ” [11] написано, что “онтология определяет общий словарь для ученых, которым нужно совместно использовать информацию в предметной области. Она включает машинно-интерпретируемые формулировки основных понятий предметной области и отношения между ними”.

Онтологии обычно используются в таких областях, как искусственный интеллект, семантическая паутина, системная инженерия, биомедицинская информатика, библиотечное дело, информационная архитектура и др. Все онтологии нужны для организации информации.

2.2. Жизненный цикл метаданных

Жизненный цикл метаданных можно разделить на четыре стадии, представленных на рис. 2. [20]

1. Оценка требований и анализ контента.
2. Спецификация системных требований.
3. Система метаданных.
4. Сервис и оценка.

Рассмотрим эти стадии подробнее.

2.2.1. Оценка требований и анализ контента

Этот этап состоит из четырех частей.

1. Выявление и получение базовых требований к метаданным.
2. Обзор релевантных стандартов и проектов по метаданным.
3. Исследование требований к глубоким метаданным.
4. Идентификация стратегий для схем метаданных.

---

[4] ГОСТ Р 52438-2005. http://www.complexdoc.ru/text/%D0%93%D0%9E%D0%A1%D0%A2%20%D0%A0%2052438-2005/1

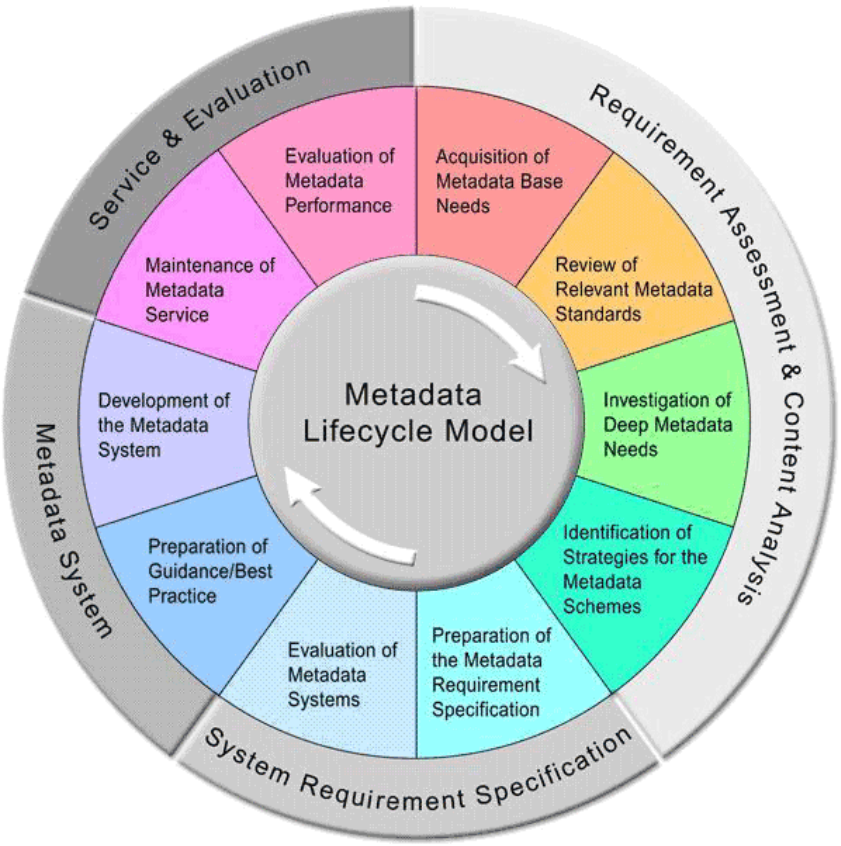

Рис. 2 – Жизненный цикл метаданных

Выявление и получение базовых требований к метаданным

На этом этапе жизненного цикла метаданных проводится опрос экспертов и специалистов по предметной области на тему требований к метаданным в конкретном проекте и анализ атрибутов. Целью опроса или интервьюирования экспертов является получение предварительной информации о проекте и установление контакта между специалистами по метаданным и провайдерами данных (или контента). На этой стадии должны быть уточнены и разъяснена и уточнена область метаданных, контекст метаданных, действующая система метаданных, роль и функция метаданных. [20]

Обзор релевантных стандартов и проектов по метаданным

Этот этап включает определение стандартов метаданных, потенциально полезных для проекта, изучение существующих схем метаданных и вариантов их использования. Этот этап нужен для того, чтобы можно было лучше узнать, какие различия существуют среди других аналогичных проектов, а также пересмотреть цели проекта. [20]

Исследование требований к глубоким метаданным

На этом этапе определяются требования к метаданным более детально и глубоко. Для этого используется концепция контент-анализа. Здесь важно уточнить область и контент метаданных, а также проработан вопрос планируемых в работе СУБД и информационных систем, в которых планируется использовать метаданные. В дальнейшем эти наработки могут быть использованы в качестве методики для масштабирования. [20]

Идентификация стратегий для схем метаданных

На этом этапе формулируется стратегия использования метаданных на основе всех имеющих предыдущих результатов. Стратегия включает принятие одного или нескольких существующих стандартов метаданных и разработку на основе этих стандартов схемы метаданных. [20]

2.2.2. Спецификация системных требований

Этот этап состоит из двух частей.

1. Подготовка спецификации требований к метаданным
2. Оценка систем метаданных

Подготовка спецификации требований к метаданным

Спецификация требований к метаданным (Metadata Requirement Specification, MRS) является связующим звеном между участниками проекта, специалистами по метаданным и разработчиками систем. [20]

Оценка систем метаданных

На этом этапе производится оценка систем метаданных, которые потенциально могут использоваться в проекте в дальнейшем. Участники проекта могут выбрать из существующих систем метаданных. [20]

2.2.3. Система метаданных

Этот этап состоит из двух частей.

1. Подготовка руководства по лучшей практике.
2. Разработка системы метаданных.

Подготовка руководства по лучшей практике

Этот этап включает в себя создание документации и разработку руководящих принципов “лучшей практики” для отдельных элементов метаданных. Руководство может использоваться как контрольный список или как средство обеспечения контроля качества записей метаданных в проекте. [20]

Разработка системы метаданных

Разработчики системы разрабатывают инструменты и системы метаданных на основе спецификации требований к метаданным. [20]

2.2.4. Сервис и оценка

Этот этап состоит из двух частей.

1. Поддержка службы метаданных.
2. Оценка качества метаданных.

Поддержка службы метаданных

Цель разработки служб метаданных – гарантировать качество метаданных и механизмов работы с ними. Модель сервиса метаданных состоит из трех основных элементов: служебный механизм, роли и отношения между ролями. [20]

Оценка эффективности метаданных

Последний этап жизненного цикла метаданных направлен на рассмотрение результатов всего процесса работы с метаданными и качества самих метаданных. Интегральная оценка состоит из оценки качества метаданных, оценки эффективности работы схемы метаданных, оценки результатов использования инструментов создания метаданных и оценки применения модели жизненного цикла метаданных на каждом этапе. [20]

Для более подробного ознакомления с метаданными и работу с ними прочтите книги [25]-[28] из Списка информационных источников для самостоятельной работы, приведенного в конце данного учебного пособия.

Вопросы для проверки

1. Что такое метаданные?
2. Какие ГОСТы есть для метаданных?
3. Что такое онтология?
4. Что такое жизненный цикл метаданных?
5. Какие этапы жизненного цикла метаданных вы знаете?
6. Зачем нужен этап “Оценка требований и анализ контента” (один из этапов жизненного цикла метаданных)?
7. Зачем нужен этап “Спецификация системных требований” (один из этапов жизненного цикла метаданных)?
8. Зачем нужен этап “Система метаданных” (один из этапов жизненного цикла метаданных)?
9. Зачем нужен этап “Сервис и оценка” (один из этапов жизненного цикла метаданных)?

3. Большие данные. Системы управления Большими данными

Если давать краткое определение, то *Большие данные* – это данные, которые не помещаются в оперативную память компьютера.

По сути это определение обозначает то, что свойство “быть большим” является не самостоятельным свойством данных, а зависит от характеристики системы, применяемой для их обработки.

Например, обычному человеку затруднительно запомнить какая именно температура была в нашем городе каждый день за прошедший месяц. Таким образом, три десятка значений вполне могут быть примером Больших данных. Однако вот человек уверенно сообщает “прошедший месяц был холодным”. Это сообщение несет информацию об обработанных данных: по мнению собеседника, средняя температура за прошедший месяц была ниже, чем обычно в этом месяце за несколько десятков лет.

Другим примером могут быть данные об объектах, которые теоретически несут важную информацию, однако имеющие такой размер, что эти данные практически невозможно не только обработать или сохранить, но даже собрать. Рассмотрим к примеру набор данных, содержащий координаты и скорости молекул в воздушном столбе над территорией аэропорта. Имеются также метаданные с описанием в какой момент проводилось измерение и что это за молекула. Такой набор данных несет информацию о погодных условиях над аэропортом, включая температуру, давление, влажность, облачность, особые погодные условия – проходящий торнадо или падающий град. С другой стороны, для корректной обработки данные для всех молекул должны быть достаточно полны и репрезентативны для статистической обработки.

В результате такого мысленного эксперимента мы понимаем, что для эффективной работы с большими данными нужна модель данных, позволяющая сформировать методы работы с данными.

Данные могут быть различных типов. Информацию, полученную в результате учёта или измерения каких-либо объектов или параметров, называют *мастер-данными* (Master Data). Например, учёт количества, замеры координат и скоростей конкретных молекул – это мастер-данные.

Транзакционные данные (в англоязычной литературе применяются термины Transactional Data, Application Specific Data, Operational Data) – это данные, отображающие результат выполнения каких-либо операций. Например, данные о взаимодействии молекул между собой, а именно о пересечении границ рассматриваемой области, о траектории конкретной молекулы, об испарении капель дождя – это транзакционные данные. Транзакционные данные описывают взаимодействие объектов друг с другом или с окружающим миром, которые можно получить при помощи обработки мастер-данных.

Ретроспективные данные (Historical data) – это данные, снабженные метками времени. Например, с одной стороны мы можем сохранять данные о координате и векторе скорости каждой молекулы, но если у нас есть набор координат в зависимости от времени, то скорость молекулы становится лишней, она вычисляется исходя из модели, описываемой ньютоновской механикой.

Ссылочные данные (справочники, НСИ, нормативно-ссылочная информация, Reference Data, Lookup Data, Dictionaries) – это базовые неизменяемые данные, заранее известные из внешних источников, такие как нормативы, сокращения, акронимы, словари, стандарты. Например, удельные веса молекул, зависимость температуры замерзания и кипения от давления, зависимость средней скорости молекул (скорости звука) от температуры.

Формат данных. Структурированные данные имеют заранее определенный формат. *Полуструктурированные* или *слабоструктурированные данные* – это данные, зачастую собранные из различных источников. Структура данных документирована, но в зависимости от источника данных конкретный формат представления информации может быть разным. Неструктурированные данные требуют обязательной обработки и последующей валидации перед использованием.

Например, данные о координатах и скоростях молекул, в которых некоторые координаты пропущены или некоторые записи повторяются, являются полуструктурированными. Нам нужно понять, почему так произошло и перед использованием либо исключить такие данные (что может привести к систематической ошибке), либо, исходя из модели данных, восстановить пропущенные значения.

Данные, в которых координаты измеряются в разных единицах измерения, числа иногда записаны словами, иногда латинскими цифрами, а иногда в виде сканированного изображения почерка лаборанта, являются *неструктурированными данными*.

Обычно Большие данные описываются при помощи следующих характеристик. [3]

1. *Объем* (Volume) – количество сгенерированных и хранящихся данных. Размер данных определяет значимость и потенциал данных, а также то, могут ли они быть рассмотрены как Большие данные.
2. *Разнообразие* (Variety) – тип данных. Большие данные могут состоять из текста, изображений, аудио, видео. Большие данные при сопоставлении друг с другом могут дополнять отсутствующие данные.
3. *Скорость* (Velocity) – скорость. Здесь подразумевается скорость, с которой данные генерируются и обрабатываются. Очень часто Большие данные используются в режиме реального времени.
4. *Изменчивость* (Variability) – противоречивость наборов данных может препятствовать их обработке и управлению ими.

5. *Достоверность* (Veracity) – качество данных напрямую влияет на точность проведения анализа данных.

Большие данные могут быть классифицированы в соответствии с несколькими главными компонентами. Интеллект-карта, представленная на рис. 3, была составлена на основе [2].

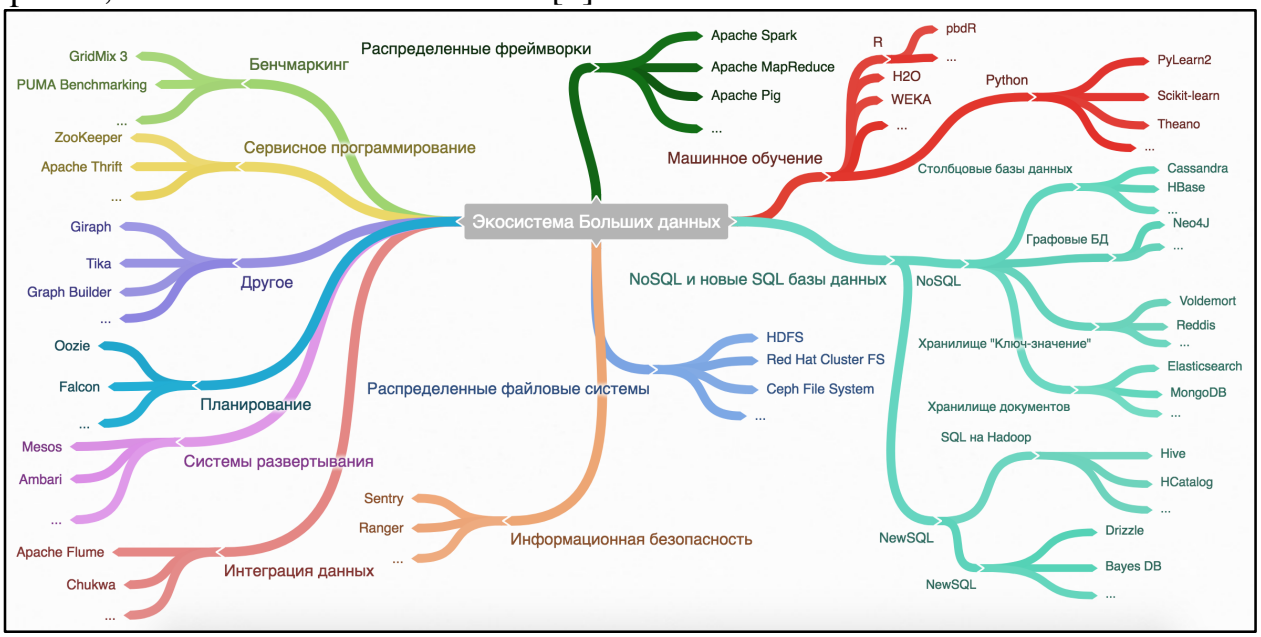

Рис. 3 – Интеллект-карта экосистемы Больших данных

На рис. 3 при помощи интеллект-карты показаны компоненты экосистемы Больших данных. Рассмотрим эти компоненты подробнее.

3.1. Распределенные файловые системы

Для хранения и обработки Больших данных созданы распределенные системы хранения данных, в том числе *распределенные файловые системы*, позволяющие использовать внешнее файловое пространство системы хранения для обработки данных на нодах, входящих в вычислительных кластер.

Зачастую удобно использовать распределенные файловые системы, арендуемые как отдельный облачный сервис, например, Google Colossus[5], Amazon S3[6], Yandex Disk[7].

3.2. Распределенные фреймворки

Обработка находящихся на распределенных системах хранения данных ведется параллельно на компьютерах, составляющих узлы (nodes)

[5] Google Colossus. https://cloud.google.com/bigtable/docs/overview
[6] Amazon S3. https://aws.amazon.com/ru/s3/
[7] Yandex Disk. https://tech.yandex.com/disk/

вычислительного кластера. Для организации вычислений разработчики систем обработки используют *распределенные фреймворки*. Большинство фреймворков доступны по лицензии Apache и ориентированы на работу в кластерах на базе Linux. Существуют также облачные фреймворки, арендуемые как отдельный облачный сервис. Некоторые из них описаны в **разделе 4 "Архитектура систем обработки Больших данных"** и в **разделе 6 "Программные платформы и системы для Больших данных"**.

3.3. Бенчмаркинг

Этот класс инструментов был разработан для оптимизации инсталляции Больших данных при помощи использования стандартизированных профилей (Profiling suites). *Бенчмаркинг* и оптимизация инфраструктуры Больших данных зачастую не является сферой ответственности дата-ученых, это область ответственности для отдельных профессионалов, специализирующихся на IT-инфраструктуре. Использование оптимизированной инфраструктуры может существенно снизить стоимость используемого оборудования. [2]

3.4. Серверное программирование

Предположим, что вы сделали приложение для прогнозирования результатов футбольных матчей мирового класса на платформе Hadoop, и вы хотите разрешить другим использовать прогнозы, сделанные вашим приложением. Тем не менее, вы не имеете представления об архитектуре или технологии всех, кто стремится использовать ваши прогнозы. Сервисные инструменты позволяют предоставлять приложения на Больших данных другим приложениям в качестве службы. Дата-ученым иногда приходится предоставлять свои модели через службы. Наиболее известным примером здесь является *REST-сервис*; REST означает репрезентативную передачу состояния (REpresentational State Transfer, REST). Она часто используется в качестве обмена данными с веб-сайтами. [2]

3.5. Планирование

Инструменты *планирования* позволяют автоматизировать повторяющиеся задачи и запускать задания на основе таких событий, как добавление нового файла в папку. Они похожи на такие инструменты, как CRON в Linux, но специально разработаны для работы в отказоустойчивом кластере. Вы можете использовать их, например, для запуска задачи MapReduce всякий раз, когда в каталоге имеется новый набор данных. [2]

3.6. Системы развертывания

Настройка инфраструктуры Больших данных – непростая задача, и развертывание новых приложений в кластере Больших данных – это зона ответственности инженеров по Большим данным. Они в значительной степени автоматизируют установку и настройку компонентов Больших данных. [2]

3.7. Интеграция данных

Допустим, что уже есть распределенная файловая система, и теперь необходимо перенести данные из одного источника в другой. В таких случаях используют *фреймворки для интеграции данных*, такие как Apache Sqoop и Apache Flume. Этот процесс похож на процесс извлечения, преобразования и загрузки (Extract, Transform and Load, ETL) в традиционном хранилище данных. [2]

3.8. Информационная безопасность

Средства обеспечения безопасности Больших данных позволяют осуществлять централизованный контроль доступа к данным. Безопасность Больших данных стала самостоятельной дисциплиной, и дата-ученые обычно сталкиваются с ней только как потребители данных. Безопасностью Больших данных занимаются эксперты по информационной безопасности. [2]

3.9. Машинное обучение

Если у вас есть Большие данные, то было бы неплохо получить из них полезный контент. Это можно сделать при помощи использования методов машинного обучения, статистики и прикладной математики. [2]

Еще перед Второй мировой войной многие трудоемкие вычисления производились вручную, что естественным образом ограничивало возможности анализа данных. После Второй мировой войны стала активно развиваться вычислительная техника и научные вычисления. Появилась возможность писать программы с формулами и алгоритмами, а затем загружать в программы различные данные.

На сегодняшний день, когда появилось огромнейшее количество данных, один компьютер уже не в состоянии справиться с задачей их обработки. Некоторые алгоритмы, разработанные в прошлом веке, увы, не смогут справиться с этой задачей, даже если теоретически можно было бы подключить к решению задачи все компьютеры Земли. Это связано с временной сложностью алгоритма[8].

[8] Временная сложность алгоритма.
https://ru.wikipedia.org/wiki/Временная_сложность_алгоритма

С примерами временной сложности можно ознакомиться на Stack Overflow[9].

Одна из самых больших проблем со старыми алгоритмами заключается в том, что они недостаточно масштабируются. Учитывая объем данных, которые необходимо анализировать сегодня, это становится проблематичным. Для обработки этого объема данных требуются специализированные структуры и библиотеки. Например, в языке Python есть следующие библиотеки: Scikit-learn (библиотека машинного обучения), PyBrain (для работы с нейронными сетями), NLTK (для обработки естественного языка), Pylearn2 (еще одна библиотека машинного обучения), TensorFlow (библиотека глубокого обучения, есть программный интерфейс API для языка Python), Keras (библиотека для работы с нейронными сетями) и другие. [2]

Существует также Apache Spark – программный каркас с открытым исходным кодом для реализации распределенной обработки неструктурированных и слабоструктурированных данных[10].

3.10. Базы данных NoSQL и новые SQL базы данных

Использование реляционных баз данных для обработки Больших данных крайне неэффективно из-за высоких накладных расходов. Традиционно для обработки Больших данных используются базы данных типа “ключ – значение” (Key value database или HashDB). Одна из первых баз этого типа DBM была реализована Ken Thompson для AT&T Unix 7 в 1979 году.

База данных вида “ключ – значение”, по сути, представляет собой ассоциативный массив (Hash, Dict), то есть множество, состоящее из пар (Key, Value). В некоторых реализациях на множестве ключей вводится отношение порядка, и мы можем получить значения последовательно по мере возрастания ключа. В других случаях сортировка по ключу неустойчива при одинаковых ключах и при неоднократных выборках можно получить различную последовательность пар.

Большое количество баз данных можно разделить на следующие типы:

- *Столбцовые базы данных* (Column databases). Данные хранятся в столбцах, что позволяет алгоритмам выполнять гораздо более быстрые запросы.
- *Хранилища документов* (Document stores) Хранилища документов больше не используют таблицы, но сохраняют каждое наблюдение в документе. Это позволяет использовать гораздо более гибкую схему данных.

---

[9] Real-world example of exponential time complexity. http://stackoverflow.com/questions/7055652/real-world-example-of-exponential-time-complexity

[10] Apache Spark. http://spark.apache.org/

- *Потоковые данные* (Streaming data). Данные собираются, преобразуются и агрегируются не в партиях, а в реальном времени.
- *Хранилища для ключей* (Key-value stores). Данные не хранятся в таблице; для каждого значения назначается ключ (как рассказано об этом выше).
- *SQL на Hadoop* – пакетные запросы на Hadoop, использующие фреймворк MapReduce в фоновом режиме.
- *Новый SQL* (New SQL). Этот тип сочетает масштабируемость баз данных NoSQL с преимуществами реляционных баз данных. Здесь используется интерфейс SQL и реляционная модель данных.
- *Графовые базы данных* (Graph databases). Это тип баз данных, использующих графовые структуры для семантических запросов с узлами и ребрами и свойствами для представления и хранения данных. Классическим примером этого типа является социальная сеть.

Доступ к базам данных можно арендовать в виде сервиса над распределёнными системами хранения, например, у Google это PostgreSQL[11] и NoSQL Datastore[12].

Вопросы для проверки

1. Что такое Большие данные?
2. Какие пять характеристик присущи Большим данным?
3. Какие существуют базовые принципы обработки Больших данных?
4. Оцените какое количество 10 Тб HDD необходимо для хранения набора данных, содержащего координаты, скорости и метаинформацию (тип молекулы и время измерения по конкретной молекуле) для всех молекул на территории аэропорта.
5. Что такое столбцовые базы данных?
6. Что такое хранилища документов?
7. Что такое потоковые данные?
8. Что такое хранилища для ключей?
9. Что такое SQL на Hadoop?
10. Что такое новый SQL?
11. Что такое графовые базы данных?

[11] Google PostgreSQL. https://cloud.google.com/sql/
[12] NoSQL Datastore. https://cloud.google.com/datastore/

4. Архитектура системы обработки Больших данных

Для работы с Большими данными используются сложные системы, в которых можно выделить несколько компонентов или слоёв (Layers). Обычно выделяют четыре уровня компонентов таких систем: прием, сбор, анализ данных и представление результатов (рис. 4). Это деление является в значительной мере условным так как, с одной стороны, каждый компонент в свою очередь может быть разделен на подкомпоненты, а с другой некоторые функции компонентов могут перераспределяться в зависимости от решаемой задачи и используемого программного обеспечения, например, выделяют хранение данных в отдельный слой.

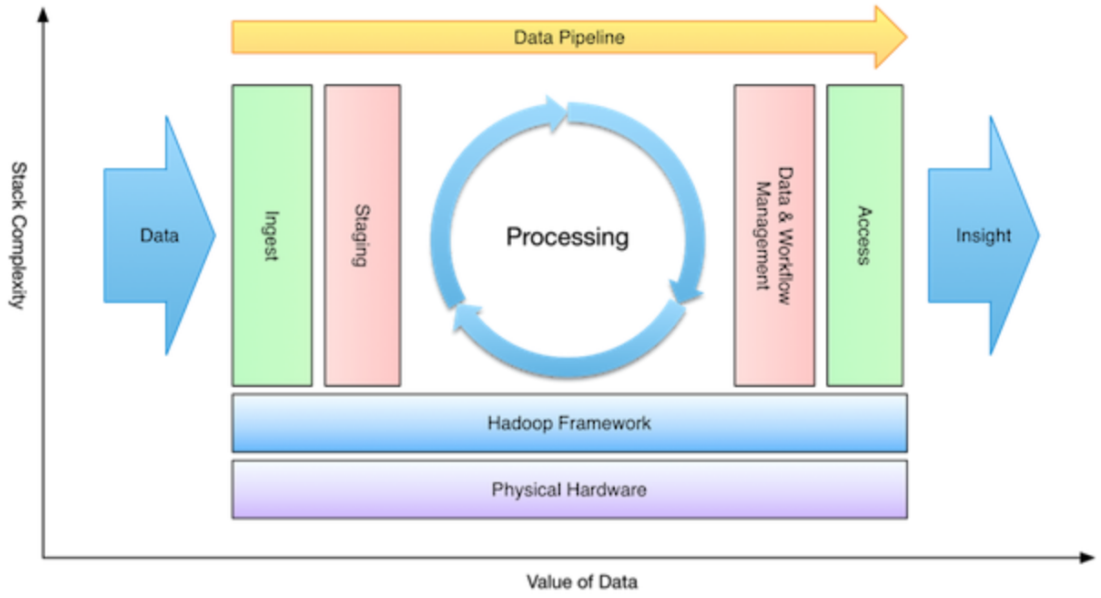

Рис. 4 – Стек работы с Большими данными. [15]

Для работы с Большими данными разработчиками систем создаются модели данных, содержательно связанные с реальным миром. Разработка адекватных моделей данных представляет собой сложную аналитическую задачу, выполняемую системными архитекторами и аналитиками. Модель данных позволяет создать математическую модель взаимодействий объектов реального мира и включает в себя описание структуры данных, методы манипуляции данными и аспекты сохранения целостности данных. Описание разработки моделей данных не является задачей настоящего руководства.

Для хранения данных используются распределенные системы различных типов. Это могут быть файловые системы, базы данных, журналы, механизмы доступа к общей виртуальной памяти. Большинство систем хранения ориентированы исключительно на работу с Большими данными, они имеют крайне ограниченное число функций (например, может отсутствовать возможность

не только модификации, но и удаления поступивших данных) что объясняется внутренней сложностью создания высокоэффективных распределенных систем. В конце текста приведены ссылки на несколько используемых в настоящий момент систем.

Для того, чтобы работа с данными происходила быстрее системы хранения и обработки данных распараллеливаются в кластере (cluster, группа компьютеров, объединенных сетью для выполнения единой задачи). Однако, согласно гипотезе Брюера невозможно обеспечить одновременную согласованность (непротиворечивость) данных, доступность данных и устойчивость системы к отделению отдельных узлов. Гипотеза доказана для транзакций типа ACID (Atomic, Consistent. Isolated, Durable) и известна под названием CAP теоремы (Consistency, Availability, Partition tolerance).

4.1. Прием данных (Data Ingestion)

Источники данных имеют различные параметры, такие как частоту поступления данных из источника, объём порции данных, скорость передачи данных, тип поступающих данных и их достоверность.

Для эффективного сбора данных необходимо установить источники данных. Это могут быть хранилища данных, поставщики агрегированных данных, API каких-либо датчиков, системные журналы, сгенерированный человеком контент в социальных сетях, в корпоративных информационных системах, геофизическая информация, научная информация, унаследованные данные из других систем. Источники данных определяют исходный формат данных.

Например, мы можем самостоятельно проводить погодные на территории аэропорта, использовать данные, поступающие с взлетающих и садящихся самолетов, закупить данные со спутников, пролетающих над аэропортом и у местной метеослужбы, а также найти их где-то в сети в другом месте. В общем случае для каждого источника необходимо создавать собственный сборщик (Data Crawler для сбора информации в сети и Data Acquisition для проведения измерений).

Прием данных заключается в начальной подготовке данных от источников с целью приведения данных к общему формату представления данных. Этот единый формат выбирается в соответствии с принятой моделью данных. Выполняются преобразования систем измерения, типов (типизация), верификация. Обработка данных содержательно не затрагивает имеющуюся в данных информацию, но может изменять ее представление (например, приводить координаты к единой системе координат, а значения к единой размерности).

4.2. Сбор данных (Data Staging)

Этап сбора данных характеризуется непосредственным взаимодействием с системами хранения данных. Устанавливается точка сбора, в которой собранные данные снабжаются локальными метаданными и помещаются в хранилище либо передаются для последующей обработки. Данные, по каким-либо причинам не прошедшие точку сбора, игнорируются.

Для структурированных данных проводится преобразование из исходного формата по заранее заданным алгоритмам. Это наиболее эффективная процедура в случае, если структура данных известна. Однако если данные представлены в двоичном виде, структура и связи между данными утеряны, то разработка алгоритмов и основанного на них программного обеспечения для обработки данных может оказаться крайне затруднительной.

Для полуструктурированных данных требуется интерпретация поступающих данных и использование программного обеспечения, умеющего работать с используемым языком описания данных. Существенным плюсом полуструктурированных данных является то, что в них зачастую содержатся не только сами данные, но метаданные в виде информации о связях между данными и способах их получения.

Разработка программного обеспечения для обработки полуструктурированных данных представляет собой достаточно сложную задачу. Однако имеется значительное количество готовых конвертеров, которые могут, например, извлечь данные из формата XML в сформированное табличное представление.

Наибольшего объема работ требует обработка неструктурированных данных. Для их перевода к заданному формату может потребоваться создание специального ПО, сложная ручная обработка, распознавание и выборочный ручной контроль.

На этапе сбора проводится контроль типов данных и может выполняться базовый контроль достоверности данных. Например, координаты молекул газа, содержащихся в какой-либо области, не могут лежать за пределами этой области, а скорости – существенно превышать скорость звука. Для того, чтобы избежать ошибок типизации, необходимо проверять правильно ли заданы единицы измерения. Например, в одном наборе данных высота может измеряться в километрах, а в другом – в футах. В этом случае необходимо произвести преобразование высоты в те единицы измерения, которые приняты в используемой модели.

При сборе данные систематизируются и снабжаются метаданными, хранимыми в связанных метаданных. При наличии большого количества источников данных может потребоваться управление сбором данных для того, чтобы сбалансировать объемы информации, поступающие из различных источников.

Собранные данные либо сохраняются в системах хранения, либо (в особенности, для потоковых данных) передаются для анализа в реальном времени.

4.3. Анализ данных (Analysis Layer)

Анализ данных, в отличии от сбора данных, использует информацию, содержащуюся в самих данных. Анализ может проводиться как в реальном времени, так и в пакетном режиме. Анализ данных составляет основную по трудоемкости задачу при работе с Большими данными.

Существует множество методик обработки данных: предиктивный анализ, запросы и отчетность, реконструкция по математической модели, трансляция, аналитическая обработка и другие. Методики используют специфические алгоритмы в зависимости от поставленных целей. Например, аналитическая обработка может являться анализом изображений, социальных сетей, географического местоположения, распознавания по признакам, текстовым анализом, статистической обработкой, анализом голоса, транскрибированием.

Алгоритмы анализа данных также, как и алгоритмы обработки данных, опираются на модель данных. При этом при анализе может быть использовано несколько моделей, задающих общий формат данных, но по-разному моделирующие содержательные процессы, данные о которых мы обрабатываем. При использовании при анализе методов искусственного интеллекта, в частности нейронных сетей, производится динамическое обучение моделей на различных наборах данных.

При анализе данных производится идентификация сущностей, описываемых данными на основании имеющейся в данных информации и используемых моделей. Сущностью анализа является аналитических механизм, использующий аналитические алгоритмы, управление моделями и идентификацию сущностей для получения новой содержательной информации, являющейся результатом анализа.

Для анализа данных также используются методы искусственного интеллекта на нейронных сетях, не рассматриваемые в данном учебном пособии.

4.4. Представление результатов (Consumption Layer)

Результаты анализа данных предоставляются на уровне потребления. Имеется несколько механизмов, позволяющих использовать результаты анализа больших данных.

- Мониторинг метаинформации.

Подсистема отображения в реальном времени существенных параметров работы системы, загруженности вычислителей, распределение задач в кластере, распределение информации в хранилищах, наличие свободного места

в хранилищах, поступление данных от источников, активности пользователей, отказов оборудования и тд.

- Мониторинг данных.

Подсистема отображения в реальном времени процессов приема, сбора и анализа данных, навигация по данным.

- Генерация отчетов, запросы к данным, представление данных в виде визуализации на дэшбордах (Dashboard), в формате PDF, инфографике, сводных таблицах и кратких справках
- Преобразование данных и экспорт в другие системы, интерфейс с BI-системами.

Вопросы для проверки

1. Какие уровни можно выделить в системах обработки Больших данных?
2. Зачем компьютеры объединяют в кластер?
3. В чём заключается приём данных от источников?
4. В чём может быть сложность обработки структурированных данных?
5. Приведите пример полуструктурированных данных.
6. Какие задачи может решать анализ Больших данных?
7. В каких целях используется слой/уровень компонентов системы “Прием данных”?
8. В каких целях используется слой/уровень компонентов системы “Сбор данных”?
9. В каких целях используется слой/уровень компонентов системы “Анализ данных”?
10. В каких целях используется слой/уровень компонентов системы “Представление результата”?

5. Параллельные алгоритмы для работы с данными

5.1. Операторы Map и Reduce

Для параллельной обработки данных в интерфейсе MPI (Message Passing Interface), являющимся распространённым стандартом для обмена данными при организации параллельных вычислений, была предложена ныне широко используемая во множестве реализаций парадигма параллельной обработки наборов данных при помощи использования операторов Map и Reduce.

5.2. Оператор Reduce (свертка)

Свертка в простейшем случае получает на входе функцию от двух параметров из набора данных, состоящего из элементов одного типа. На выходе она возвращает один элемент такого же типа. В общем случае от функции требуется коммутативность и ассоциативность на множестве определения. В этом случае порядок вычислений несущественен, алгоритм позволяет эффективно распараллеливаться, так как есть возможность выбрать произвольные пары элементов набора данных, от выбранных пар при помощи переданной в Reduce функции параллельно вычислить значения и исходные пары элементов заменить на вычисленные значения. Алгоритмы, в которых необходимая нам функция свертки коммутативна и ассоциативна, высокоэффективна при обработке Большого неупорядоченного (или упорядоченного, для вычисления это несущественно) набора данных.

Для ассоциативной, но некоммутативной функции свертка определена для набора данных, элементы которого упорядочены. В этом случае мы можем разбить исходный набор на несколько упорядоченных между собой последовательных поднаборов, например, по числу имеющихся вычислителей в кластере. С математической точки зрения эта процедура соответствует расстановке скобок. Затем, получив по одном элементу данных из каждого поднабора, мы имеем промежуточный упорядоченный набор данных, и выполняем свертку над ним и далее рекурсивно до получения итогового значения.

В более сложном случае мы хотим при работе Reduce получить данные другого типа, например, на входе мы имеем набор с массой и скоростями молекул, а на выходе нам нужно получить суммарную массу и импульс (чтобы узнать среднюю скорость ветра разделив импульс на массу). Для решения такой задачи функция, передаваемая в свертку, существенно видоизменяется следующим образом: она получает один параметр результирующего типа (называемый аккумулятором) и один параметр исходного типа (итератор). В самом начале аккумулятор имеет некоторое начальное значение. Для нашего примера

с молекулами передаваемое в свертку начальное значение – это структура из четырёх элементов: масса равна нулю и три компонента (по координатам *x*, *y* и *z*) импульса тоже равны нулю.

Затем элементы данных перебираются и производятся вычисления, определённые функцией. В нашем примере масса молекулы, переданной в итераторе, прибавляется к массе в аккумуляторе, далее вычисляется импульс молекулы (вектор из произведений массы на компоненты скорости) и прибавляется к компонентам импульса аккумулятора.

Разбив исходный набор данных на поднаборы мы для каждого вычислим массу и импульс. Однако далее у нас появляется проблема: имеющаяся функция умеет добавлять к данным имеющим типа аккумулятора (масса, импульс) данные имеющие тип итератора (масса, скорость). Чтобы произвести итоговые вычисления, нам необходимо произвести переопределение функции: в том случае, если передаётся два параметра, имеющих тип аккумулятора, то функция начинает вести себя совсем по-другому. А именно складывает массы и импульсы. Иначе говоря, по сути у нас появляются две функции: одна работает с мастер-данными, а другая с промежуточными данными. Таким образом, мы можем провести эффективное распараллеливание. Обратим внимание что этот метод также допускает работу с некоммутирующими, но упорядоченными данными в случае ассоциативных функций.

Иногда для удобства вычислений, кроме описанной выше прямой (левой) свертки, при которой функция вычисляется от первых двух элементов (либо от начального значения и первого элемента данных), затем от полученного аккумулятора и следующего элемента данных (итератора) и так далее до завершения вычисления, вводят также правую (обратную) свертку. При обратной свертке функция вычисляется сначала от начального значения (если оно задано) и последнего элемента, затем от аккумулятора и предпоследнего элемента, и так далее, пока не дойдём до вычисления от аккумулятора первого элемента.

Эффективное распараллеливание вычисления свертки от некоммутативных неассоциативных функций в общем случае невозможно. Заметим, что Гвидо ван Россум (BDFL, Benevolent Dictator For Life, великодушный пожизненный диктатор языка Python) сознательно перевел свертку из общего пространства имен в модуль functools поскольку в большинстве приложений ее использовали либо для таких тривиальных вещей, как суммирование массивов, либо для получения совершенно непонятного нечитаемого кода.

5.3. Оператор Map

Этот оператор известен во многих языках программирования под различными именами, кроме Map используется также названия Select и Transform.

В простейшем случае Map создаёт из переданной ему функции и упорядоченного набора данных другой упорядоченный набор данных, в котором каждый элемент является значением функции от соответствующего элемента исходного набора данных.

Существуют расширения оператора Map на несколько наборов данных, при этом функция должна быть определена от соответствующего количества элементов. Возвращается набор данных, элементы которого являются значениями функции от соответствующих наборов. При этом есть два способа определить поведение расширенного оператора Map в случае различия в размерах передаваемых наборов данных – либо результирующий набор имеет размер равный размеру наименьшего набора из передаваемых, либо размеру максимального, при этом функции передаются неопределённые значения.

5.4. Лямбда-архитектура

Архитектура, позволяющая проводить работу с данными при помощи эквивалентных алгоритмов одновременно в пакетном режиме и в реальном времени. Аналитика в реальном времени может быть неточной, выполняется в оперативной памяти, но предоставляются быстро. Расчеты в пакетном режиме обеспечивают хранение полученных результатов и выдает достоверные данные, но выполняется долго.

Вопросы для проверки

1. Приведите пример ассоциативной, но некоммутативной функции, определённой на множестве целых чисел
2. Приведите пример коммутативной, но неассоциативной функции, определённой на множестве целых чисел.
3. Какой тип данных может возвращать оператор свертки, примененный к начальному значению, функции и набору данных?
4. В каком случае свертка данных неэффективна?
5. Какой тип данных возвращает оператор Map, примененный к функции и набору данных?
6. Какая обработка данных обеспечивает более высокую достоверность результатов – в реальном времени или в пакетном режиме?

6. Программные платформы и системы для Больших данных

В настоящее время используется значительное количество платформ и систем Больших данных. Системы обработки больших данных являются фреймворками, то есть каркасами, для использования которых необходимо состыковать их с другими фреймворками, прикладным программным обеспечением пользователя и системой хранения данных.

В аналитическом отчете Big Data Analytics Market Study, 2017 Edition [18] приводится следующая диаграмма инфраструктур Больших данных, внедренных на предприятиях, представленная в разрезе размеров предприятий (рис. 5).

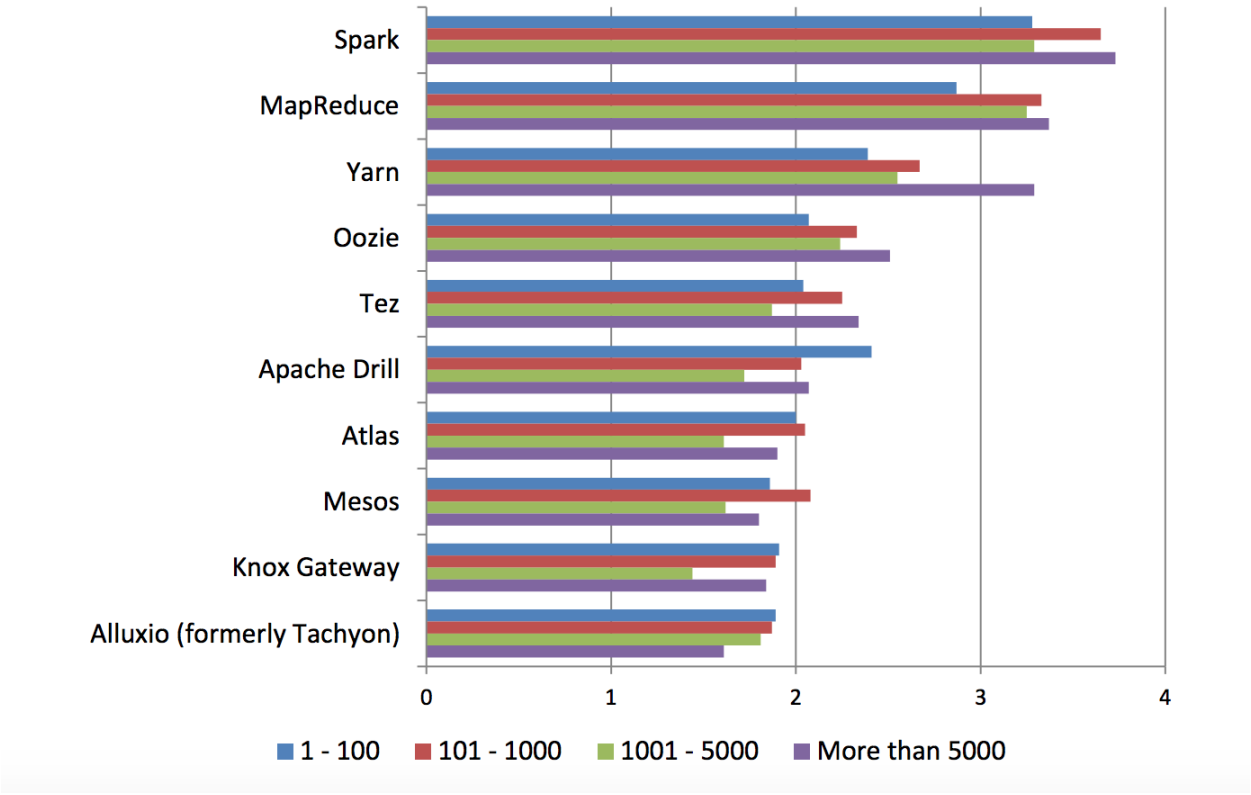

Рис. 5 – Инфраструктуры Больших данных в разрезе размера предприятий [18]

Большинство используемых платформ доступно по лицензии Apache 2.0 и расположены на сайте фонда программного обеспечения Apache.

6.1. Системы управления потоками данных

Flume

https://flume.apache.org/
Разработана в 2017 году.
Система управления потоками данных.

Apache Kafka

https://kafka.apache.org/
Разработан в LinkedIn в 2011 году.
Масштабируемый отказоустойчивый журнал коммитов.

Niagara Files (NiFi)

https://nifi.apache.org/
Разработан в NSA в 2014.
Система управления потоками данных.

6.2. Системы хранения Больших данных

HDFS (Hadoop Distributed File System)

https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html
Файловая система, входящая в проект Hadoop.

OpenStack Swift

https://docs.openstack.org/swift/latest/
Платформа хранения, входящая в проект OpenStack.

Cassandra

http://cassandra.apache.org/
Табличная СУБД, написана на языке Java.

HBase

https://hbase.apache.org/
Табличная СУБД, написана на языке Java.

Apache Drill

https://drill.apache.org/

SQL-интерфейс к NoSQL базам данным (HBase, MongoDB, MapR-DB, HDFS, MapR-FS, Amazon S3, Azure Blob Storage, Google Cloud Storage, Swift, NAS and local files).

6.3. Платформы Больших данных

Hadoop

http://hadoop.apache.org/

Предоставляет интерфейс к Java (а через коннекторы и к другим языкам), свободно распространяется под лицензиями Apache License 2.0 и GNU GPL пакет программного обеспечения, состоящий из управляющего модуля Hadoop Common, распределенной файловой системы HDFS, планировщика заданий YARN и вычислительной платформы Hadoop MapReduce. Развивается с 2005 года.

Spark

https://spark.apache.org/

Предоставляет интерфейсы к Scala, Java, Python и R, распространяется под лицензией Apache License 2.0. Вычислительная платформа, развивающаяся с 2014 года.

Elasticsearch

https://www.elastic.co/products/elasticsearch

Совместно с системой сбора Logstash и платформой аналитики Kibana составляют интегрированную систему сбора, хранения, поиска и аналитики данных.

Solr

http://lucene.apache.org/solr/

Еще одна система поиска и анализа в базах Больших данных.

Hortonworks Data Platform (HDP)

https://hortonworks.com/products/data-platforms/hdp/

Платформа управления данными, включающая HDFS, Hadoop, HBase, HCatalog, Pig, Hive, Oozie, Zookeper, Ambari, WebHDFS, TalentOS, Sqoop, Flume, и Mahout.

Windows Azure HDInsight

https://azure.microsoft.com/en-ca/services/hdinsight/

Система от Microsoft для развёртывания hadoop на Azure.

6.4. Обработка данных в реальном времени

Apache Storm

https://storm.apache.org/

По предварительно созданной топологии вычислений управляющий узел создаёт в кластере спауты (spout), генерирущие кортежи (tuple) ключ-значение, и болты (bolt), выполняющие обработку. Любые языки (JVM, Ruby, Python, Perl).

Apache Spark

https://spark.apache.org/streaming/

Потоки разбиваются на пакеты DStream (дискретизированный поток) состоящие из нескольких RRD (resilient distributed dataset, отказоустойчивый распределенный набор данных). Обработка при помощи технологии sliding window (скользящих окон). Поддерживает Scala, Java и Python.

Apache Samza

http://samza.apache.org/

Распределение сообщений на разделы, каждый из которых независимо обрабатывается по мере их получения. Поддерживает JVM: Scala и Java.

6.5. Системы управления Большими данными

Ambari

https://ambari.apache.org/

webUI провизионинга, управления и мониторинга кластера Hadoop.

Atlas

https://atlas.apache.org/

Система обмена метаданными для Hadoop-стека.

Cloudbreak

https://hortonworks.com/open-source/cloudbreak/

Система развёртывания платформы Hortonworks в коммерческом облаке.

Knox

https://knox.apache.org/

Система аутентификации и авторизации для Hadoop-кластера.

Ranger

https://ranger.apache.org/

Система обеспечения безопасности данных на платформе Hadoop.

ZooKeeper

https://zookeeper.apache.org/

Централизованный сервис для управления конфигурационной информацией, именованием, распределенной синхронизацией и групповыми сервисами.

6.6. Аналитические платформы

RapidMiner

https://rapidminer.com/

Система прогнозной аналитики, поддерживает глубинный анализ, проверку, оптимизацию и визуализацию, имеет графический интерфейс программирования.

IBM SPSS Modeler

https://www.ibm.com/products/spss-modeler

Коммерческая аналитическая система автоматизированного моделирования, геопространственной аналитики, анализа текстовой информации. Плохо подходит для больших объёмов информации.

KNIME

https://www.knime.com/

Бесплатная система анализа данных, имеющая глубинный анализ, веб-анализ, обработку изображений, анализ социальных сетей, обработку текстов.

Qlik Analytics Platform

https://www.qlik.com/us/products/qlik-analytics-platform

Система визуальной аналитики, предоставляет доступ к ассоциативной машине индексации данных QIX Engine.

STATISTICA Data Miner

http://statsoft.ru/products/STATISTICA_Data_Miner/data-mining-tools.php

Система от российского производителя

IBM Watson Analytics

https://www.ibm.com/watson-analytics

Мощная облачная система от IBM (применяемая в том числе twitter'ом)

Dell EMC Analytic Insights Module

https://www.dellemc.com/en-us/big-data/index.htm

Многокомпонентная система от Dell EMC

SAP Predictive Analytics
https://www.sap.com/products/predictive-analytics.html
Система от мирового лидера ERP, интегрируется с SAP HANA

Oracle Big Data Preparation
https://cloud.oracle.com/bigdatapreparation
Облачное решение от мирового лидера баз данных

Вопросы для проверки

1. Какие языки программирования используются для работы с фреймворками данных?
2. Позволяет ли лицензия Apache 2.0, под которой выпущены некоторые фреймворки, вносить собственные исправления в код программного обеспечения?
3. Перечислите несколько фреймворков, обеспечивающих обработку данных в реальном времени.
4. Перечислите несколько фреймворков, обеспечивающих аналитическую обработку данных.
5. Перечислите несколько фреймворков, обеспечивающих хранение данных.
6. Перечислите несколько фреймворков, обеспечивающих управление потоками данных.

7. Оборудование для обработки Больших данных

Комплект оборудования для обработки Больших данных монтируется в ЦОД. Основными компонентами системы являются система управления, вычислительные ресурсы, система хранения данных, локальная сеть. Электропитание, мониторинг, доступ к интернету и другие внешние ресурсы предоставляют центры обработки данных (ЦОД).

Система управления предназначена для общего управления системой, внешнего доступа, обеспечения аутентификации, авторизации и аккаунтинга и представления результатов пользователям. Выполняется на базе обычной серверной платформы, специфических требований не имеет.

На рис. 6 представлен кластер Hadoop от Yahoo 10 летней давности. [16]

Рис. 6 – Кластер Hadoop от Yahoo 10 летней давности

Вычислительные ресурсы кластера состоят из узлов (nodes), основными параметрами которых является объем оперативной памяти с контролем четности (ECC) и максимальное количество ядер CPU. Расчётным параметрами является размер оперативной памяти на ядро и скорость работы процессора. Высокая отказоустойчивость узлов желательна, но в целом не требуется, некоторое количество отказов является нормальным и компенсируется при помощи программного обеспечения – отказавший узел автоматически выводится из эксплуатации, и его работа перераспределяется на другие узлы.

В некоторых случаях для обработки Больших данных, особенно при использовании нейронных сетей, используют вычислители на базе GPU, аналогичные используемым для HPC.

Распределённая система хранения данных состоит из дисковых полок, обеспечивающих максимально быстрый доступ к данным. Резервирование выполняется при помощи программного обеспечения систем хранения и в общем

случае не требуется. Зачастую также используются компьютеры, в которых подключено большое количество локальных дисков.

Рис. 7 – Центр хранения данных производства Titan Power [21]

Сетевая инфраструктура связывает вычислительные ресурсы, систему хранения и систему управления в единое целое. Основное требование к локальной сети – низкие задержки. При большом количестве узлов используются сетевые коммутаторы, начинающие передачу пакета данных сразу после обработки заголовка пакета данных. При малом количестве узлов распространено прямое соединение компьютеров по схеме гиперкуб, где размерность куба соответствуют числу портов на интерфейсных платах. Ранее массово использовались сети на базе различных вариантов интерфейса Infiniband, сейчас, в основном, используется 100 Gigabit Ethernet. Сравнительно недавно появился 200 и 400 Gigabit Ethernet[13].

[13] ConnectX®-6 EN Dual-Port Adapter Supporting 200Gb/s Ethernet. http://www.mellanox.com/page/products_dyn?product_family=266&mtag=connectx_6_en_card

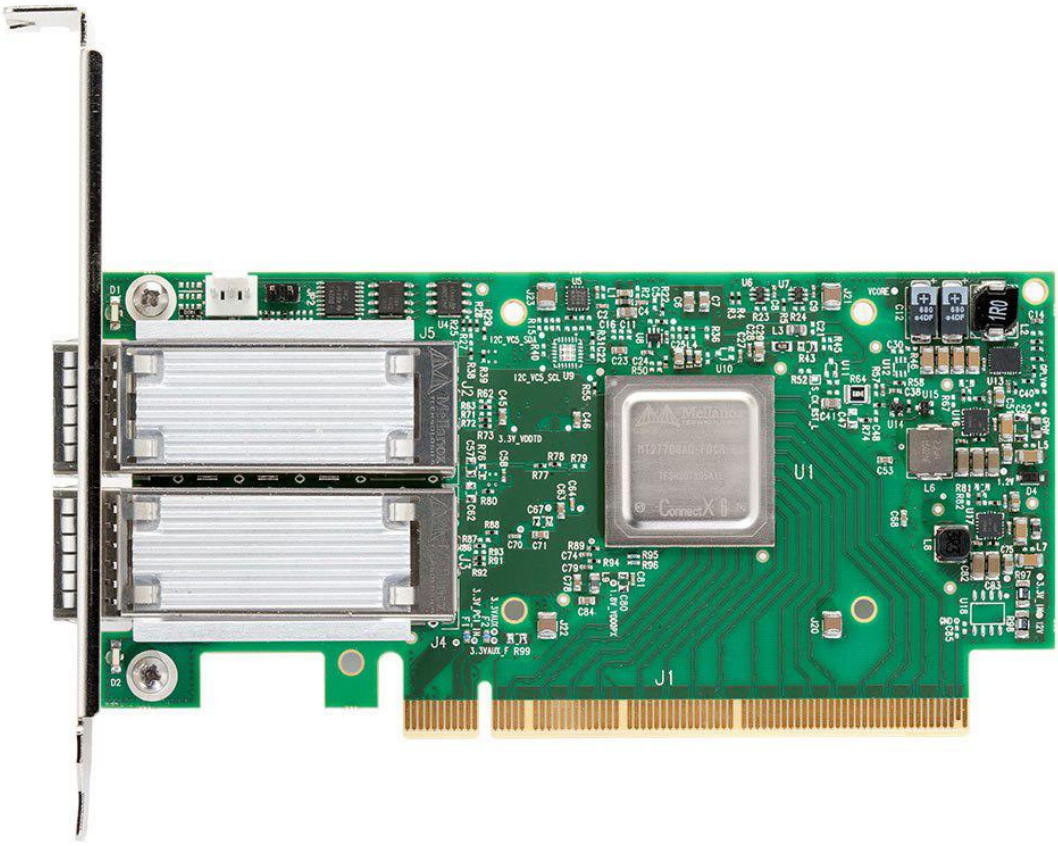

Рис. 8 – ConnectX®-6 EN двухпортовой сетевой адаптер 200Gb/s Ethernet [22]

Вопросы для проверки

1. Где монтируется оборудование для обработки Больших данных?
2. Из каких компонентов состоит оборудование для обработки Больших данных?
3. Какие параметры существенны для вычислительных узлов кластера?
4. Какие параметры существенны для системы хранения данных?
5. Какие параметры существенны для сетевой инфраструктуры?
6. Сколько максимально узлов может иметь кластер, использующий архитектуру подключения узлов гиперкуб, при наличии четырех быстродействующих сетевых портов в каждом узле?

8. Центры обработки Больших данных

Большие данные обрабатывают в центрах обработки данных (ЦОД). В 2005 году ассоциация телекоммуникационной индустрии (TIA) и инженерный комитет по телекоммуникационным кабельным системам (TR-42) выпустили стандарт ANSI/TIA-942 на телекоммуникационную инфраструктуру центров обработки данных (Telecommunications Infrastructure Standards for Data Centres), который регламентирует требования к инфраструктуре ЦОД.

Текущая версия стандарта TIA-942-A устанавливает требования к сетевой инфраструктуре, организации электроснабжения, файловому хранению, сохранению резервных копий и архивированию, отказоустойчивости систем, контролю доступа к сети и сетевой безопасности, управлению базами данных, веб-хостингу, хостингу аппликаций, распределению контента, управлению климатом, защите от внешних физических разрушений таких как пожару, наводнению, погодным явлением, управлению энергопитанием.

При построении датацентров также применяются стандарты ANSI/TIA/EIA-568-B.1, ANSI/TIA/EIA-568-B.2, ANSI/TIA/EIA-568-B.3, ANSI/TIA-569-B, ANSI/TIA/EIA-606-A, ANSI/TIA/EIA-J-STD-607, ANSI/TIA-758-A, национальные правила по безопасному устройству электроустановок NESC IEEE C2, национальный электрический код NEC NFPA 70 (в РФ их роль играют правила устройства электроустановок 6 и 7 редакции, ПУЭ-6 и ПУЭ-7), защита ИТ-оборудования, стандарт NFPA 75, инженерные требования к универсальной телекоммуникационной стойке ANSI T1.336, рекомендованные правила запитки и заземления электронного оборудования IEEE 1100, рекомендованные правила для систем аварийного и резервного энергоснабжения промышленного и коммерческого применения IEEE 446, технические требования Telcordia GR-63-CORE (NEBS) и GR-139-CORE и другие отраслевые стандарты.

Проектирование, строительство, монтаж и ввод в эксплуатацию ЦОД осуществляется высококвалифицированными специалистами, имеющими соответствующее профильное образование и снабженными специализированным поверенным контрольно-измерительным оборудованием.

Стандарт регламентирует четыре уровня ЦОД, от первого уровня, практически не обеспечивающего надежность, до четвёртого.

Уровень 1 — базовый

- Коэффициент постоянной готовности 99,671%.
- Подвержен нарушениям хода работы от плановых или внеплановых действий.
- Имеет один канал распределения электропитания и охлаждения без резервирования.
- Может иметь или не иметь фальшпол, ИБП или генератор.
- Развертывается за 3 месяца.
- Годовое время простоя – 28,8 часов.

- Полностью останавливается для выполнения работ по планово-предупредительному обслуживанию и профилактическому ремонту.

Уровень 2 — с резервированием

- Коэффициент постоянной готовности 99,741%.
- Менее подвержен нарушениям работы от плановых и от внеплановых действий.
- Имеет один канал распределения электропитания и охлаждения, но с резервированными компонентами (N+1).
- Имеет фальшпол, ИБП и генератор.
- Время развертывания – от 3 до 6 месяцев.
- Годовое время простоя – 22,0 часов.
- Техническое обслуживание и ремонт канала электропитания и других частей инфраструктуры объекта требует остановки процесса обработки данных.

Уровень 3 — с возможностью параллельного проведения ремонтных работ

- Коэффициент постоянной готовности 99,982%.
- Плановые действия не нарушает работы, инцидент может привести к нарушению.
- Имеет несколько каналов распределения электропитания и охлаждения, но лишь один из них активен; имеет резервированные компоненты (N+1).
- Время развертывания – от 15 до 20 месяцев.
- Годовое время простоя – 1,6 часов.
- Имеет достаточно мощности и распределительных возможностей для того, чтобы при загруженности одного канала можно было обслуживать или тестировать другой.

Уровень 4 — отказоустойчивый дата-центр: коэффициент постоянной готовности 99,995%

- Плановые действия не нарушают работы, толерантен к одному инциденту наихудшего свойства без последствий для критически важной нагрузки.
- Имеет несколько активных каналов распределения нагрузки и охлаждения с резервными компонентами (2 (N+1), т.е. 2 ИБП с избыточностью N+1 каждый).
- Время развертывания – от 15 до 20 месяцев.
- Годовое время простоя – 0,4 часа.

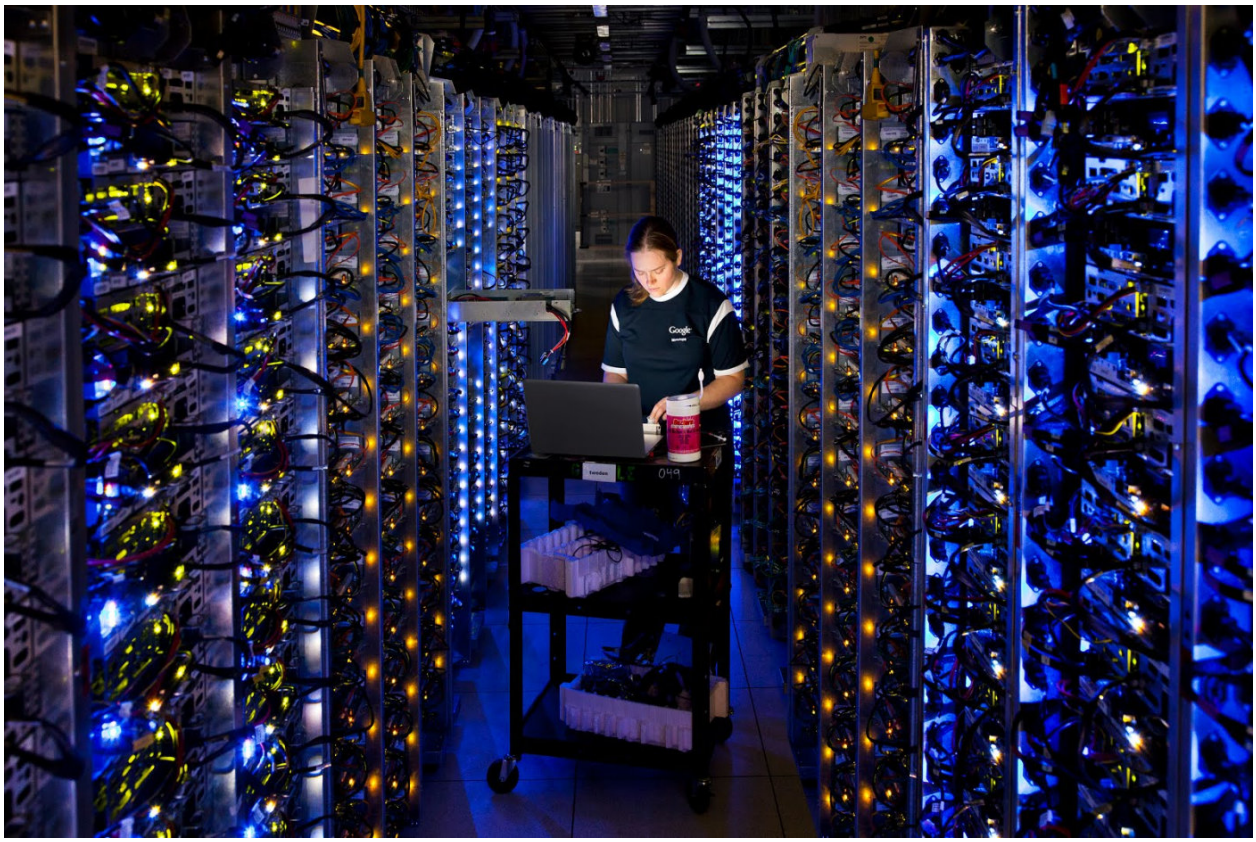

Рис. 9 – Инженер обслуживает оборудование в центре обработки данных [23]

Вопросы для проверки

1. Какое оборудование требуется для обработки Больших данных?
2. Центр обработки данных какого уровня обеспечивает максимальную надежность?
3. Центр обработки данных какого уровня обеспечивает резервирование?
4. Центр обработки данных какого уровня позволяет проводить обслуживание оборудования одновременно с обработкой данных?
5. Как много времени необходимо для создания центра обработки данных "под ключ"?

Заключение

В данном учебном пособии были рассмотрены основные понятия и технологии, связанные с Большими данными. Обработка Больших данных является сложным технологическим процессом, требующим глубоких программно-инженерных знаний для разработки модели данных, выбора соответствующих программно-аппаратных средств и оценки совокупной стоимости управления данными. Во многих случаях обработка данных может быть проведена достаточно скромными средствами, при помощи аренды систем хранения и обработки в облачной среде, в других случаях требуется аренда или даже строительство собственного ЦОД и установка собственного оборудования, в третьих случаях – стоимость работы с данными может превысить доход от их обработки и обработка данных своими силами нецелесообразна, однако может быть выполнена при помощи подрядчика.

Библиографический список

1. Дивакар Майсор, Шрикант Кхупат, и Швета Джайн
   Архитектура и шаблоны больших данных. [Электронный ресурс] Режим доступа: https://www.ibm.com/developerworks/ru/library/bd-archpatterns1/index.html
2. Davy Cielen, Arno D. B. Meysman, and Mohamed Ali. Introducing Data Science. Big data, machine learning, and more, using Python tools https://www.manning.com/books/introducing-data-science
3. Hilbert, M. (2016). Big Data for Development: A Review of Promises and Challenges. Development Policy Review, 34(1), 135–174. http://doi.org/10.1111/dpr.12142
4. Environmental Protection Agency. Subpart A—General Provisions. https://www.gpo.gov/fdsys/pkg/CFR-2011-title40-vol1/pdf/CFR-2011-title40-vol1-part3-subpartA.pdf
5. Seth Gilbert and Nancy Lynch. Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-Tolerant Web Services. https://doi.org/10.1145/564585.564601
6. Steven J. Plimpton and Karen D. Devine (2011), MapReduce in MPI for Large-scale Graph Algorithms, https://doi.org/10.1016/j.parco.2011.02.004
7. Инфосфера общественных наук России : монография / А. Б. Антопольский, Д. В. Ефременко ; под ред. В. А. Цветковой. – М. ; Берлин : Директ-Медиа, 2017. – 676 с.
8. Федеральный закон от 27 июля 2006 г. № 149-ФЗ "Об информации, информационных технологиях и о защите информации" с изменениями и дополнениями от: 27 июля 2010 г., 6 апреля, 21 июля 2011 г., 28 июля 2012 г., 5 апреля, 7 июня, 2 июля, 28 декабря 2013 г., 5 мая, 21 июля 2014 г. [Электронный ресурс] Режим доступа: http://base.garant.ru/12148555/
9. Hari Shreedharan, 2014, Using Flume: Flexible, Scalable, and Reliable Data Streaming
   ISBN-13: 978-1449368302
10. Neha Narkhede, Gwen Shapira, Todd Palino, 2017
    Kafka: The Definitive Guide: Real-Time Data and Stream Processing at Scale
    ISBN-13: 978-1491936160
11. Д.И. Муромцев. Онтологический инжиниринг знаний в системе Protégé. – СПб: СПб ГУ ИТМО, 2007. – 62 с
12. Data is the new oil in the digital economy. https://www.wired.com/insights/2014/07/data-new-oil-digital-economy/
13. Data. Cambridge dictionary. https://dictionary.cambridge.org/dictionary/english/data

14. M.Rouse. Data Life Cycle. https://whatis.techtarget.com/definition/data-life-cycle
15. L. George. Getting Started With Big Data Architecture. http://blog.cloudera.com/blog/2014/09/getting-started-with-big-data-architecture/
16. Noll M.G. Running Hadoop on Ubuntu Linux (Multi-Node Cluster). http://www.michael-noll.com/tutorials/running-hadoop-on-ubuntu-linux-multi-node-cluster/
17. 7 phases for Data Life Cycle. https://www.bloomberg.com/professional/blog/7-phases-of-a-data-life-cycle/
18. Dresner Advisory Services, LLC. Big Data Analytics Market Study 2016. https://www.microstrategy.com/getmedia/cd052225-be60-49fd-ab1c-4984ebc3cde9/Dresner-Report-Big_Data_Analytic_Market_Study-WisdomofCrowdsSeries-2017
19. Приказ 11 февраля 2013 г. N 17 "При выводе из эксплуатации машинных носителей информации, на которых осуществлялись хранение и обработка информации, осуществляется физическое уничтожение этих машинных носителей информации": https://fstec.ru/normativnye-pravovye-akty-tzi/110-deyatelnost/tekushchaya/tekhnicheskaya-zashchita-informatsii/normativnye-pravovye-akty/prikazy/703-prikaz-fstek-rossii-ot-11-fevralya-2013-g-n-17
20. Metadata Life Cycle. http://metadata.teldap.tw/design/lifecycle_eng.htm
21. Increasing Rate of Data Production Prompts Google to Rethink Data Center Storage. http://www.titanpower.com/blog/increasing-rate-of-data-production-prompts-google-to-rethink-data-center-storage/
22. Mellanox Scale-Out SN3000 Ethernet Switch Series. http://www.mellanox.com/page/products_dyn?product_family=280&mtag=sn3000_label
23. Take a 360-degree video tour of Google's Oregon data center. https://www.engadget.com/2016/03/24/google-360-video-tour-data-center/

Список информационных источников для самостоятельной работы

1. Blackwell M., Sen M. Large Datasets and You: http://www.mattblackwell.org/files/papers/bigdata.pdf
2. Ross N. FasteR! HigheR! StrongeR! – A Guide to Speeding Up R Code for Busy People: http://www.noamross.net/blog/2013/4/25/faster-talk.html
3. Ryan R. Rosario. Taking R to the Limit, Part II: Working with Large Datasets: http://www.bytemining.com/wp-content/uploads/2010/08/r_hpc_II.pdf
4. Big Data Specialization. https://www.coursera.org/specializations/big-data
5. Mining Massive Datasets. Online course.https://online.stanford.edu/course/mining-massive-datasets-self-paced

6. G. Press. 12 Big Data Definitions. What's yours? https://www.forbes.com/sites/gilpress/2014/09/03/12-big-data-definitions-whats-yours/
7. Big Data awesome list. https://github.com/onurakpolat/awesome-bigdata
8. Keuper F., Schmidt D., Schomann M. Smart Big Data Management. ISBN-10: 3832537686. 2014. https://www.amazon.com/Smart-Data-Management-Frank-Keuper/dp/3832537686
9. Mayer-Schönberger V. Big Data: A Revolution That Will Transform How We Live, Work, and Think. 2013. ISBN-10: 1848547927 https://www.amazon.com/Big-Data-Revolution-Transform-Think/dp/1848547927/
10. Smith M.D., Telang R. Streaming, Sharing, Stealing: Big Data and the Future of Entertainment (MIT Press). 2016. https://www.amazon.com/Streaming-Sharing-Stealing-Future-Entertainment/dp/0262034794/
11. Karimi H.A. Big Data: Techniques and Technologies in Geoinformatics. 2014. ISBN-10: 1138073199 https://www.amazon.com/Big-Data-Techniques-Technologies-Geoinformatics-ebook/dp/B00HZNQKMM/
12. Marz N., Warren J. Big Data: Principles and best practices of scalable realtime data systems. 2015. ISBN-10: 1617290343 https://www.amazon.com/Big-Data-Principles-practices-scalable/dp/1617290343/
13. Separation of storage and compute in BigQuery. https://cloud.google.com/blog/big-data/2017/11/separation-of-storage-and-compute-in-bigquery
14. Mayer-Schonberger V., Ramge T. Reinventing Capitalism in the Age of Big Data. 2018. ISBN-10: 046509368X
15. https://www.amazon.com/Reinventing-Capitalism-Age-Big-Data/dp/046509368X/
16. Bahga A., Madisetti V. Big Data Science & Analytics: A Hands-On Approach. 2016.ISBN-10: 0996025537. https://www.amazon.com/Big-Data-Science-Analytics-Hands/dp/0996025537/
17. Marr B. Big Data: Using SMART Big Data, Analytics and Metrics To Make Better Decisions and Improve Performance. 2015. ISBN-10: 1118965833. https://www.amazon.com/Big-Data-Analytics-Decisions-Performance/dp/1118965833/
18. Marr B. Data Strategy: How to Profit from a World of Big Data, Analytics and the Internet of Things. 2017.ISBN-10: 074947985X. https://www.amazon.com/Data-Strategy-Profit-Analytics-Internet/dp/074947985X/
19. Jones H. Data Analytics: An Essential Beginner's Guide To Data Mining, Data Collection, Big Data Analytics For Business, And Business Intelligence Concepts. 2018. ISBN-10: 1985097974.

20. https://www.amazon.com/Data-Analytics-Essential-Collection-Intelligence/dp/1985097974/
21. Sawchik T. Big Data Baseball: Math, Miracles, and the End of a 20-Year Losing Streak. 2015. ISBN-10: 1250063507. https://www.amazon.com/Big-Data-Baseball-Miracles-20-Year/dp/1250063507/
22. Ferguson A.G. The Rise of Big Data Policing: Surveillance, Race, and the Future of Law Enforcement. 2017. ISBN-10: 1479892823. https://www.amazon.com/Rise-Big-Data-Policing-Surveillance/dp/1479892823/
23. O'Neil C. Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy. 2016. ISBN-10: 0553418815. https://www.amazon.com/Weapons-Math-Destruction-Increases-Inequality/dp/0553418815/
24. Tenner E. The Efficiency Paradox: What Big Data Can't Do. 2018. ISBN-10: 1400041392. https://www.amazon.com/Efficiency-Paradox-What-Data-Cant/dp/1400041392/
25. Lei Zeng M., Qin J. Metadata. 2016. ISBN-10: 1555709656 https://www.amazon.com/Metadata-Second-Marcia-Lei-Zeng/dp/1555709656/
26. Gartner R. Metadata: Shaping Knowledge from Antiquity to the Semantic Web. 2016. ISBN-10: 3319408917
27. https://www.amazon.com/Metadata-Shaping-Knowledge-Antiquity-Semantic/dp/3319408917/
28. Baca M. Introduction to Metadata. 2016. ISBN-10: 1606064797 https://www.amazon.com/Introduction-Metadata-Third-Murtha-Baca/dp/1606064797/

Сведения об истории кафедры

УНИВЕРСИТЕТ ИТМО

Миссия университета – генерация передовых знаний, внедрение инновационных разработок и подготовка элитных кадров, способных действовать в условиях быстро меняющегося мира и обеспечивать опережающее развитие науки, технологий и других областей для содействия решению актуальных задач.

КАФЕДРА ИНФОРМАТИКИ И ПРИКЛАДНОЙ МАТЕМАТИКИ

В 1976 году из кафедры Вычислительной техники выделяется кафедра Прикладной математики, на которую возлагается задача по подготовке специалистов в области программирования и методов вычислений. Кафедру возглавляет д.т.н., профессор Немолочнов О.Ф., работающий в области создания систем автоиатизации проектирования ЭВМ.

На кафедре работают д.т.н., проф. Цейтлин Я.М., специализирующийся в области цифровой фильтрации, и д.т.н., профессор Крыжановский И.И., специализирующийся в области машинного эксперимента.

Сотрудники кафедры проводят большую работу по организации учебного процесса – разрабатываются новые учебные программы и циклы лабораторных работ, ориентированные на привитие студентам всех специальностей навыков практической работы на ЭВМ. Параллельно с этим кафедра организует подготовку преподавателей и сотрудников института в области программирования и применения средств ВТ в учебном процессе. В институте создается студенческий вычислительный зал и методическое руководство этим залом возлагается на кафедру ИПМ. Большой вклад в эту работу внесли доценты кафедры Голованевский Г.Л., Кармазиненко В.В., Троицкая М.П., Шипилов П.А.

После смерти профессора Я.М.Цейтлина и перехода профессора И.И.Крыжановского на заведование кафедрой в академию ГВФ, а доцента Г.А.Петухова на заведование кафедрой МАП, основным научным направлением кафедры стало создание систем автоматизированного контроля цифровой аппаратуры. Разрабатываются методы автоматического построения контролирующих тестов (доц. Усвятский А.Е., доц. Звягин В.Ф.), создаются

системы цехового контроля схем ЭВМ (доц. Блохин В.Н., доц. Голованевский Г.Л.), исследуются методы моделирования неисправностей в цифровых схемах и методы оценки качества спроектированных тестов (доц. Кукушкин Б.А., доц. Голыничев В.Н.), проводятся работы по моделированию больших схем (доц. Зыков А.Г.). В это время кафедра активно сотрудничает с многими организациями, такими как НИЦЭВТ в Москве, НПО Электроавтоматика в Ленинграде, КБЭ в Харькове, завод САМ в Минске. Благодаря работам, которые кафедра проводит совместно с кафедрами МАП проф. Петухов Г.А.) и ТОП (проф. Родионов С.А.), удается организовать Вычислительный центр, оснащенный современной техникой и обслуживающий все подразделения института.

Кафедра принимает активное участие в повышении квалификации преподавателей и некоторые курсы, созданные сотрудниками кафедры, тиражируются на всю страну. Сотрудники кафедры разрабатывают новые учебные программы и циклы лабораторных работ, ориентированные на привитие студентам всех специальностей навыков практической работы на ЭВМ. Параллельно с этим кафедра организует подготовку преподавателей и сотрудников института в области программирования и применения средств ВТ в учебном процессе.

В последнее десятилетие на кафедре продолжаются работы по исследованию методов построения контролирующих и диагностических тестов (доц. Т.А. Павловская, доц. Э.В. Денисова), по автоматизации проектно-конструкторских работ в оптике (проф. А.В. Демин), по информационному обеспечению САПР (доц. В.Н. Блохин), по проблемам микропроцессоров.

На кафедре ведутся работы по внедрению методов дистанционного обучения, создаются учебники (доц. А.В. Лаздин, доц. Т.А. Павловкая) и комплекты тестов по отдельным дисциплинам и разделам дисциплин.

Кафедра Прикладной математики в 1994 году была переименована в кафедру Информатики и прикладной математики.

Радченко Ирина Алексеевна, Николаев Игорь Николаевич

Технологии и инфраструктура Big Data

Учебное пособие

В авторской редакции
Редакционно-издательский отдел Университета ИТМО
Зав. РИО Н.Ф. Гусарова
Подписано к печати
Заказ №
Тираж
Отпечатано на ризографе