Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И.Ульянова (Ленина)

Аналитическая обработка данных в задачах информационной безопасности

г. Санкт-Петербург 2022 г.

Вводная лекция

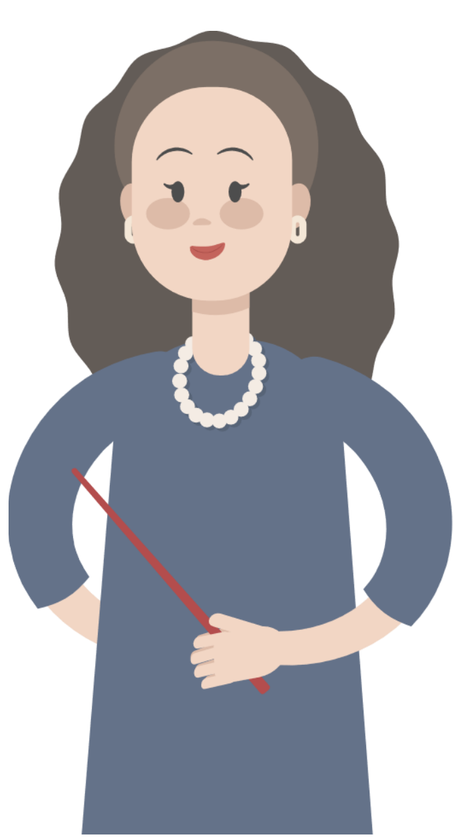
РПД

Приложение к ОПОП
«Безопасность и этика искусственного интеллекта»

СПбГЭТУ «ЛЭТИ»
ПЕРВЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ

МИНОБРНАУКИ РОССИИ

федеральное государственное автономное образовательное учреждение высшего образования
«Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И.Ульянова (Ленина)»
(СПбГЭТУ «ЛЭТИ»)



РАБОЧАЯ ПРОГРАММА

дисциплины

«АНАЛИТИЧЕСКАЯ ОБРАБОТКА ДАННЫХ В ЗАДАЧАХ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ»

для подготовки магистров

по направлению

09.04.01 «Информатика и вычислительная техника»

по программе

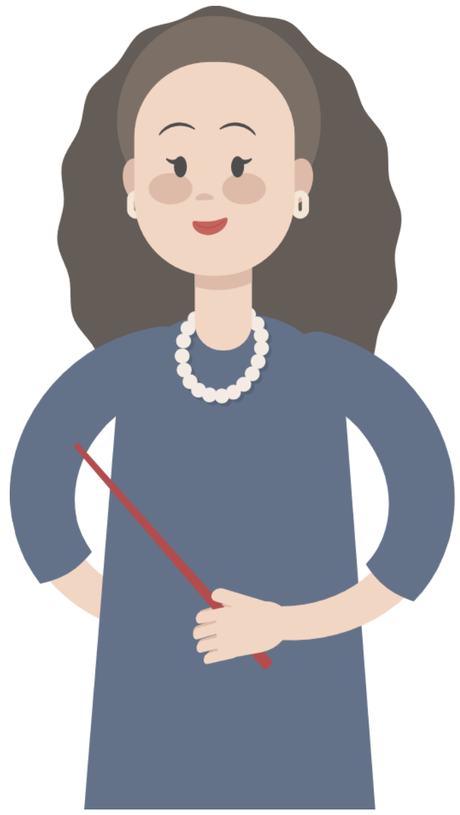
«Безопасность и этика искусственного интеллекта»

РПД

1 СТРУКТУРА ДИСЦИПЛИНЫ

Обеспечивающий факультет	ФКТИ
Обеспечивающая кафедра	ИС
Общая трудоемкость (ЗЕТ)	2
Курс	1
Семестр	2

Виды занятий	
Лекции (академ. часов)	17
Практические занятия (академ. часов)	17
Все контактные часы (академ. часов)	34
Самостоятельная работа, включая часы на контроль (академ. часов)	38
Всего (академ. часов)	72
Вид промежуточной аттестации	
Дифф. зачет (курс)	1



Аннотация

- Дисциплина посвящена изучению безопасности личности в цифровом пространстве, разведке в информационном пространстве, информационных войнах и применению больших объемов данных в задачах информационной безопасности.
- В рамках данной дисциплины рассматриваются основные подходы к сбору больших объемов информации из открытых источников, их накоплению, обработке и анализу.
- Дисциплина формирует знания студентов о современных технологиях анализа цифрового следа личности, дает понимание природы утечек информации и применения больших данных в SIEM (также необходимости самих SIEM).
- Также дисциплина формирует умения и навыки работы с самыми современными технологиями контейнеризации и их аспектами безопасности, нереляционными и графовыми базами данных, стеком Apache Hadoop и Apache NiFi (для хранения и обработки больших объемов данных), технологиями машинного обучения и графовыми нейронными сетями, а также применения данных технологий в информационной безопасности.

Цели и задачи

- **Цель** – изучить основные подходы к **сбору**, **обработке**, **накоплению** и **анализу** больших объемов информации из открытых источников для их последующего применения в задачах искусственного интеллекта и информационной безопасности.

Знания

- Дисциплина формирует знания студентов о современных технологиях анализа цифрового следа личности, дает понимание природы утечек информации и применения больших данных.

- Дисциплина формирует умения и навыки работы с самыми современными технологиями контейнеризации и их аспектами безопасности, нереляционными и графовыми базами данных, технологиями машинного обучения и графовыми нейронными сетями, а также применения данных технологий в информационной безопасности

Навыки

- Результатом освоения дисциплины является приобретение практических навыков в анализе существующих методов и средств, применяемых для контроля и защиты информации, разработке математических моделей, реализуемых в средствах защиты информации и выполнении анализа защищенности сетевых сервисов с использованием средств автоматического реагирования на попытки несанкционированного доступа к ресурсам компьютерных систем и сетей, разрабатывать программное и аппаратное обеспечение технологий и систем искусственного интеллекта для решения профессиональных задач с учетом требований информационной безопасности в различных предметных областях, модернизировать программное и аппаратное обеспечение технологий и систем искусственного интеллекта для решения профессиональных задач с учетом требований информационной безопасности в различных предметных областях.

Компетенции

ПК-30

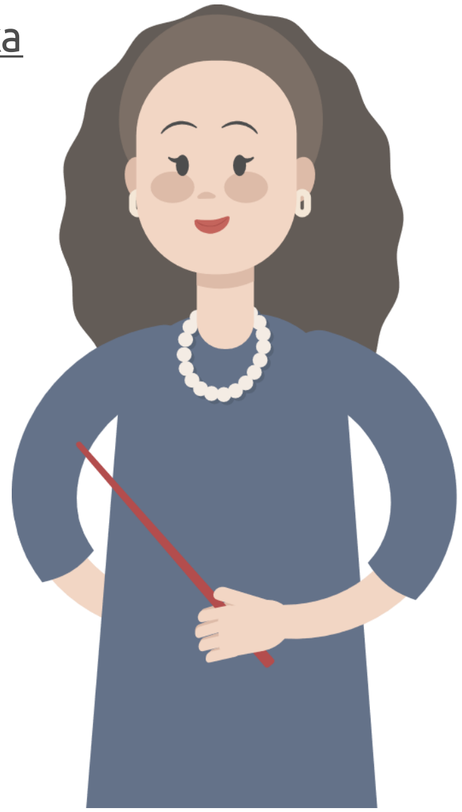
- Способен разрабатывать и модернизировать программное и аппаратное обеспечение технологий и систем искусственного интеллекта с учетом требований информационной безопасности в различных предметных областях

Индикаторы

- ПК-30.1. Разрабатывает программное и аппаратное обеспечение технологий и систем искусственного интеллекта для решения профессиональных задач с учетом требований информационной безопасности в различных предметных областях
- ПК-30.2. Модернизирует программное и аппаратное обеспечение технологий и систем искусственного интеллекта для решения профессиональных задач с учетом требований информационной безопасности в различных предметных областях

Содержание курса:

1. Вводная лекция
2. Тема 1. Влияние социальных сетей, медиа и всеобщего проникновения сети Интернет на жизнь современного человека. Проблемы безопасности личности в цифровом пространстве. Цифровой след личности в медиапространстве
3. Тема 2. Проблемы классических подходов к обработке, накоплению и анализу данных, разработка новых подходов. Изменчивость информационных систем
4. Тема 3. Хранение больших объемов данных. Стек технологий Apache Hadoop. Файловая система HDFS. Потоковая обработка данных с помощью Apache NiFi. Архитектурные решения хранения больших объемов данных, примененные в Apache Hadoop
5. Тема 4. Вычисления в памяти как единственный способ обработки больших данных в реальном времени. Современные технологии вычислений в памяти
6. Тема 5. Модель анализа текстов BERT. Модель анализа текстов CatBoost
7. Тема 6. Графовые нейронные сети.
8. Тема 7. Применение изученных подходов для хранения и анализа событий информационной безопасности.



Введение. Основные понятия.

Большие данные — это разнообразные данные, которые поступают с постоянно растущей скоростью и объем которых постоянно растет. Таким образом, три основных свойства больших данных — разнообразие, высокая скорость поступления и большой объем.

Если говорить простыми словами, **большие данные** — это более емкие и сложные наборы данных, особенно из нестандартных источников. Размер этих наборов данных настолько велик, что традиционные программы для обработки не могут с ними справиться. Но эти огромные объемы данных можно использовать для решения бизнес-задач, которые раньше казались слишком сложными.

Введение. Источники больших данных.

Введение. Свойства больших данных

- **Объем (Volume).** Количество данных — важный фактор. При работе с большими данными обрабатываются внушительные объемы неструктурированных данных низкой плотности. Ценность таких данных не всегда известна. Это могут быть данные каналов Twitter, данные посещаемости веб-страниц, а также данные мобильных приложений, сетевой трафик, данные датчиков. В некоторые компании могут поступать десятки терабайт данных, в другие — сотни петабайт.

Введение. Свойства больших данных

- **Скорость (Velocity)**. Скорость в данном контексте — это скорость приема данных и, возможно, действий на их основе. Обычно высокоскоростные потоки данных поступают прямо в оперативную память, а не записываются на диск. Некоторые «умные» продукты, функционирующие на основе Интернета, работают в режиме реального или практически реального времени. Соответственно, такие данные требуют оценки и действий в реальном времени.

Введение. Свойства больших данных

- Разнообразие (Variety). Разнообразие означает, что доступные данные принадлежат к разным типам. Традиционные типы данных структурированы и могут быть сразу сохранены в реляционной базе данных. С появлением больших данных данные стали поступать в неструктурированном виде. Такие неструктурированные и полуструктурированные типы данных как текст, аудио и видео требуют дополнительной обработки для определения их значения и поддержки метаданных.

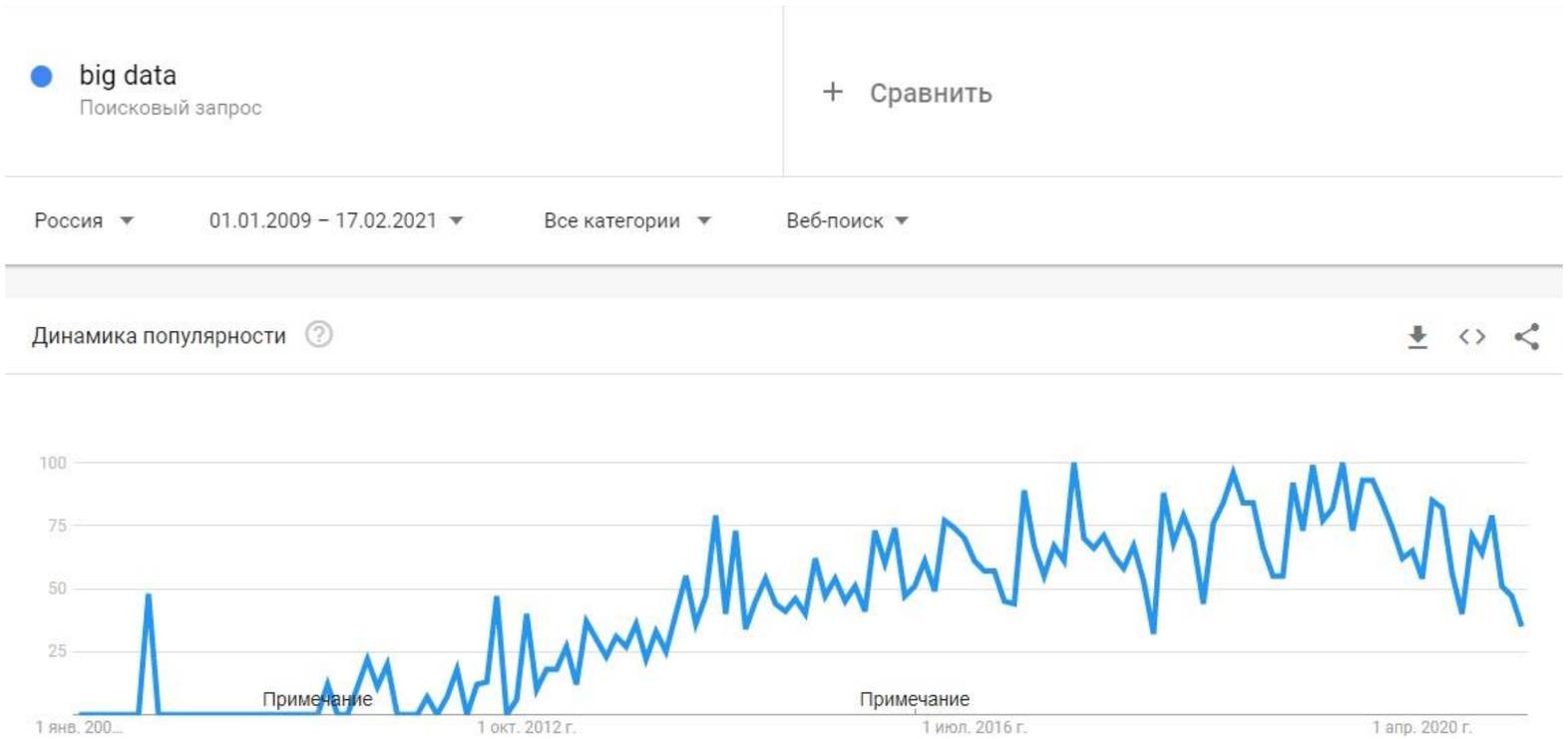
Введение. История больших данных

- Хотя сама по себе концепция больших данных не нова, изначально большие наборы данных начали использовать в 1960-70-х гг., когда появились первые в мире ЦОД и реляционные базы данных.
- К 2005 году бизнес начал осознавать, насколько велик объем данных, которые пользователи создают при использовании Facebook, YouTube и других интернет-сервисов. В том же году появилась платформа Hadoop на основе открытого кода, которая была создана специально для хранения и анализа наборов больших данных. В то же время начала набирать популярность методология NoSQL.
- Появление платформ на основе открытого кода, таких как Hadoop и позднее Spark, сыграло значительную роль в распространении больших данных, так как эти инструменты упрощают обработку больших данных и снижают стоимость хранения. За прошедшие годы объемы больших данных возросли на порядки. Огромные объемы данных—по-прежнему появляются в результате деятельности пользователей, но теперь данные производят не только они.

Введение. История больших данных

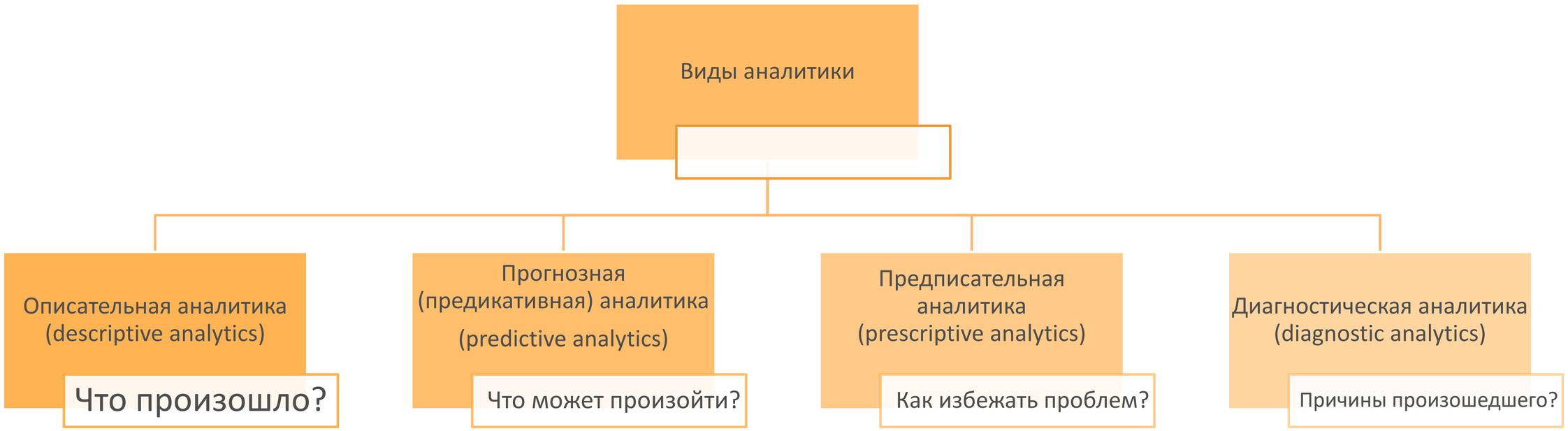
- С появлением **Интернета вещей (IoT)** все большее число устройств получает подключение к Интернету, что позволяет собирать данные о моделях действий пользователей и работе продуктов. А когда появились **технологии машинного обучения**, объем данных вырос еще больше.
- Большие данные имеют долгую историю развития, однако их потенциал еще далеко не раскрыт. Облачные вычисления раздвинули границы применения больших данных. Облачные технологии обеспечивают по-настоящему гибкие возможности масштабирования, что позволяет разработчикам развертывать кластеры для тестирования выборочных данных по требованию. **Графические базы данных** также становятся все более важными благодаря их способности отображать огромные объемы данных для предоставления быстрой и всеобъемлющей аналитики.

Введение. История больших данных



Всплеск интереса к большим данным в Google Trends

Введение. Виды аналитики.



Лекция 1.
Влияние социальных сетей, медиа и всеобщего проникновения Интернет на жизнь современного человека.

Проблемы безопасности личности в цифровом пространстве. Цифровой след личности в медиапространстве.

Охват распространения сети Интернет

- По данным исследования проекта WEB-Index, в феврале-ноябре 2020 года интернетом в России хотя бы раз в месяц пользовались в среднем 95,6 млн человек или 78,1% населения всей страны старше 12 лет. В среднем за день в интернет выходили 87,1 млн человек или 71,1% населения России.
- Проникновение интернета в России среди более молодого населения (до 44 лет) в 2020 году превысило 90%, а среди самых молодых россиян (12-24 лет) приблизилось к 100%. В группе населения 45-54 лет интернетом хотя бы раз в месяц пользовались 84,2% россиян, а среди самых старших жителей страны (55+ лет) в интернет выходит только половина – 49,7%.

Влияние распространения сети Интернет

Из положительных сторон можно выделить:

- Современный Интернет является одним из наиболее эффективных средств коммуникации между людьми, предоставляет различные платформы для общения, включая электронную почту и мгновенный обмен сообщениями через социальные сети.
- Сегодня Интернет является главным источником информации, главным ресурсным полем для образования. Глобальная сеть послужила катализатором в развитии человечества, расширив охват коммуникации, а также её интенсивность.

Из негативных сторон можно выделить:

- Последние медицинские исследования показывают, что зависимость от Интернета (порождаемая обилием развлекательного контента) вызывает личные, профессиональные, а также социальные проблемы.
- Интернет формирует новые социальные практики: троллинг, киберзапугивание, приватность в социальных сетях и пр. Однако, способы профилактики и противодействия им развиваются с некоторым опозданием – «на шаг позади» деструктивного поведения пользователей Интернета.

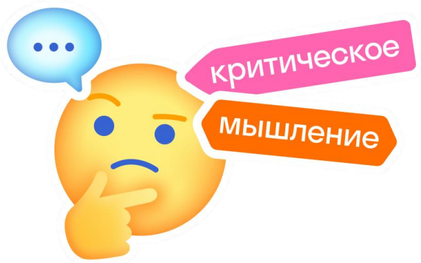
Влияние социальных сетей

- Социальные сети (даже самого простого типа) имеют узловую структуру. Центрами «притяжения» могут являться сообщества или отдельные пользователи – производители публичного контента, значимого для конкретных групп пользователей. Также последние создают социальные связи между собой, для передачи и получения информации.
- Ведущие социальные сети, отличаются высоким количеством учётных записей пользователей. За первый квартал 2022 г. аудитория Twitter насчитывала 229 млн ежедневных активных пользователей, а аудитория ВКонтакте73 млн. 550 млн пользователей были к июлю 2021 г. в Телеграме.

Влияние социальных сетей

Пользователей можно разделить на три группы: «активные» (те, которые создают публичный контент), «пассивные» (читатели чужого контента, безобидные) и можно выделить ещё одну группу «трансляторы», которые не создают никакой информации, но активно распространяют найденное в сети другим пользователям – адресными сообщениями или перепостом на свою страничку. Последние зачастую не умеют проверять достоверность пересылаемых данных. Будучи подверженными чужому влиянию, могут выполнять роль послушного “стада” и способствовать распространению дезинформации и слухов.

Специалист по ИБ обязан уметь проверять информацию, а также обладать критическим мышлением.



Цифровой след личности в медиапространстве

Цифровым следом - называют любые данные, оставляемые вами при использовании компьютерных систем и сети Интернет. Это может быть не только оставленный телефон в социальной сети или же информация о ваших предпочтениях, но и информация о размере экрана, вашем устройстве, с которого вы посещаете веб-ресурс, user-agent, электронные письма, список посещаемых веб-сайтов…

Цифровой след личности в медиапространстве

Активный цифровой след

- Пользователь сам оставляет информацию о себе в соц сетях, делает публикации, оставляет комментарии на форумах или в каналах Телеграма.
- Также активный цифровой след остается при заполнении онлайн-форм, например, подписке на информационные рассылки, или при согласии принимать файлы cookie в браузере.

Пассивный цифровой след

- Информация о пользователе собирается без его ведома. Сколько раз пользователи посещали сайт, откуда эти пользователи (местоположение, а также сайт, с которого они перешли на текущий) и их IP-адреса.

Проблемы безопасности личности в цифровом пространстве

Почему важны цифровые следы?

- Они относительно постоянны. Как только информация становятся общедоступной (полностью или частично), как, например, публикация в Facebook, автор практически не может контролировать, как она будет использоваться другими людьми.
- Цифровой след может отражать цифровую репутацию человека, которая теперь считается такой же важной, как и репутация за пределами сети.
- Прежде чем принимать решения о найме, работодатели могут проверять цифровые следы своих потенциальных сотрудников, особенно их социальные сети. Колледжи и университеты могут проверять цифровые следы своих будущих студентов перед зачислением на учебу.
- Публикуемые в интернете сообщения и фотографии могут быть неверно истолкованы или изменены, что может привести к непреднамеренному оскорблению.
- Контент, предназначенный для узкой группы, может распространиться на более широкий круг и испортить отношения и дружбу.
- Киберпреступники могут использовать ваш цифровой след в целях фишинга для доступа к учетной записи или для создания ложных профилей на основе ваших данных.

Проблемы безопасности личности в цифровом пространстве

- Очевидно, что результаты цифрового анализа данных о личности могут стать объектом неправомерных действий, например, номер телефона, имя, иных сведений из жизни конкретного человека, его голос, фотографии. Любая даже безобидная на первым взнляд информация о человеке может быть использована для мошеннических действий.
- Из «открытой публичности» данных о пользователях извлекают материальную и нематериальную выгоду частные лица, маркетологи, экономисты и рекрутеры. По этой причине популярная фраза «мне нечего скрывать» в корне неверна. Безответственное, неряшливое открытие личной информации сродни обнажанию своего тела.

Законодательство

- К персональным данным можно отнести любую информацию, которой достаточно, чтобы однозначно определить физическое лицо и получить о нём какую-либо дополнительную информацию. Любая организация, работающая с данными физических лиц, должна защитить информационные системы и получить документы, подтверждающие соответствие этих систем требованиям закона.
- По закону каждой информационной системе, в которой хранятся и обрабатываются персональные данные, необходимо присвоить класс, в соответствии с которым будет обеспечиваться защита этих данных. За несоблюдения требований можно быть привлечённым к ответственности.
- Работа с персональными данными регламентируются Федеральным законом "О персональных данных" от 27.07.2006 N 152-ФЗ (**ФЗ-152**).
- **GDPR (General Data Protection Regulation)** - это постановление, которое требует от предприятий защищать персональные данные и конфиденциальность граждан ЕС для транзакций, которые происходят в государствах-членах ЕС. И несоблюдение может дорого обойтись компаниям. Вот что каждая компания, которая ведет бизнес в Европе, должна знать о GDPR.

Конкурентная разведка

Конкурентная разведка — это целенаправленный сбор информации о прямых конкурентах с помощью общедоступных источников информации.

Извлечение общедоступной информации о конкурентах позволяет получить:

- данные о текущих тенденциях рынка (поможет в составлении долгосрочной стратегии в развитии компании);
- вероятные инструменты давления, которые использует конкурент (и в дальнейшем составление разумных ответных действий для защиты собственных интересов);
- опыт конкурентов (используются сильные стороны для укрепления собственных позиций);
- общая емкость потребительского рынка (это указывает финансовое развитие предприятия);
- потенциал на развитие за пределами текущего региона;
- степень выгодности сотрудничества с текущими клиентами и поставщиками (или специалистами, нанятыми в рамках аутсорсинга).

Контроль утечек информации в открытых источниках

- **ГосСОПКА** (Государственная система обнаружения, предупреждения и ликвидации последствий компьютерных атак) создаётся для обмена информацией о кибератаках на информационные системы, нарушение или прекращение работы которых крайне негативно скажется на экономике страны или безопасности граждан.
- К ГосСОПКА должны подключиться владельцы объектов критической информационной инфраструктуры. К ним относятся организации здравоохранения, науки, транспорта, связи, энергетики, банковской сферы (системно значимые кредитные организации, операторы платёжных систем, системно значимые инфраструктурные организации финансового рынка), топливно-энергетического комплекса, в области атомной энергии, оборонной, ракетно-космической, горнодобывающей, металлургической и химической промышленности, российские юридические лица и (или) индивидуальные предприниматели, которые обеспечивают взаимодействие указанных систем или сетей.
- Если ваши персональные данные слили в сеть, вы можете обратиться в Роскомнадзор.

Информационные войны

С помощью информационных технологий изменились тактики ведения военных и других конфликтов. Информация и дезинформация превращаются в опасное оружие сообразно тому, в чьих руках оказались сведения и с какой целью они применяются.

- **Командно-управленческая война** — ставит перед собой цель лишить контроля налаженную связь между командованием и исполнителем.
- **Разведывательная война** — предусматривает сбор ценной информации для нападения и собственной защиты. Электронная война — целью является вывод из строя всех электронных коммуникаций.
- **Психологическая война** — пропаганда и информационное зомбирование населения противником.
- **Хакерская война** — взлом и доступ к любым данным (электронная почта, банковские карты, личные файлы, переписки и так далее) и несанкционированное их использование.
- **Экономическая война** — информационная блокада (ограничение коммерческой деятельности) или информационный империализм (политическая информационная атака).
- **Кибервойна** — ставит перед собой цель захватить компьютерные данные, выследить объект, нарушить работу инфраструктуры, полагающейся на информационные технологии.

Лекция 2.
Проблемы классических подходов к обработке, накоплению и анализу данных, разработка новых подходов. Изменчивость информационных систем.

Классические и новые подходы к обработке данных

Разница между классическими и новыми методами к обработке данных заключается в том, какие данные необходимо проанализировать, отобрать, хранить.

Провести разницу между большими и традиционными данными можно с помощью нескольких характеристик:

1. Размер данных
2. Способ организации данных, тип данных
3. Необходимая архитектура для управления данными
4. Источники, из которых поступают данные (один или несколько, какие)
5. Методы анализа данных

Классических методы анализа данных

- Структурный анализ – выявление структуры как относительно устойчивой совокупности отношений. **Возникают проблемы при попытке структурировать очень неоднородные данных.**
- Диаграммы потоков данных – графическое средство для изображения информационного потока и преобразований, которым подвергаются данные при движении от входа к выходу системы. **Данный метод применим при обработке больших массивов данных, однако труден в выборе необходимой модели и её описании.**
- Описание потоков данных – выделение словаря требований описаний данных. Описание псевдонимов (alias) данных, содержания данных, их влияния на те или иные процессы.
- Метод анализа Джексона (объектно-ориентированный подход) – выделение свойств объектов, объектов-действий, объектов-структур, доопределение функций, определение характеристик будущих процессов.

Новые подходы к обработке данных

- Машинное обучение и нейронные сети
- Смешение и интеграция данных
  Процесс приведения разнородной информации к единому виду.
- Предиктивная аналитика
  Прогнозирование
- Имитационное моделирование
  Чтобы не экспериментировать с реальным бизнесом, можно построить симуляцию.
- Статистический анализ
- Data mining
  Поиск полезные закономерности, глубинный анализ данных.
- Визуализация аналитических данных (диаграммы потоков данных)
- (Краудсорсинг)
  Использование большого количества волонтёров для выполнения рутинных задач.

Разница в анализе данных

Традиционная аналитика

- Постепенный анализ небольших пакетов данных
- Редакция и сортировка данных перед обработкой
- Старт с гипотезы и ее тестирования относительно данных
- Данные собираются, обрабатываются, хранятся и лишь затем анализируются

Big data аналитика

- Обработка сразу всего массива доступных данных
- Данные обрабатываются в их исходном виде
- Поиск корреляций по всем данным до получения искомой информации
- Анализ и обработка больших данных в реальном времени, по мере поступления

Разница в хранении данных

- В зависимости от объёма данных меняется и используемая технология хранения самих этих данных. Чем больше у вас данных, тем полезнее будет нереляционная база данных, потому что она не будет накладывать ограничений на входящие данные, что позволит вам быстрее делать запись в базу.
- Если у вас меньше 1 ТБ данных, то с PostgreSQL вы получите хорошую производительность. Но она замедляется на объемах около 6 ТБ. Если вам нравится MySQL, но нужны немного большие масштабы, Aurora (собственная версия Amazon) может достичь 64 ТБ. Для размера в петабайт Amazon Redshift обычно является хорошим выбором, поскольку он оптимизирован для выполнения аналитики до 2PB. Для параллельной обработки, скорее всего, пора взглянуть на Hadoop.
- Также стоит отметить, что для больших объёмов данных не подойдёт классический подход хранения данных на твердотельных накопителях (вместо этого можно использовать хранение в оперативной памяти, но об этом в след. лекциях).

Database Options by Scale				
Data Size	< 1TB	2TB-64TB	64TB-2PB	#alloftheData
Database that's a good fit	Postgres MySQL	Amazon Aurora	Amazon Redshift Google BigQuery	Hadoop

Изменчивость информационных систем

- С ростом технологий меняются информационных системы, объёмы и виды оперируемой информации, протоколы передачи данных, регулирующие их правила. Существенные изменения могут происходить как на уровне одной компании, так и на уровне целого государства. Так, к примеру может меняться весь технологический процесс в погоне за лучшей бизнес-моделью, может меняться устройство сетей в стране, осуществляющий переход на высокоскоростное оборудование и решения. Не стоит также забывать и о том, что с обновлением технологий и ПО могут появляться новые уязвимости. Не стоит с полным доверием относиться и к ПО с открытым исходным кодом. Есть масса примеров, когда даже в открытых проектах находили бэкдоры и уязвимости. Также, открытое ПО открыто для анализа злоумышленником, и при использовании таких решений, ваша система может стать простой целью, для таких исследователей.
- По этой причине, важно уметь адаптироваться в условиях быстрой изменчивости информационных систем. Сделать это можно посредством построения модульной системы, а также использовании технологий контейнеризации.

Технология контейнеризации

- **Контейнеризация** — метод виртуализации, при котором ядро операционной системы поддерживает несколько изолированных экземпляров пространства пользователя вместо одного. Эти экземпляры, с точки зрения выполняемых в них процессов, идентичны отдельному экземпляру операционной системы. Для систем на базе Unix эта технология похожа на улучшенную реализацию механизма chroot. Ядро обеспечивает полную изолированность контейнеров, поэтому программы из разных контейнеров не могут воздействовать друг на друга.
- Широко известна технологий контейнеризации **Docker**. Контейнеры упрощают разворачивание приложений, обеспечивают их изоляцию от реальной машины, а значит обеспечивает некоторый уровень безопасности (обойти виртуализацию крайне сложно) и избавляет от разбирательств с неразберихой между версиями используемых в приложежнии библиотек. Docker позволяет обеспечивает качественную гибкую настройку разворачиваемых контейнеров (вы можете выбрать, сколько уделить памяти контейнеру, включить перезапуск в случаях падения сервисов), их масштабируемость вплоть до множества контейнеров на различных хостах. Контейнеризация позволяет вам создать исключительно необходимую среду для работы разрабатываемого вами приложения. Для подключения различных ресурсов с реальной машины предназначены так называемые volumes, которые подключаются в контейнер.

Лекция 3.
Хранение больших объемов данных. Стек технологий ApacheHadoop. Файловая система HDFS. Потоковая обработка данных с помощью Apache Ni-Fi. Архитектурные решения хранения больших объемов данных, примененные в Apache Hadoop.

Хранение больших объемов данных

В отличие от традиционных данных, большие данные имеют намного больше требований для хранения. По этой причине хранилище таких данных должно:

- обеспечивать быстродоступность к данным;
- уметь хранить неоднородные наборы данных;
- уметь хранить большие объёмы данных, не утрачивая скорости обращения к данным.

Нереляционные базы данных. Достоинства и недостатки.

Нереляционные базы данных. Достоинства и недостатки.

Плюсы:

- Простота работы. Многие NoSQL решения, в основном хранилища вида “ключ-значение” имеют по сравнению с реляционными базами данных очень сильно урезанную функциональность, которая им просто не требуется для выполнения поставленных задач. В таком случае оператору базы данных не требуется глубоких знаний достаточно мощного и гибкого механизма работы с SQL-запросами.
- Более простой синтаксис запросов - меньше ошибок.

Минусы:

- Приложение сильно привязывается к конкретной СУБД. Язык SQL универсален для всех реляционных хранилищ.
- Ограниченная емкость встроенного языка запросов. SQL имеет очень богатую историю и множество стандартов.
- Низкая ценность и узкопрофильность знаний - специалистов с хорошим знанием SQL гораздо проще найти, в то время когда спецификой работы API некоторых NoSQL решений на серьёзном уровне мало кто увлекается - это значит, что многие специфические моменты оператору базы данных придется осваивать “на ходу”.

ApacheHadoop

Библиотека программного обеспечения **Apache Hadoop** - это платформа, которая позволяет распределять обработку больших наборов данных между кластерами компьютеров с использованием простых моделей программирования. Он предназначен для масштабирования от отдельных серверов до тысяч машин, каждая из которых предлагает локальные вычисления и хранение. Вместо того, чтобы полагаться на аппаратное обеспечение для обеспечения высокой доступности, сама библиотека предназначена для обнаружения и обработки сбоев на прикладном уровне, поэтому она предоставляет высокодоступную службу поверх кластера компьютеров, каждый из которых может быть подвержен сбоям.

Где и зачем используется Hadoop?

- обработка больших данных, поступающих из открытых источников;
- обработка массивов телеком данных (биллинги, смс, чаты);
- поисковые и контекстные механизмы высоконагруженных веб-сайтов и интернет-магазинов (Yahoo!, Facebook, Google, AliExpress, Ebay и т.д.), в т.ч. для аналитики поисковых запросов и пользовательских логов;
- хранение, сортировка огромных объемов данных и разбор содержимого чрезвычайно больших файлов;
- быстрая обработка графических данных, например, газета New York Times с помощью Hadoop и Web-сервиса Amazon Elastic Compute Cloud (EC2) всего за 36 часов преобразовала 4 терабайта изображений (TIFF-картинки размером в 405 КБ, SGML-статьи размером в 3.3 МБ и XML-файлы размером в 405 КБ) в PNG-формат размером по 800 КБ.

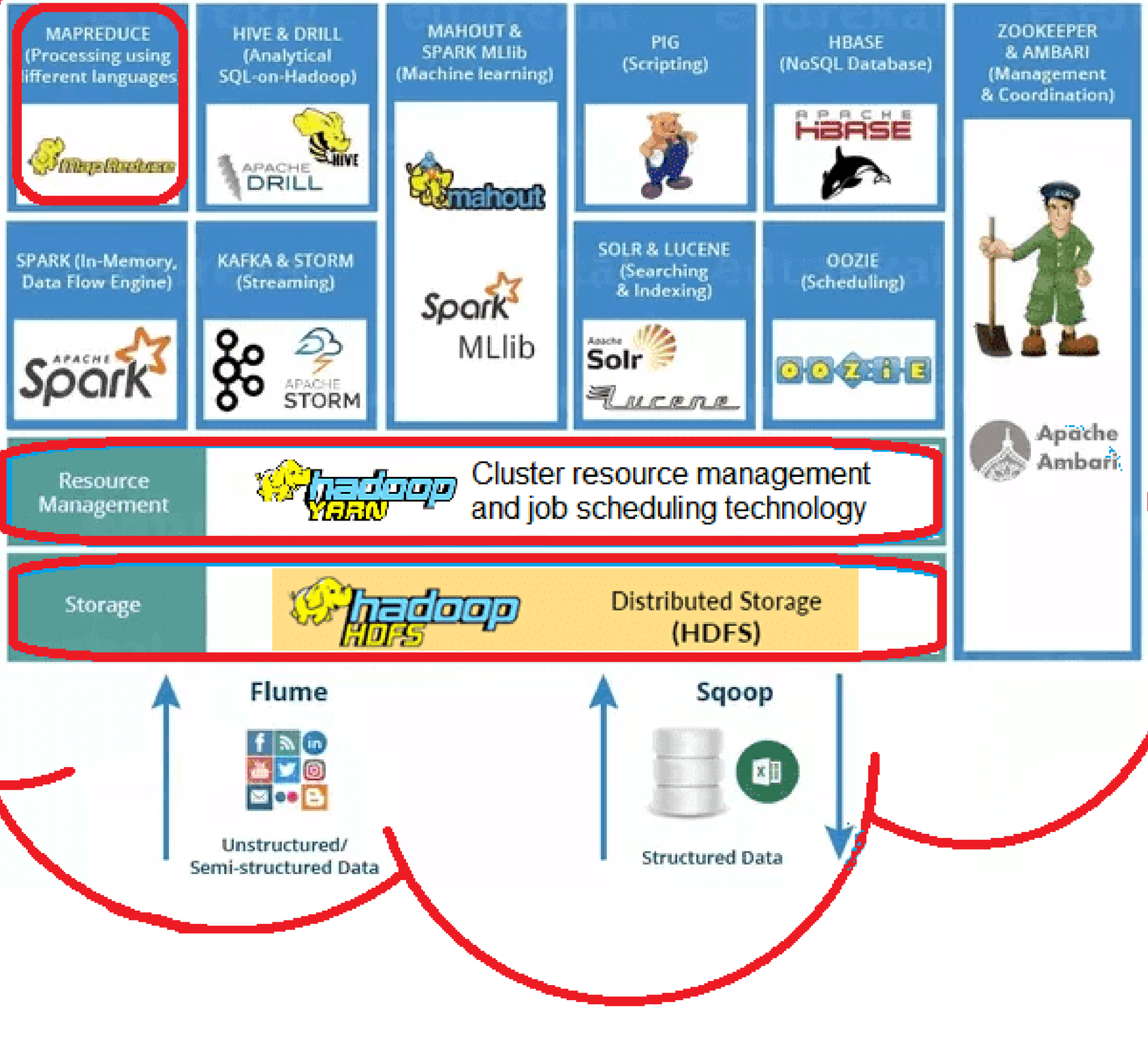
Стек технологий ApacheHadoop

Проект состоит из основных 4-х модулей:

- **Hadoop Common** – набор инфраструктурных программных библиотек и утилит, которые используются в других решениях и родственных проектах, в частности, для управления распределенными файлами и создания необходимой инфраструктуры.
- **HDFS** – распределённая файловая система, Hadoop Distributed File System – технология хранения файлов на различных серверах данных (узлах, DataNodes), адреса которых находятся на специальном сервере имен (мастере, NameNode). За счет дублирования (репликации) информационных блоков, HDFS обеспечивает надежное хранение файлов больших размеров, поблочно распределённых между узлами вычислительного кластер.
- **YARN** – система планирования заданий и управления кластером (Yet Another Resource Negotiator), которую также называют **MapReduce** 2.0 (MRv2) – набор системных программ (демонов), обеспечивающих совместное использование, масштабирование и надежность работы распределенных приложений. Фактически, YARN является интерфейсом между аппаратными ресурсами кластера и приложениями, использующих его мощности для вычислений и обработки данных
- **Hadoop MapReduce** – платформа программирования и выполнения распределённых параллельных MapReduce-вычислений, с использованием большого количества компьютеров (узлов, nodes), образующих кластер.

Hadoop Common

Hadoop Common - связующее программное обеспечение, набор инфраструктурных программных библиотек и утилит, используемых для других модулей и родственных проектов



Подробнее о MapReduce

Парадигма MapReduce предполагает, что данные организованы в виде некоторых записей. Обработка данных происходит в 3 стадии:

1. **Стадия Map**. На этой стадии данные предобрабатываются при помощи функции map(), которую определяет пользователь. Работа этой стадии заключается в предобработке и фильтрации данных. Работа очень похожа на операцию map в функциональных языках программирования – пользовательская функция применяется к каждой входной записи. Функция map() примененная к одной входной записи и выдаёт множество пар ключ-значение. Множество – т.е. может выдать только одну запись, может не выдать ничего, а может выдать несколько пар ключ-значение. Что будет находится в ключе и в значении – решать пользователю, но ключ – очень важная вещь, так как данные с одним ключом в будущем попадут в один экземпляр функции reduce.

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

3. **Стадия Reduce**. Каждая «корзина» со значениями, сформированная на стадии shuffle, попадает на вход функции reduce(). Функция reduce задаётся пользователем и вычисляет финальный результат для отдельной «корзины». Множество всех значений, возвращённых функцией reduce(), является финальным результатом MapReduce-задачи.

Подробнее о MapReduce

Преимущества:

- прост в программировании;
- хорошая масштабируемость;
- высокая отказоустойчивость.

Недостатки:

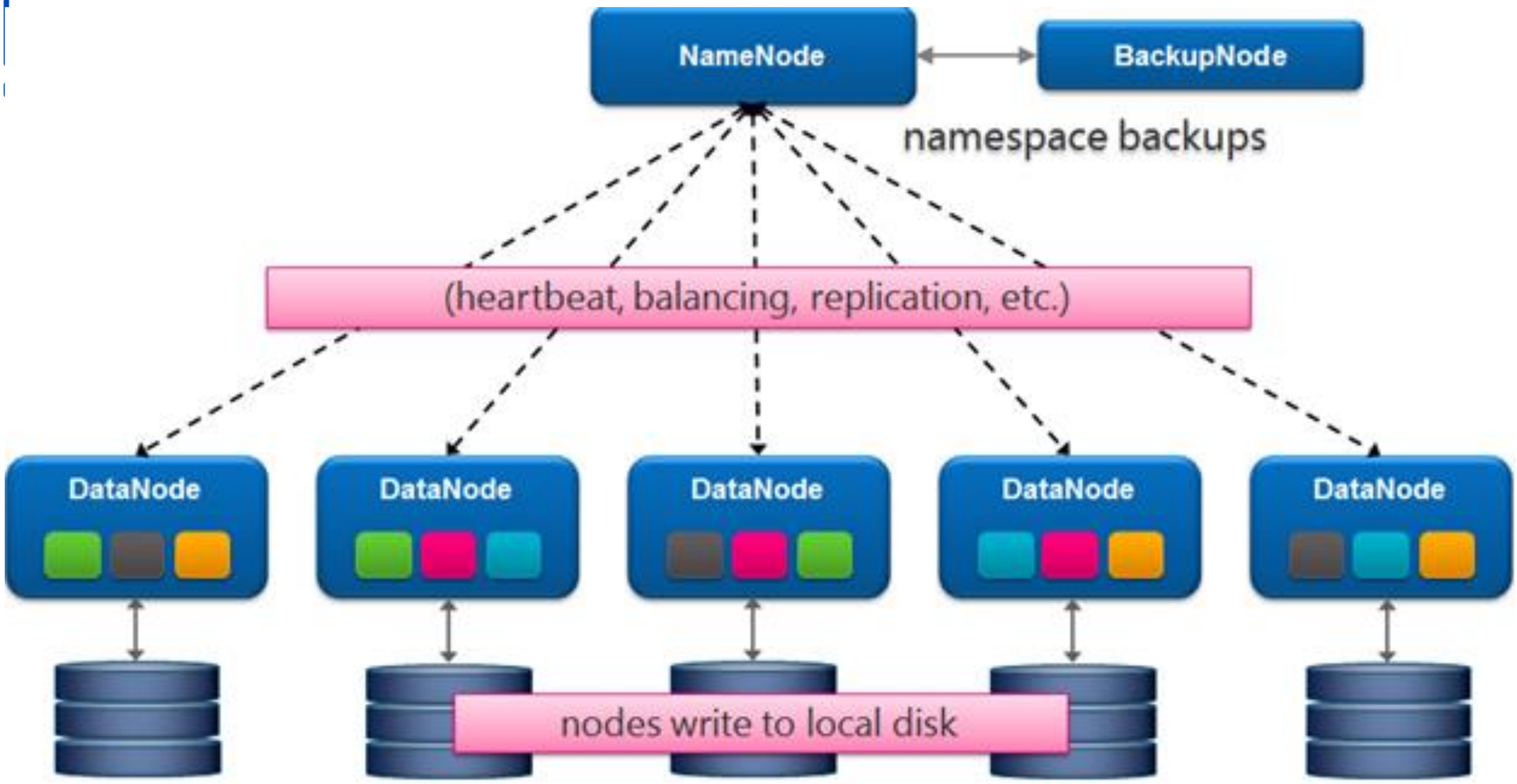
- не подходит для расчета в реальном времени;
- не очень хорош для вычислений потоковых данных;
- не очень хорош при расчете зависимостей DAG (направленных графов).

Подробнее о HDFS

Кластер HDFS включает следующие компоненты:

- **Управляющий узел, узел имен (NameNode)** – отдельный, единственный в кластере, сервер с программным кодом для управления пространством имен файловой системы, хранящий дерево файлов, а также мета-данные файлов и каталогов. **NameNode** отвечает за открытие и закрытие файлов, создание и удаление каталогов, управление доступом со стороны внешних клиентов и соответствие между файлами и блоками, дублированными (реплицированными) на узлах данных.
- **Secondary NameNode** — вторичный узел имен, отдельный сервер, единственный в кластере, который копирует образ HDFS и лог транзакций операций с файловыми блоками во временную папку, применяет изменения, накопленные в логе транзакций к образу HDFS, а также записывает его на узел NameNode и очищает лог транзакций. Secondary NameNode необходим для быстрого ручного восстанавления NameNode в случае его выхода из строя.
- **Узел или сервер данных (DataNode, Node)** – один их множества серверов кластера с программным кодом, отвечающим за файловые операции и работу с блоками данных. **DataNode** отвечает за запись и чтение данных, выполнение команд от узла **NameNode** по созданию, удалению и репликации блоков, а также периодическую отправку сообщения о состоянии (**heartbeats**) и обработку запросов на чтение и запись, поступающих от клиентов файловой системы **HDFS**.
- **Client** – специальное API для взаимодействия с файловой системой.

Структура взаимодействия компонентов HDFS



Apache Hbase & Cassandra

Apache HBase и Cassandra считаются наиболее популярными нереляционными базами данных в мире Big Data.

Особенности:

- **специфическая модель данных**, не ограничивающая число столбцов, которые можно сгруппировать в группы или семейства (column families) — информация хранится по столбцам, при не нужно хранить пустые значения (равные нулю), поэтому HBase хорошо подходит для разреженных наборов данных;
- **прослеживаемая аналогия с реляционными СУБД** в плане индекса первичного ключа — в HBase данные в таблицах упорядочены по строковым ключам при динамическом секционировании (partitioning) диапазона строк;
- **встроенный механизм временных меток** (timestamp), которые добавляются автоматически, но могут быть изменены вручную;
- **наличие инструментов расширяемости** (REST и другие API-интерфейсы Java и шлюзов) и внешних SQL-решений, позволяющих работать с данными, хранящимися в HBase, как с реляционными таблицами.
- **высокая производительность и быстрота работы**, в т.ч. в режиме реального времени, за счет кэширования в памяти и обработки данных на стороне сервера через фильтры и сопроцессоры. Например, тест на таблице из 3-х миллиардов строк при 300 параллельных запросов в секунду показал, что чтение будет занимать примерно 18 миллисекунд, запись выполнится почти в 3 раза быстрее и займет примерно 8 миллисекунд.
- **высокая доступность и отказоустойчивость**, обеспечиваемые с помощью репликации через центр обработки данных, неделимые и согласованные операции на уровне строк, а также автоматическое распределение нагрузки и балансировку таблиц за счет механизма регионирования. В качестве отказоустойчивого хранилища данных используются распределенная файловая система Hadoop (HDFS) и Amazon S3.
- **способность к масштабированию —** Apache HBase рассчитана на поддержание высокой производительности даже при увеличении кластера до сотен узлов для работы с миллиардами строк и миллионами столбцов.

При всех вышеперечисленных достоинствах, рассматриваемой нереляционной СУБД класса «семейство столбцов» свойственны **следующие недостатки**:

- некоторые, особенно сложные запросы (примерно 1% от общего количества) могут выполняться медленнее (порядка 300 миллисекунд);
- индексация возможна только по одному полю (Row Key). Впрочем, Apache Phoenix позволяет ввести вторичный индекс

Apache Hive

Программное обеспечение хранилища данных Apache Hive облегчает чтение, запись и управление большими наборами данных, находящимися в распределенном хранилище, с использованием SQL. Структура может быть проецирована на данные, уже находящиеся в хранилище. Для подключения пользователей к Hive предоставляется инструмент командной строки и драйвер JDBC.

Обеспечивает

- Масштабируемость MapReduce
- Удобство использования SQL для выборок из данных.

Потоковая обработка данных с помощью Apache Ni-Fi

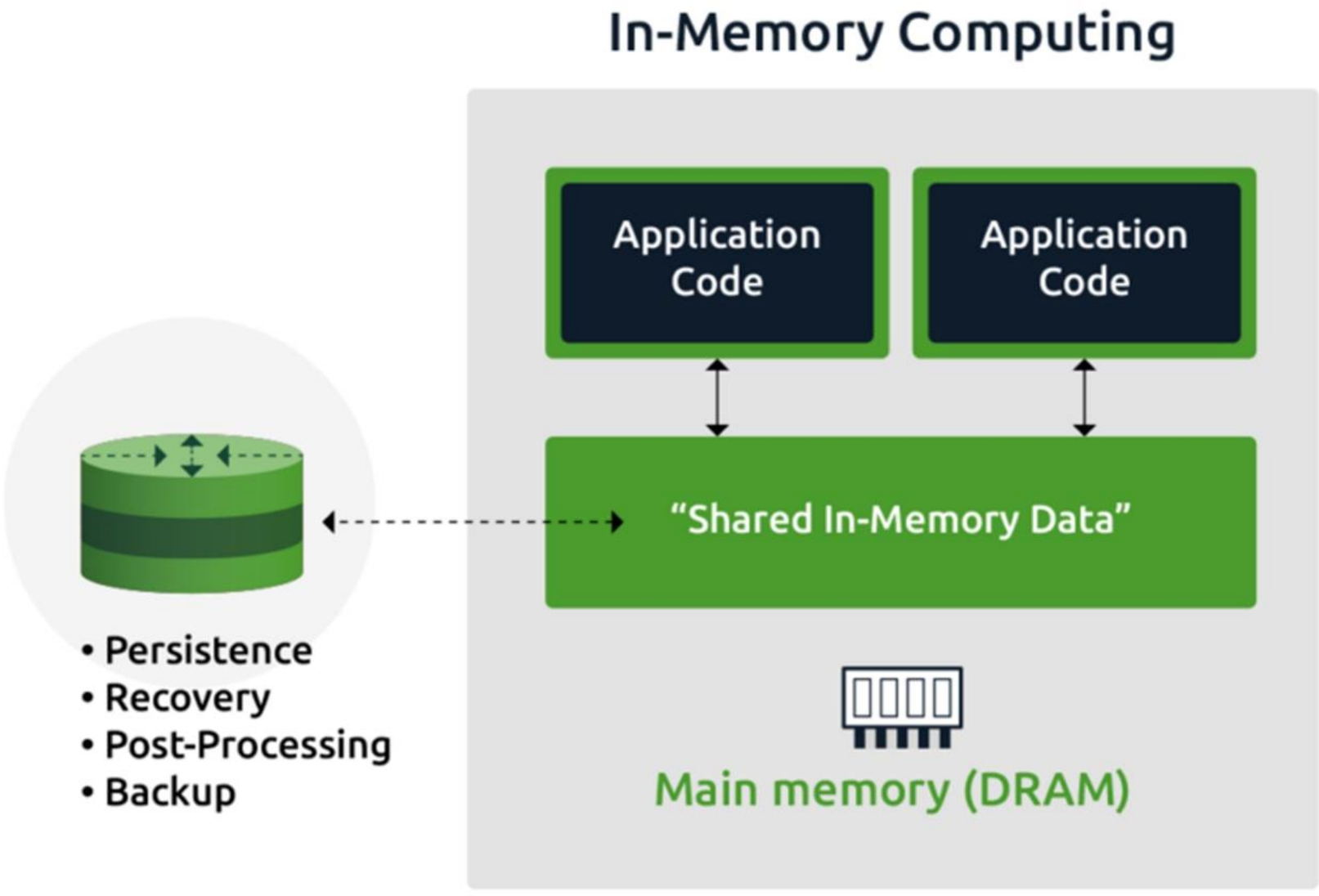
- **Apache NiFi** – популярного ETL-инструмента потоковой дата-инженерии, который предоставляет наглядный веб-интерфейс для проектирования конвейеров обработки потока данных в режиме реального времени. Он также поддерживает мощные и масштабируемые средства маршрутизации и преобразования данных, которые можно запускать на одном сервере или в кластерном режиме на нескольких узлах.
- Потоковые данные непрерывно генерируются тысячами источников, которые отправляют записи одновременно и в небольших размерах (порядка килобайт). Чаще всего такими данными являются лог-файлы от клиентов мобильных или веб-приложений, покупки в электронной торговле, события пользовательского поведения на сайтах, действия онлайн-игроков, информация из соцсетей, финансовых торговых площадок или геопространственных сервисов, а также телеметрия с IoT-устройств или оборудования в дата-центрах. Чтобы проанализировать эти данные, можно составить конвейер потокового маршрутизатора Apache NiFi из следующих шагов:
- генерация потоковых данных;
- анализ потоковых данных с помощью **Spark Streaming** (фреймворк с открытым исходным кодом для распределённой пакетной и потоковой обработки неструктурированных и слабоструктурированных данных, входящий в экосистему проектов **Hadoop**);
- запись проанализированных данных в топики **Apache Kafka** (распределенная платформа потоковой передачи событий с открытым исходным кодом, используемая тысячами компаний для высокопроизводительных конвейеров данных, потоковой аналитики, интеграции данных и критически важных приложений);
- визуализация результатов анализа на наглядном дэшборде **Kibana** в реальном времени.

Лекция 4.
Вычисления в памяти как единственный способ обработки больших данных в реальном времени. Современные технологии вычислений в памяти.

Вычисления в памяти

- Как известно, большие данные необходимо быстро уметь обрабатывать, хранить, отбирать.
- Для решения всех этих проблем используется технологии построенные на вычислениях в памяти, на методе методе, позволяющем выполнять компьютерных вычислений полностью в памяти компьютера (например, в оперативной памяти). Этот термин обычно подразумевает крупномасштабные, сложные вычисления, которые требуют специализированного системного программного обеспечения для выполнения вычислений на нескольких компьютерах, работающих вместе в кластере (их ещё называют сетками данных). Такой подход позволяет разбивать задачи на более мелкие и выполнять их параллельно.

Принцип работы вычислений в памяти



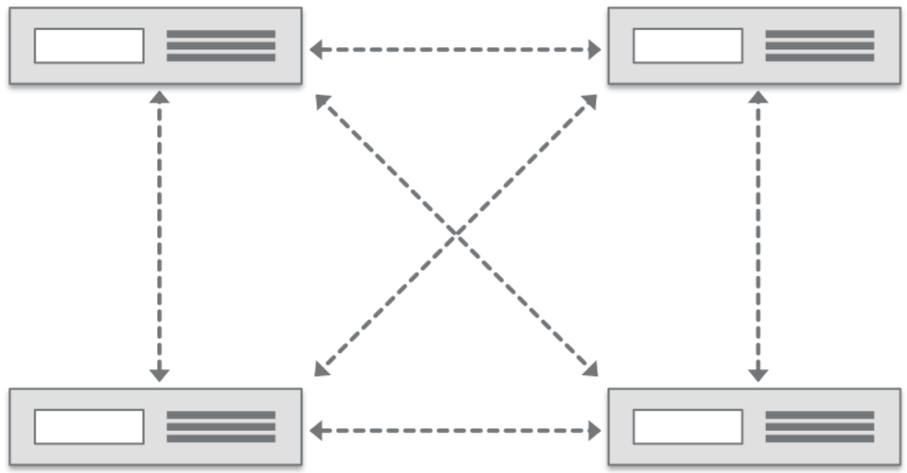
Вычисления в памяти

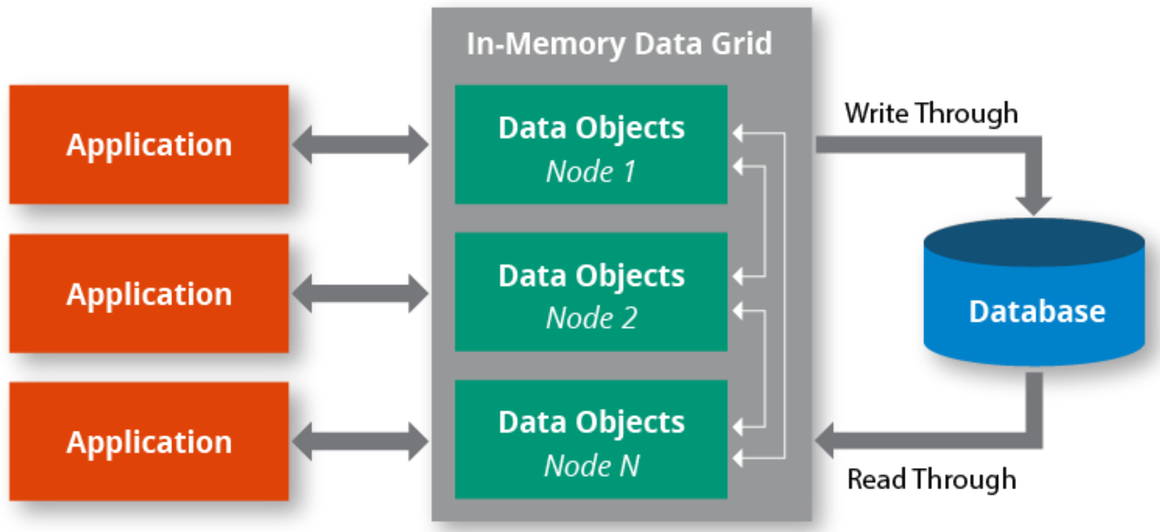
- Вычисления в памяти хранят данные в ОЗУ, а не в базах данных, размещенных на дисках. Это устраняет требования к транзакциям ввода-вывода и экспоненциально ускоряет доступ к данным, поскольку данные, хранящиеся в оперативной памяти, доступны мгновенно, в то время как данные, хранящиеся на дисках, ограничены скоростью сети и диска. Вычисления в памяти может кэшировать огромные объемы данных, обеспечивая чрезвычайно быстрое время отклика, и хранить данные сеанса, что может помочь достичь оптимальной производительности.

Распределенные вычисления

Распределенные вычисления - это метод объединения нескольких компьютерных серверов по сети в кластер для обмена данными и координации вычислительной мощности. Такой кластер называют “распределенной системой”.
Распределенные вычисления предлагают следующие преимущества:

- **Масштабируемость**
- **Производительность**
- **Устойчивость (репликация данных)**
- **Экономическая эффективность (недорогое оборудование)**



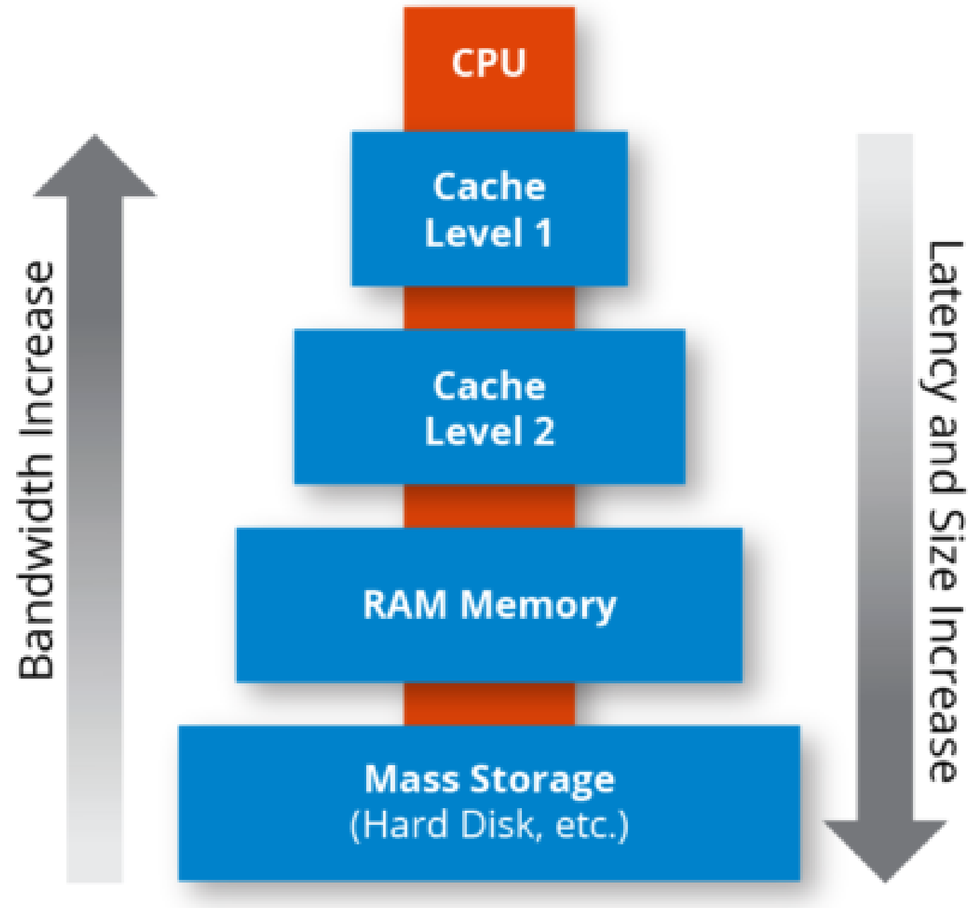


База данных в памяти

- База данных в памяти (IMDB) среда, в которой данные хранятся в ОЗУ по сравнению с дисками или твердотельными накопителями. IMDB по существу заменяют компонент доступа к диску баз данных на основе дисков доступом к ОЗУ. В некоторых IMDB дисковый компонент остается нетронутым, но ОЗУ является основным носителем данных. Поскольку ОЗУ энергозависимо (например, данные теряются, если компьютер теряет питание), некоторые IMDB также хранят данные на диске в качестве превентивной меры, чтобы минимизировать риск потери данных.
- Большинство IMDB также защищают от потери данных в одном центре обработки данных (возможность, известная как “высокая доступность”), сохраняя копии (“реплики”) всех записей данных на нескольких компьютерах в кластере. Эта избыточность данных гарантирует, что любая запись данных не будет потеряна при сбое любого данного компьютера.

Кэширование памяти

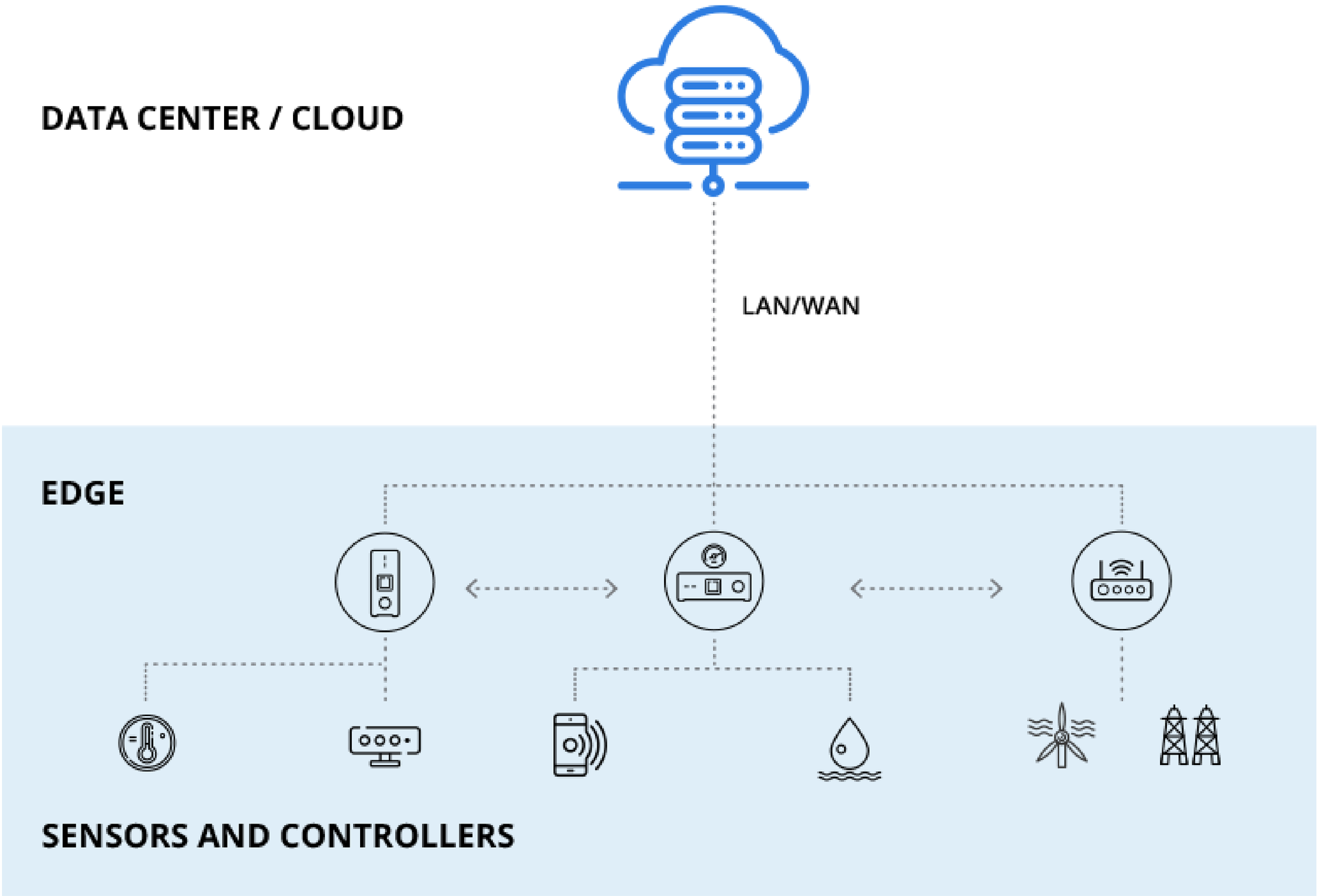
- Метод, подразумевающий использование кэша позволяет ускорить доступ к маленькому по сравнению со всем хранилищем объёму памяти, располагающемуся в оперативной памяти.
- Работает это следующим способом: когда приложение пытается прочитать данные, обычно из системы хранения данных, такой как база данных, оно проверяет, существует ли нужная запись в кэше.
- Проблема с кэшами заключается в том, как минимизировать “промахи кэша”.



Пограничные вычисления

- **Пограничные вычисления** (или пограничная обработка **IoT**, **EDGE**) относятся к действиям с данными как можно ближе к источнику, а не в центральном удаленном центре обработки данных, чтобы уменьшить задержку и использование полосы пропускания. Перенося вычисления на край сети, предприятия либо фильтруют / агрегируют необработанные данные, чтобы уменьшить объем, который должен передаваться по сети, либо запускают аналитику на месте, чтобы немедленно получить важную информацию. Обе эти стратегии помогают уменьшить или устранить задержку, присущую передаче данных на большие расстояния.

Принцип работы пограничных вычислений



Лекция 5.
Модель анализа текстов BERT.
Модель анализа текстов CatBoost

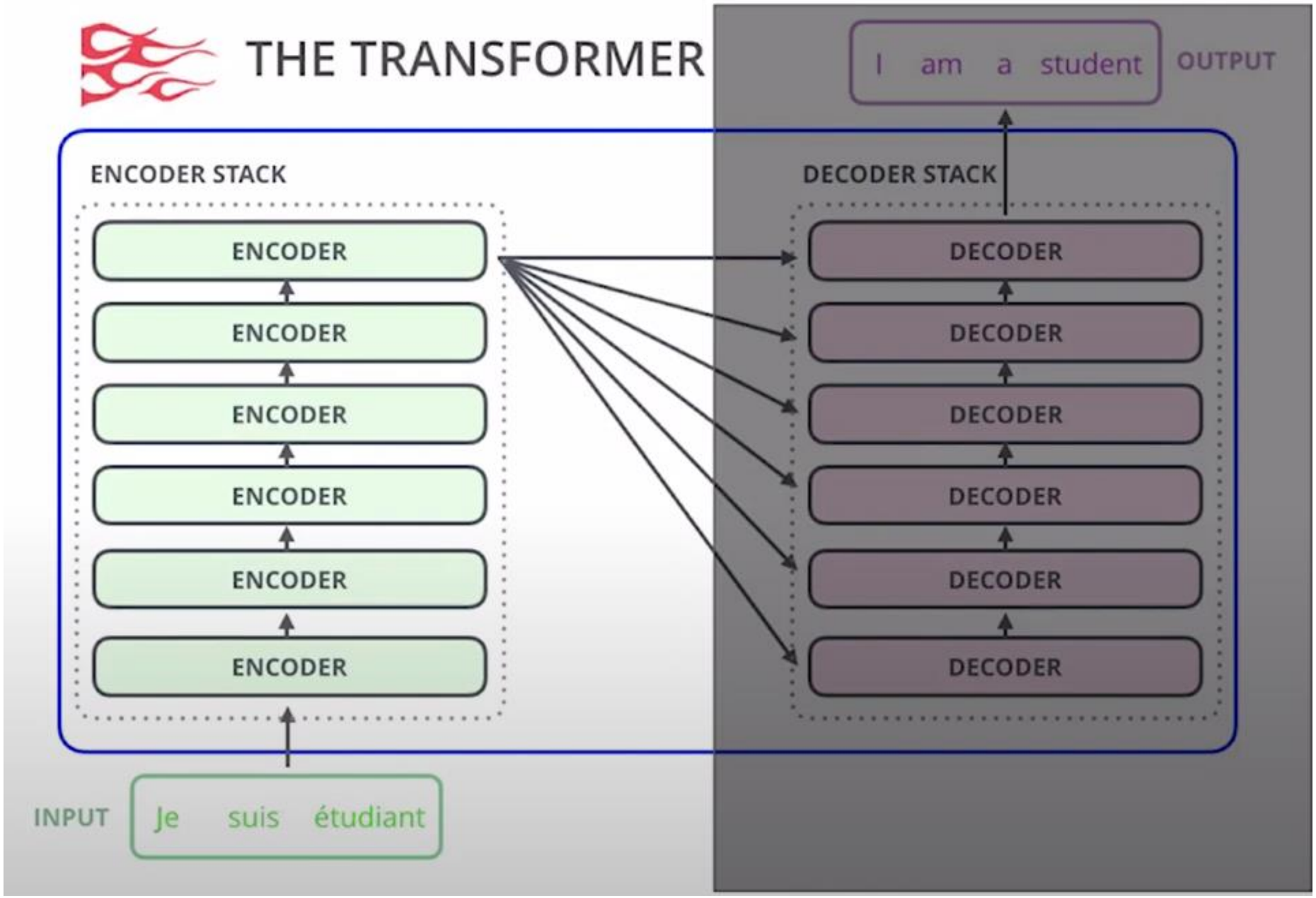
Некоторые сведения

- При обработке естественного языка (NLP) приходится сталкиваться с проблемой недостатка данных об обучении. Современные модели глубокого изучения естественного языка требуют миллионы и миллиарды учебных данных с аннотациями.
- Градиентный бустинг — это техника машинного обучения для задач классификации и регрессии, которая строит модель предсказания в форме ансамбля слабых предсказывающих моделей, обычно деревьев решений.
- Ансамбль — это набор предсказателей, которые вместе дают ответ (например, среднее по всем). Причина почему мы используем ансамбли — несколько предсказателей, которые пытаюсь получить одну и ту же переменную дадут более точный результат.

BERT

- **BERT** (Bidirectional Encoder Representations from **Transformers**) – языковая модель, основанная на архитектуре трансформер (технологии глубоких нейросетей от Google Brain) и предназначенная для предобучения языковых представлений с целью их последующего применения в широком спектре задач обработки естественного языка.
- BERT представляет собой нейронную сеть, основу которой составляет композиция кодировщиков трансформера. BERT является автокодировщиком, то есть является искусственной нейросетью, применяющая обучение без учителя при использовании метода обратного распространения ошибки. В каждом слое кодировщика применяется двустороннее внимание (техника, позволяющая искать связи между частями входных и выходных данных), что позволяет модели учитывать контекст с обеих сторон от рассматриваемого токена, а значит, точнее определять значения токенов.
- Таким образом, BERT решает огромную проблему нехватки данных для обучения нейросетей.

Архитектура Transformer



CatBoost

- CatBoost – библиотека с открытым исходным кодом, разработанная Яндексом. Он предоставляет структуру повышения градиента, которая среди других функций пытается решить для категориальных функций, используя альтернативу, управляемую перестановками, по сравнению с классическим алгоритмом.
- Основным преимуществом является то, что CatBoost может использовать категориальные и текстовые фичи без дополнительной предварительной обработки. Для тех, кто ценит скорость — прогнозы CatBoost в 20–40 раз быстрее, чем у других библиотек бустинга с открытым исходным кодом, и это делает CatBoost полезным для задач, критичным к скорости работа нейросети.

BERT и CatBoost

Данные библиотеки позволяют решать такие задачи, как

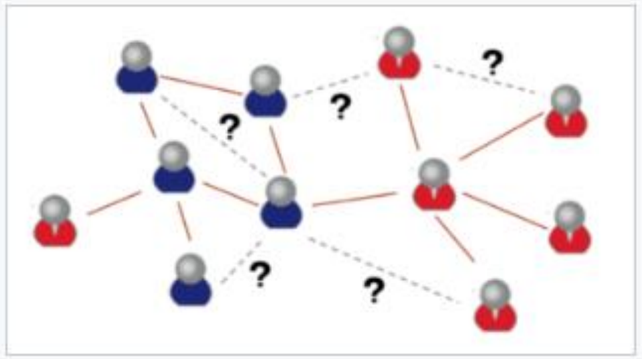
- распознавание спама;
- автоматическая сортировка мнений в отзывах, обсуждениями в социальных сетях, комментариями
- полное понимание написанных человеком статей;
- понимание команд, выданных роботам и голосовым помощникам;
- улучшенный перевод;
- поиск неправильных конфигураций;
- поиск шаблонных ошибок в коде.

Лекция 6.

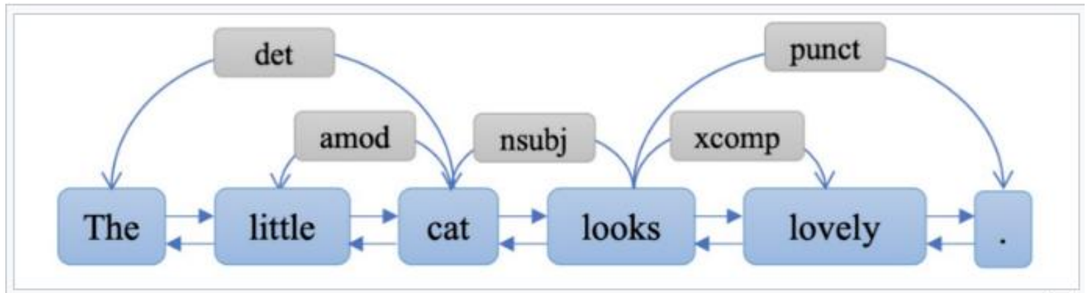
Графовые нейронные сети.

Графовые нейронные сети

- **Графовая нейронная сеть** – тип нейронной сети, которая напрямую работает со структурой графа. При работе с естественными языками, обработке и анализе изображений, построении моделей веб-сетей и еще широком спектре прикладных задач, бывает удобно представлять данные в виде графов. Графовые нейросети используют для обработки текста, а также в компьютерном зрении.



Социальные сети



Анализ текста

Графовые нейронные сети

- Обратим внимание на то, что любое изображение является графом, т.к. это не просто множество пикселей, для каждого пикселя есть соседние и это важная информация, поскольку их значения, как правило, сильно коррелируют, а их сильное различие может означать наличие на изображении в соответствующем месте контрастных переходов: границ объектов. Текст также является графом, правда очень простым, ведь текст это конечное множество слов (и знаков пунктуации, которыми иногда пренебрегают при анализе), на которых введён линейный порядок (есть первое слово, второе и т.д.).

Устройство графовых нейронных сетей

- Один слой графовой нейросети — это обычный полносвязный слой (fully-connected layer) нейронной сети, но веса в нём применяются не ко всем входным данным, а только к тем, которые являются соседями конкретной вершины в графе, в дополнение к ее собственному представлению с предыдущего слоя. Веса для соседей и самой вершины могут задаваться общей матрицей весов или двумя отдельными. Могут добавляться нормализации для ускорения сходимости; могут меняться нелинейные функции активаций, но общая конструкция остается похожей. При этом графовые сверточные сети получили свое название благодаря агрегации информации от своих соседей, хотя гораздо ближе к этому определению стоят графовые механизмы внимания (GAT) или индуктивная модель обучения (GraphSAGE).

Графовые нейронные сети

- Графы развиваются в контексте взаимодействия пользователей с продуктами на платформах электронной торговли. В результате многие компании используют графовые нейросети для создания рекомендательных систем. Обычно с помощью графов моделируют взаимодействие пользователей с товарами, обучают эмбеддингам с учетом правильно подобранной отрицательной выборки, и с помощью ранжирования результатов выбирают персонализированные предложения по товарам и в реальном времени показывают конкретным пользователям. Одним из первых сервисов с таким механизмом стал Uber Eats: нейросеть GraphSage подбирает рекомендации продуктов питания и ресторанов.

Графовые нейронные сети

- Хотя в случае с рекомендациями продуктов питания графы получаются относительно небольшими из-за географических ограничений, однако в некоторых компаниях применяются нейросети с миллиардами связей. Например, китайский гигант Alibaba запустил в эксплуатацию графовые модели и графовые нейросети применительно к миллиардам пользователей и товаров. Одно только создание таких графов — кошмар для разработчиков. Но благодаря конвейеру Aligraph можно всего за пять минут построить граф на 400 млн узлов.

Графовые нейронные сети

- Объекты в реальном мире глубоко взаимосвязаны, поэтому изображения этих объектов можно успешно обрабатывать с помощью графовых нейросетей. Например, можно воспринимать содержимое изображения через графы сцены — набор объектов на картинке с их взаимосвязями. Графы сцен применяются для поиска изображений, понимания их содержимого и осмысления, добавления субтитров, ответов на визуальные вопросы и генерирования изображений. Эти графы позволяют сильно повысить производительность моделей.

Графовые нейронные сети

- Фармацевтические компании активно ищут новые методы разработки лекарств, жёстко конкурируя друг с другом и тратя на исследования миллиарды долларов. В биологии можно с помощью графов представлять взаимодействия на разных уровнях. Например, на молекулярном уровне связи между узлами будут обозначать межатомные силы в молекуле, или взаимодействие между аминокислотными основаниями в белке. В более крупном масштабе графы могут представлять взаимодействие между протеинами, и РНК или продуктами обмена веществ

Лекция 7.
Применение изученных подходов для хранения и анализа событий информационной безопасности.

Обеспечение безопасности корпоративных сетей

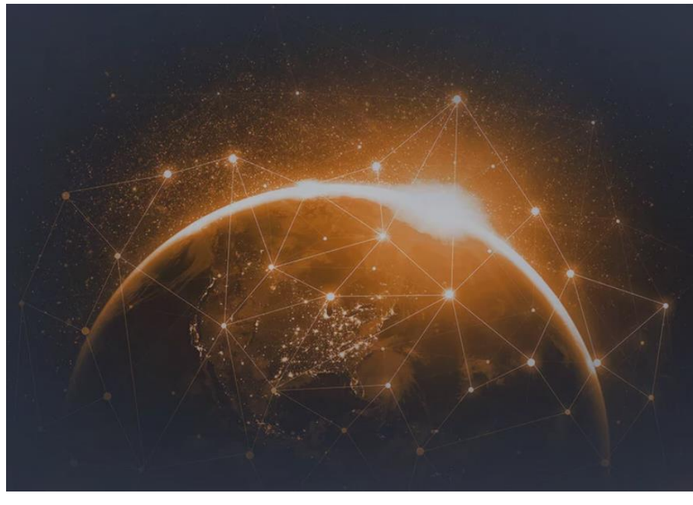
Для обеспечения безопасности целой компании необходимо уметь проводить анализ подозрительной активности в реальном времени. В этом могут помочь нейросети. Достичь этого можно путём глубинного анализа высокоскоростного трафика. Традиционно процесс управления киберзащитой был наукоемким и трудоемким. Благодаря быстро растущей глубине анализа больших данных время, затрачиваемое на корреляцию данных для целей криминалистики и создание действенных мер безопасности, должно значительно сократиться.

Анализ сетевого трафика позволяет решать следующие задачи:

- выявлять проблемы в работе сети (в том числе, несанкционированную активность);
- восстанавливать потоки данных («прослушивание»);
- предотвращать различные сетевые атаки;
- вести статистику и определять характеристики сетевых соединений.

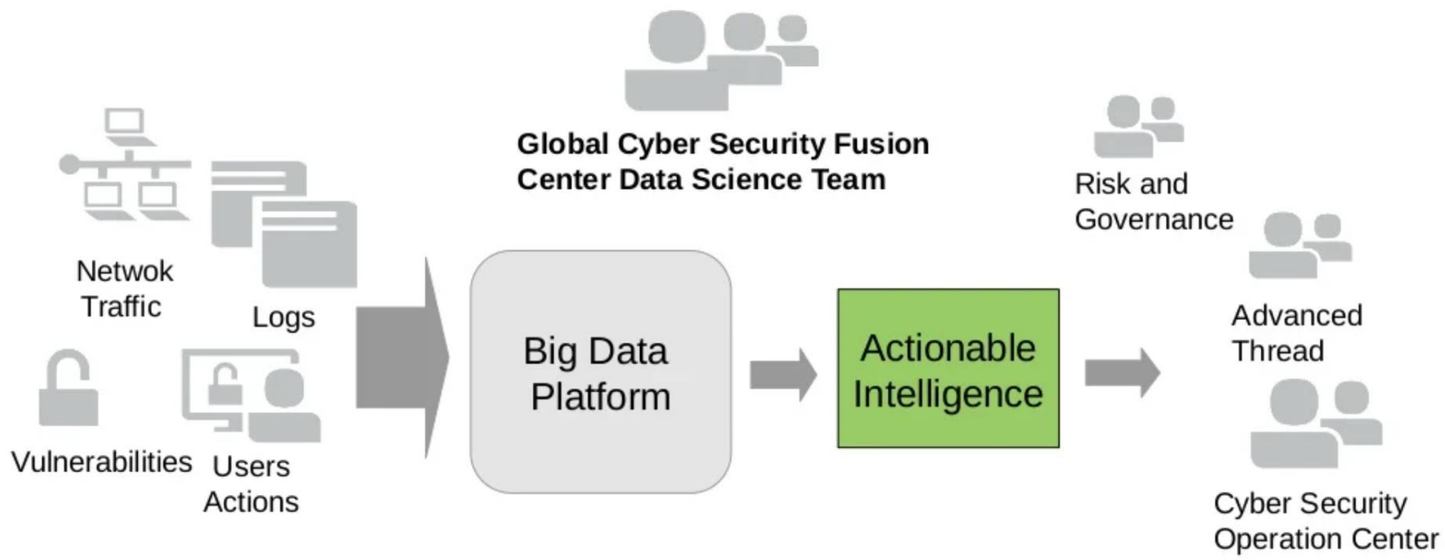
OSINT

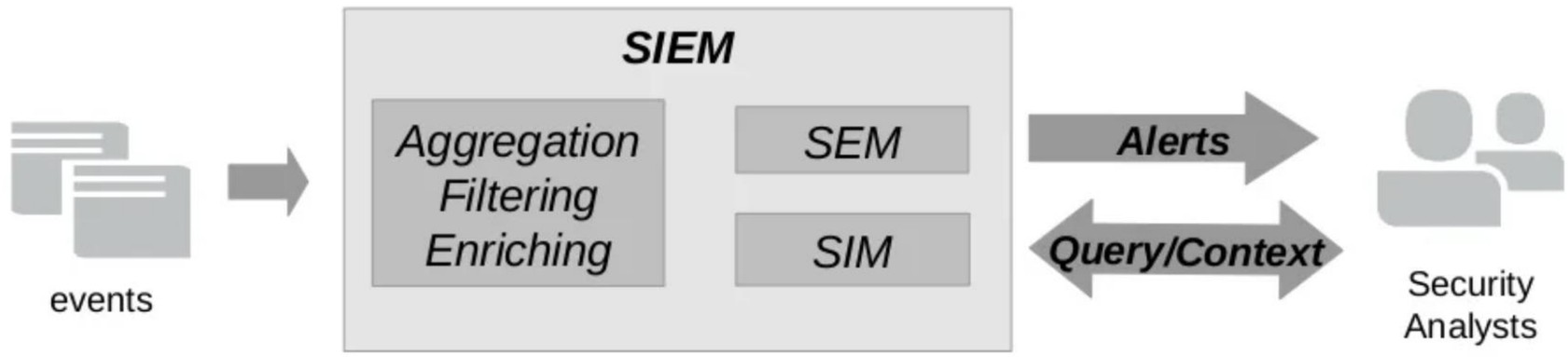
- **OSINT** (разведка на основе открытых источников) является неотъемлемой частью наступательной безопасности. Из себя он представляет поиск, сбор, хранение и анализ разнородных данных. Разумеется, такая задача порой может занимать большое количество времени и сил. Поэтому хорошим решением могут выступать методы, применяемые в big data.
- Также не обойтись и без big data и компьютерного зрения при анализе фото и видео кадров с городских камер. Такие технологии применяют государственные службы для поиска и выявления преступников. Анализ таких данных позволяет выполнять поиску по лицам, манере хотьбы, одежде.



Apache Hadoop

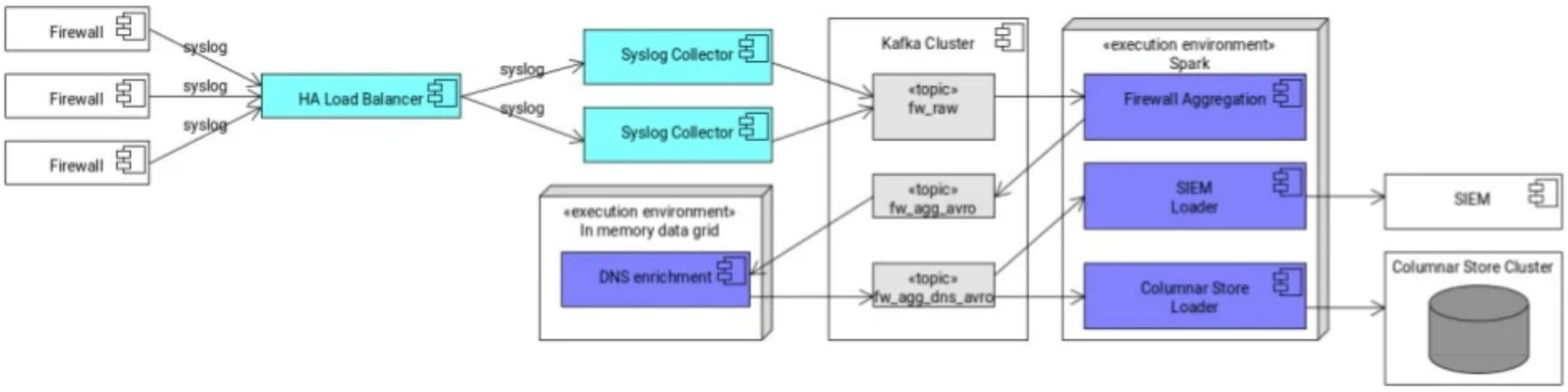
- В настоящее время многие операции по кибербезопасности используют программное обеспечение Security information and event Management (**SIEM**) для понимания и реагирования на ситуацию с безопасностью. Использование традиционной SIEM в крупной компании, такой как HP Enterprise, является сложной задачей из-за увеличения объема и скорости передачи данных. Для решения данной проблемы **Apache Hadoop** предлагает следующую структуру программных комплексов:





Big Data Processing

Firewall logs aggregation



Apache Hadoop

- Таким образом, предложенный программный комплекс **Apache Hadoop** позволяет собрать всю необходимую информацию.
- Сначала результаты логов файерволов, снифферов, систем обнаружения вторжений проходят через **HA Load Balancer**. Данный балансировщик в зависимости от источников, а также типов данных генерирует соответствующие события для **Syslog Collector** (системного сборщика). Данные, обработанные системными сборщиками, в свою очередь при необходимости дополняются данными, собранными **Dns Enrichment**, и отправлются в **Kafka Cluster**. После происходит финальная обработка всех собранных данных и формирования отчётов для специалистов по кибербезопасности.

NLP (Обработка естественного языка)

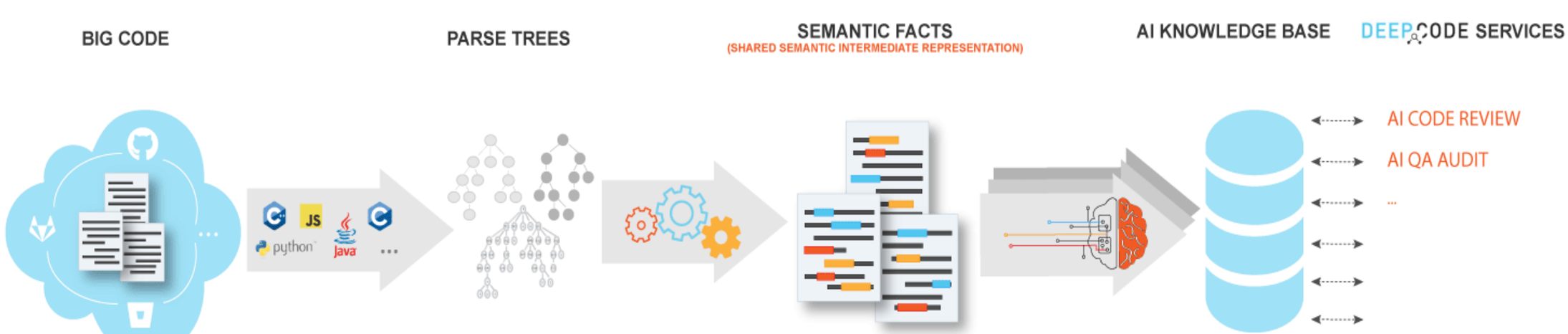
- NLP было разработано, чтобы позволить машинам научиться общаться как люди с людьми. Многие сервисы, которые мы используем сегодня, используют машинные коммуникации либо друг с другом, либо в переводе, чтобы стать понятными людям. NLP имеет место и в безопасности.
- Обработка языка может позволить защититься от попыток фишинга, а также спама. Не остаётся без внимания и обычный анализ документов и отчётов по кибербезопасности. Если NLP может позволить выделить ключевые идеи в обычной статье, почему не может в отчёте? Благодаря автоматизации анализа отчётов, реагирование на инциденты может значительно ускориться, а порой даже и принять характер реагирования в реальном времени.
- Также, NLP может помочь с обнаружением неправильных конфигураций, ошибок в коде.

Статический анализ кода

Статический анализ кода позволяет выполнять сканирование кода на наличие опечаток, ошибок и багов без непосредственного выполнения самого кода.

Выделяют несколько типов уровней статического анализа кода:

- **Единичный уровень**
  Анализ, который выполняется в рамках определенной программы или подпрограммы, без подключения к контексту этой программы.
- **Технологический уровень**
  Анализ, который учитывает взаимодействия между единичными программами, чтобы получить более целостное и семантическое представление о программе в целом, чтобы найти проблемы и избежать очевидных ложных срабатываний. Например, можно статически анализировать стек технологий Android, чтобы найти ошибки разрешений.
- **Системный уровень**
  Анализ, который учитывает взаимодействия между единичными программами, но не ограничивается одной конкретной технологией или языком программирования.
- **Бизнес-уровень**
  Анализ, учитывающий условия, правила и процессы бизнес-уровня, которые реализуются в программной системе для ее функционирования в рамках деятельности уровня предприятия или программы / миссии. Эти элементы реализуются, не ограничиваясь одной конкретной технологией или языком программирования, и во многих случаях распределяются по нескольким языкам, но статически извлекаются и анализируются для понимания системы для обеспечения миссии.



STEP 1:
PARSING

This is the only language specific part of our platform and enables us to add support for any/custom language in a matter of weeks.

STEP 2:
SOLVERS
(DECLARATIVE STATIC ANALYSIS)

Our custom language-independent linear-complexity Datalog solvers allow us to analyze huge repositories in a matter of seconds.

STEP 3:
ML ALGORITHMS

Our custom Semantic Facts representations allows us to run powerful ML algorithms to understand the structure, function, and intent of the code.

Доклад окончен!
Спасибо за внимание!