Введение в облачные вычисления

Алексей Островский

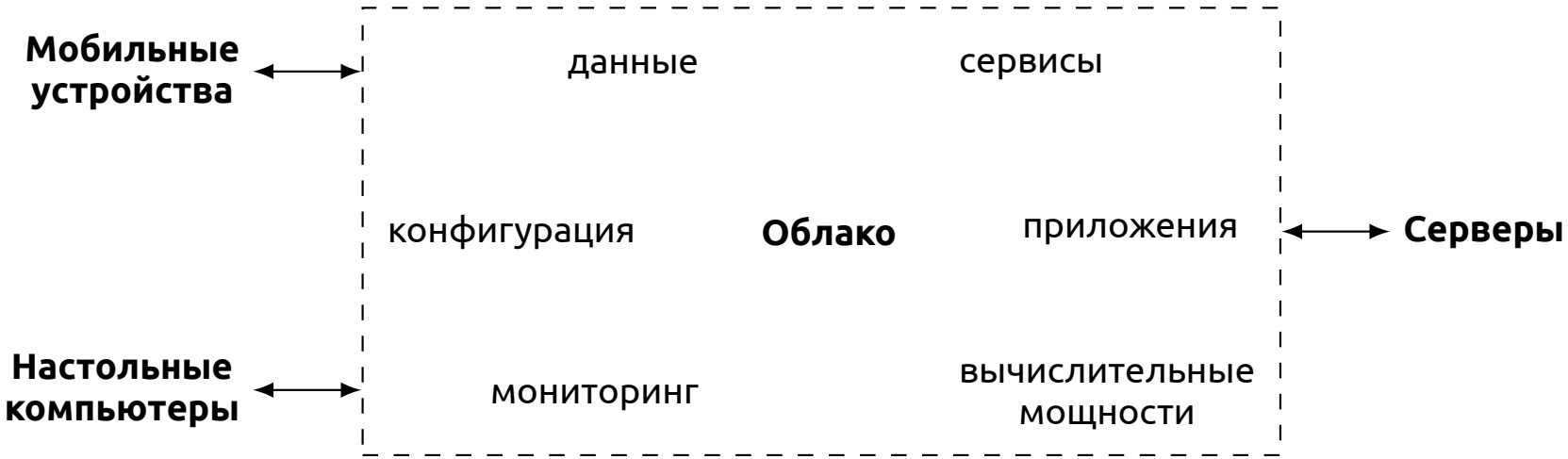
Физико-технический учебно-научный центр НАН Украины

21 мая 2015 г.

Облачные вычисления

Определение (NIST — Национальный институт стандартов и технологий США)

Облачные вычисления (англ. *cloud computing*) — модель для предоставления повсеместного удобного сетевого доступа к конфигурируемым вычислительным ресурсам, которые могут быть введены в использование быстро и с минимальными затратами на управление или взаимодействие с провайдером.



Облачная архитектура скрывает детали распределения ресурсов от пользователя

Характеристики облачных вычислений

- **Самообслуживание** — возможность заказа потребителями дополнительных ресурсов автоматически с помощью системы управления (без взаимодействия со службой поддержки провайдера).
- **Широкий сетевой доступ** — возможность доступа к ресурсам через стандартные сетевые протоколы для широкого круга устройств (смартфоны, планшеты, настольные компьютеры).
- **Пулинг (объединение) ресурсов** — динамическое распределение мощностей оборудования провайдера для обслуживания актуальных запросов потребителей. Прозрачность распределения ресурсов для заказчиков (с возможностью спецификации высокого уровня, напр., страны и информационного центра, на котором хостится приложение).

Характеристики облачных вычислений (продолжение)

- **Эластичность** — возможность быстрого выделения дополнительных ресурсов (по заказу или автоматически) для обеспечения производительности при повышении потребления; иллюзия бесконечных ресурсов.
- **Измерение предоставляемых услуг** — автоматизированный контроль и оптимизация использования ресурсов; модель оплаты на основе количества предоставленных услуг (вычислительных мощностей, объема данных, …). Возможность мониторинга и предоставления отчетности потребителям.
- **Минимизация затрат** на приобретение, конфигурацию и поддержку оборудования.

Предшествующие технологии

Интернет-технологии:

- сервисная архитектура приложений (SOA);
- веб-сервисы (SOAP / WSDL и REST);
- объединения сервисов (англ. *mashup*);
- Web 2.0.

Распределенные вычисления:

- грид-вычисления (англ. *grid computing*) — форма распределенной архитектуры, представленная виртуальным суперкомпьютером, составленным из слабо связанных и географически разделенных независимых компьютеров или кластеров;
- утилитарные вычисления (англ. *utility computing*) — модель предоставления ресурсов по заказу потребителей.

Предшествующие технологии (продолжение)

Виртуализация:

- виртуализация на уровне операционной системы (англ. *OS-level virtualization*) — создание множества изолированных пространств пользователя (контейнеров) с поддержкой со стороны ядра ОС;
- поддержка виртуализации оборудованием (напр., многоядерные процессоры) и ПО (гипервизоры — приложения для управления ВМ).

Управление:

- автономные вычислительные узлы — компьютеры с возможностью самостоятельной конфигурации, оптимизации, восстановления после сбоев и защиты;
- инструменты для мониторинга и управления распределенными конфигурациями (напр., распределением вычислительных мощностей; предупреждением и устранением неполадок).

Уровни облачной архитектуры

Уровень	Средства доступа и управления	Содержимое
ПО как сервис (SaaS)	Веб-браузер	**Облачные приложения:** социальные сети, офисные приложения, системы управления содержимым, интеллектуальная обработка данных.
Платформа как сервис (PaaS)	Облачная среда разработки	**Облачная платформа:** языки программирования, библиотеки, утилиты конфигурирования композиций сервисов, структурированные данные.
Инфраструктура как сервис (IaaS)	Система управления виртуальной инфраструктурой	**Облачная инфраструктура:** вычислительные сервера, хранилища данных, организация сетевых соединений (брандмауэры, балансировка нагрузки).

Уровни облачной архитектуры

Определение

Инфраструктура как сервис (англ. *infrastructure as a service, IaaS*) — предоставление потребителям спецификации базовых ресурсов (для вычислений, хранения и передачи данных, …) с возможностью развертывания произвольного ПО (ОС и приложений).

Степень контроля:

- ОС;
- системы хранения данных;
- развернутые приложения;
- (частично) сетевые компоненты, напр., брандмауэры.

Пример: Amazon EC2.

Уровни облачной архитектуры

Определение

Платформа как сервис (англ. *platform as a service, PaaS*) — предоставление возможности развертывания пользовательских и аналитических приложений на основе поддерживаемых провайдером ЯП, библиотек, сервисов и инструментов.

Степень контроля:

- развернутые приложения;
- (частично) конфигурация среды выполнения.

Пример: Google AppEngine.

Уровни облачной архитектуры

Определение

ПО как сервис (англ. *software as a service, SaaS*) — предоставление возможности использовать приложения, работающие в облачном окружении и доступные с помощью клиентов (напр., веб-браузера) или сетевого API.

Степень контроля:

- (возможно) конфигурация приложения;
- создаваемые пользователями данные.

Пример: Microsoft Office 365.

Модели развертывания

- **Частное облако** (англ. *private cloud, enterprise cloud*) — выделение инфраструктуры для эксклюзивного использования некоторой организацией.
- **Общественное облако** (англ. *community cloud*) — выделение инфраструктуры для пользования объединением организаций (напр., из соображений безопасности).
- **Публичное облако** (англ. *public cloud, Internet cloud*) — выделение облачной инфраструктуры для открытого использования частными лицами или организациями.
- **Гибридное облако** (англ. *hybrid cloud*) — композиция нескольких видов инфраструктуры, объединенных средствами коммуникации для обмена данными и приложениями.

Достоинства и недостатки облачных вычислений

Достоинства:

- минимизация затрат на создание, конфигурацию и поддержку распределенной инфраструктуры;
- эластичность — возможность быстрой адаптации к росту нагрузки (в т. ч. географически неоднородной);
- доступность сервисов для широкого круга пользователей.

Недостатки:

- угрозы безопасности данных, злонамеренного использования сервисов и т. п.;
- возможная ограниченность средств разработки, необходимость адаптации к принципам распределенной / облачной архитектуры;
- отсутствие общепринятых стандартов разработки.

Технологии облачных вычислений

Оборудование:

- аппаратная виртуализация;
- информационные и вычислительные центры, кластеры;
- сетевые соединения, Интернет.

Программное обеспечение:

- распределенные файловые системы;
- облачные базы данных и другие технологии хранения (напр., распределенные системы кэширования);
- средства распределения нагрузки в узлах сети;
- инструменты для обработки данных в облачном окружении;
- веб-API и веб-сервисы.

Примеры облачных платформ

Amazon Web Services (ссылка):

- предоставляет услуги IaaS, PaaS;
- Elastic Compute Cloud (EC2) — масштабируемые сервера для вычислений;
- Elastic MapReduce (EMR) — аналитика;
- Simple Storage Service (S3) — хранилище данных на основе веб-сервисов;
- DynamoDB, SimpleDB — базы данных.

Google App Engine (ссылка):

- предоставляет услуги PaaS;
- автоматическое масштабирование в зависимости от количества запросов;
- поддерживаются ЯП Java (+ другие, использующие JVM, напр., Scala), Python, Go и PHP;
- ограниченный перечень API: БД BigTable, HTTP-запросы, обработка изображений, …

Примеры облачных платформ (продолжение)

Microsoft Azure (ссылка):

- предоставляет услуги PaaS и IaaS;
- управление виртуальными машинами под управлением Windows Server и Linux;
- БД SQL Azure (облачная версия MS SQL Server);
- веб-приложения на основе ASP.NET, PHP, Node.js, Python;
- аналитика при помощи доступных SDK, в частности, Hadoop и машинное обучение.

Heroku (ссылка):

- предоставляет услуги PaaS;
- веб-интефейс и интерфейс командной строки для большинства операций, поддержка быстрого добавления модулей (Heroku Elements);
- поддержка ЯП Ruby, Java, JavaScript / Node.js, Scala, Clojure, Python, PHP;
- БД PostgreSQL (реляционная), MongoDB, Redis (нереляционные).

Big Data

Определение

Большие данные (англ. *big data*) — наборы данных, характеризующиеся большим объемом, высокой скоростью прироста и слабой структурированностью, для которых невозможны или затруднены традиционные методы хранения и обработки (напр., реляционные БД).

Источники данных:

- мобильные устройства;
- Web 2.0 (социальные сети, поиск данных, …);
- наука (метеорология, биоинформатика, физика, …);
- коммерческие организации (данные клиентов).

Характеристики Big Data

Основные характеристики (3V):

- объем (англ. *volume*) — большой размер данных, влияющий на выбор средств их обработки;
- разнообразие (англ. *variety*) — отсутствие общей для данных структуры, наличие различных типов данных из многих источников; неструктурированные (естественный текст) или полуструктурированные (XML, JSON) данные.
- скорость (англ. *velocity*) — высокие темпы накопления данных и требования к скорости их обработки.

Характеристики Big Data (продолжение)

Дополнительные характеристики:

- изменчивость (англ. *variability*) — несогласованность между данными из различных источников;
- (не)достоверность (англ. *veracity*) — возможность существенных различий в качестве исходных данных;
- сложность (англ. *complexity*) — незаурядные требования к алгоритмам и реализациям систем анализа данных для связывания и извлечения полезной информации.

Теория распределенных хранилищ

Теорема (CAP-теорема, Э. Брюэр, С. Гильберт, 2002)

Не существует распределенной компьютерной системы, удовлетворяющей одновременно трем условиям:

- **согласованность данных** (англ. *consistency*) — все узлы системы имеют доступ к одним и тем же данным в произвольный момент времени;
- **доступность** (англ. *availability*) — на каждый запрос к данным будет получен ответ об успешности его выполнения;
- **масштабируемость** (англ. *partition tolerance*) — система продолжает функционировать, несмотря на возможную потерю сообщений между узлами или отказ части системы.

Следствие

При разработке распределенных систем хранения данных выбираются два из трех требований (чаще всего — AP или CP), в зависимости от условий использования.

MapReduce

Определение

MapReduce — программная модель для параллельной обработки больших объемов данных в распределенных системах, сходная с применением функций `map` и `reduce` в функциональном программировании.

Этапы вычисления:

1. подготовка данных для процедуры `Map()` на узлах системы, устранение дублирующихся данных;
2. выполнение кода `Map()`, заданного пользователем;
3. реорганизация данных (англ. *shuffle*) для выполнения функции `Reduce()`;
4. выполнение кода `Reduce()`, заданного пользователем;
5. вывод полученного результата.

Пример MapReduce

Задача. Определить файл (один из файлов) с максимальным количеством слов.

Вход: набор имен файлов.

Выход: словарь с одним вхождением (файл → количество слов).

Псевдокод (Python):

```
1  def user_map(fname):
2      """ Возвращает словарь, включающий количество слов для входного файла. """
3      wc = 0
4      with open(fname, 'r') as f:
5          for line in f: wc += len(line.split())
6      return { fname: wc }
7
8  def user_reduce(records):
9      """ records — коллекция словарей, возвращенных функцией map.
10         Возвращает словарь, соответствующий файлу с макс. числом слов. """
11     max_wc = max([rec.values()[0] for rec in records])
12     for rec in records:
13         if rec.values()[0] == max_wc: return rec
```

Пример MapReduce

Задача. Определить файл (один из файлов) с максимальным количеством слов.

Вход: набор имен файлов.

Выход: словарь с одним вхождением (файл $\rightarrow$ количество слов).

Псевдокод (Python):

```
# Локальное выполнение
map_result = map(user_map, filenames)
result = reduce(lambda x, y: user_reduce((x, y)), map_result)
# Распределенное выполнение
def node_exec(node, local_files):
    """ Выполняется на каждом вычислительном узле node """
    map_result = map(user_map, local_files)
    return user_reduce(map_result)
# На главном сервере
local_files = [distribute(filenames, node) for node in nodes]
local_results = [node_exec(node, files) for node, files \
    in zip(nodes, local_files)]
result = user_reduce(local_results)
```

Характеристики MapReduce

Особенности:

- Процедура Reduce может выполняться в несколько этапов по мере поступления данных на каждом узле и при агрегации данных на различных узлах.
- Метод MapReduce эффективен для обработки больших объемов данных ($\sim$ Гб–Тб).

Области применения:

- распределенный поиск и индексирование;
- распределенная сортировка;
- получение статистики по документам в распределенных хранилищах;
- математические приложения (напр., сингулярное разложение матриц);
- машинное обучение (напр., кластеризация документов, машинный перевод, …).

Облачные файловые системы

Определение

Облачная файловая система (англ. *distributed file system for cloud*) — файловая система с распределенной архитектурой, предоставляющая пользователям одновременный полноценный сетевой доступ к данным / файлам.

Цели:

- оптимизация пакетной обработки данных (напр., с помощью MapReduce);
- высокая доступность (англ. *high availability*) — доступ к данным при возможности отказа узлов системы;
- поддержка сложной топологии системы (географически разделенные узлы и кластеры);
- поддержка больших файлов (до нескольких Тб) и большого количества файлов;
- использование TCP/IP и удаленного вызова процедур для доступа к данным.

Облачные файловые системы

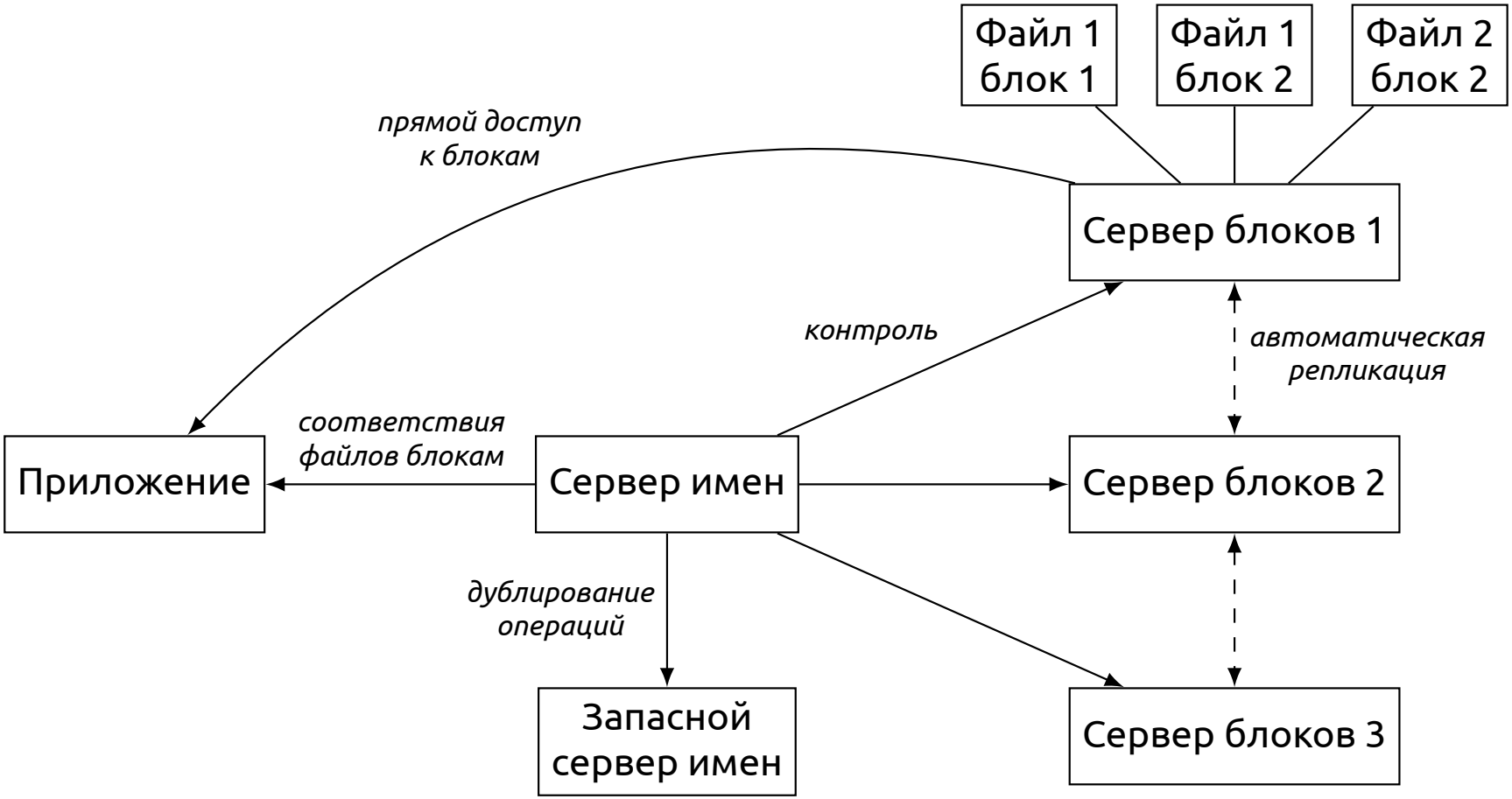
Характеристики облачных ФС:

- разделение файлов на блоки ($\sim$ несколько Мб) для оптимизации доступа;
- дублирование блоков на нескольких узлах для отказоустойчивости. Часто подбираются географически разделенные узлы для оптимизации скорости доступа.
- выделенные серверы для хранения метаданных (соответствия блоков файлам, положение файлов в директориях, …).

Примеры облачных ФС:

- Google File System;
- Hadoop Distibuted File System (HDFS);
- Lustre;
- IBM General Parallel File System (GPFS).

Архитектура облачных ФС



Типичная архитектура облачных файловых систем

Распределенные БД

Недостатки реляционных БД для облачных приложений:

- невозможность линейного горизонтального масштабирования (*scaling out* — линейный рост производительности при увеличении количества узлов), слабая совместимость с распределенными системами;
- отсутствие или недостаточность встроенных механизмов кэширования;
- фрагментация при хранении больших объемов данных;
- жесткость схемы данных, необходимость структуризации входящей информации;
- транзакции для соблюдения согласованности данных, замедляющие работу системы;
- уменьшающие производительность операции нормализации данных и объединения таблиц (оператор SQL **`JOIN`**).

NoSQL

Определение

NoSQL (not only SQL) — модель данных для распределенного хранения данных, отличающаяся от реляционной алгебры традиционных СУБД.

Характеристики:

- отсутствие жесткой схемы данных, проектирование структур данных согласно заранее заданным шаблонам запросов (а не наоборот, как в РСУБД);
- упрощение структуры данных по сравнению с реляционными таблицами, отсутствие нормализации;
- отказ от транзакций в пользу отложенной согласованности (англ. *delayed consistency*);
- встроенная поддержка распределенной архитектуры и (часто) кэширования.

Структуры данных в NoSQL

Типы баз данных (от простых к сложным):

- **Пары «ключ — значение».** Используются для кэширования; зачастую данные хранятся исключительно в оперативной памяти.

  Примеры: Redis; memcached.

- **На основе столбцов** (англ. *column-oriented*). Используются для хранения просто структурированных данных при необходимости быстрого доступа.

  Примеры: Apache Cassandra, Apache HBase.

- **Графовые.** Хранят отношения между сущностями (напр., followers / followed by в Twitter).

  Примеры: Neo4j; OrientDB.

- **Документно-ориентированные.** Используются для хранения произвольных документов со схемой, задающейся форматом сериализации (напр., JSON).

  Примеры: Apache CouchDB, MongoDB.

Обработка данных

Машинное обучение — извлечение полезной информации из данных при помощи методов оптимизации / мат. статистики:

- регрессия (полиномиальная, MARS);
- классификация (байесовские методы, решающие деревья, SVM, бустинг, …);
- структурное распознавание (обработка изображений, текста, …);
- кластеризация (k-means, гауссовские смеси);
- сокращение размерности (сингулярное разложение и другие методы).

Обратное индексирование данных — подготовка индекса для полнотекстового поиска в большом объеме документов.

Apache Hadoop

Определение

Apache Hadoop — оболочка для распределенного хранения и обработки данных, написанная на ЯП Java.

Модули:

- HDFS — распределенная файловая система для хранения данных;
- система выполнения заданий MapReduce — JobTracker (центральный модуль управления заданиями), TaskTracker (выполнение процедур `Map` и `Reduce`);
- планировщик заданий.

Подключаемые модули Hadoop

- **Файловые системы** (доступны через плагины).
- **NoSQL-база данных** HBase (устанавливается поверх HDFS).
- Apache Spark — архитектура для выполнения **распределенной обработки данных**, альтернативная MapReduce.
- Apache Mahout — библиотека **машинного обучения**, написанная на Java.
- Apache Pig — высокоуровневая **платформа для создания заданий типа MapReduce** на основе процедурного ЯП Pig Latin ($\sim$ SQL) и пользовательских функций на Python, Java, JavaScript.
- Apache Hive — **инфраструктура для обработки данных** с использованием языка запросов HiveQL, который транслируется в набор заданий для Hadoop.
- Apache ZooKeeper — **централизованный сервер координации** в распределенных системах.

Выводы

1. Облачные вычисления — платформа для выполнения распределенных приложений (как веб-, так и аналитических), основанная на принципе горизонтальной масштабируемости.
2. Есть три уровня облачной архитектуры: инфраструктура как сервис (IaaS), платформа как сервис (PaaS) и ПО как сервис (SaaS).
3. Основой облачной архитектуры является хранение данных (при помощи распределенных ФС и NoSQL-баз данных) и их обработка (напр., с помощью инструментов типа MapReduce).
4. Основные характеристики распределенных хранилищ данных — структура хранимых объектов и характеристики из набора доступность, согласованность данных и масштабируемость. Согласно CAP-теореме, выполнение трех характеристик одновременно невозможно.

Материалы

Sommerville, Ian

Software Engineering.

Pearson, 2011. — 790 p.

Voorsluys W., Broberg J., Buyya R.

Introduction to Cloud Computing.

http://media.johnwiley.com.au/product_data/excerpt/90/04708879/0470887990-180.pdf

Cloud Security Alliance

Big Data Taxonomy.

https://downloads.cloudsecurityalliance.org/initiatives/bdwg/Big_Data_Taxonomy.pdf

Спасибо за внимание!