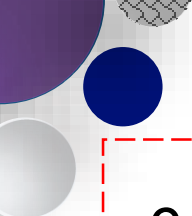




9

# 大数据计算体系



## 9 大数据计算体系

9.1 大数据计算体系综述

9.2 数据存储系统

9.3 数据处理系统

9.4 数据应用系统

## 9.1 大数据计算体系综述

### **体系主要组成**

- 数据存储系统
- 数据处理系统
- 数据应用系统

## 9.1 大数据计算体系综述

### 数据存储系统

#### 主要组成

- 数据采集层
- 数据清洗、抽取与建模
- 数据存储架构

#### 地位

- 大数据计算的基础

## 9.1 大数据计算体系综述

### 数据处理系统

- 主要组成
  - 针对不同类型数据的计算模型
  - 针对应用需求的各类数据分析算法
  - 数据计算处理各种开发工具包
  - 运行支持环境的计算平台

## 9.1 大数据计算体系综述

### **数据应用系统**

- 基于上述存系统和计算处理平台提供各行业各领域的大数据应用技术解决方案



## 9.2 数据存储系统

9.2.1 数据清洗与建模

9.2.2 分布式文件系统

9.2.3 NoSQL数据库

9.2.4 统一数据访问接口

## 9.2.1 数据清洗与建模

### 大数据数据特点

- 多种数据源
- 异构型数据
- 非结构型数据
- 影响：
  - 原始数据很难直接存入数据库



## 9.2.1 数据清洗与建模

- 数据清洗：
  - 合并或删除重复数据项，消除数据错误
- 数据抽取
  - 从多个数据源的数据项中抽取不同值域构成目标数据库的数据结构
  - 从一个数据源抽取数据项分解成多个结构装载入目标数据库

## 9.2.1 数据清洗与建模

### 数据建模

- 定义
  - 对实体数据（或用户对数据功能的描述）建立一个抽象模型
  - 模型主要包括元数据、数据结构、属性、值域、关联关系、一致性、时效性等元素。
- 意义
  - 为进一步的数据存储结构设计、数据库设计和计算模型提供了参考依据

## 9.2.1 数据清洗与建模

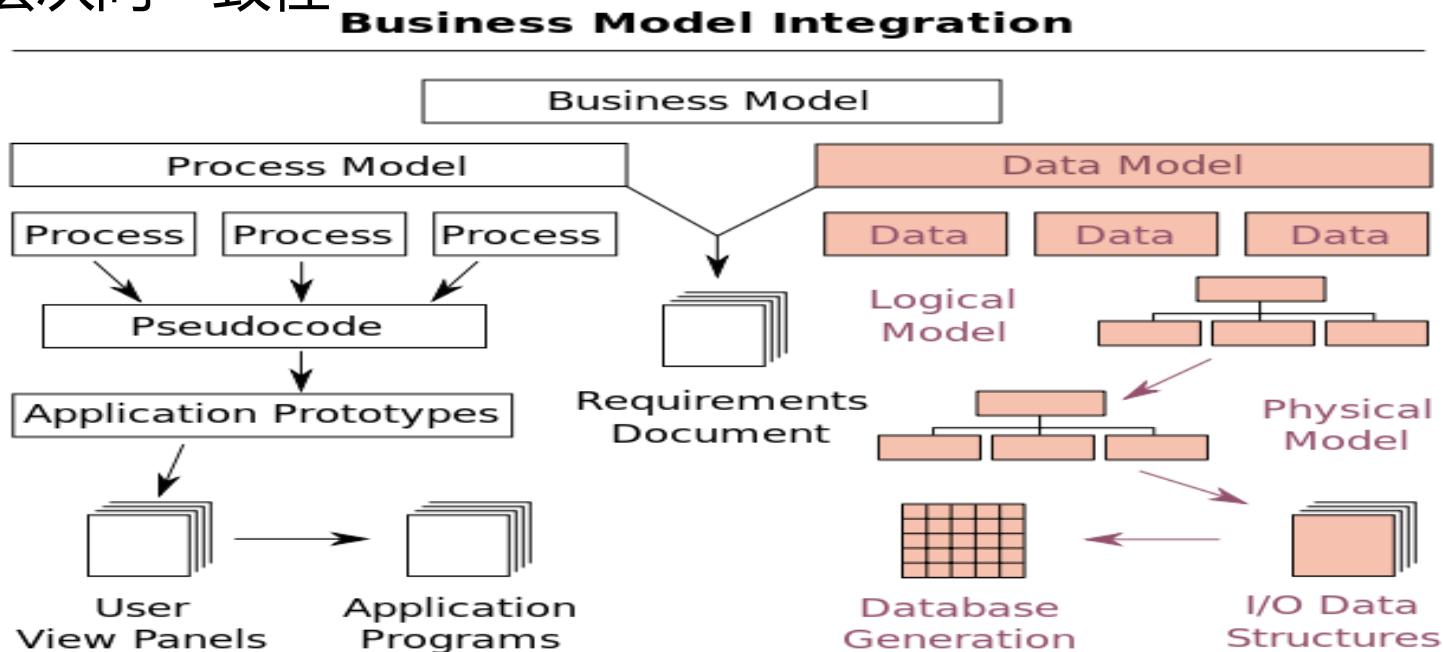
### 数据建模

- 主要包含层次
  - 概念模型 (conceptual model)
  - 逻辑模型 (logic model)
  - 物理模型 (physical model)
- 层次间关系
  - 相对独立
  - 一致性

## 9.2.1 数据清洗与建模

### 数据建模

- 层次间一致性



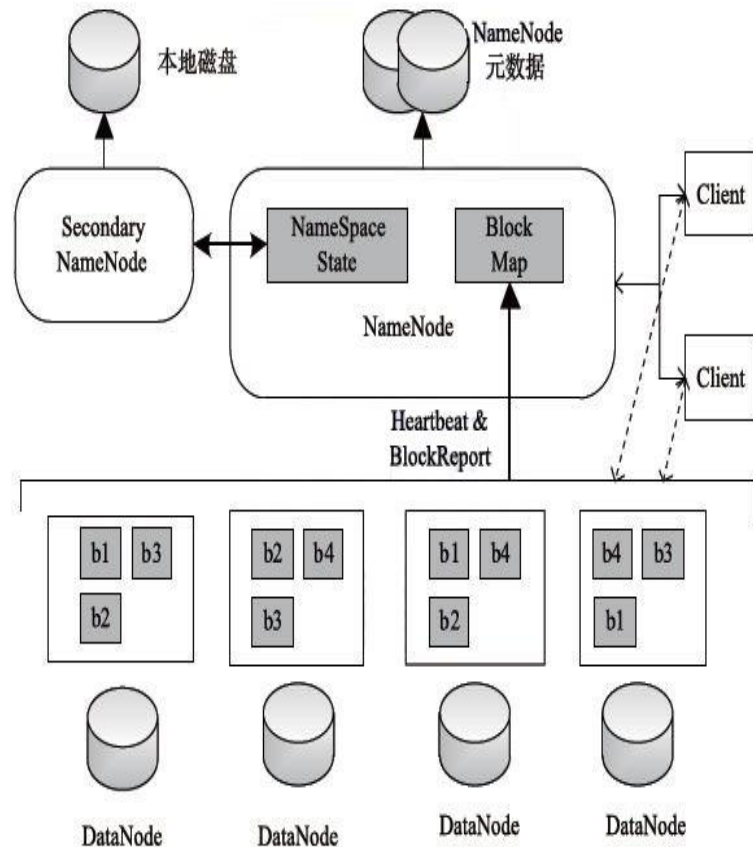
## 9.2.2 分布式文件系统

- 作用
  - 提供了数据的物理存储架构
- 主要种类
  - HDFS (Hadoop Distributed File System)
  - GFS (Google File System) (目前已演化成Colossus系统)

## 9.2.2 分布式文件系统

### HDFS

- 结构
  - 主从结构
- 组成
  - 一个名称节点(主节点)
  - 若干个数据节点(从节点)



## 9.2.2 分布式文件系统

名称节点(NameNode)	数据节点(DataNode)
管理文件系统命名空间	存储文件blocks
保存文件到block到数据节点的映射关系	实现blocks到数据节点本地文件系统的映射
调度客户端对文件的访问	blocks存储在本地磁盘上
元数据存储在内存，便于快速访问	

## 9.2.2 分布式文件系统

### HDFS

- 流程
  - 每个存储文件被划分为固定长度（64MB）的多个数据块（blocks）
  - blocks按照一定法则分布存储到不同的DataNode上，一个DataNode可以存储来自不同文件的blocks
  - 每个数据节点都运行一个节点程序或进程，负责处理文件系统客户端的读/写请求
  - 在名称节点的统一调度下进行blocks的创建、删除和复制等操作。



## 9.2.2 分布式文件系统

### GFS

- 特点
  - 主节点有多个备份节点，甚至是分布式部署
  - 从节点上存储固定长度（GFS是64MB，后来的Colossus是1MB）的数据单元(chunk)

## 9.2.2 分布式文件系统

### HDFS优劣

- 优：
  - 开源实现
  - 实现更简单
  - 部署灵活
  - 扩展性好
  - 可以运行在廉价硬件设备上
- 劣
  - 数据访问时延较长
  - 处理小文件效率低
  - 不支持多用户写入及任意修改文件

## 9.2.2 分布式文件系统

### GFS优劣

- 优：
  - 快速响应
  - Master节点容错能力强
  - 处理小文件能力强
- 劣
  - Google技术作为商业产品的成本较高

### 9.2.3 NoSQL数据库

#### 特点

- 不需要预定义数据格式
- 无共享架构
- 弹性可扩展
- 数据分区
- 异步复制

### 9.2.3 NoSQL数据库

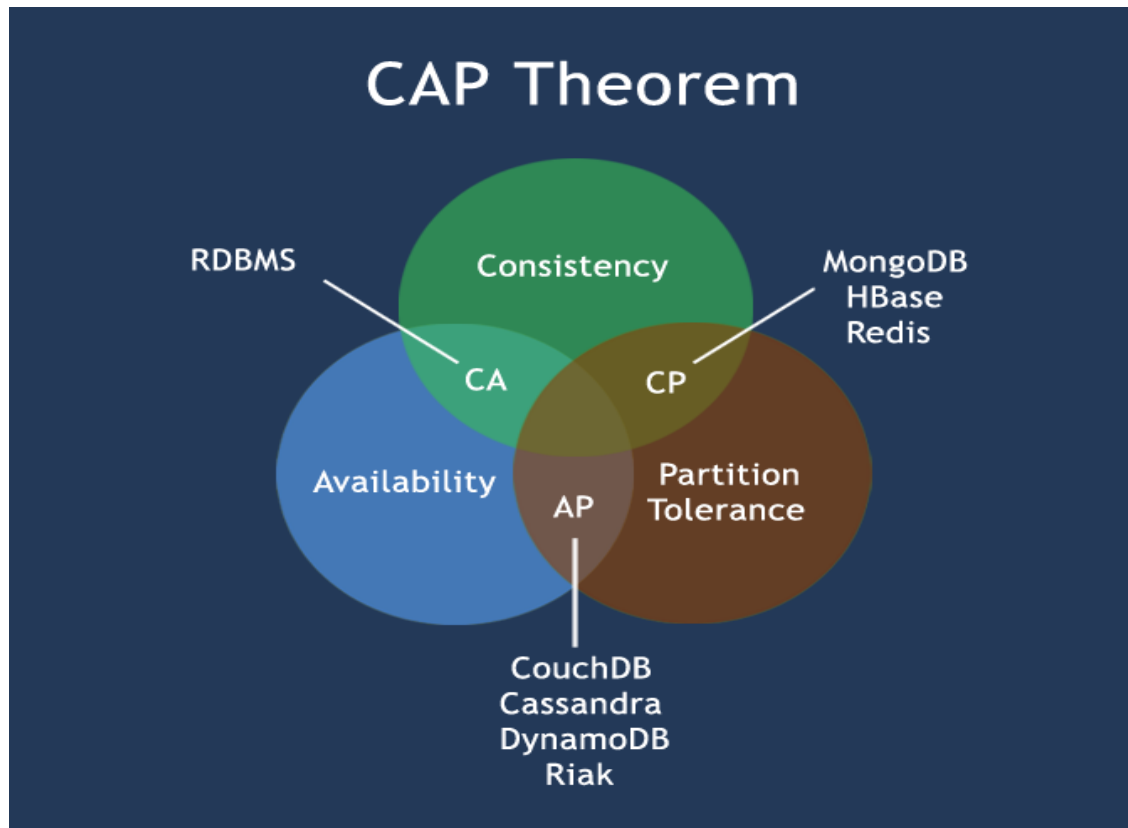
#### **NoSQL数据库性能要求——CAP理论**

- C: 一致性 (Consistency)
- A: 可用性 (Availability)
- P: 分区容忍性 (Partition-tolerance)
- 理论内容: 一个分布式系统在运行中其数据读写操作只能满足三条中的两条, 而无法同时满足三条
- 理论意义: 对CAP三原则的取舍导致了关系型数据库与NoSQL数据库的区别。

## 9.2.3 NoSQL数据库

### CAP理论

- 取舍



### 9.2.3 NoSQL数据库

#### **NoSQL数据库四大种类**

- 键值数据库(key-value store database)
- 列存储数据库(column family-oriented database)
- 文档数据库(document-oriented database)
- 图形数据库(graph-oriented database)

## 9.2.3 NoSQL数据库

### 四类数据库比较

分类	应用场景	数据模型	优点	缺点
键值 (key-value)	内容缓存, 主要用于处理大量数据高访问负载。	Key 指向 Value 的键值对, 通常用 hash table 来实现	查找速度快	数据无结构化, 通常只被当作字符串或者二进制数据
列存储数据库	分布式的文件系统	以列簇式存储, 将同一列数据存在一起	查找速度快, 易进行分布式扩展	功能相对局限
文档型数据库	Web应用结构化的数据	Key-Value对应的键值对, Value为结构化数据	数据结构要求不严格, 不需预先定义表结构	查询性能不高, 而且缺乏统一的查询语法。
图形)数据库	社交网络, 推荐系统等。专注于构建关系图谱	图结构	利用图结构相关算法。比如最短路程寻址等。	很多时候需要对整个图做计算才能得出需要的信息。



## 9.2.4 统一数据访问接口

### ODBC

- 定义：是一组数据库访问API（应用程序编程接口），由一组函数调用组成，核心是SQL语句
- 特性：
  - 用户直接将SQL语句传送给ODBC
  - ODBC对数据库的操作不依赖任何DBMS，不直接与DBMS打交道，所有的数据库操作由对应的ODBC驱动程序完成

## 9.2.4 统一数据访问接口

### ODBC

- 组成：
  - ODBC API编程接口
  - 驱动程序管理器
  - 驱动程序
- 缺点：
  - ODBC API是C语言编写的底层函数，不利于高级语言调用
  - 只能访问关系型数据库——OLE DB

## 9.2.4 统一数据访问接口

### JDBC

- 定义：是一个面向对象的数据库的接口规范，定义了一个支持标准SQL查询的通用程序编程接口(API)
- 特性：
  - 由Java 语言编写的类和接口组成
  - 用于支持Java应用程序对各类数据库的访问
  - 支持同时建立多个数据库连接
  - 可以用SQL语句同时访问多个异构数据库
  - 具有对硬件平台和操作系统的跨平台支持。

## 9.2.4 统一数据访问接口

### UDAL

- 定义：基于统一数据接口用于支持分布式环境中对跨平台异构数据库访问的数据访问层（DAL）
- 功能：
  - 统一的数据展示、存储和管理
  - 访问接口与实现代码分离的原则，底层数据库连接的更改不影响统一数据访问接口
  - 屏蔽了数据源的差异和数据库操作细节，使得应用层专注于数据应用
  - 提供一个统一的访问界面和一种统一的查询语言

## 9.2.4 统一数据访问接口

### **UDAL**

- 组成
  - 统一数据访问界面/统一查询语言
  - 数据模型/元数据/服务模型
  - 数据转换引擎/数据访问引擎/数据源管理器
  - 数据源包装器

## 9.3 数据处理系统

9.3.1 数据分析算法

9.3.2 计算处理模型

9.3.3 计算平台与引擎

### 9.3.1 数据分析算法

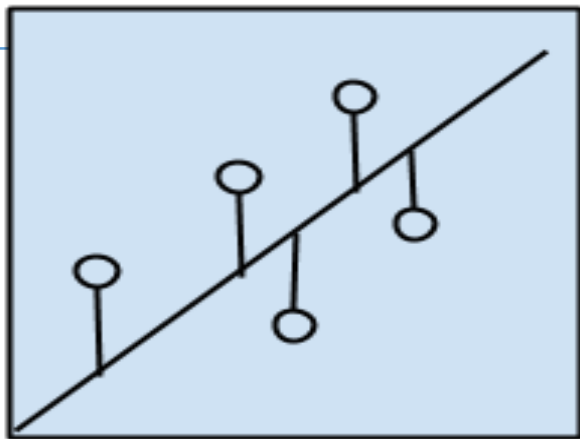
#### **机器学习算法主要种类：**

- 监督学习 (supervised learning)
- 无监督学习 (unsupervised learning)
- 半监督学习 (semi-supervised learning)

## 9.3.1 数据分析算法

### 回归分析类

- 是一种通过最小化预测值与实际结果值之间的差值而得到输入特征之间关系的一类算法。对于连续值预测有线性回归算法，而对于离散值/类别预测也可以把逻辑回归等也视作回归算法的一种

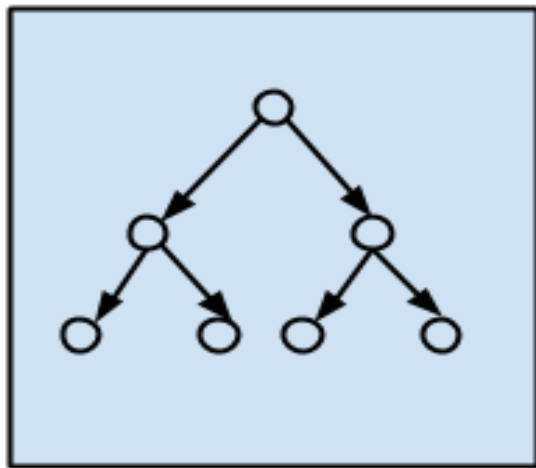


Regression Algorithms



### 9.3.1 数据分析算法

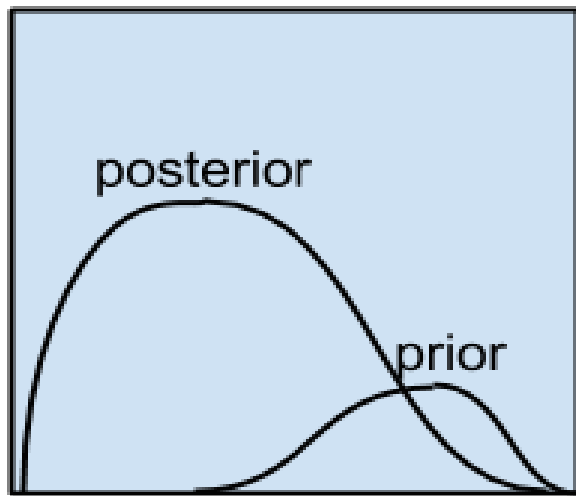
- 决策树类算法
- 基于原始数据特征，构建一颗包含很多决策路径的树。预测阶段选择路径进行决策。



Decision Tree  
Algorithms

### 9.3.1 数据分析算法

- 贝叶斯类算法
- 在分类和回归问题中，隐含使用了贝叶斯原理的算法。



Bayesian Algorithms

### 9.3.2 计算处理模型

- MapReduce
- 交互式处理
- 流计算

### 9.3.2 计算处理模型

#### **MapReduce**

- 是一种支持分布式计算环境的并行处理模型
- 运行在由多台计算机组成的Master/Slave集群架构上
- 一个Master节点，负责任务调度和管理
- 多个Slave节点，执行具体的计算任务

## 9.3.2 计算处理模型

### MapReduce

- 计算流程
  - Split (数据划分)
  - Map (映射)
  - Collect&Sort (聚合排序, 也称Shuffle)
  - Reduce (简化)
  - Store (数据存储)
- 特点
  - 分治策略
- 缺点
  - 硬盘数据读取频繁, 处理时效性较差

### 9.3.2 计算处理模型

#### 流计算模型

- 是一种处理实时动态数据的计算模型
- 特性：针对动态连续性数据流进行实行分析计算，获得计算结果后，数据要么导入静态数据库要么丢弃，即数据一次性使用。
- 主要步骤：
  - 数据实时采集
  - 数据实时计算
  - 实时查询服务

### 9.3.3 计算平台与引擎

#### 计算平台

- Hadoop
- Cloudera
- Spark
- Storm

#### 计算引擎

- MapReduce计算引擎
- 交互式计算引擎
- 图并行计算引擎
- S4 (Simple Scalable Streaming System)

## 9.4 数据应用系统

### 大数据应用领域

- 政府
- 互联网
- 电信
- 金融
- .....



## 9.4 数据应用系统

### 大数据解决方案

- IBM 大数据计算处理InfoSphere平台
- Oracle 软硬一体化的大数据技术解决方案
- 微软 大数据解决方案Microsoft Azure HDInsight
- .....