Саратовский национальный исследовательский государственный
университет
имени Н.Г. Чернышевского

В.Ф. Кабанов, А.В. Бурмистров

ОСНОВНЫЕ ПРИНЦИПЫ ОБРАБОТКИ ДАННЫХ. СОВРЕМЕННЫЙ АСПЕКТ

Учебное пособие
для студентов
Саратовского национального исследовательского государственного
университета

Саратов 2018

УДК 004.41/.42
ББК 32.973 – 018.1

С

Кабанов В.Ф., Бурмистров А.В.

С ОСНОВНЫЕ ПРИНЦИПЫ ОБРАБОТКИ ДАННЫХ. СОВРЕМЕННЫЙ АСПЕКТ: учебное пособие для студ. Саратовского гос. ун-та. – Саратов, 2018. - 49 с.

ISBN

В пособии в доступной форме рассмотрены основные понятия и принципы функционирования постреляционных баз данных, облачных технологий и больших данных. Рассмотрены особенности реализации современных технологий обработки данных в России.

Учебное пособие предназначено для студентов факультета нано- и биомедицинских технологий, а также для студентов физических и инженерных специальностей других факультетов и институтов Саратовского национального исследовательского государственного университета.

Рекомендует к публикации

Кафедра физики полупроводников
факультета нано- и биомедицинских технологий
Саратовского национального исследовательского государственного университета

УДК 004.41/.42
ББК 32.973 – 018.1

ISBN

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

В настоящее время по данным аналитиков, пока именно реляционные системы управления базами данных (СУБД) используются в абсолютном большинстве крупных проектов, связанных с внедрением систем управления базами данных. СУБД – одна из фундаментальных составляющих компьютерного обеспечения информационных процессов, являющаяся основой для построения большинства современных информационных систем. Главной функцией СУБД является эффективное хранение и предоставление данных в интересах конкретных прикладных задач.

Коммерческие СУБД ведут свою историю с середины 60-х годов, когда компанией IBM был выпущен первый продукт данного класса – иерархическая СУБД IMS. В начале 70-х годов Эдгаром Коддом были заложены основы реляционной модели данных, был разработан структурированный язык запросов SQL, а в 80-х годах были созданы промышленные СУБД, которые в скором времени заняли доминирующее положение. В настоящее время ведущая тройка игроков – Microsoft, Oracle и IBM – полностью контролируют рынок, а их флагманские продукты Microsoft SQL Server, Oracle Database и IBM DB2 вместе занимают долю рынка около 90%. Рынок СУБД активно растет и, по мнению аналитиков Forrester, к 2013 году его общий объем достиг 32 млрд. долл.

В настоящее время СУБД в основном приобретаются для использования со сложными и дорогостоящими программными продуктами, ориентированными на автоматизацию корпоративных бизнес-процессов - типичным примером здесь являются системы класса ERP **(**Enterprise Resource Planning - планирование ресурсов предприятия). Большинство таких продуктов имеют высокую критичность для бизнеса, что обуславливает зависимость компаний от функционирования СУБД, серверов баз данных и качества обслуживания инфраструктуры. В свою очередь это выдвигает повышенные требования к выбору СУБД, который зависит не только от ее функциональных возможностей, но и от набора приложений, с которыми она будет взаимодействовать, а также от стоимости лицензий и наличия подготовленного персонала для ее администрирования.

Основными тенденциями, давшими повод для проведения различных масштабных исследований в области баз данных стали:

1. **Экспоненциальный рост данных**. Объем данных, в том числе синтетических, генерируемых автоматизированными системами, значительно возрос. Увеличилось и число прикладных областей, в которых требуется обработка больших объемов данных. К таким областям теперь относятся не только традиционные

корпоративные приложения и поиск в Web, но также и научные исследования, обработка естественных языков, анализ социальных сетей и т.п.

2. **Значительное усложнение структур используемых данных**. Простые виды данных в виде чисел и символьных строк стали дополнятся многочисленной мультимедийной информацией, пространственными, процедурными данными и большим количеством прочих сложных форматов.
3. **Широкое распространение дешевых высокопроизводительных аппаратных средств**. Ежегодно мы наблюдаем рост вычислительных возможностей микропроцессоров, увеличение емкости и снижение стоимости доступных и удобных в эксплуатации устройств дисковой и оперативной памяти.
4. **Активное развитие средств коммуникации и «всемирной паутины» WorldWideWeb**. WWW становится единой информационной средой, пронизывающей весь мир и объединяющей огромное число пользователей и электронных устройств.
5. **Появление новых важных областей применения СУБД**. В первую очередь, это связано с интеллектуальным анализом данных, хранилищами данных, а в последнее время – с параллельными вычислениями и «облачными» технологиями.

ГЛАВА 1. ОСНОВНЫЕ ПОНЯТИЯ БАЗ ДАННЫХ

1.1. Базы данных. Основные модели представления данных

Одной из наиболее распространенных форм реализации информационных технологий в настоящее время являются **базы данных.**

База данных (БД) представляет собой совокупность специальным образом организованных данных, хранимых в памяти вычислительной системы и отображающих состояние объектов и их взаимосвязей в рассматриваемой предметной области.

Другими словами базу данных можно определить как совокупность взаимосвязанных данных, характеризующихся возможностью использования для большого количества приложений, возможностью быстрого получения и модификации необходимой информации, минимальной избыточностью информации, независимостью от прикладных программ, общим управляемым способом поиска.

Также важными понятиями информационных технологий при работе с базами данных являются:

- система управления базами данных (СУБД);
- словарь данных;
- администратор баз данных и др.

Система управления базами данных (СУБД) – это комплекс языковых и программных средств, предназначенный для создания, ведения и совместного использования БД пользователями.

Словарь данных предназначен для централизованного хранения информации о структурах данных, взаимосвязях файлов БД друг с другом, типах данных и форматах их представления, кодах защиты, разграничения доступа и т.д.

Администратор БД – это лицо или группа лиц, отвечающих за выработку требований к БД, ее проектирование, создание, использование и сопровождение.

Можно сформулировать основные ***требования к базам данных*** с точки зрения пользователя.

- БД должна соответствовать актуальным информационным потребностям пользователя.
- БД должна обеспечивать получение требуемых данных за приемлемое время (т.е. отвечать заданным требованиям производительности).
- БД должна легко расширяться при реорганизации и расширении предметной области.

- БД должна легко изменяться при изменении программной и аппаратной среды.
- Помещенные в БД корректные данные должны оставаться корректными.
- Данные до включения в БД должны проверяться на достоверность.
- Доступ к данным, размещаемым в БД, должны иметь только лица с соответствующими полномочиями.

Моделью представления данных или моделью базы данных называют *логическую структуру хранимых в базе данных.* Структуры данных в существующих моделях БД обладают относительной устойчивостью. Минимальная избыточность и возможность быстрой модификации позволяют поддерживать данные на соответствующем уровне актуальности. Одно из основных свойств БД – независимость данных и использующих их программ (т.е. изменение данных не приводит к изменению программ и наоборот). Структура базы данных предполагает формирование логических записей, их элементов и взаимосвязей между ними. Принято выделять следующие типы взаимосвязей:

- **один к одному** (1 : 1) – одна запись может быть связана с одной записью,
- **один ко многим** (1 : N) – одна запись взаимосвязана со многими другими,
- **многие ко многим** (M : N) – одна запись может входить в отношения со многими другими записями в различных вариантах.

Хранимые в базе данные имеют определенную логическую структуру (модель). В соответствии с применением определенного вида взаимосвязей выделяют следующие важнейшие **модели данных:**

- иерархическая;
- сетевая;
- реляционная;
- объектно-ориентированная.

Иерархическая модель данных строится по принципу иерархии типов объектов, т.е. один объект является **главным**, а остальные, находящиеся на низших уровнях иерархии, - **подчиненными**. Между главным и подчиненным объектами устанавливается взаимосвязь «один ко многим». Данная модель удобна для работы с иерархически упорядоченной информацией и громоздка для информации со сложными логическими связями.

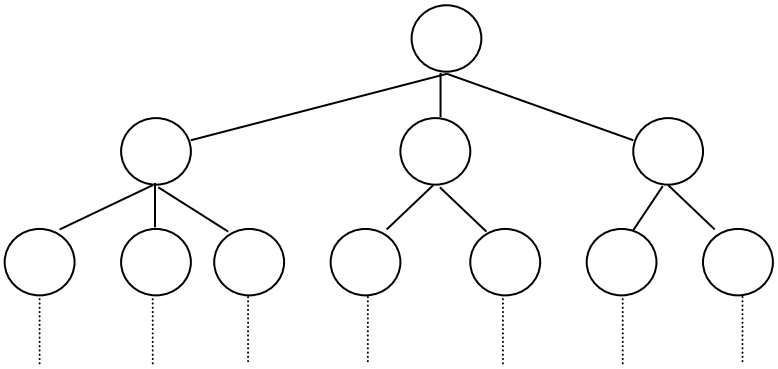


Рис.1.1. Схема иерархической модели данных

Достоинством иерархической модели является достаточно очевидное описание структуры данных на логическом и физическом уровне. *Недостаток* модели – жесткая структура взаимосвязей между элементами данных (потеря информационной гибкости).

В ***сетевой модели*** данных понятия главного и подчиненных объектов несколько расширены. Любой объект может быть и главным и подчиненным, то есть могут быть реализованы связи «многие ко многим».

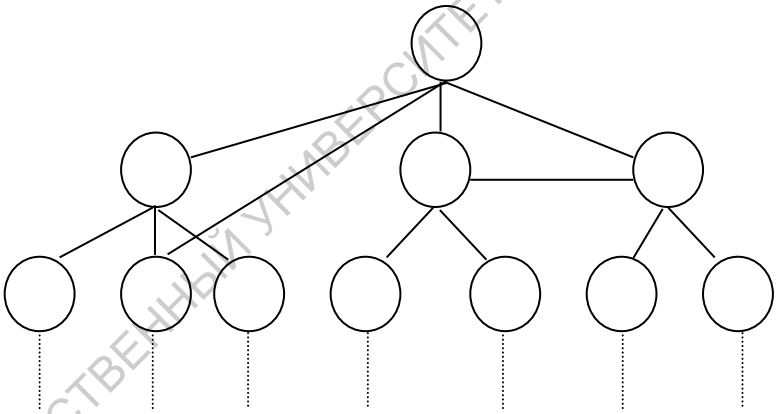


Рис.1.2. Схема сетевой модели данных

Достоинством сетевой модели данных является более высокая информационная гибкость по сравнению с иерархической моделью, эффективность затрат памяти и оперативность. *Недостаток* модели – достаточно сложное математическое описание системы и, как следствие, высокая сложность схемы базы данных, построенной на ее основе.

В ***реляционной модели*** данных объекты и взаимосвязи между данными представляются с помощью двумерных таблиц (рис. 1.3.). Каждая таблица представляет собой один объект и состоит из строк (записей) и столбцов (полей). Основное *достоинство* реляционной модели – простота, понятность для пользователя и удобство физической реализации на компьютере. Благодаря этому реляционная модель получила

наибольшее распространение в СУБД для персональных компьютеров. *Недостатком* модели является сложность описания иерархических и сетевых связей.

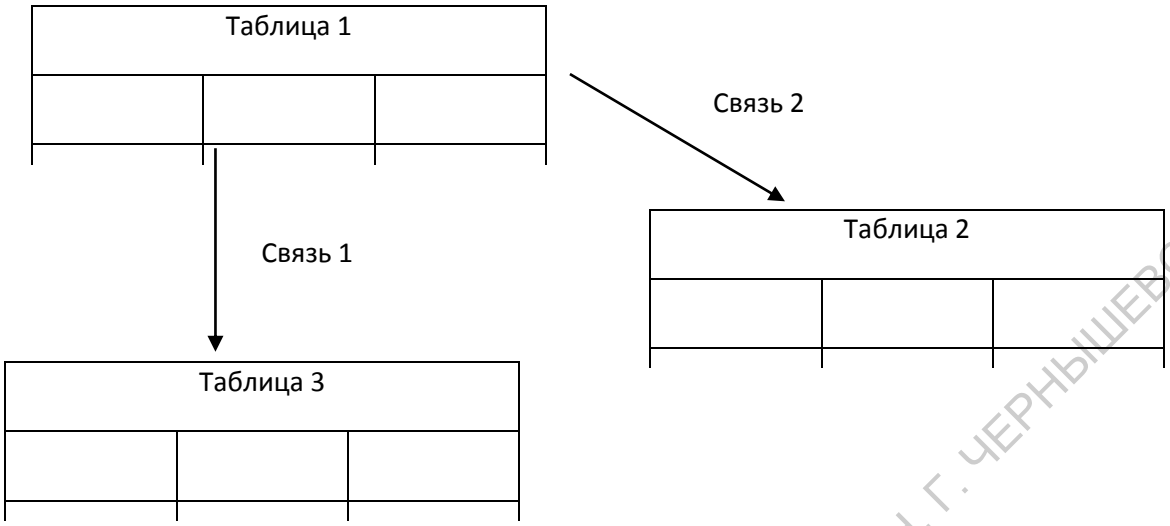


Рис. 1.3. Схема реляционной модели данных

Объектно - ориентированная модель в определенном смысле объединяет в себе две модели данных: реляционную и сетевую, и используется для создания крупных баз данных со сложными структурами данных.

Объектные базы данных строятся на основе объектно-ориентированного программирования, где акцент сделан не на программные структуры, а на объекты. Под объектом понимается достаточно крупный блок функционально взаимосвязанных данных. Типы и структуры данных, из которых состоит объект, могут быть различными у разных объектов и создаваться программистом на основе стандартных типов данных выбранного языка программирования. Объект можно определить как программно-связанный набор методов (функций) и свойств, выполняющих одну функциональную задачу. (Примерами использования объектных баз данных являются программные продукты **1С**).

Важнейшими понятиями, связанными с объектом являются *свойство, событие и метод*.

Свойство – это характеристика, с помощью которой описываются внешний вид и работа объекта.

Событие – это действие, которое связано с объектом. Событие может быть инициировано пользователем, прикладной программой или операционной системой.

Метод – это функция или процедура, управляющая работой объекта при его реакции на событие.

Для выполнения действий над данными в рассматриваемой модели БД применяются логические операции с использованием объектно-ориентированных механизмов *инкапсуляции, наследования и полиморфизма***.**

- ***Инкапсуляция*** ограничивает область видимости имени свойства пределами того объекта, в котором оно определено. Смысл такого свойства будет определяться тем объектом, в который оно инкапсулировано.
- ***Наследование*** распространяет область видимости свойства на всех потомков объекта.
- ***Полиморфизм*** в объектно-ориентированных языках программирования означает способность одного и того же программного кода работать с разнотипными данными. Другими словами, он означает допустимость в объектах разных типов иметь методы (процедуры или функции) с одинаковыми именами.

Создание объектов – достаточно трудоемкая задача для программистов. В настоящее время объектные модели данных не имеют строгой теоретической основы, что затрудняет их создание и использование. Однако увеличение возможностей ПК, развитие средств мультимедиа и компьютерных сетей предполагает реализацию надежных систем объектных баз данных.

Основным *достоинством* объектно-ориентированной модели данных является возможность отображения информации о сложных связях объектов. Объектно-ориентированная модель данных позволяет идентифицировать отдельную запись базы данных и определять функции их обработки.

Недостатками объектно-ориентированной модели являются высокая понятийная сложность, неудобство обработки данных и низкая скорость выполнения запросов.

1.2. Состав и основные функции систем управления базами данных

Для работы с базами данных используются системы управления базами данных (**СУБД)**.

Современная СУБД включает в свой ***состав***:

- программные средства создания баз данных,
- средства для работы с данными,
- сервисные средства.

С помощью средств создания БД проектировщик (используя язык описания данных) переводит логическую модель БД (то есть некоторое формализованное отображение структуры данных предметной области) в физическую структуру; разрабатывает программы, реализующие основные операции с данными (применяя язык манипулирования данными). При проектировании используются визуальные средства и программа – отладчик, с помощью которой соединяются и тестируются отдельные блоки программы управления конкретной базы данных.

Средства работы с данными (предназначены для пользователя) позволяют установить удобный интерфейс с пользователем, создать необходимую функциональную конфигурацию экранного представления выводимой и вводимой информации, производить операции с данными, текстовыми и графическими экранными объектами.

Сервисные (дополнительные) средства позволяют при проектировании и использовании БД привлечь другие системы (например, использование программ офиса или сетевых серверных ресурсов).

К основным ***функциям*** СУБД с точки зрения пользователя можно отнести:

- **Определение данных**, т.е. описание таблиц базы данных. Прежде чем заносить данные в таблицы, необходимо определить структуру (поля) этих таблиц: название поля, его тип.
- **Обработка данных**. Любая СУБД позволяет выполнять четыре простейшие операции с данными:
    - добавить в таблицу одну или несколько записей;
    - удалить из таблицы одну или несколько записей;
    - обновить значение полей в одной или нескольких записях;
    - найти одну или несколько записей, удовлетворяющих заданному условию.
- **Управление данными**. Под управлением данными обычно понимают защиту данных от несанкционированного доступа, поддержку многопользовательского режима, работу с данными и обеспечение целостности и согласованности данных.

Перечисленные выше функции СУБД используют следующие основные функции более низкого уровня:

- управление данными во внешней памяти;
- управление буферами оперативной памяти;
- управление транзакциями;
- ведение журнала изменений в БД;

- обеспечение целостности и безопасности БД.

СУБД различаются по моделям БД, с которыми они работают. Например, если модель реляционная, то и используемая СУБД реляционная. Наиболее популярными персональными (или настольными) реляционными СУБД (то есть обеспечивающих создание персональных БД и приложений, работающих с ними) в настоящее время являются: ***Paradox***, различные версии ***dBASE***, ***Ms Access***.

ГЛАВА 2. ПОСТРЕЛЯЦИОННЫЕ СУБД

Главным недостатком реляционных СУБД считается присущая этим системам ограниченность использования в областях, в которых требуются достаточно сложные структуры данных. Одним из основных аспектов традиционной реляционной модели данных является атомарность (единственность и неделимость) данных, которые хранятся на пересечении строк и столбцов таблицы. Такое правило было заложено в основу реляционной алгебры при ее разработке как математической модели данных. Кроме того, специфика реализации реляционной модели не позволяет адекватно отражать реальные связи между объектами в описываемой предметной области. Данные ограничения существенно мешают эффективной реализации современных приложений, которые требуют уже несколько иных подходов к организации данных.

Основной принцип реляционной модели – устранять повторяющиеся поля и группы с помощью процесса, который называется нормализацией. Плоские нормализованные таблицы универсальны, просты в понимании и теоретически достаточны для представления данных любой предметной области. Они хорошо подходят для приложений, связанных с хранением и отображением данных в традиционных отраслях, таких как банковские или учетные системы, но их применение в системах, основанных на более сложных структурах данных, часто является затруднительным. В основном, это связано с примитивностью механизмов хранения данных, лежащих в основе реляционной модели. Опыт разработки прикладных информационных систем показал, что отказ от атомарности значений ведет к качественно полезному расширению модели данных. Введение в реляционную модель возможности использовать многозначные поля как самостоятельные вложенные таблицы, при условии, что вложенная таблица удовлетворяет общим критериям, позволяет естественным образом расширить возможности реляционной алгебры. В классическом понимании именно такая модель данных называется **постреляционной.**

Постреляционная модель использует многомерные структуры, позволяющие хранить в полях таблицы другие таблицы. В связи с этим ее часто называют "не первой нормальной формой" или "многомерной базой данных" (рис. 2.1.). В качестве языка в данной модели запросов используется расширенный SQL, позволяющий извлекать сложные объекты из одной таблицы без операций соединения. Реляционные и постреляционные СУБД различаются способами хранения и индексирования данных, во всем остальном они схожи. Первыми постреляционными СУБД, получившими достаточно

большую известность, стали Universe компании Ardent (впоследствии купленной Informix, которую, в свою очередь, приобрела IBM) и ADABAS компании Software AG.

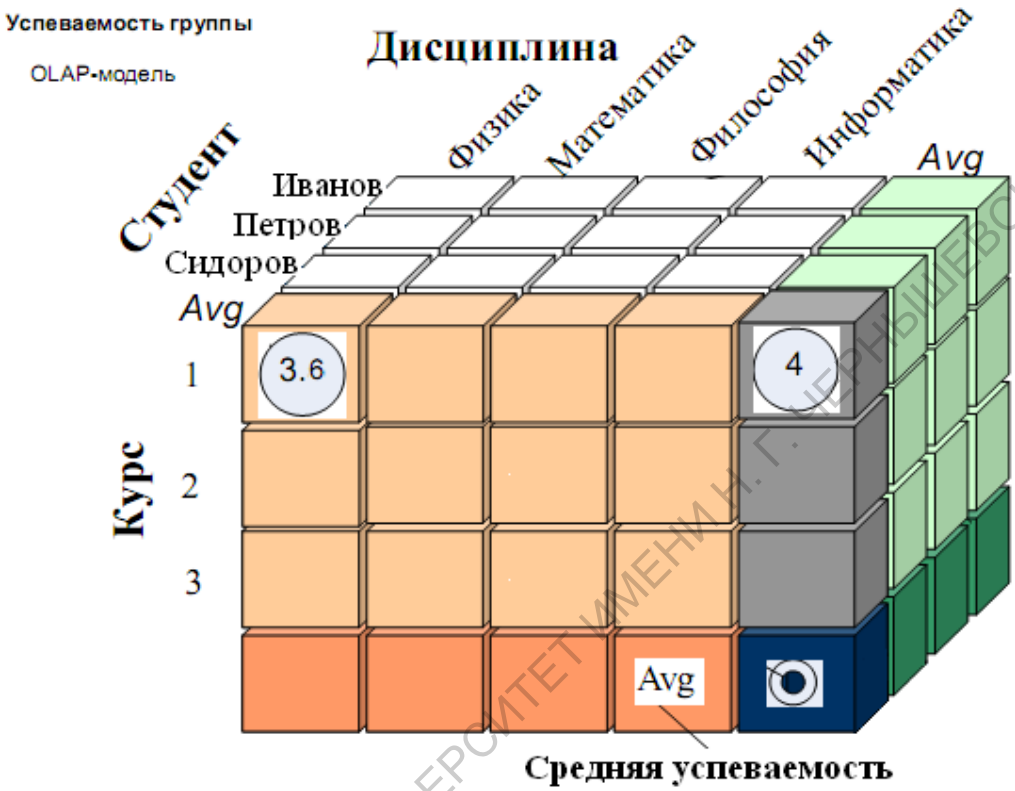


Рис. 2.1. Модель многомерной БД

2.1. Понятие объектно-реляционных баз данных

До недавнего времени выбор типа СУБД ограничивался только реляционными и объектно-реляционными СУБД. Но в настоящее время многие системы не подходят для создания сложных специализированных приложений. Для того, чтобы обеспечить возможность применения реляционных СУБД в этих приложениях, необходимо расширить функциональные возможности существующих систем. При рассмотрении новых сложных специализированных приложений баз данных, можно заметить, что в них широко используются такие объектно-ориентированные компоненты, как расширяемая пользователем система типов, инкапсуляция, наследование, полиморфизм, динамическое связывание методов, использование составных объектов, а также поддержка идентичных объектов. Наиболее очевидный способ преодоления ограничений реляционной модели заключается в ее расширении указанными функциями. Именно этот подход был предпринят во многих прототипах расширенных реляционных систем, хотя в каждой из них реализован свой собственных и отличный от других набор функциональных возможностей. Однако не существует какой-то общепринятой расширенной реляционной

модели, а скорее имеется несколько таких моделей, характеристики которых зависят от способа и степени реализации внесенных расширений. Во всех моделях используются одинаковые базовые реляционные таблицы и язык запросов, включено понятие объекта, а в некоторых дополнительно реализована возможность сохранения методов (или процедур, или триггеров) таким же способом, что и базе данных.

Для систем с расширенной реляционной моделью данных используются самые разные термины. Сначала применялся термин расширенная реляционная СУБД (Extended Relational DBMS - ERDBMS). Однако в последние годы используется более информативный термин объектно-реляционная СУБД, или *ОРСУБД* (Object-Relational DBMS - ORDBMS), в котором содержится указание на использование понятия объект. Чаще всего используется термин Объектно-реляционная СУБД или ОРСУБД. Три ведущих фирмы в области разработки ОРСУБД, а именно Oracle, Informix, IBM, расширили свои системы до уровня ОРСУБД, хотя их функциональные возможности немного отличаются. Концепция ОРСУБД, как комбинации ООСУБД и РСУБД, считается достаточно привлекательной благодаря применению знаний и опыта, которые были накоплены за время работы с РСУБД.

Рис. 2.2. Представление ОРСУБД

Разработка стандартов в этой сфере построена на расширении стандарта языка SQL. Основными преимуществами расширенной реляционной модели данных являются повторное и совместное использование компонентов. Например, в приложении может понадобиться использование данных пространственного типа, представляющие собой точки, линии, и многоугольники, со связанными с ними функциями, которые вычисляют расстояние между точками, расстояние между точкой и линией, проверяют наличие точки

в многоугольнике и т.д. При правильном проектировании с учетом новых возможностей подобный подход позволяет организациям воспользоваться преимуществами новых расширений эволюционным путем без утраты преимуществ, получаемых от использования компонентов и функций уже существующей базы данных.

Очевидным недостатком подхода с использованием ОРСУБД являются сложность и связанные с ней повышенные расходы. Простота и ясность, свойственная реляционной модели, утрачивается при использовании подобных типов расширения. Однако, есть мнение, что расширения РСУБД предназначены для незначительного количества приложений, причем в последних не может быть достигнута оптимальная производительность при использовании имеющейся реляционной технологии.

2.2. Объектно-реляционные СУБД

Понятие СУБД третьего поколения, которыми и являются объектно-реляционные СУБД, появилось после опубликования группой известных специалистов в области баз данных «Манифеста систем баз данных третьего поколения». Основные принципы СУБД третьего поколения, обозначенные в манифесте:

1. ***Помимо традиционных услуг по управлению данными, СУБД третьего поколения должны обеспечить поддержку более развитых структур объектов и правил.*** Более развитая структура объектов характеризует средства, необходимые для хранения и манипулирования нетрадиционными элементами данных (тексты, пространственные данные, мультимедиа).
2. ***СУБД третьего поколения должны включить в себя СУБД второго поколения.*** Системы второго поколения внесли решающий вклад в двух областях – непроцедурный доступ с помощью языка запросов SQL и независимость данных. Эти достижения обязательно должны учитываться в системах третьего поколения.
3. ***СУБД третьего поколения должны быть открыты для других подсистем.*** Это включает оснащение разнообразными инструментами поддержки принятия решений, доступом из многих языков программирования, интерфейсами к существующим популярным системам и бизнес-приложениям, возможностью запуска приложений из базы данных на другой машине и распределенной СУБД. Весь набор инструментов и СУБД должен эффективно функционировать на разнообразных аппаратных платформах с различными операционными системами. Кроме того, СУБД, рассчитывающая на широкую сферу применения, должна быть оснащена языком четвертого поколения (4GL).

Помимо отказа от нормализации, постреляционные СУБД позволяют хранить в полях отношений данные абстрактных, определяемых пользователями типов. Это дает возможность решать задачи нового уровня, хранить объекты и массивы данных, ориентированные на конкретные предметные области, а также делает сходным постреляционные СУБД с еще одним классом – объектно-ориентированными СУБД. Внедрение объектного подхода в традиционную реляционную модель дало возможность появлению еще одного направления – объектно-реляционных СУБД. Первым представителем данного класса систем принято считать систему Informix Universal Server одноименной компании.

Известно, в основе объектно-ориентированного подхода к моделированию предметных областей лежат такие понятия, как объект и свойства инкапсуляции, наследования и полиморфизма. В отличие от реляционных СУБД при проектировании объектно-ориентированных БД не требуется декомпозиция и нормализация объектов, выделенных на этапе концептуального проектирования. Объекты представляются в том же виде, в каком они существуют в реальности, что наделяет объектно-ориентированные структуры наглядностью и позволяет значительно сократить время на их проектирование и разработку.

Одной из наиболее известных постреляционных СУБД является система Postgres, созданная в середине 80-х годов прошлого века под руководством одного из ведущих разработчиков СУБД Майкла Стоунбрейкера. В Postgres традиционная реляционная модель была расширена за счет внедрения механизмов управления объектами, которые позволяли хранить и эффективно управлять нетрадиционными типами данных. Также в Postgres поддерживалась многомерная темпоральная модель хранения и доступа к данным. Все основные идеи и разработки Postgers были продолжены и развиты в свободно распространяемой СУБД Postgre SQL, которая в настоящее время является наиболее развитой открытой СУБД. Часто постреляционными называют также СУБД, которые позволяют представлять данные, как в виде реляционных таблиц, так и классов объектов. Типичным представителем данного вида СУБД является система Cache компании Inter Systems. По мнению ее разработчиков, в данной системе наиболее эффективно совмещены реляционный и объектный подходы, основанные, соответственно, на стандартах SQL-92 и ODMG 2.0. Механизмы работы с объектами и реляционными таблицами находятся на одном логическом уровне, что обеспечивает более высокую скорость доступа и работы с данными и функциональную полноту. Также Cache использует многомерную модель хранения данных и оптимизирована для обработки транзакций в системах с большими и сверхбольшими БД (сотни гигабайт, терабайты) и большим количеством (тысячи, десятки

тысяч) одновременно работающих пользователей, при этом позволяя получать очень высокую производительность.

2.2.1. Общие службы объектно-реляционной среды

В разработке объектно-реляционных приложений намечаются общие тенденции. Специалисты, неоднократно занимавшиеся интеграцией подобных систем, выявляют определенные структуры и свойства удачных объектно-реляционных приложений. Эти структуры и свойства были формализованы в высокоуровневых спецификациях CORBA (которые относятся и к системам на основе COM/DCOM).

При разработке объектно-реляционных систем следует принять во внимание следующие спецификации служб CORBA:

- Хранение
- Запросы
- Транзакции
- Параллелизм
- Взаимосвязи

Хранение

Термин "хранение" применяется для описания механизма, который используется для сохранения состояния объектов в промежутках между сеансами. Среда хранения позволяет пользователю сохранять объекты в конце сеанса работы, а затем обращаться к ним из следующего сеанса. При обращении к объектам из следующего сеанса их состояние (например, набор атрибутов) будет точно соответствовать их состоянию на момент сохранения в предыдущем сеансе. В многопользовательских системах это правило может не выполняться, поскольку разные пользователи могут изменять одни и те же объекты. Хранение взаимосвязано с другими службами. В частности, взаимосвязь с параллелизмом и другими категориями намеренна и соответствует структуре классификации служб CORBA.

Ниже перечислены некоторые службы, предоставляемые системой хранения:

- ***Управление соединением с источником данных***: объектно-реляционные приложения должны устанавливать соединение с физическим источником данных. Как правило, реляционные СУБД представляют собой комплекс из сервера и базы данных. Особенности управления соединениями зависят от разновидности СУБД, и среда должна быть достаточно гибкой для поддержки различных систем.

- ***Извлечение объектов***: извлечение объектов из базы данных заключается в том, что данные считываются из базы данных и преобразуются в объекты. Данный процесс предполагает извлечение данных из структур базы данных, маршрутизацию данных в соответствующие классы и типы объектов и формирование необходимых атрибутов объектов.
- ***Сохранение объектов***: процесс сохранения объектов обратен процессу их извлечения. Значения атрибутов извлекаются из объекта, затем создается структура базы данных со значениями атрибутов (строка SQL, хранимая процедура или особый вызов RPC), а затем полученная структура заносится в базу данных.
- ***Удаление объектов***: объекты удаляются из системы, а соответствующие им данные - из базы данных. Для удаления объектов необходимо извлечь из них соответствующую информацию, создать запрос на удаление (это может быть строка SQL, хранимая процедура или особый вызов RPC), а затем передать запрос в базу данных. Нужно учитывать, что в некоторых языках (например, Smalltalk и Java) явное удаление не поддерживается. Вместо этого там применяется стратегия ***сбора мусора***. В системах хранения, поддерживающих эти языки, должен быть предусмотрен альтернативный способ удаления данных из базы после того, как приложения перестают пользоваться ими. Один из распространенных способов заключается в учете количества ссылок на объект из других объектов. Как только количество ссылок падает до нуля, на этот объект больше не ссылаются другие объекты, и возможно, этот объект больше не нужен. В некоторых случаях можно удалять объекты с нулевым количеством ссылок, поскольку даже если на объект нет ссылок, он может использоваться. В любом случае нужна централизованная стратегия удаления объектов, действующая в масштабах всей базы данных.

Запрос

От системы хранения мало пользы в отсутствие механизма поиска и извлечения нужных объектов. Средства поиска применяются для нахождения объектов, удовлетворяющих определенным условиям. Стандартные операции поиска в объектно-реляционных средах - find и findunique. Операция findunique возвращает конкретный объект, а операция find - набор объектов, соответствующих заданным критериям.

Набор средств выборки данных очень сильно зависит от разновидности СУБД. В простых файловых хранилищах данных может поддерживаться минимальный набор необходимых операций, тогда как реляционные СУБД зачастую снабжены

гибким языком управления данными. Среды объектно-реляционных преобразований расширяют модель реляционных запросов и смещают ее акцент с данных на объекты. Кроме того, для обеспечения достаточной гибкости и поддержки расширений, зависящих от конкретной реализации, применяются сквозные интерфейсы (например, хранимые процедуры).

Существует потенциальный конфликт между механизмами запросов к базам данных и объектной парадигмой: механизмы запросов работают на основе значений атрибутов (столбцов) таблицы. Принцип инкапсуляции объектной модели не позволяет видеть значения атрибутов, так как они ***инкапсулированы*** за операциями класса. Инкапсуляция применяется для того, чтобы упростить изменение приложений - она дает возможность изменять внутреннюю структуру классов, не беспокоясь о зависимых классах постольку, поскольку не меняются внешние интерфейсы изменяемого класса. Механизм запросов зависит от внутренней реализации классов и таким образом нарушает инкапсуляцию. Задача среды заключается в том, чтобы сохранить независимость приложений от особенностей запросов и сохранить свободу изменения классов.

Транзакции

Поддержка транзакций позволяет разработчикам определить понятие единицы (кванта) работы. В терминологии баз данных это значит, что система должна уметь либо внести сразу набор изменений к базе данных, либо гарантировать, что ни одно из этих изменений не было внесено. В рамках транзакции либо все операции выполняются успешно, либо транзакция не выполняется вовсе. В объектно-реляционных средах, как минимум, должны поддерживаться операции фиксации и отката. Разработка объектно-реляционной среды в многопользовательской системе может быть сопряжена с различными сложностями и должна быть тщательно продумана. Помимо среды хранения, приложение должно уметь обрабатывать ошибки. Если транзакция будет прервана или завершится с ошибкой, система должна быть в состоянии вернуться в стабильное рабочее состояние, например путем считывания информации о предыдущем состоянии из базы данных. Поэтому среда хранения и среда обработки ошибок должны работать в тесном взаимодействии.

Параллелизм

Многопользовательские объектно-ориентированные системы должны поддерживать управление одновременным доступом к объектам. Если с объектом одновременно работают несколько пользователей, в системе должен быть

предусмотрен последовательный механизм управления изменением объекта в системе хранения. В объектно-реляционных средах могут применяться пессимистичный и оптимистичный подходы к управлению параллелизмом.

- ***Пессимистичное управление параллелизмом*** заключается в том, что разработчик обязан указать цель, с которой объект извлекается из хранилища (для чтения, для записи и т.д.). Если объект блокирован, он недоступен другим пользователям до тех пор, пока он не будет разблокирован. Пессимистичное управление следует реализовывать с осторожностью, поскольку оно способствует возникновению тупиковых ситуаций.
- ***Оптимистичное управление параллелизмом*** заключается в предположении о том, что несколько пользователей почти наверняка не будут работать с одним и тем же объектом одновременно. Конфликты параллелизма обнаруживаются при сохранении изменений в базе данных. Как правило, если объект был изменен другим пользователем с момента его извлечения, в приложение будет отправлено сообщение об ошибке с объяснением причины сбоя в операции изменения. Ответственность за обработку этой ошибки возлагается на приложение. Для реализации такой среды необходим механизм кэширования текущих значений объектов и их сравнения со значениями в базе данных. Оптимистичное управление параллелизмом требует меньшего количества ресурсов при небольшом количестве конфликтов, но может стать неэффективным при большом количестве конфликтов (за счет необходимости повторно выполнять работу при возникновении конфликтов).

Все приложения, работающие с общими данными, должны пользоваться одной и той же стратегией управления параллелизмом. Смешение разных стратегий управления в пределах одного набора данных может привести к повреждению данных. С обеспечением единства стратегии управления параллелизмом лучше всего справится среда хранения.

Взаимосвязи

У объектов есть взаимосвязи с другими объектами. У объекта Order есть несколько объектов LineItem. У объекта Book есть много объектов Chapter. Объект Employee принадлежит ровно одному объекту Company. В реляционных системах взаимосвязи между объектами реализованы с помощью ссылок на основные и внешние ключи. В объектно-ориентированных системах такие взаимосвязи обычно явно заданы с помощью атрибутов. Если у объекта Order есть объекты LineItems, объект Order будет содержать атрибуты lineItems. Атрибут lineItems объекта Order может содержать произвольное количество объектов LineItem. Аспекты

взаимосвязей в объектно-реляционной среде тесно связаны с вопросами реализации хранения, транзакций и выполнения запросов. При сохранении и выборке объектов, выполнении транзакций и запросов необходимо учитывать следующие факторы:

- Нужно ли вместе с объектом извлекать из базы данных объекты, с которыми он связан? Простой ответ на этот вопрос - "да", но цена извлечения ненужных объектов может быть очень большой. В хорошей среде могут поддерживаться несколько стратегий на этот счет.
- Нужно ли при сохранении объекта заодно сохранять и связанные с ним объекты, которые также были изменены? Ответ на этот вопрос тоже зависит от контекста.

Считается, что правильнее рассматривать общие службы объектно-реляционной среды независимо друг от друга; в реальности эти службы будут сильно зависеть друг от друга. Они должны быть единообразно реализованы не только в масштабах всех подразделений организации, но и во всех приложениях, работающих с одними и теми же данными. Единственный экономичный способ решения этой задачи заключается в формировании общей среды.

2.2.2. Объектно-реляционные методы

Объектно-ориентированный подход при создании систем управления базами данных дает возможность упаковывать вместе данные и код для их обработки. Тем самым, фактически снимаются ограничения на типы данных, что позволяет работать на любом уровне абстракции.

Объектно-реляционные адаптеры. Применение объектно-реляционного адаптера позволяет автоматически выделять программные объекты и сохранять их в реляционных БД. В данном случае объектно-ориентированное приложение работает как рядовой пользователь СУБД. Такой подход дает программистам возможность полностью сконцентрироваться на объектно-ориентированной разработке, несмотря на небольшое снижение производительности. Также, все имеющиеся на предприятии приложения могут по-прежнему обращаться к данным, которые хранятся в реляционной форме. Например, объектно-реляционный адаптер Odapter фирмы Hewlett-Packard для СУБД Oracle, можно с успехом применять во многих областях, например в качестве связующего программного обеспечения, которое объединяет объектно-ориентированные приложения с реляционными базами данных.

Объектно-реляционные шлюзы. В данном случае пользователь взаимодействует с базой данных с применением языка ООСУБД, а используемый шлюз производит замену всех объектно-ориентированных элементов такого языка на их реляционные компоненты.

Недостатком данного способа также является снижение производительности. К примеру, объектно-реляционный шлюз должен осуществить преобразование объектов в набор связей, генерацию оригинальных идентификаторов (OID) объектов и потом передать их в реляционную базу данных. Далее шлюз должен каждый раз при использовании интерфейса реляционной базы данных преобразовывать OID, найденный в базе, в соответствующий объект, который сохранен в реляционной СУБД.

Гибридные СУБД. Еще одним решением является создание гибридных объектно-реляционных СУБД, которые могут хранить как традиционные табличные данные, так и объекты. Ведущие поставщики реляционных СУБД начинают добавлять к своим продуктам объектно-ориентированные средства. Например, Sybase и Informix ввели в поддержку объектов СУБД. Подобные разработки вводят и другие независимые компании. К примеру, компания Shores оснастила объектно-ориентированными средствами СУБД Oracle8.

Перспективы развития

Современные промышленные СУБД представляют собой сложные комплексы, состоящие из различных элементов, технологий и подходов. Данные составляющие объединяются и совершенствуются, исходя из потребностей обеспечения идеальных условий для решения проблем управления большими объемами данных в различных условиях. При этом все разработчики проводят масштабные исследовательские работы. Многолетний опыт разработки СУБД показал, что для обеспечения эффективной, надежной и безошибочной работы нового функционала требуется достаточно много времени. Жесткая конкуренция на рынке СУБД заставляет производителей тщательно следить за продуктами конкурентов, выявлять новые тенденции, а появление важных новых возможностей у одного из конкурентов вынуждает остальных реализовывать аналогичный функционал в своих разработках. В свою очередь, растут и потребности разработчиков современных баз данных. В первую очередь, это связано с бурным развитием интернета, активным использованием мультимедиа и необходимостью обрабатывать слабоструктурированные данные.

Все вышеперечисленное говорит о том, что стратегия развития, выбранная ведущими игроками рынка СУБД, позволит и в дальнейшем сохранять лидерские позиции. Их основные продукты будут совершенствоваться, будет реализовываться новый функционал, а разработчики и в дальнейшем будут выбирать универсальные и проверенные временем традиционные решения.

ГЛАВА 3. ОСНОВНЫЕ ПРИНЦИПЫ ОБЛАЧНЫХ ТЕХНОЛОГИЙ

Облачные вычисления (**cloudcomputing**) - это технология распределённой обработки данных, в которой компьютерные ресурсы и мощности предоставляются пользователю как интернет-сервис. Другими словами, облачные технологии, - это различные аппаратные, программные средства, методологии и инструменты, которые предоставляются пользователю, как интернет-сервисы, для реализации своих целей, задач, проектов.

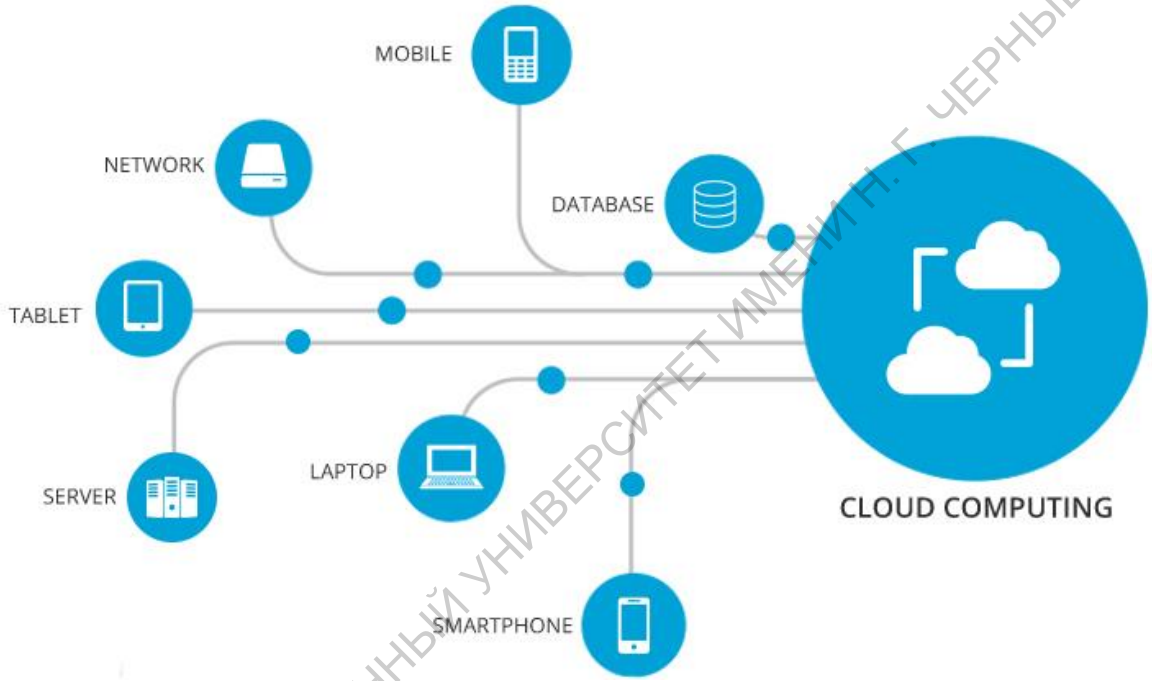


Рис.3.1. Облачные вычисления (cloudcomputing)

3.1. Возможности облачных вычислений

Можно выделить следующие наиболее привлекательные возможности облачных вычислений.

- Доступ к личной информации с любого компьютера, подключённого к Интернету.
- Можно работать с информацией с разных устройств (ПК, планшеты, телефоны и т.п.).
- Не важно, в какой операционной системе происходит работа, - Web-сервисы работают в браузере любых ОС.
- Одну и туже информацию можно просматривать и редактировать одновременно с разных устройств.
- Многие платные программы стали бесплатными (или более дешёвыми) Web - приложениями.

- Если что-то случится с устройством (ПК, планшетом, телефоном), то не произойдет потеря важной информации, так как она теперь не хранится в памяти устройств.
- Всегда свежая и обновлённая информация.
- Самая последняя версия программ (не надо следить за выходом обновлений).
- Можно свою информацию объединять с другими пользователями.
- Легко можно делиться информацией с близкими людьми или с людьми из любой точки земного шара.

3.2. Основные типы услуг

Storage-as-a-Service ("хранение как сервис").

Это самый простой из **CC**-сервисов, представляющий собой дисковое пространство по требованию. Услуга **Storage-as-a-Service** дает возможность сохранять данные во внешнем хранилище, в "облаке". Оно будет выглядеть, как дополнительный логический диск или папка.

Database-as-a-Service ("база данных как сервис").

Возможность работать с базами данных, как если бы **СУБД** была установлена на локальном ресурсе.

Information-as-a-Service ("информация как сервис").

Дает возможность удаленно использовать любые виды информации, которая может меняться ежеминутно или даже ежесекундно.

Process-as-a-Service ("управление процессом как сервис").

Представляет собой удаленный ресурс, который может связать воедино несколько ресурсов (таких как услуги или данные, содержащиеся в пределах одного "облака" или других доступных "облаков"), для создания единого бизнес-процесса.

Application-as-a-Service ("приложение как сервис").

Так же может называться **Software-as-a-Service** ("ПО как сервис"). Позиционируется как «программное обеспечение по требованию», которое развернуто на удаленных серверах и каждый пользователь может получать к нему доступ посредством Интернета.

Platform-as-a-Service ("платформа как сервис").

Пользователю предоставляется компьютерная платформа с установленной операционной системой и некоторым программным обеспечением.

Integration-as-a-Service ("интеграция как сервис").

Это возможность получать из "облака" полный интеграционный пакет, включая программные интерфейсы между приложениями и управление их алгоритмами. Сюда входят известные услуги и функции пакетов централизации, оптимизации и интеграции корпоративных приложений (**EAI**), но предоставляемые как "облачный" сервис.

Security-as-a-Service ("безопасность как сервис").

Данный вид услуги предоставляет возможность пользователям быстро развертывать продукты, позволяющие обеспечить безопасное использование Web - технологий, электронной переписки, локальной сети, что позволяет пользователям

данного сервиса экономить на развертывании и поддержании своей собственной ***системы безопасности***.

Management/Governace-as-a-Service ("администрирование и управление как сервис").

Дает возможность управлять и задавать параметры работы одного или многих "облачных" сервисов. Это в основном такие параметры, как топология, использование ресурсов, виртуализация.

Infrastructure-as-a-Service ("инфраструктура как сервис").

Пользователю предоставляется компьютерная инфраструктура, обычно виртуальные платформы (компьютеры), связанные в сеть, которые он самостоятельно настраивает под собственные цели.

Testing-as-a-Service ("тестирование как сервис").

Дает возможность тестирования локальных или "облачных" систем с использованием тестового ПО из "облака"

3.3. Модели развертывания облачных технологий

По модели развертывания облака разделяют на частные, общедоступные (публичные) и гибридные (рис. 3.2.)..

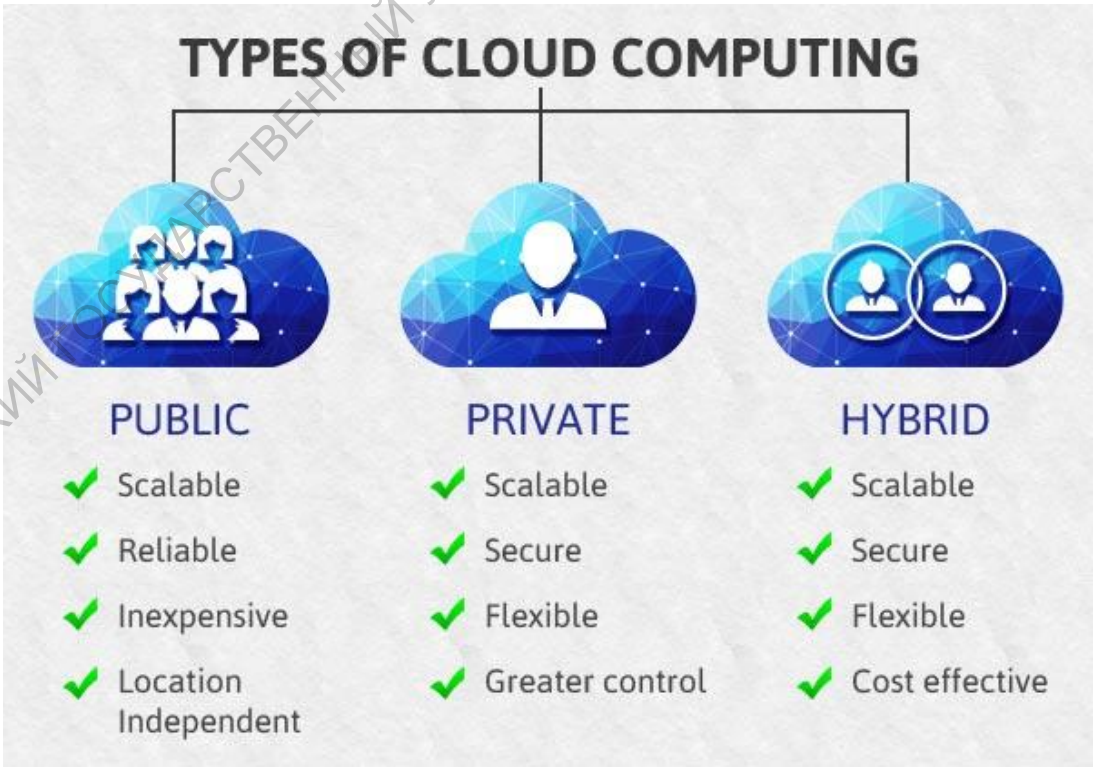


Рис.3.2. Модели развертывания облачных технологий

Частные облака – это внутренние облачные инфраструктура и службы предприятия. Эти облака находятся в пределах корпоративной сети. Организация может управлять частным облаком самостоятельно или поручить эту задачу внешнему подрядчику. Инфраструктура может размещаться либо в помещениях заказчика, либо у внешнего оператора, либо частично у заказчика и частично у оператора. Идеальный вариант частного облака – облако, развернутое на территории организации, обслуживаемое и контролируемое ее сотрудниками. Предприятие само занимается установкой и поддержкой облака. Сложность и стоимость создания внутреннего облака могут быть очень высоки, а расходы на его эксплуатацию могут превышать стоимость использования общедоступных облаков.

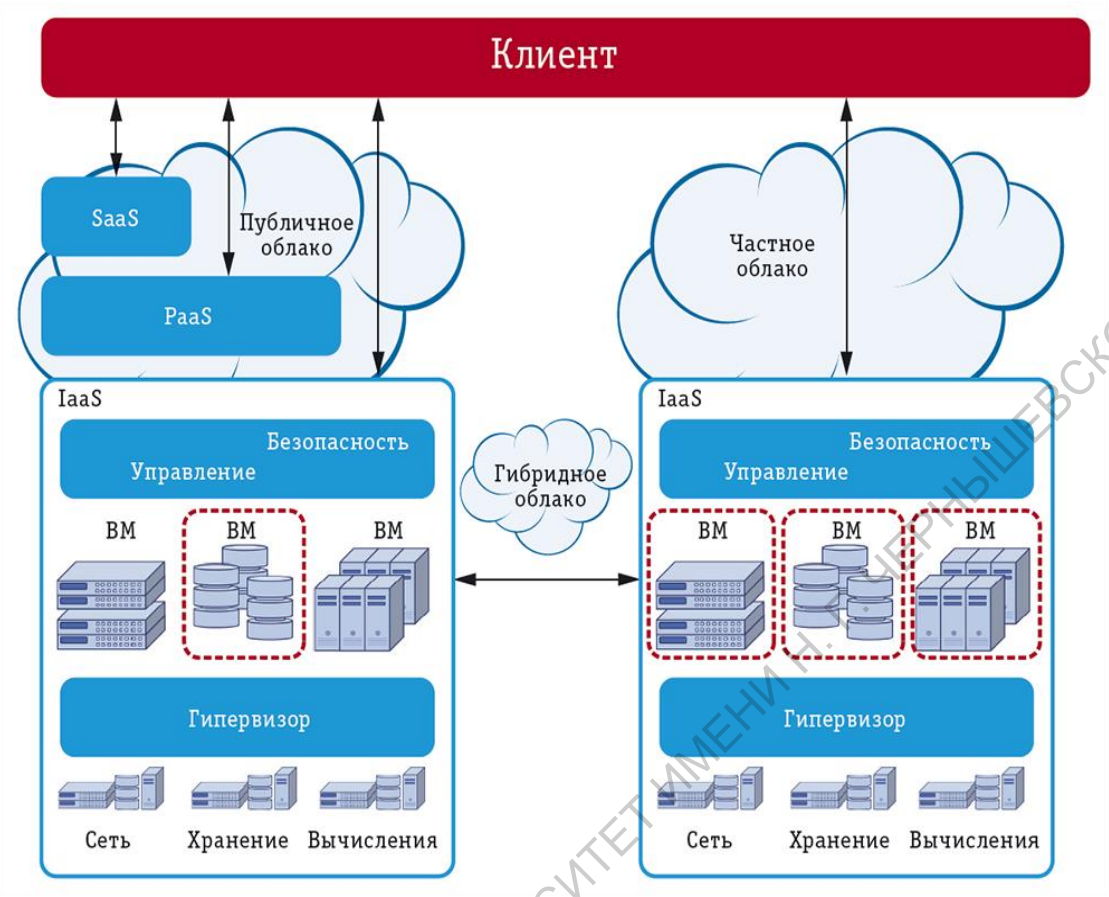
У частных облаков: более детальный контроль над различными ресурсами облака обеспечивает компании любые доступные варианты конфигурации. Кроме того, частные облака идеальны, когда нужно выполнять работы, которые нельзя доверить общедоступному облаку из соображений безопасности.

Общедоступные (публичные) облака – это облачные услуги, предоставляемые поставщиком. Они находятся за пределами корпоративной сети. Пользователи данных облаков не имеют возможности управлять данным облаком или обслуживать его, вся ответственность возложена на владельца этого облака. Поставщик облачных услуг принимает на себя обязанности по установке, управлению, предоставлению и обслуживанию программного обеспечения, инфраструктуры приложений или физической инфраструктуры. Клиенты платят только за ресурсы, которые они используют. Абонентом предлагаемых сервисов может стать любая компания и индивидуальный пользователь. Они предлагают легкий и доступный по цене способ развертывания веб-сайтов или бизнес-систем с большими возможностями масштабирования, которые в других решениях были бы недоступны. Примеры: онлайн-сервисы Amazon EC2 и Amazon Simple Storage Service (S3), Google Apps/Docs, Salesforce.com, Microsoft Office Web.

Услуги публичных облаков в основном предоставляются в виде стандартных конфигураций, то есть исходя из условий наиболее распространенных случаев использования. Это значит, что у пользователя остается меньше возможностей по выбору конфигурации по сравнению с системами, в которых ресурсами управляет сам потребитель. Также надо иметь в виду, что, поскольку потребители слабо контролируют инфраструктуру, процессы, требующие строгих мер безопасности и соответствия нормативным требованиям, не всегда подходят для реализации в общедоступном облаке.

Гибридные облака представляют собой сочетание общедоступных и частных облаков. Обычно они создаются предприятием, а обязанности по управлению ими распределяются между предприятием и поставщиком общедоступного облака. Гибридное облако предоставляет услуги, часть которых относится к общедоступным, а часть – к частным. Обычно такой тип облаков используется, когда организация имеет сезонные периоды активности. Другими словами, как только внутренняя ИТ-инфраструктура не справляется с текущими задачами, часть мощностей перебрасывается на публичное облако (например, большие объемы статистической информации, которые в необработанном виде не представляют ценности для предприятия), а также для предоставления доступа пользователям к ресурсам предприятия (к частному облаку) через публичное облако. Хорошо продуманное гибридное облако может обслуживать как требующие безопасности критически важные процессы, такие как получение платежей от клиентов, так и более второстепенные.

Основным недостатком этого типа облака является сложность эффективного создания подобных решений и управления ими. Необходимо получать услуги из разных источников и организовать их так, как если бы это был единый источник. Взаимодействие между частным и общедоступным компонентами может еще больше усложнить решение. Поскольку это относительно новая архитектурная концепция в сфере облачных вычислений, для этой модели появляются все новые и новые практические рекомендации и инструменты, и ее широкое распространение может затянуться до тех пор, пока она не будет лучше изучена.



3.3.1. Преимущества использования облачных технологий

Доступность.

Доступ к информации, хранящейся на облаке, может получить каждый, кто имеет компьютер, планшет, любое мобильное устройство, подключенное к сети интернет.

Мобильность.

У пользователя нет постоянной привязанности к одному рабочему месту. Из любой точки мира менеджеры могут получать отчетность, а руководители — следить за производством.

Экономичность.

Одним из важных преимуществ называют уменьшение затрат. Пользователю не надо покупать дорогостоящие, большие по вычислительной мощности компьютеры и ПО, а также он освобождается от необходимости нанимать специалиста по обслуживанию локальных IT-технологий.

Арендность.

Пользователь получает необходимый пакет услуг только в тот момент, когда он ему нужен, и платит, собственно, только за количество приобретенных функций.

Гибкость.

Все необходимые ресурсы предоставляются провайдером автоматически.

Высокая технологичность.

Большие вычислительные мощности, которые предоставляются в распоряжение пользователя, которые можно использовать для хранения, анализа и обработки данных.

Надежность.

Некоторые эксперты утверждают, что надежность, которую обеспечивают современные облачные вычисления, гораздо выше, чем надежность локальных ресурсов, аргументируя это тем, что мало предприятий могут себе позволить приобрести и содержать полноценный ЦОД.

3.3.2. Недостатки использования облачных технологий

Необходимость постоянного соединения.

Для получения доступа к услугам «облака» необходимо постоянное соединение с Интернет.

Программное обеспечение и его «кастомизация».

Существуют ограничения по ПО, которое можно разворачивать на «облаках» и предоставлять его пользователю. Пользователь имеет ограничения в используемом обеспечении и иногда не имеет возможности настроить его под свои собственные цели.

Конфиденциальность.

Конфиденциальность данных, хранимых в публичных «облаках», в настоящее время, вызывает много споров, но в большинстве случаев эксперты сходятся в том, что не рекомендуется хранить наиболее ценные для компании документы на

публичном “облаке”, так как в настоящее время нет технологии, которая бы гарантировала **100%** конфиденциальность данных.

Безопасность.

"Облако” само по себе является достаточно надежной системой, однако при проникновении в него злоумышленник получает доступ к огромному хранилищу данных. Еще один минус, - это использование систем **виртуализации** в которых, в качестве гипервизора, используются ядра стандартных ОС (например **Windows**), что позволяет использовать вирусы и уязвимости системы.

Дороговизна оборудования.

Для построения собственного облака необходимо выделить значительные материальные ресурсы, что не выгодно только что созданным и малым компаниям.

Дальнейшая монетизация ресурса.

Вполне возможно, что компании в дальнейшем решат брать плату с пользователей за предоставляемые услуги.

3.4. Облачные технологии в России

Отечественный рынок развивается в русле общемировых трендов: для России характерны аналогичные темпы роста, преобладание в объеме продаж продуктов SaaS и высокая динамика развития сегмента IaaS.

Примерно одна шестая всех затрат на ИТ-инфраструктуру приходится на облака. Все больше предприятий при выборе ИТ предпочитают облачные технологии, при этом разработчики переходят от стратегии cloud-first (облако прежде всего) к принципу cloud-only (только облачные решения). В будущем облака будут основным способом потребления ИТ, однако этот процесс растянется на многие годы, что обеспечит стабильный рост направлениям IaaS и SaaS в ближайшие годы. Расходы на SaaS сейчас превышают IaaS бюджеты, однако последний сегмент развивается быстрее. При этом для России характерна постепенная миграция с иностранных сервисов IaaS на российские.

Глобальные поставки облачных решений продолжают расти опережающими тепами по отношению к остальному ИКТ-рынку. Эксперты Gartner прогнозируют, что в 2017 г. выручка от продажи облаков (включая IaaS, SaaS, PaaS, BPaaS, инструменты управления облаками и облачные средства безопасности) увеличится на 20% до $155 млрд., при том что общий рост поставок инфотелекоммуникационных решений по итогам текущего года должен составить только 2,4% (таким образом, рынок облаков растет более чем 8 раз быстрее). В ближайшие годы рост облачных продаж стабилизируется в районе 17–21%, что говорит о формировании более зрелого рынка, полагают аналитики. Желание оптимизировать издержки и необходимость трансформации бизнеса будут играть на пользу облачным провайдерам: согласно прогнозу Gartner, к 2020 г. более половины сделок по ИТ-аутсорсингу будут предполагать развертывание облаков. В 2016 г. на облака пришлась примерно одна шестая или около 17% всех затрат на инфраструктуру, приложения, межплатформенное ПО и сервисы по автоматизации бизнес-процессов, при этом к 2021 г. эта цифра возрастет до 28%.

По итогам 2016 г. глобальные продажи SaaS (Software-as-a-Service, доставка программного обеспечения в качестве онлайн сервсиа через веб-бразуер) достигли $48,2 млрд, в 2016–2020 гг. темпы роста этого рынка составят 22–18%, с тенденцией к постепенному замедлению. На данный момент в виде SaaS можно получить практически любой функционал, доступный через ПО, устанавливаемое на рабочую станцию. Наибольшей зрелостью отличаются продукты SaaS в области систем для взаимоотношений с клиенатми (CRM, Customer Relationship Management) и по управлению персоналом (HCM, human capital management).

Объем поставок IaaS (Infastructure-as-a-Service, аренда мощностей удаленного ЦОДа) по итогам 2016 г. оказался почти в два раза меньше, чем SaaS ($25,4 млрд против 48,2 млрд), однако именно продажи инфраструктуры как сервиса будут расти быстрее всего: по итогам 2017 г. поставки в этом сегменте увеличатся на 37%, полагают аналитики Gartner. Переходу на эту модель потребления ИТ-ресурсов способствует распространение новых технологий, которые предъявляют высокие требования к вычислительным мощностям и СХД, например, искуственный интеллект (artificial intelligence, AI), а также интернет вещей (Internet of things, IoT) и связанные с ним инструменты аналитики. Кроме того, распространение услуги PaaS (Platform-as-a-Service, предоставление средств разработки по облачной модели) также стимулирует спрос на продукты IaaS.

Результаты глобальных продаж по различным облачным направлениям, в млрд $

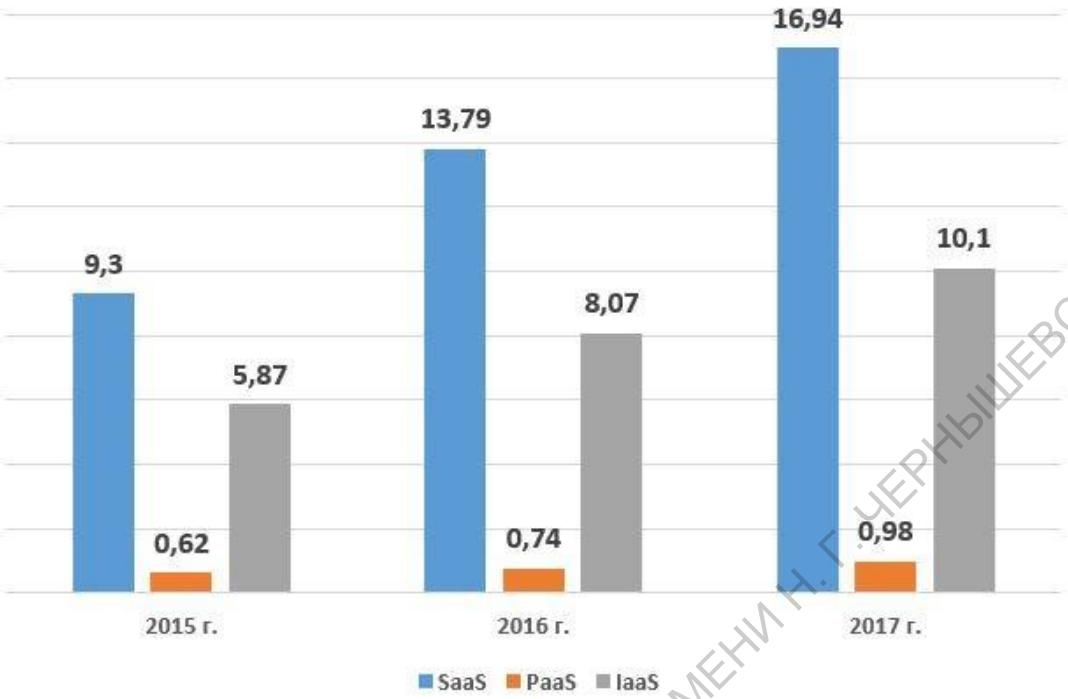
	2016 г.	2017 г.	2018 г.	2019 г.	2020 г.
SaaS	48,2	58,6	71,2	84,8	99,7
IaaS	25,4	34,7	45,8	58,4	72,4
BPaaS	39,6	42,2	45,8	49,5	53,6
PaaS	9	11,4	14,2	17,3	20,8
управление облаками и сервисы безопасности	7,1	8,3	10,3	12	13,9

Источник: Gartner, 2017

По данным IDC, общий объем российского рынка по итогам 2016 г. составил $422 млн., что на 20% больше, чем годом ранее (с учетом частных облаков, при этом доля публичных облаков составила 87%). Наибольшая доля этого «пирога» пришлась на SaaS (68%), далее следуют IaaS (28%) и PaaS (8%). При этом наибольший вклад в рост рынка сделали крупные предприятия из финансовой, производственной и торговой отраслей, которые внедряли инструменты интернета вещей и аналитики больших данных.

При этом есть существенная разница в парадигме потребления облаков между крупным бизнесом и сектором СМБ. Большие компании имеют большой штат ИТ-персонала для решения задач функциональных заказчиков. На данном этапе такие компании не готовы отказаться от собственных ИТ-специалистов и объясняют свой выбор слабой подготовкой сервисных ИТ-компаний. На рынке среднего и малого бизнеса преобладает спрос на SaaS – это проще, связано с меньшими рисками и позволяет сохранить гибкость бизнеса.

Объем российского рынка облачных технологий (IaaS, PaaS, SaaS) , в млрд. руб.*



**2017 г. – прогноз*

Источник: Forrester/SAP, 2017

По данным Forrester, в рублевом исчислении по итогам 2016 г. российские продажи SaaS составили 13,79 млрд.руб., что почти в половину (на 48%) больше, чем годом ранее. Объем рынка IaaS составил 8,07 млрд.руб. при росте на 37% по сравнению с предыдущим годом. В 2017 г. аналитики прогнозировали некоторое замедление темпов развития и предсказывает рост IaaS на 25% и SaaS на 23%. Тем не менее, запас роста остается значительным. Подавляющее большинство заказчиков осведомлены об облачных технологиях, однако реально используют или планируют внедрение пока меньше половины представителей бизнеса. Согласно опросу, проведенному Forrester, среди представителей крупного российского бизнеса 19,4% уже используются облака, 21,2% планируют внедрять и 55,4% представляют, что такое облако хотя бы в теории. Для среднего бизнеса эти показатели составляют 18,4%, 20,8% и 52% соответственно. Несколько хуже ситуация обстоит среди малых предприятий: здесь 14,8% респондентов уже используют облака, 21,4% планируют начать, при этом 13,4% вообще ничего не слышали об облачных технологиях.

Крупнейшие поставщики SaaS в России 2017

№2016	Название компании	Город	Выручка от оказания услуг SaaS в 2016 г., ₽тыс., включая НДС	Выручка от оказания услуг SaaS в 2015 г., ₽тыс., включая НДС	Рост выручки 2016/2015, в %	Доля выручки SaaS в совокупной выручке компании
1	СКБ Контур	Екатеринбург	8 600 000	6 970 000	23%	н/д
2	Softline	Москва	1 797 140	1 034 000	74%	3%
3	Манго Телеком (1)	Москва	1 688 484	1 402 465	20%	100%
4	B2B-Center	Москва	1 364 641	1 158 604	18%	100%
5	Корус Консалтинг СНГ	Санкт-Петербург	1 056 597	783 511	35%	99%
6	Ай-Теко	Москва	676 950	237 700	185%	5%
7	МойСклад*	Москва	600 000	395 000	52%	н/д
8	Техносерв	Москва	563 755	483 942	16%	н/д
9	Телфин	Санкт-Петербург	483 700	398 500	21%	100%
10	amoCRM (Qsoft)	Москва	450 000	220 000	105%	100%

Источник: Gartner, 2017

Среди трудностей при переходе в облака представители российского бизнеса чаще всего называют нежелание передавать контроль над данными третьим лицам, неготовность передавать вовне конфиденциальные данные, а также трудности в обосновании преимущество облаков перед руководством, необходимость повышать квалификацию персонала и неприятие облаков сотрудниками (рис. 3.3.).

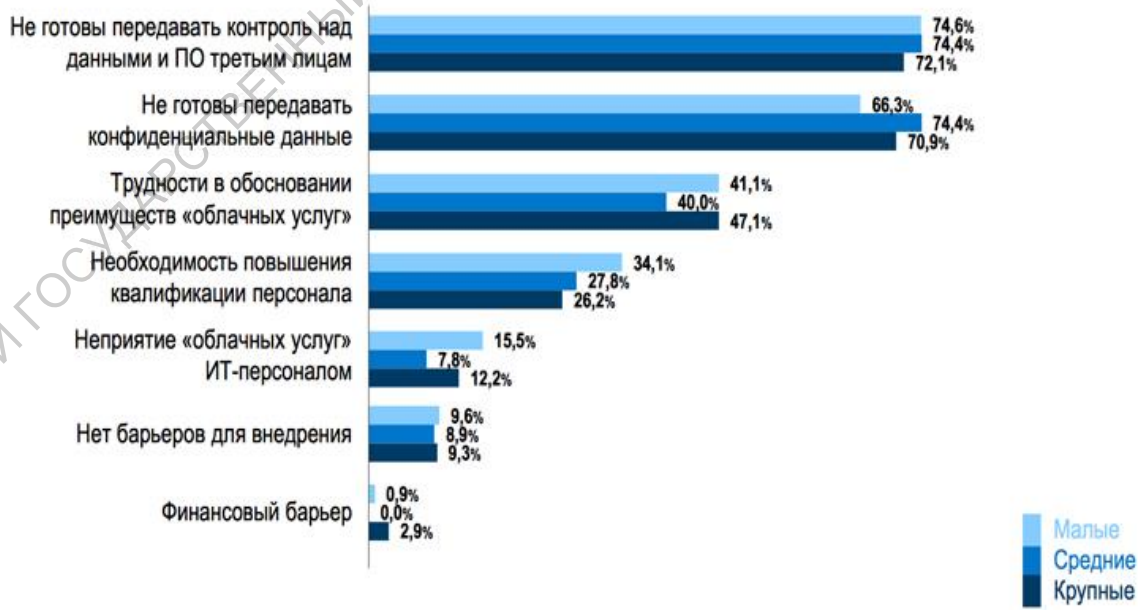


Рис. 3.3. Основные барьеры для внедрения облачных услуг

Необходимо отметить, что заметная часть российского потребления облаков приходится на иностранные сервисы. По данным iKS-Consulting, четверть используемых в России сервисов IaaS предоставляются иностранными провайдерами (например, компанией Amazon, которая является мировым лидером по поставкам инфраструктуры как сервиса). При этом в некоторых случаях потребление иностранных облаков происходит в обход официальных каналов. Однако в целом заказчики постепенно переходят на российские сервисы. Росту доверия заказчиков к отечественным решениям способствуют стремительное развитие российского рынка IaaS, внедрение новых технологий, улучшение качества предоставляемых сервисов.

ГЛАВА 4. БОЛЬШИЕ ДАННЫЕ. ОСНОВНЫЕ ПОНЯТИЯ

В первом приближении можно предположить, что термин «большие данные» относится просто к управлению и анализу больших объемов данных. Согласно отчету McKinsey Institute «Большие данные: новый рубеж для инноваций, конкуренции и производительности» (Bigdata: Thenext frontier for innovation, competition and productivity), термин «большие данные» относится к наборам данных, размер которых превосходит возможности типичных баз данных по занесению, хранению, управлению и анализу информации. Тем не менее «большие данные» предполагают нечто большее, чем просто анализ огромных объемов информации. Проблема не в том, что организации создают огромные объемы данных, а в том, что бóльшая их часть представлена в формате, плохо соответствующем традиционному структурированному формату БД, — это Web-журналы, видеозаписи, текстовые документы, машинный код или, например, геопространственные данные. Всё это находится во множестве разнообразных хранилищ, может быть даже за пределами организации. В результате корпорации могут иметь доступ к огромному объему своих данных и не иметь необходимых инструментов, чтобы установить взаимосвязи между этими данными и сделать на их основе значимые выводы. Кроме того, данные сейчас обновляются все чаще и чаще, и поэтому традиционные методы анализа информации не могут угнаться за огромными объемами постоянно обновляемых данных, что в итоге и открывает дорогу технологиям больших данных.

В сущности, понятие больших данных подразумевает работу с информацией огромного объема и разнообразного состава, часто обновляемой и находящейся в разных источниках в целях увеличения эффективности работы, создания новых продуктов и повышения конкурентоспособности. Консалтинговая компания Forrester дает краткую формулировку: «Большие данные объединяют техники и технологии, которые извлекают смысл из данных на экстремальном пределе практичности».

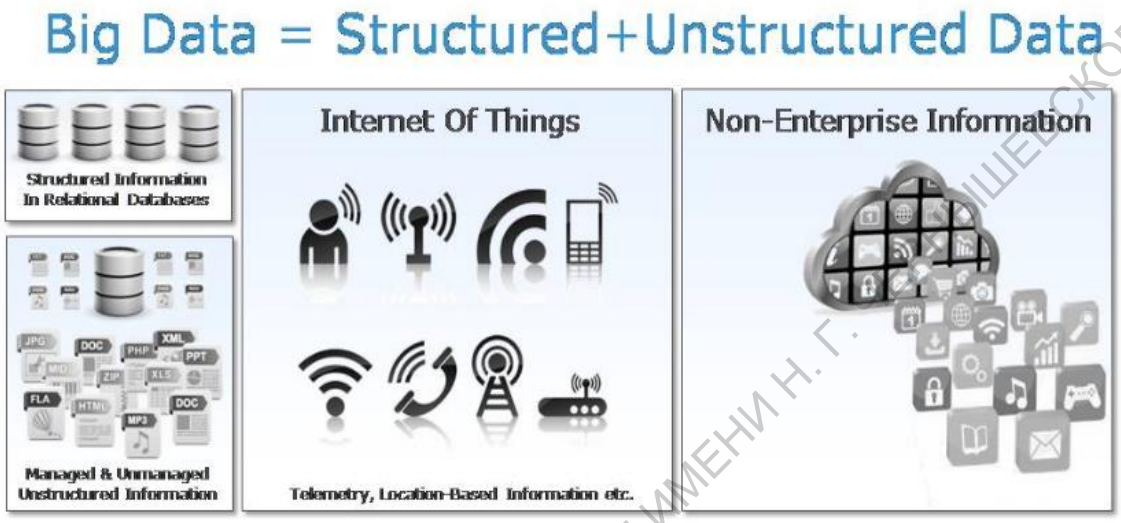
4.1. Бизнес-аналитика и большие данные

Бизнес-анализ (BI) *является описательным процессом анализа результатов, достигнутых бизнесом в определенный период времени*, между тем как скорость обработки больших данных позволяет сделать анализ предсказательным, способным предлагать бизнесу рекомендации на будущее. Технологии больших данных позволяют также анализировать больше типов данных в сравнении с инструментами бизнес-аналитики, что дает возможность фокусироваться не только на структурированных хранилищах. Согласно исследованиям IDC, мировой рынок бизнес-аналитики и больших данных планомерно растет: в 2015 г. он достиг $122 млрд, в 2016 г. – уже $130 млрд. К 2020 г. аналитики прогнозируют рост объема рынка до $203 млрд.

Большие данные и бизнес-аналитика имеют одинаковую цель (поиск ответов на вопрос), они отличаются друг от друга по трем аспектам.

- Большие данные предназначены для обработки более значительных объемов информации, чем бизнес-аналитика, и это, конечно, соответствует традиционному определению больших данных.
- Большие данные предназначены для обработки более быстро получаемых и меняющихся сведений, что означает глубокое исследование и интерактивность. В некоторых случаях результаты формируются быстрее, чем загружается Web - страница.

- Большие данные предназначены для обработки неструктурированных данных, способы использования которых начинают изучаться только после того, как налажен их сбор и хранение, и требуются алгоритмы и возможность диалога для облегчения поиска тенденций, содержащихся внутри этих массивов.



При работе с большими данными подход к информации осуществляется иначе, чем при проведении бизнес-анализа. Работа с большими данными не похожа на обычный процесс бизнес-аналитики, где простое сложение известных значений приносит результат: например, итог сложения данных об оплаченных счетах становится объемом продаж за год. При работе с большими данными результат получается в процессе их очистки путём последовательного моделирования: сначала выдвигается гипотеза, строится статистическая, визуальная или семантическая модель, на ее основании проверяется верность выдвинутой гипотезы и затем выдвигается следующая. Этот процесс требует от исследователя либо интерпретации визуальных значений или составления интерактивных запросов на основе знаний, либо разработки адаптивных алгоритмов `машинного обучения`, способных получить искомый результат. Причём время жизни такого алгоритма может быть довольно коротким.

4.1.1. Методики анализа больших данных

Существует множество разнообразных методик анализа массивов данных, в основе которых лежит инструментарий, заимствованный из статистики и информатики (например, машинное обучение). Список не претендует на полноту, однако в нем отражены наиболее востребованные в различных отраслях подходы. При этом следует понимать, что исследователи продолжают работать над созданием новых методик и

совершенствованием существующих. Кроме того, некоторые из перечисленных них методик вовсе не обязательно применимы исключительно к большим данным и могут с успехом использоваться для меньших по объему массивов (например, A/B-тестирование, регрессионный анализ). Безусловно, чем более объемный и диверсифицируемый массив подвергается анализу, тем более точные и релевантные данные удается получить на выходе.

Data mining.

Набор методик, который позволяет определить наиболее восприимчивые для продвигаемого продукта или услуги категории потребителей, выявить особенности наиболее успешных работников, предсказать поведенческую модель потребителей.

A/B testing.

Методика, в которой контрольная выборка поочередно сравнивается с другими. Тем самым удается выявить оптимальную комбинацию показателей для достижения, например, наилучшей ответной реакции потребителей на маркетинговое предложение. Большие данные позволяют провести огромное количество итераций и таким образом получить статистически достоверный результат.

Association rule learning.

Набор методик для выявления взаимосвязей, т.е. ассоциативных правил, между переменными величинами в больших массивах данных. Используется в **data mining**.

Classification.

Набор методик, которые позволяет предсказать поведение потребителей в определенном сегменте рынка (принятие решений о покупке, отток, объем потребления и проч.). Используется в **data mining**.

Cluster analysis.

Статистический метод классификации объектов по группам за счет выявления наперед не известных общих признаков. Используется в **data mining**.

Crowdsourcing.

Методика сбора данных из большого количества источников.

Data fusion and data integration. Набор методик, который позволяет анализировать комментарии пользователей социальных сетей и сопоставлять с результатами продаж в режиме реального времени.

.**Ensemble learning**.

В этом методе задействуется множество предикативных моделей за счет чего повышается качество сделанных прогнозов.

Genetic algorithms.

В этой методике возможные решения представляют в виде «хромосом», которые могут комбинироваться и мутировать. Как и в процессе естественной эволюции, выживает наиболее приспособленная особь.

Machine learning.

Направление в информатике (исторически за ним закрепилось название `искусственный интеллект`), которое преследует цель создания алгоритмов самообучения на основе анализа эмпирических данных.

Natural language processing (**NLP**).

Набор заимствованных из информатики и лингвистики методик распознавания естественного языка человека.

Network analysis.

Набор методик анализа связей между узлами в сетях. Применительно к социальным сетям позволяет анализировать взаимосвязи между отдельными пользователями, компаниями, сообществами и т.п.

Optimization.

Набор численных методов для редизайна сложных систем и процессов для улучшения одного или нескольких показателей. Помогает в принятии стратегических решений, например, состава выводимой на рынок продуктовой линейки, проведении инвестиционного анализа и проч.

Pattern recognition.

Набор методик с элементами самообучения для предсказания поведенческой модели потребителей.

Predictive modeling.

Набор методик, которые позволяют создать математическую модель наперед заданного вероятного сценария развития событий. Например, анализ базы данных CRM-системы на предмет возможных условий, которые подтолкнут абоненты сменить провайдера.

Regression.

Набор статистических методов для выявления закономерности между изменением зависимой переменной и одной или несколькими независимыми. Часто применяется для прогнозирования и предсказаний. Используется в datamining.

Sentiment analysis.

В основе методик оценки настроений потребителей лежат технологии распознавания естественного языка человека. Они позволяют вычленить из общего информационного потока сообщения, связанные с интересующим предметом (например, потребительским продуктом). Далее оценить полярность суждения (позитивное или негативное), степень эмоциональности и проч.

Signal processing.

Заимствованный из радиотехники набор методик, который преследует цель распознавания сигнала на фоне шума и его дальнейшего анализа.

Spatial analysis.

Набор отчасти заимствованных из статистики методик анализа пространственных данных – топологии местности, географических координат, геометрии объектов. Источником больших данных в этом случае часто выступают геоинформационные системы (ГИС).

Statistics.

Наука о сборе, организации и интерпретации данных, включая разработку опросников и проведение экспериментов. Статистические методы часто применяются для оценочных суждений о взаимосвязях между теми или иными событиями.

Supervised learning.

Набор основанных на технологиях машинного обучения методик, которые позволяют выявить функциональные взаимосвязи в анализируемых массивах данных.

Simulation.

Моделирование поведения сложных систем часто используется для прогнозирования, предсказания и проработки различных сценариев при планировании.

Time series analysis.

Набор заимствованных из статистики и цифровой обработки сигналов методов анализа повторяющихся с течением времени последовательностей данных. Одни из очевидных применений – отслеживание рынка ценных бумаг или заболеваемости пациентов.

Unsupervised learning.

Набор основанных на технологиях машинного обучения методик, которые позволяют выявить скрытые функциональные взаимосвязи в анализируемых массивах данных. Имеет общие черты с **ClusterAnalysis**.

4.2. Бизнес-аналитика и большие данные в России

Согласно исследованиям IDC, мировой рынок бизнес-аналитики и больших данных планомерно растет: в 2015 г. он достиг $122 млрд, в 2016 г. – уже $130 млрд. К 2020 г. аналитики прогнозируют рост объема рынка до $203 млрд. Отечественные поставщики BI также положительно оценивают динамику российского рынка, исходя из внутренних показателей роста выручки по направлению.

Ключевой тенденцией рынка BI и больших данных в 2016 г. и начале 2017 г. стало то, что бизнес начал более широко смотреть на применение новых технологий. Соответственно, разработчиков и консультантов просят решить не локальные, а глобальные задачи, которые оказывают влияние на эффективность всей организации, а не отдельных ее участков. Таким образом, решаются не узкие задачи, а ставится четкий запрос на построение сквозного управление данными. С точки зрения роста выручки такая тенденция выгодна поставщикам бизнес-аналитики. Даже в условиях спада количества проектов, средний чек будет расти за счет увеличения их качества. Еще одним подтверждением повышения ценности данных для бизнеса стали частые запросы в ИТ-компании на приведение данных в порядок. В этот период наблюдался новый всплеск интереса к систематизации нормативно-справочной информации (НСИ). Однако рост интереса к данным выявил новую проблему рынка – отсутствие высококвалифицированных специалистов, способных грамотно с ними работать.

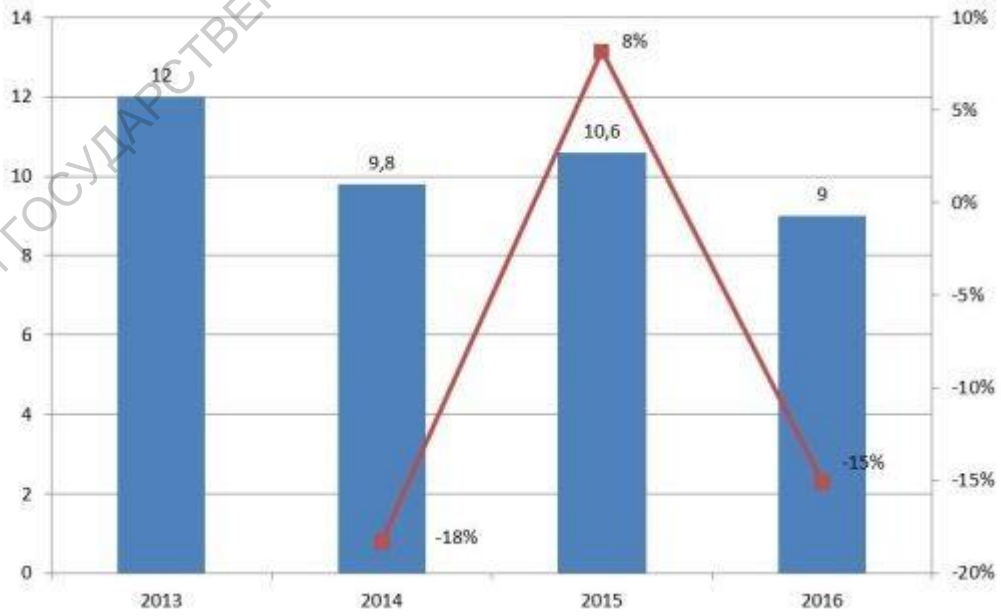
Несмотря на открытость бизнеса к инновациям, на рынке BI все же сохраняется ряд тенденций, которые уже можно назвать традиционными. В первую очередь, это набор отраслей, в которых наиболее широко применяются инструменты BI. Также на рынке бизнес-аналитики сохраняется спрос на визуализацию данных. Более того, согласно исследованию Gartner от 2017 г., именно визуальная аналитика является драйвером развития мирового рынка BI. Очевидным трендом 2016 г. стала демократизация BI-систем. Пользователи не хотят ждать, пока служба ИТ обработает необходимые данные, им нужны инструменты, которые позволят самостоятельно в режиме реального времени

готовить аналитику, используя любые источники данных. Современные BI-системы все больше уходят в сторону self-service, то есть дают пользователю простые и наглядные инструменты, позволяющие подключать новые источники данных и работать с ними. Все большую популярность набирают системы, позволяющие ежедневно добавлять новые измерения, показатели, изменять форму и представление отчета. Эксперты также отмечают рост потребности в гибких и функциональных BI-решениях, предоставляющих бизнес-пользователям свободу в аналитической работе с информацией. Такая ситуация складывается на фоне роста числа пользователей BI-систем внутри организаций.

Особенности BI в России

Отечественный рынок BI становится более концентрированным. Подобная ситуация сложилась из-за ужесточения требований со стороны бизнеса к срокам внедрения и получения отдачи от этого внедрения. Несмотря на то, что рынок в финансовом выражении стабилизировался, клиенты чаще стали «голосовать рублем за узкоспециализированные системы», одно из преимуществ которых как раз и заключается в быстром внедрении. Изменился сам подход: заказчики хотят быстрых внедрений систем BI и быстрого получения желаемого результата, связанного с внедрением, за очень короткий промежуток времени.

Динамика общей выручки топ-15 рейтинга «Крупнейшие поставщики BI-решений в России 2017», рмлрд



Источник: CNews Analytics, 2017

Повсеместная цифровизация бизнеса стала ключевым фактором роста рынка бизнес-аналитики и больших данных, поскольку практически все его процессы сводятся к работе с данными. Кроме того, снижение стоимости владения и доступность технологий постоянно подогревают рынок. Также нужно учесть вклад лидеров мнений в популяризацию BI как в бизнесе, так и в госсекторе. Политика и экономика страны по-прежнему оказывают мощное воздействие на развитие отечественного рынка бизнес-аналитики. Подводя промежуточные итоги политики импортозамещения, игроки по-разному оценивают ее влияние. С одной стороны, российским разработчикам предоставлены хорошие условия для развития. С другой стороны, российские решения пока не достигли того уровня возможностей, чтобы заставить бизнес последовать за госсектором и отказаться от зарубежных разработок. Экономическая ситуация в стране, когда практически все отрасли стали высококонкурентными, превратила BI в одно из главных оружий в борьбе за клиента. В будущее поставщики BI смотрят с оптимизмом. Судя по динамике спроса на BI, в ближайшие годы российский рынок бизнес-аналитики будет продолжать интенсивно расти. При этом рост рынка будет идти в основном за счет повышения доли облачных продуктов бизнес-аналитики, и снижения спроса на классическую отчетную аналитику.

СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

1. Информационные системы [Электронный ресурс]: Учебное пособие / Ольга Леонидовна Голицына, Игорь Иванович Попов, Николай Вениаминович Максимов. - Москва: Издательство "ФОРУМ" ; Москва : ООО "Научно-издательский центр ИНФРА-М", 2014. - 448 с.

2. Основные принципы работы с базами данных [Электронный ресурс] / учебное пособие / В. Ф. Кабанов, А. В. Бурмистров. - Саратов : [б. и.], 2015. - 94 с. - Б. ц. ID= 1272

3. Лапытова, Р. Р. Базы данных : курс лекций : учебное пособие / Р. Р. Лапытова. - Москва : Проспект, 2016. - 96 с.

4. Базовые и прикладные информационные технологии : учебник / В. А. Гвоздева. - Москва : ИД "ФОРУМ" : ИНФРА-М, 2015. – 382 с.

5. Компьютерная архитектура. Количественный подход / Дж. Л. Хеннесси, Д. А. Паттерсон ; пер. с англ. М. В. Таранчевой под ред. А. К. Кима. - 5-е изд. - Москва : Техносфера, 2016. – 935 с.

6. Просто о больших данных = BigDataForDummies : перевод с английского / Д. Гурвиц [и др.]. - Москва : Эксмо, 2015. – 393 с.

7. http://www.cnews.ru/reviews/free/marketBD/articles/articles2.shtml

8. http://www.cnews.ru/reviews/free/marketBD/articles/articles5.shtml

9. http://www.cnews.ru/reviews/cloud2017/articles/odna_shestaya_vseh_zatrat_na_itinfrastrukturu_prihoditsya_na_oblaka

10. http://www.datacenterknowledge.com/archives/2009/05/14/whos-got-the-most-web-servers

11. http://blogs.msdn.com/b/windows-embedded/archive/2013/09/06/the-internet-of-things-is-here.aspx

12. http://www.ibm.com/developerworks/websphere/techjournal/0904_amrhein/0904_amrhein.html

13. http://www.tadviser.ru/index.php/Статья:Облачные_вычисления_(Cloud_computing)

14. http://www.crn.ru/news/detail.php?ID=73064

15. http://www.ibm.com/developerworks/ru/library/cloudservices1iaas

16. http://www.cnews.ru/reviews/bi_bigdata_2017/articles/bi_v_rossii_biznes_hochet_maksimum_polzy_iz_svezhevyzhatyh

17. Медведев А. Облачные технологии: тенденции развития, примеры исполнения // Современные технологии автоматизации. 2013. № 2. С. 6–9.

Учебное издание

Кабанов Владимир Федорович
Бурмистров Александр Валерьевич

ОСНОВНЫЕ ПРИНЦИПЫ ОБРАБОТКИ ДАННЫХ. СОВРЕМЕННЫЙ АСПЕКТ

Учебное пособие

для студентов
Саратовского национального исследовательского государственного университета

Оригинал-макет подготовил *В.Ф. Кабанов*

Компьютерная верстка *В.Ф. Кабанова*

Издано в авторской редакции