**[云计算导论1. 数据](#_Toc5901)**

[1. 云计算 2](#_Toc4709)

[1. 定义 2](#_Toc8322)

[2. 典型特征 2](#_Toc8872)

[3. 服务模型：laaS,PaaS,SaaS 3](#_Toc20466)

[4. 公有云和私有云 4](#_Toc16726)

***资源池、集中共享、虚拟化、动态可伸缩、面向服务、laaS、PaaS、SaaS、云服务***

[2. 虚拟化Virtualization 4](#_Toc1047)

[1. 定义 4](#_Toc12028)

[2. 分类：计算虚拟化，存储虚拟化，应用程序虚拟化 4](#_Toc27909)

***仿真建模、虚拟机(计算虚拟化)、存储虚拟化软件定义***

[3. 分布式文件系统 5](#_Toc27057)

1. 大数据 5

2. 定义 6

3. 关键问题 7

1. 命名空间 7

2. 元数据管理 8

3. 云存储 9

***命名空间、元数据、HDFS、HA策略、云存储***

[4. NoSQL数据库 9](#_Toc14065)

[1. 分类(基于数据模型) 9](#_Toc10892)

[2. 基本原理：CAP定理 10](#_Toc15174)

3. HBase 11

***CAP(分布式操作系统性质定理)、Apache、Hadoop、时间戳***

[5. MAP REDUCE 并行计算框架 12](#_Toc6980)

[1. 简介 12](#_Toc18139)

[2. 典型的批量大数据问题的特征 12](#_Toc24257)

3. MAP & REDUCE 13

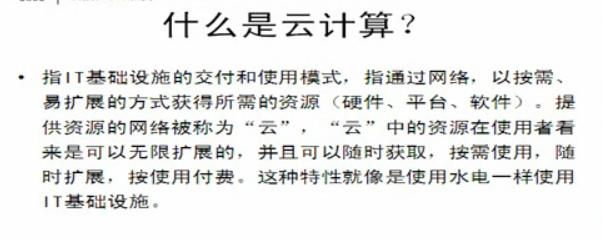
4. 应用：文档词频统计 13

***并行、海量数据处理、键值对***

# **云计算导论1. 数据处理**

### 1. 云计算

###### 定义

* 云是什么：在用户看来，云计算提供了一种大规模的资源池，资源池管理的资源包括计算、存储、平台和服务等各种资源，资源池中的资源经过了**抽象和虚拟化的处理**，并且是动态可扩展的(Foster, Zhao, Raicu, et al., 2008)
* 
* 通俗解释：大量it基础设施（服务器，硬件资源，软件资源，网络， 存储，运算设备）集中起来完成很多以前靠单个服务器完成不了的工作，这种it部署和应用的模式，叫云计算模式
* 云计算＝ 互联网:((∑资源(设备、数据、软件))×服务
* 我们的定义：云计算是分布式计算的一种形式，它强调在互联网上建立大规模数据中心等IT基础设施，通过面向服务的商业模式为各类用户提供基础设施能力，是建造和运维互联网分布系统相关技术的总称。

###### 典型特征

* 规模经济驱动的资源**集中共享**
* 资源**虚拟化**
* 系统**动态可伸缩**
* **面向服务**的商业模式

###### 服务模型

* 基础设施即服务laaS
  + 向客户提供处理、存储、网络以及其他基础计算资源，客户可以在上运行任意软件，包括操作系统和应用程序
  + 用户不管理或者控制底层的云基础架构，但是可以控制操作系统、存储、发布应用程序，以及可能限度的控制选择的网络组件（例如，防火墙）
  + 仅提供计算和存储等基础功能，应用的实现需用户自行完成
* 平台即服务PaaS
  + 客户使用云供应商支持的开发语言和工具，开发出应用程序，发布到云基础架构上
  + 客户不管理或者控制底层的云基础架构，包括网络、服务器、操作系统或者存储设备，但是能控制发布应用程序和可能的应用程序运行环境配置
* 软件即服务SaaS
  + 客户所使用的服务商提供的这些应用程序运行在云基础设施上。这些应用程序可以通过各种各样的客户端设备访问
  + 客户不管理或者控制底层的云基础架构，包括网络、服务器、操作系统、存储设备



###### 公有云和私有云

* 公有云：是指为外部客户提供服务的云，他所有的服务是供别人使用的，而不是自己使用。
* 私有云：是指企业自己使用的云，他所有的服务不是供别人使用，而是公司及内部人员或分支机构使用

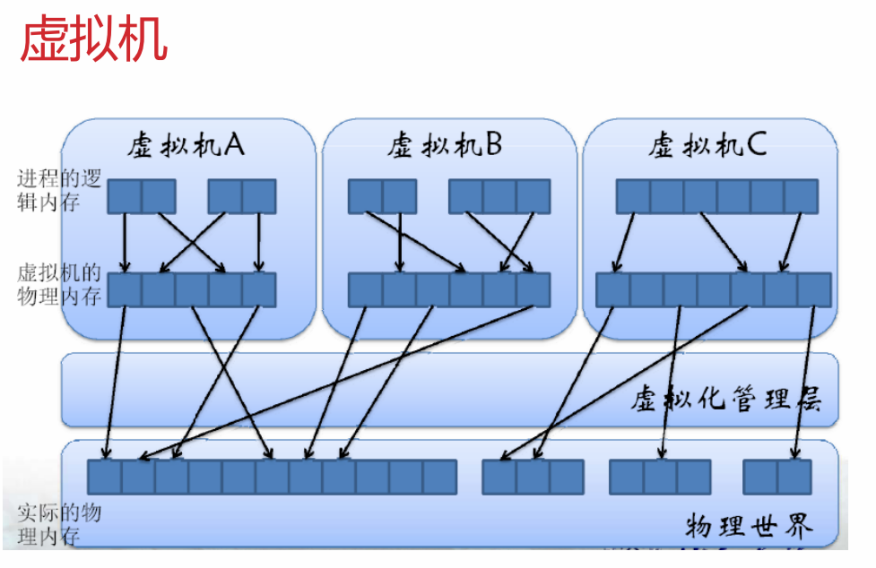
### 2. 虚拟化Virtualization

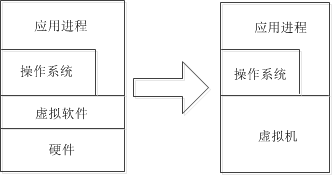
###### **定义**

* 本[[1]](#endnote-0)质：建模问题
* 资源的逻辑表示，不受物理限制约束
  + 虚拟化的对象是各种各样的资源
  + 经过虚拟化后的逻辑资源对用户隐藏了不必要的细节
  + 用户可以在虚拟环境中实现其在真实环境中的部分或全部功能

**故：虚拟化是一种仿真建模，一个变多个，多个像一个**

###### 分类

* 计算虚拟化
  + 比如一台机器上同时运行多个操作系统，每个OS的Kernel都以为独享硬件
  + 虚拟机
    - 
    - 通过虚拟化软件套件模拟的、具有完整硬件功能的、运行在一个隔离环境中的逻辑计算机程序



* 存储虚拟化
  + 我的文件可能分散存放到了不同位置
  + **新发展趋势：软件定义(Software-defined)存储**
    - Windows Server 2012 R2
    - 提供RAID配置
    - 存储分层等服务
    - 软件定义网络：在软件里定义公司或机构的所有的网络（浅）；分层、建抽象、通过补充额外的智能算法等定义附加智能（深）
* 应用程序虚拟化
  + 例如在同一台机器上同时运行office2003和office2010

### 3. 分布式文件系统

###### 大数据

* 定义
  + 很抽象，表示数据规模的庞大
  + 大数据泛指巨量的数据集，因可从中挖掘出有价值的信息而受到重视
  + 《维基百科》将**大数据是指利用常用软件工具捕获、管理和处理数据所耗时间超过可容忍时间的数据集**
* 特征
  + 大量化Volume
  + 快速化Velocity
    - 处理速度快
  + 多样化Variety
    - 数据类型繁多
  + 价值化Value
    - 价值密度低，商业价值高

###### 发展背景

* 互联网发展，海量数据存储

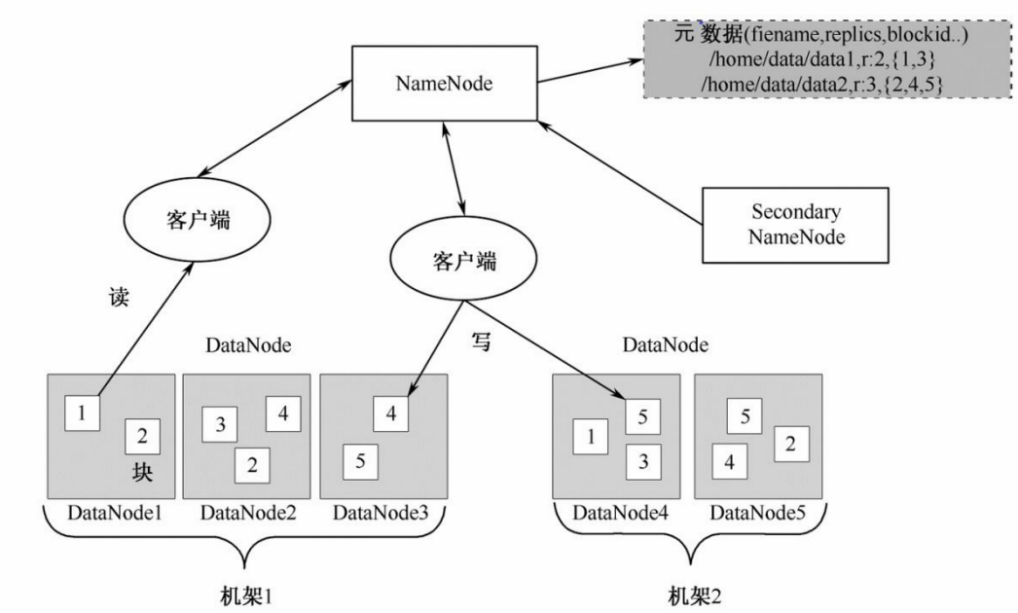
###### 定义

* 文件系统：是操作系统用来组织磁盘文件的方法和数据结构。它通过对操作系统所管理的存储空间的抽象，向用户提供统一的、抽象化的访问接口，屏蔽对物理设备的直接操作和资源管理。
* （Distributed File System）是指文件系统管理的物理存储资源不一定直接连接在本地计算机上，而是通过计算机网络与计算机相连。从内部实现来看，分布式文件系统不再和普通文件系统一样负责管理本地磁盘，它的文件内容和目录结构也不存储在本地磁盘上，而是通过网络传输到远端系统上。

###### 关键问题



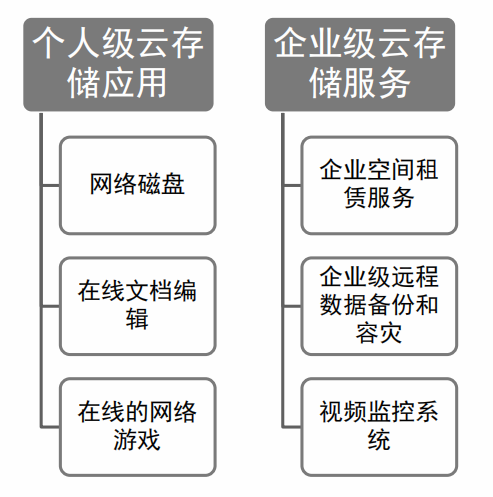
* 命名空间Namespace
  + 一个系统中实体名称的集合，命名空间指定了**名称构造的规则**，界定了**名称的范围**，在命名空间中名称唯一地标识一个实体，名称不允许重复
  + 实体名称构造方式
    - 无结构命名
      * 命名空间的命名构造规则除了**指定构成名称的符号集合**之外，**并没有指定名称的结构，对组成名称的单个符号并没有指定特殊的含义**
      * 如全局唯一标识符（Globally Unique Identifier，GUID；或者Universally Unique Identifier，UUID），全局唯一标识符是由128位二进制符号组成的，用于在全球范围内唯一标识一个数字对象，比如一个注册表项、一个软件组件、一个Word文档等。
      * GUID 的格式为“xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx”，其中每个 x 是 0-9 或 a-f 范围内的一个十六进制数。例如：6F9619FF-8B86-D011-B42D-00C04FC964FF 即为有效的 GUID 值。
    - 结构化命名
      * 结构化命名方法除了**指定名称的符号集**之外，还对**名称的结构**做了规定
      * 最常见的结构化命名方式是**分层表示**，比如计算机文件系统的命名和互联网的域名系统
* 元数据管理
  + 元数据类型
    - 文件和数据块的命名空间：主要记录目录结构、文件名称和数据块的名称
    - 文件到数据块的映射信息：一个文件通常由多个数据块组成，这些数据块按存储策略较均匀地分布在整个集群的存储空间中。这部分元数据记录组成一个文件的所有数据块和文件大小等信息
    - 每个数据块的备份位置：记录每个数据块的存储位置，以及它的备份位置
  + HDFS的体系结构



* + HDFS架构概念
    - Block：一个文件分块，默认64M
    - NameNode：保存整个文件系统的目录信息、文件信息和文件相应的分块信息
    - DataNode：用于存储Blocks（命名空间是管理者：维护文件系统树和整棵树内所有文件和目录。执行命名空间操作数据结点是工作者：存储和检索数据块，处理客户端文件读写请求）
    - HDFS的HA策略：NameNode一旦宕机，整个系统无法工作；NameNode中的数据丢失，整个文件系统丢失
* 云存储

**云存储不是存储，是服务**

* + 定义：是指通过集群应用、网格技术或**分布式文件系统**等功能，将网络中大量各种不同类型的存储设备通过应用软件集合起来协同工作，共同对外提供**数据存储**和业务访问功能的一个系统
  + 特点
    - 超强扩展性
    - 不受具体地理位置所限
    - 基于商业组件
    - 按照使用收费
    - 可跨不同应用
  + 应用



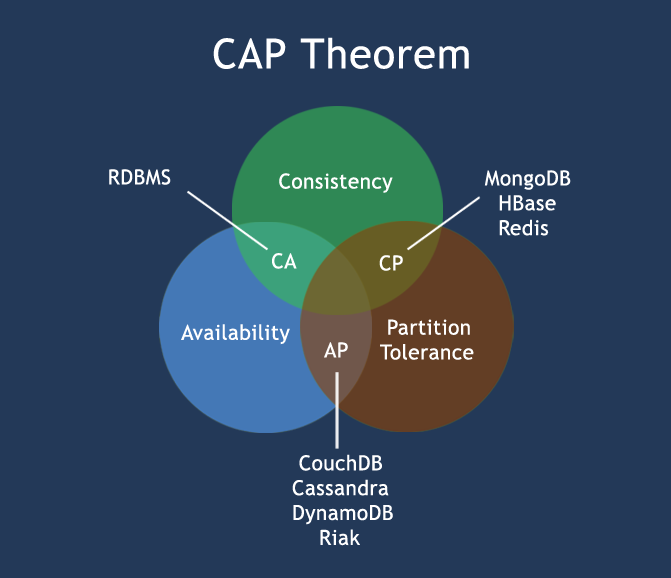
## 4. NoSQL数据库

###### **1. 分类(基于数据模型)**

* KEY/VALUE数据模型：键值对数据库
* KEY/COLUMN数据模型：列存储数据库
* KEY/DOCUMENT数据模型：文档型数据库
* 图数据模型：图关系数据库

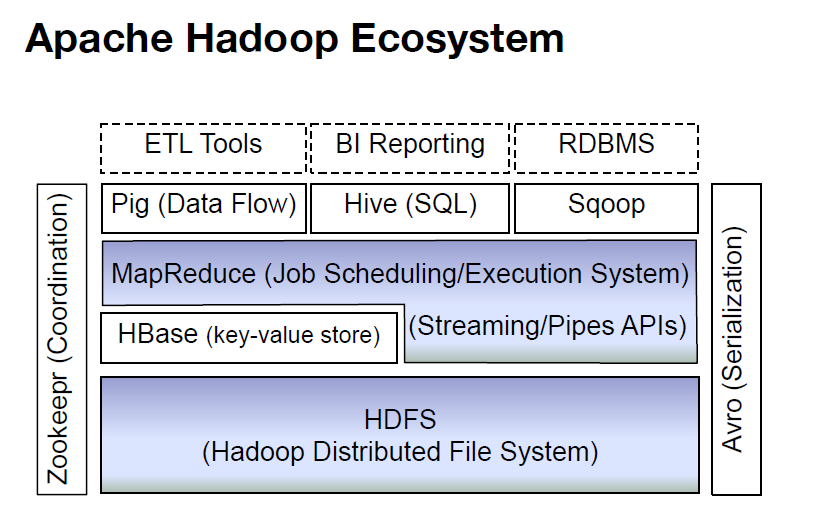
###### **2. 基本原理：CAP定理**

* 背景：CAP理论起源于**Eric Brewer**在2000年的分布式计算原则研讨会（Symposium on Principles of Distributed Computing（PODC））上提出的一个猜想。• 2002年，麻省理工学院（MIT）的Seth Gilbert和Nancy Lynch发表了Brewer猜想的证明，使之成为一个定理。
* 分布式操作系统三大性质
  + Consistency：分布式环境多个节点上副本的数据是一致
  + Availability：在确定时间内返回操作结果
  + Tolerance of Network Partition：系统中的一部分节点无法和其他节点进行通信时，系统仍然正常工作
* CAP定理内容：一个分布式系统不可能同时满足一致性、可用性和分区容忍性，最多同时满足两个

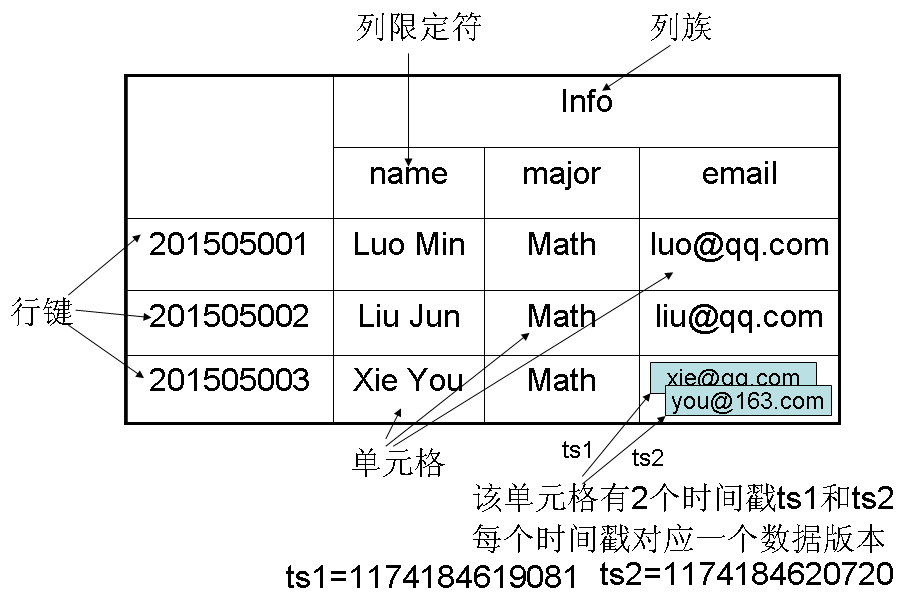


###### **3. HBase**

* Hadoop Database是谷歌BigTable的开源实现，是一个分布式存储系统(NoSql的一种)
* 在Apache Hadoop生态中的位置



* **特点**
  + 面向列Column-Oriented
  + 适合存储海量非结构化数据或半结构化数据
  + 具备高可靠性
  + 可灵活伸缩扩展
  + 高性能(支持实时数据读写)
  + 依赖于HDFS
* 逻辑模型



### MAP REDUCE 并行计算框架

###### 简介

* 谷歌公司发明的一种**面对大规模海量数据处理的高性能并行计算平台**和软件编程框架。是**目前最为成功和最易于使用**的大规模海量数据并行处理技术，广泛应用于搜索引擎（文档倒排索引，网页链接图分析与页面排序等）、Web日志分析、文档分析处理、机器学习、机器翻译等各种大规模数据并行计算应用领域
* 基于集群的高性能并行计算平台Cluster Infrastructure:允许用市场上现成的普通PC或性能较高的刀架或机架式服务器，构成一个包含数千个节点的分布式并行计算集群
* 并行程序开发与运行框架Software Framework:提供了一个庞大但设计精良的并行计算软件构架，能自动完成计算任务的并行化处理，自动划分计算数据和计算任务，在集群节点上自动分配和执行子任务以及收集计算结果，将数据分布存储、数据通信、容错处理等并行计算中的很多复杂细节交由系统负责处理，大大减少了软件开发人员的负担
* 并行程序处理模型与方法Programming Model &amp; Methodology:借助于函数式语言中的设计思想，提供了一种简便的并行程序设计方法，用Map和Reduce两个函数编程实现基本的并行计算任务，提供了完整的并行编程接口，完成大规模数据处理

###### 典型的批量大数据问题的特征

* + 关键思想：为大数据处理过程中的两个主要处理操作提供一种抽象机制
* 大量数据记录/元素进行重复处理
* 对每个数据记录/元素作特定处理获取感兴趣的中间结果信息
* 排序和整理中间结果以利后续处理
* 收集整理中间结果
* 产生最终结果输成

###### **MAP & REDUCE内容**

MapReduce借鉴了函数式程序设计语言Lisp中的思想了，定义了Map和Reduce两个抽象的编程接口，由用户去编程实现Map和Reduce为程序员提供了一个清晰的操作接口抽象描述

* map: (k1; v1) -> [(k2; v2)]
  + 输入：键值对(k1; v1)表示的数据
  + 处理：文档数据记录(如文本文件中的行，或数据表格中的行)将以“键值对”形式传入map函数；map函数将处理这些键值对，并以另一种键值对形式输出处理的一组键值对中间结果[(k2; v2)]
  + 输出：键值对[(k2; v2)]表示的一组中间数据
* reduce: (k2; [v2]) -> [(k3; v3)]
  + 输入： 由map输出的一组键值对[(k2; v2)] 将被进行合并处理将同样主键下的不同数值合并到一个列表[v2]中，故reduce的输入为(k2; [v2])
  + 处理：对传入的中间结果列表数据进行某种整理或进一步的处理,并产生最终的某种形式的结果输出[(k3; v3)]
  + 输出：最终输出结果[(k3; v3)]

###### **应用：文档词频统计**

* 题目：设有4组原始文本数据：Text 1: the weather is goodText 2: today is goodText 3: good weather is goodText 4: today has good weather
* 使用四个map节点：
  + map节点1: 输入：(text1, “the weather is good”)输出：(the, 1), (weather, 1), (is, 1), (good, 1)
  + map节点2: 输入：(text2, “today is good”)输出：(today, 1), (is, 1), (good, 1)
  + map节点3: 输入：(text3, “good weather is good”)输出：(good, 1), (weather, 1), (is, 1), (good, 1)
  + map节点4: 输入：(text3, “today has good weather”)输出：(today, 1), (has, 1), (good, 1), (weather, 1)
* 使用三个reduce节点：
  + reduce节点1: 输入：(good, 1), (good, 1), (good, 1), (good, 1), (good, 1)输出：(good, 5)
  + reduce节点2: 输入：(has, 1), (is,1), (is,1), (is, 1), 输出：(has, 1), (is, 3)
  + reduce节点3: 输入：(the, 1), (today, 1), (today, 1),(weather, 1), (weather,1), (weather, 1)输出：(the, 1), (today, 2), (weather, 3)

MapReduce伪代码（实现map和reduce两个函数）  
Class Mapper  
method map(String input\_key, String   
input\_value):  
// input\_key: text document name   
// input\_value: document contents   
for each word w in input\_value:   
EmitIntermediate(w, "1");   
  
Class Reducer  
method reduce(String output\_key,   
Iterator   
intermediate\_values):   
// output\_key: a word   
// output\_values: a list of counts   
int result = 0;   
  
for each v in intermediate\_values:   
result += ParseInt(v);

1. [↑](#endnote-ref-0)