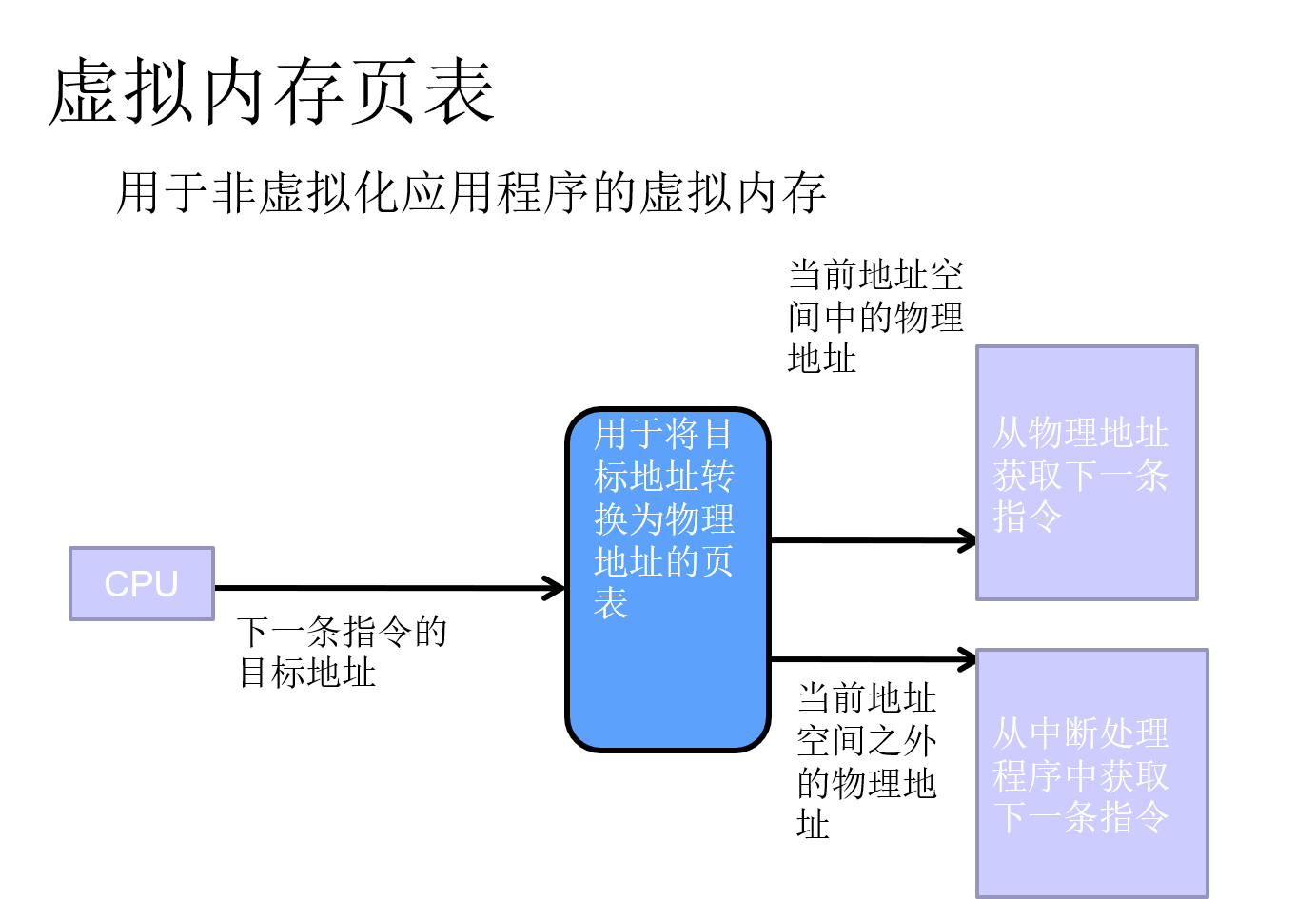
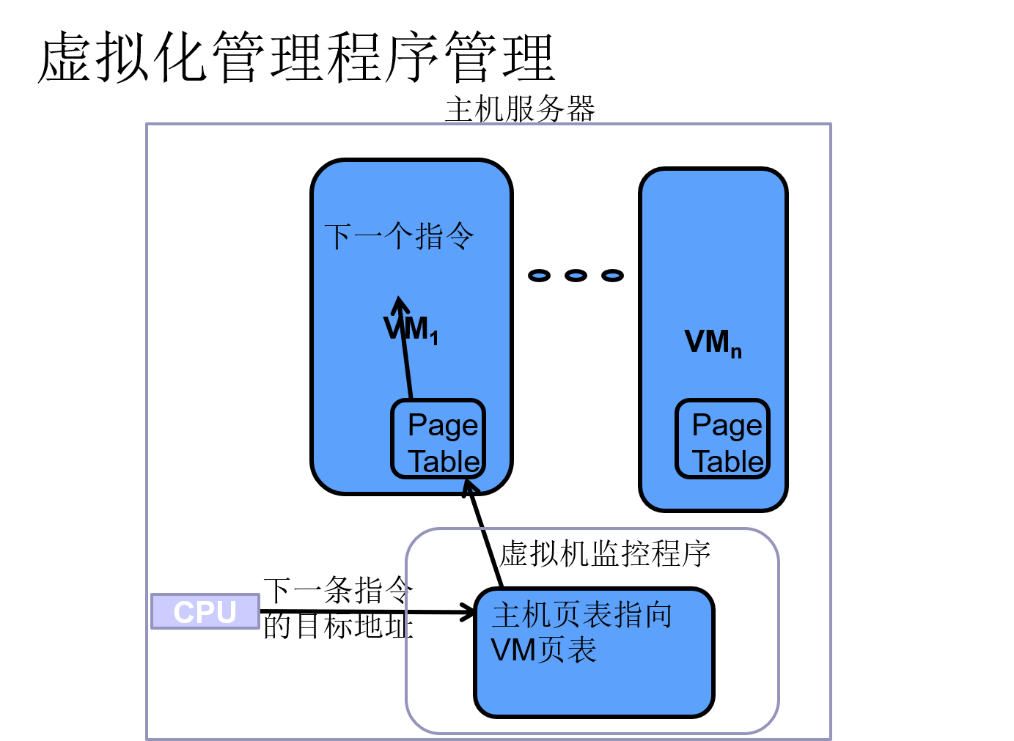
Chapter 26. Architectures for the Cloud

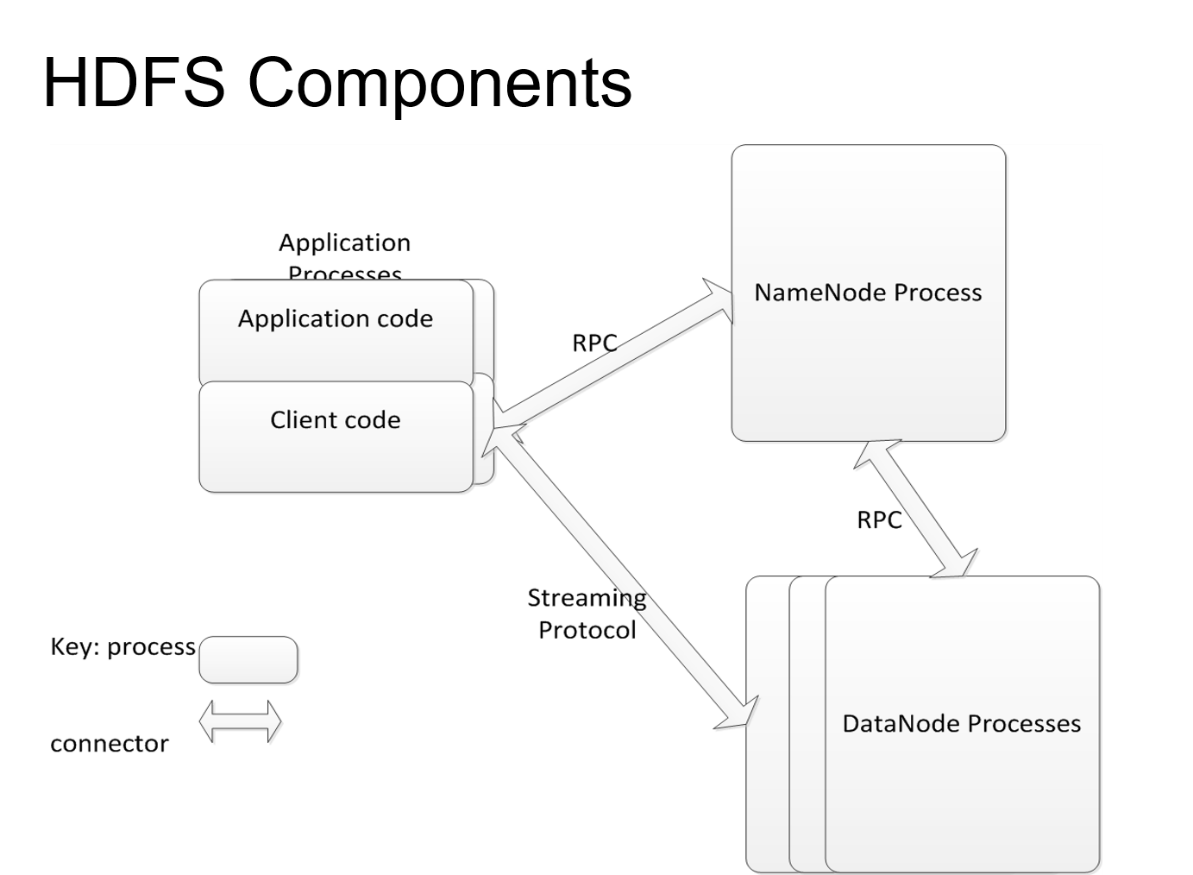
【了解：云计算的特点、主要机制、技术】

* 基本的云计算定义（Basic Cloud Definitions (from NIST)）
  + 按需自助服务On-demand self-service
  + 无处不在的网络访问Ubiquitous network access
  + 资源池Resource pooling
  + 位置独立性Location independence
  + 快速的弹性Rapid elasticity
  + 测量服务Measured service
  + 多租户Multi-tenancy
* 基本服务器模型（Basic Service Models）
  + 软件服务器(Software as a Service ——SaaS)
  + 平台服务器(Platform as a Service ——PaaS)
  + 基础设施服务器(Infrastructure as a Service ——IaaS)
  + 老师的备注：
    - 软件服务器(SaaS)。这里的消费者是终端用户。消费者使用的应用程序恰好运行在云上。例如邮件服务或数据储存服务。
    - 平台服务器(PaaS)。在这种情况下，使用者是开发人员或系统管理员。使用者使用提供者支持的编程语言和工具将应用程序部署到云基础设施上。
    - 基础架构服务器(IaaS)。在这种情况下，使用者是开发人员或系统管理员。向使用者提供的功能是提供处理、存储、网络和其他基本计算资源，使用者可以在这些资源中部署和运行任意软件，这些软件可以包括操作系统和应用程序。
* 部署模型（Deployment Models）
  + Private cloud.
  + Public cloud.
  + Community cloud.
  + Hybrid cloud.
* 经济合理性（Economic Justification）
  + 规模经济（Economies of scale）
    - 与小型数据中心相比，大型数据中心(按单位度量)的操作成本更低。
    - 大型意味着100,000或更多台服务器
    - 小型表示<10,000台服务器。
  + 利用设备（Utilization of equipment）
    - 提高服务器的利用率。
    - 可以管理工作负载的变化以提高利用率。
    - 资源使用模式。将更重的CPU服务与更重的I/O服务放在一起。
    - 不确定性。
  + 多租户（Multi-tenancy）
    - 有些应用程序(如salesforce.com)对多个不同的客户使用单个应用程序。
    - 这通过降低以下成本来降低成本：
      * 帮助台支持
      * 为所有用户同时升级一次
      * 从开发和维护的角度来看，单一版本的软件。
* 基本机制（Basic Mechanisms）
* PS：这个地方很多乱七八糟的图和引用例子
  + 管理程序Hypervisor





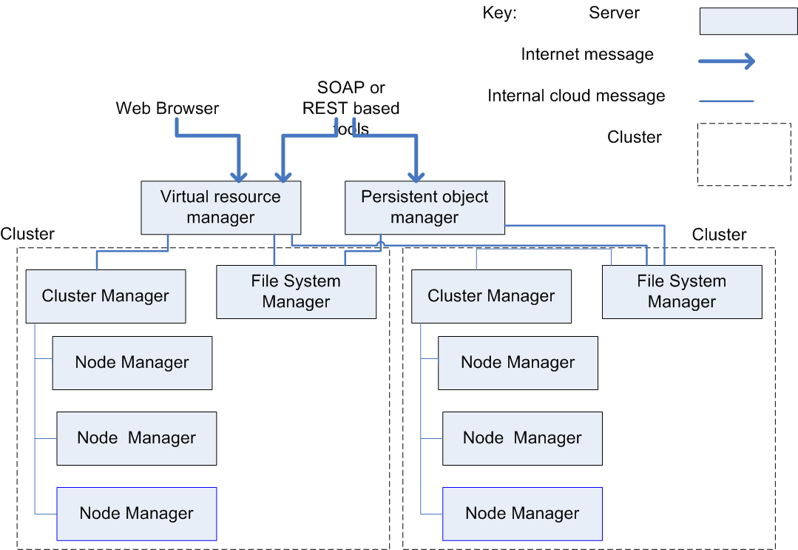
* + 虚拟机Virtual Machine
    - 虚拟机具有与任何其他虚拟机隔离的地址空间。
    - 从应用的角度看，它就像一个裸露的金属机器。
    - 分配一个IP地址并具有网络能力。
    - 可以装载任何可以在主机处理器上执行的操作系统或应用程序。
  + 文件系统File system
    - 每个虚拟机都可以访问一个文件系统。
    - 我们将介绍HDFS (Hadoop分布式文件系统)——一个广泛使用的开源云文件系统。
    - 我们描述了HDFS如何使用冗余来确保可用性。



* 基本事件流（正常操作）：
* 应用程序写入任何文件系统
* 客户端缓冲区，直到它得到64K块
* 客户端通知NameNode它希望写入一个新的块
* NameNode返回三个要保存块的DataNode的列表
* 客户端将块发送到第一个DataNode，并将其他两个副本通知DataNode。
* 第一个DataNode写入块并将其发送到第二个DataNode。第二个DataNode写入块并将其发送到最后一个DataNode。
* 每个DataNode在完成写操作后向客户机报告
* 当客户端从所有三个数据节点都接收到数据时，向NameNode提交写操作。
* 备选事件流（失败操作）：
  + - 客户端失败
      * 应用程序检测并重试
      * 写操作只有在客户端提交后才完成
    - NameNode失败
      * 备份NameNode接管
      * 维护日志文件，避免丢失信息
      * 数据节点维护它们各自拥有的块的真实列表
      * 客户端检测并重试
    - DataNode失败
      * 客户端(或管道中的早期DataNode)检测并请求NameNode提供不同的DataNode。
    - 由于每个块被复制三次，DataNode中的失败不会丢失任何数据。

（自行渣翻译兼渣审核，翻译错了切勿追责）

* + 网络Network
    - 每个虚拟机都有一个IP地址。
    - 每条使用TCP/IP的消息头都包含IP地址。
    - 云网关可以根据不同的目的调整IP地址。
* 基本技术（Sample Technologies）
  + IaaS
    - 管理基本技术的服务器的一种安排。
      * 服务器按集群排列
      * 集群中可能有数千台服务器
      * 一些服务器被用作IaaS的基础设施
      * 每个服务器都有一个hypervisor作为其基础。
    - IaaS架构



* IaaS架构组件：
* 集群管理器（Cluster Manager）负责管理每个集群
* 持久对象管理器（Persistent Object Manager）管理持久性
* 虚拟资源管理器（Virtual Resource Manager）管理其他资源。它充当消息的网关。
* 文件系统管理器类似于HDFS。它管理网络范围的文件系统。
  + IaaS提供的服务：
    - 在底层虚拟机实例失败的情况下，自动重新分配IP地址。
    - 自动缩放。根据负载创建或删除新的虚拟机。
  + PaaS
    - 为开发人员提供一个集成的堆栈。
    - 如灯堆栈
      * Linux, Apache, MySQL, Python
    - 开发人员用Python编写代码，PaaS管理分配给堆栈底层的任务。
  + DataBases
    - 为什么关系数据库会受到质疑
      * 从web系统收集大量数据。这些数据大部分是按顺序处理的，因此rdbms会带来开销，尤其是在创建和维护期间。
      * CAP定理表明，不可能同时实现一致性、可用性和分区。
      * 关系模型不是某些应用程序的最佳模型。
    - 导致了新数据模型的引入
      * 键-值（Key Value – HBase）
        + 指定为键的一列。其他的都是价值观
        + 没有模式，所以数据可以有键+任何其他值。这些值由它们的变量名标识。
        + 数据值也是时间戳
        + Hbase不支持事务。时间戳用于在事件发生后检测冲突。
      * 以文档为中心的

【掌握：云计算下的质量属性】

* 在云环境中进行架构设计（Architecting in a Cloud Environment）
  + 云中的质量属性是不同的
    - 安全（Security）
      * 多租户引入了对非云环境的额外关注。
        + 无意的信息共享。
        + 虚拟机“逃逸”。一个用户可以破坏hypervisor。到目前为止，纯粹是学术性的。
        + 侧信道攻击。
        + 拒绝服务攻击。
      * 在决定在云中托管什么应用程序时，组织需要考虑风险。
    - 性能（Performance）
      * 当负载增长时，自动伸缩提供了额外的性能。
        + 新资源的响应时间可能不够
        + 架构师需要了解应用程序的资源需求

将这些知识构建到应用程序中

应用程序可以自我感知，以便能够主动地满足资源需求。

* + - 可用性（Availability）
      * 失败在云计算中很常见
        + 随着服务器数量的增加，预计会出现故障
      * 云提供商确保云本身将保持可用，但有一些明显的例外。
      * 应用程序开发人员必须假设实例将失败，并在失败的情况下构建检测和纠正机制。

【本章总结】

* 云为具有不同特征的应用程序提供了一个新的平台。
* 架构师需要了解云集群是如何工作的，并特别注意这一点
  + 安全
  + 性能
  + 可用性