高级操作系统

3. 分布式进程与处理机

3.1 系统模型

熊焰，yxiong@ustc.edu.cn
黄文超, huangwc@ustc.edu.cn
http://staff.ustc.edu.cn/~huangwc/advancedos

参考书目：《分布式操作系统》，《分布式系统原理与范型》

本章内容

1. 分布式系统模型

2. 分布式处理机分配

3. 分布式进程调度

4. 分布式系统容错

5. 实时分布式系统

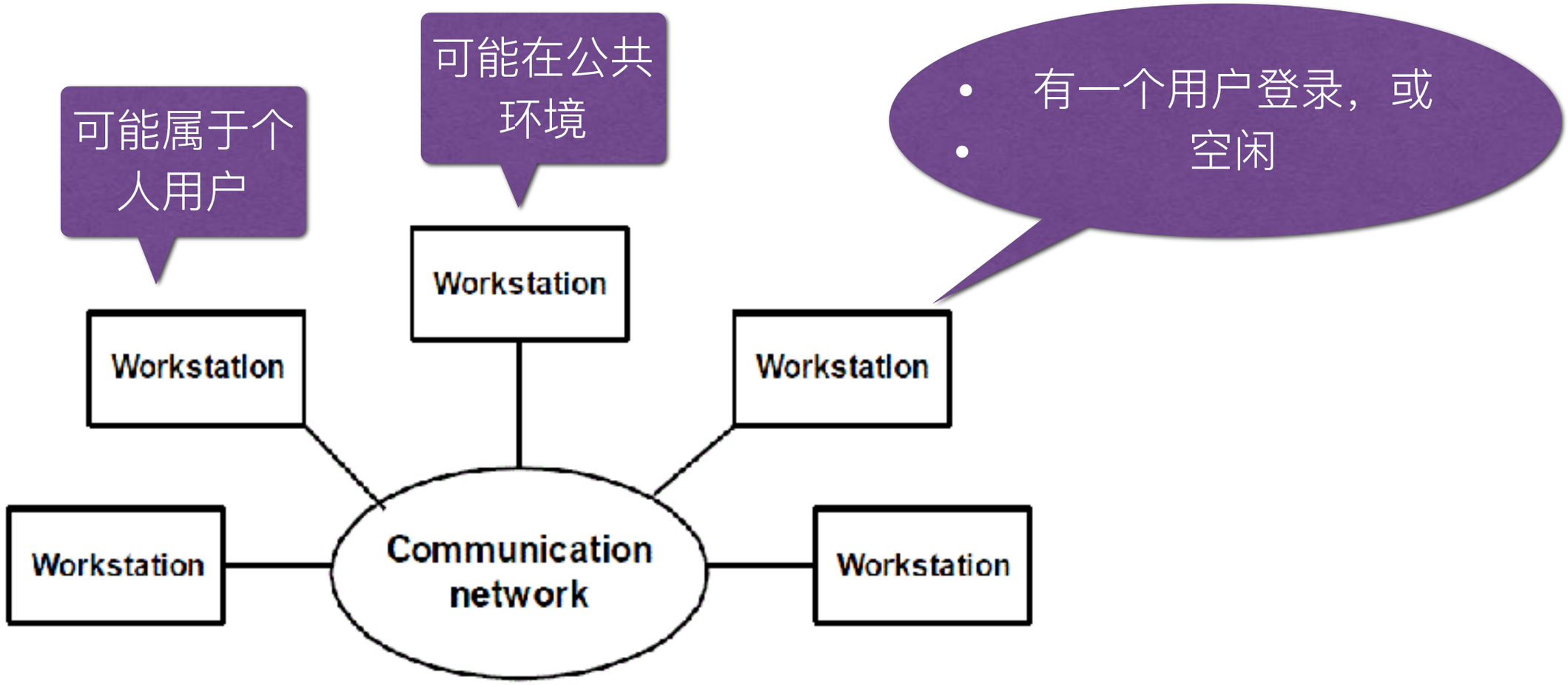
1. 分布式系统模型

- 主要模型：
  - 工作站模型 (workstation model)
  - 处理机池模型 (processor pool model)
  - 混合模型

1. 分布式系统模型

1.1 工作站模型

- 是分布在一幢大楼内或一个校园内通过**高速局域网**连接起来的**工作站（高级个人电脑）集合**



1. 分布式系统模型
1.1 工作站模型

- 工作站类型
  - 有盘工作站 (disky workstation)
  - 无盘工作站
    - 文件系统必须由一个或多个远程文件服务器实现

1. 分布式系统模型

1.1 工作站模型

- **无盘**工作站在大学和公司中流行
  - 优点
    - 价格低
    - 容易维护
    - 磁盘风扇噪音
    - 对称性和灵活性
- 缺点
  - 网络使用频繁

1. 分布式系统模型
1.1 工作站模型

- **有盘**工作站中**磁盘**的使用方式

分页+临时文件	与无盘相比，网络负担更轻
	需大量磁盘，费用较高
分页+临时文件+系统二进制文件	网络负担更轻
	费用较高，**二进制更新比较复杂**
分页+临时文件+系统二进制文件 +文件缓存	网络负担更轻、也减轻文件服务器的负担
	费用较高，**cache一致性的问题**
完整的局部文件系统 (网络操作系统)	几乎没有网络负担，无需文件服务器
	失去透明性

1. 分布式系统模型

1.1 工作站模型

- 工作站模型的优缺点
  - 优点
    - 用户有固定数量的专用计算能力
    - 有保障的响应时间
  - 问题
    - 给每个用户的计算资源与日俱增——解决方案：多处理机
    - 很多工作站是**空闲**的

1. 分布式系统模型
1.2 使用空闲工作站

- 最早尝试允许使用空闲工作站的程序
  - Berkeley UNIX
  - rsh machine command
- 缺陷：
  - 必须指定某台机器
  - 运行环境通常不同
  - 多用户登录导致的性能降低

1. 分布式系统模型
1.2 使用空闲工作站

- 空闲工作站研究的关键问题
  - 怎样找出一台空闲机器?
  - 怎样透明地运行一个远程进程?
  - 如果空闲工作站的主人回来重新使用它，怎么办?

1. 分布式系统模型
1.2 使用空闲工作站

- 怎样找出一台空闲工作站?
  - 空闲工作站的**定义**
    - 几分钟内没有人接触键盘或鼠标
    - 没有用户启动的进程在运行
- 方法种类
  - 服务器驱动
  - 客户端驱动

1. 分布式系统模型
1.2 使用空闲工作站

- 服务器驱动
  - 用户输入命令
    - remote command
  - 远程程序查看**注册文件**，以寻找空闲工作站
    - 注：如何维护注册文件?
      - 方法1: 当一个工作站空闲时，将自己登记到**公用**注册文件
      - 方法2: 当一个工作站空闲时，发送广播，其它工作站更新**私用**注册文件
        - 优缺点：查询开销低，但维护成本大

1. 分布式系统模型
1.2 使用空闲工作站

- 服务器驱动

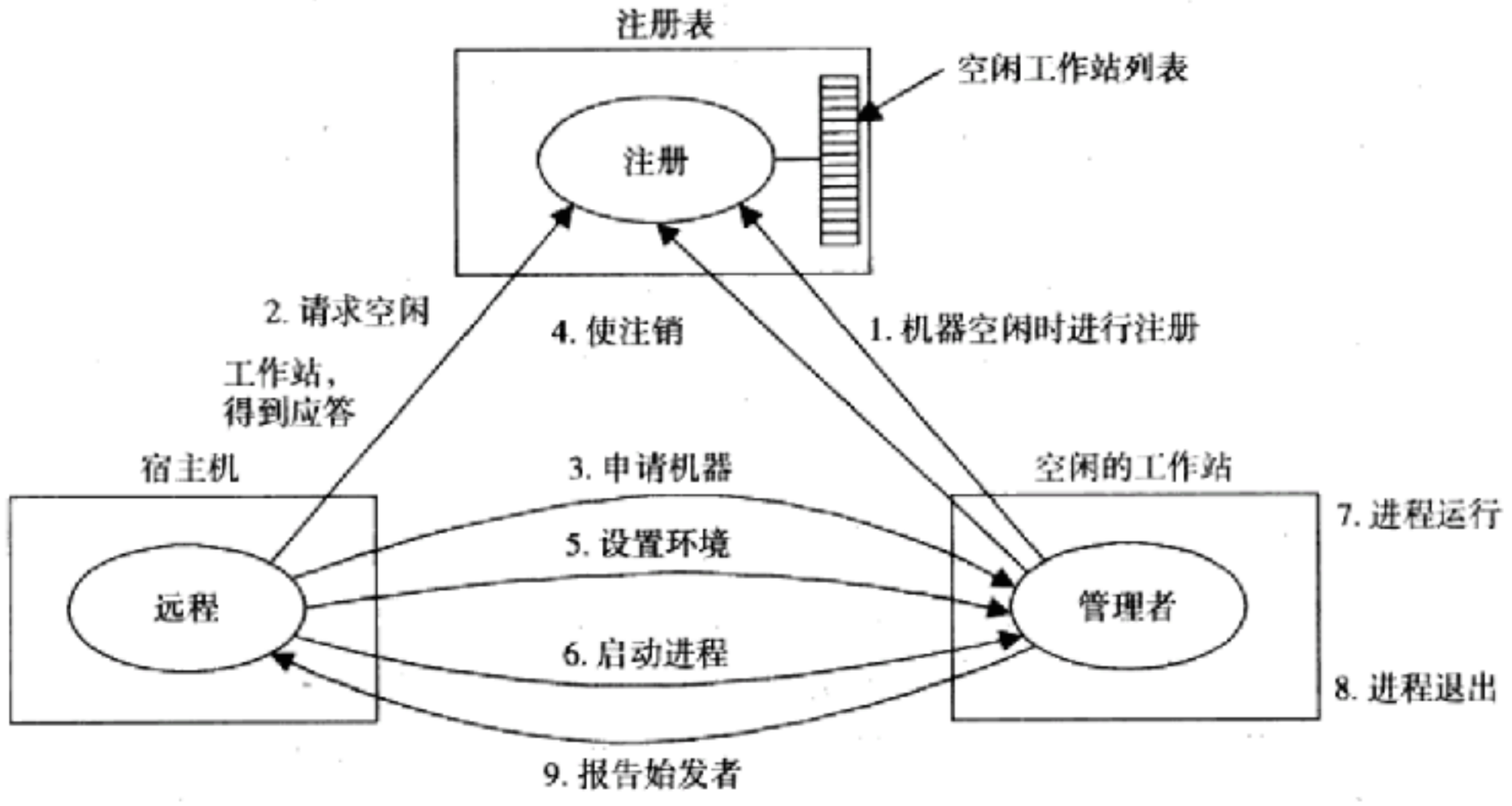


图 4.10　查找与使用空闲工作站的基于注册表算法

1. 分布式系统模型
1.2 使用空闲工作站

- 客户端驱动方法
  - 当调用remote时，广播发送请求，说明
    - 程序ID、内存需求、是否浮点运算等
- 空闲工作站返回应答
  - 如何挑选?
    - 使用延迟：使应答的延迟与工作站的负载成正比

1. 分布式系统模型
1.2 使用空闲工作站

- 怎样透明地运行一个远程进程?
  - 设置与本工作站一样的环境
    - 文件系统、环境变量、工作目录
  - 系统调用时的处理
    - 处理系统调用的错误
    - 处理只能在本机/远程主机运行的系统调用
      - 读键盘、写屏幕/调整数据段大小、程序计数器计数等
    - 处理时间有关的系统调用
    - 更多复杂的问题，如：写一个临时文件，在哪个主机上更高效?

1. 分布式系统模型
1.2 使用空闲工作站

- 如果有人注册进入、或者未激活的用户碰了鼠标或键盘，怎么办?
  - 方法1: 什么都不做
  - 方法2: 杀掉正在进入的进程
    - 缺点：工作信息丢失
    - 解决方法：给适当的警告
- 方法3: 将运行的进程迁移到另一台机器上
  - 很少使用，因为实现机制复杂，难点在于：
    - 如何寻找和收集相关的内核数据结构
    - 转移相应的子进程等

1. 分布式系统模型
1.3 处理机池模型

- 当提供相当于用户十倍甚至百倍数量的CPU时，怎么办?
  - 处理机池 （processor pool）

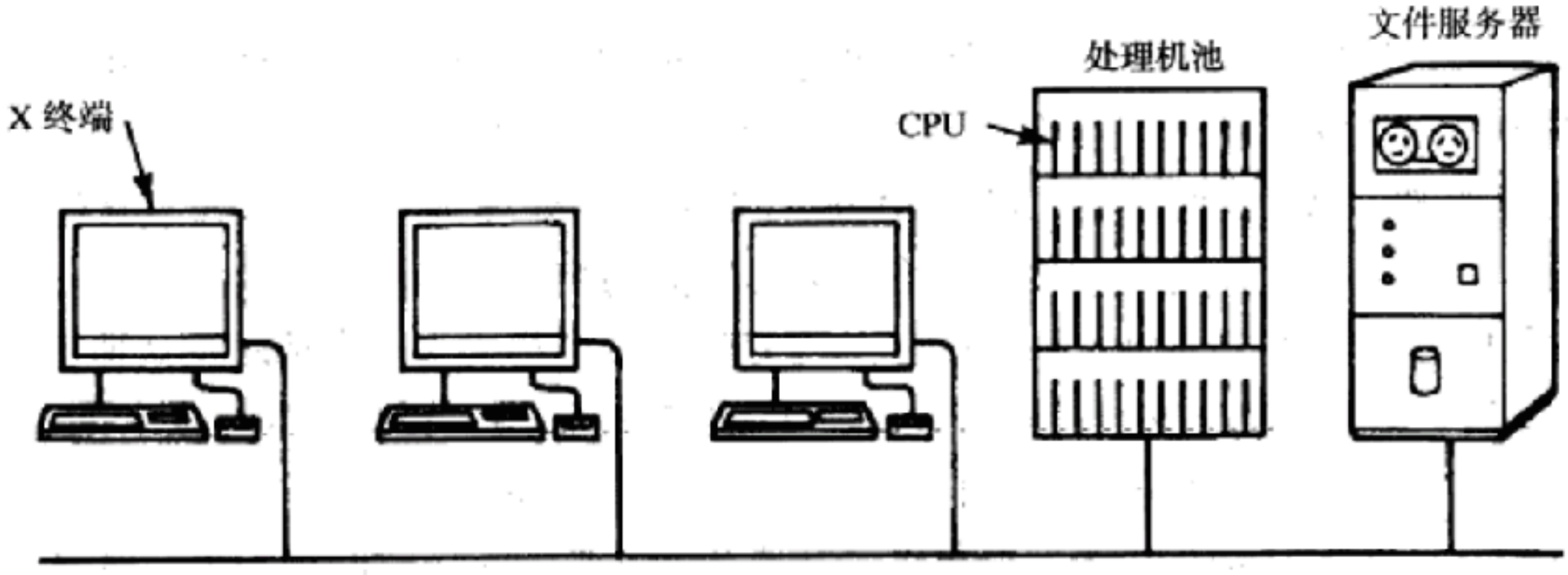


图 4.11　一个基于处理机池模型的系统

1. 分布式系统模型
1.3 处理机池模型

- 使用处理机池的好处
  - 减少用电量
  - X终端更廉价
  - 在给定资金的情况下提供更大的计算能力
  - 理论根据：排队论

1. 分布式系统模型

1.3 处理机池模型

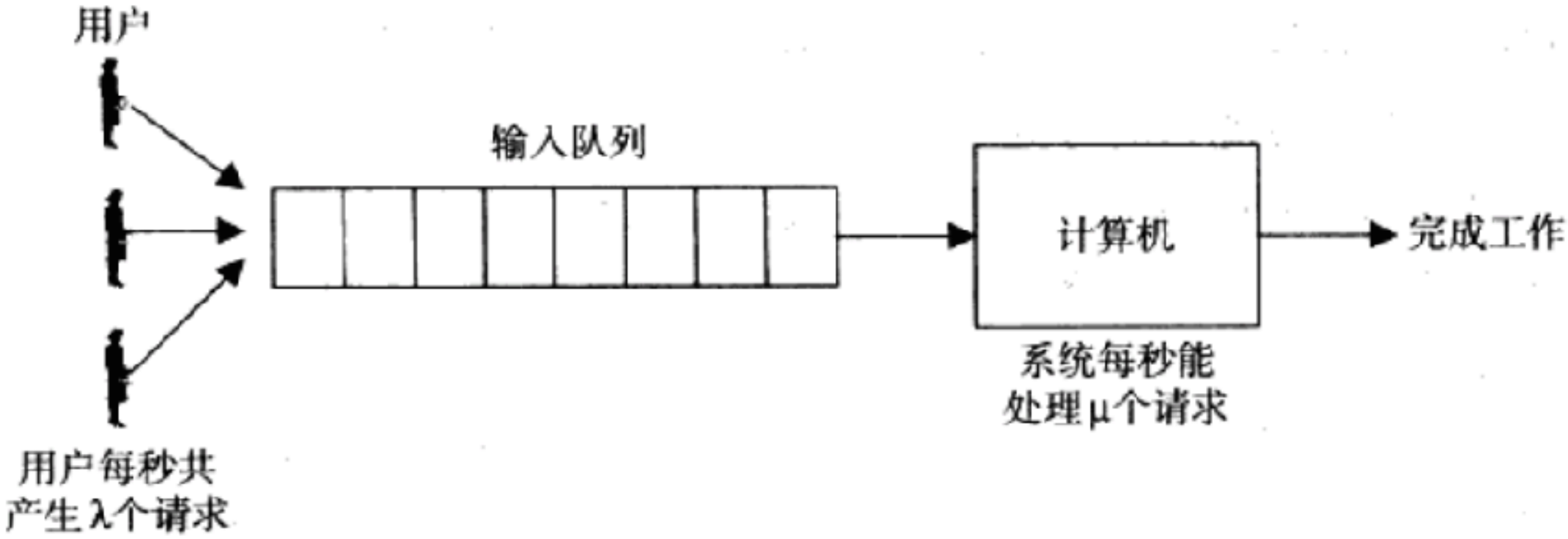


图 4.12　一个基本的排队系统

- 平均响应时间T与λ、μ的关系：**T=1/(*μ*-λ)**

1. 分布式系统模型
1.3 处理机池模型

- 例：一个服务器
- 每秒可以处理50个请求
- 但每秒接受40个请求
- 则平均响应时间T为1/（50-40）=0.1秒

1. 分布式系统模型
1.3 处理机池模型

- 假设有**n个**处理机组成的**独立**排队系统，每个处理机：
  - 请求到达速率为λ，
  - 处理机处理速率为μ，
  - 它的平均响应时间仍为**$T=1/(\mu-\lambda)$**
- 假设有**一个**排队系统
  - 请求到达速率为nλ，
  - 处理机处理速率为nμ，
  - 它的平均响应时间仍为**$T_1=1/n(\mu-\lambda)=T/n$**

反对分布式系统的论据之一…
也可避免浪费CPU周期

1. 分布式系统模型
1.3 处理机池模型

- 然而，平均响应时间**并不是**唯一指标
- 分布式计算的优点：
  - 价格低——不可能造过于巨大的机器
  - 对许多用户： 响应时间的稳定性>平均响应时间

1. 分布式系统模型
1.3 处理机池模型

- 总结：模型的选择取决于工作负载的本质
  - 适用于工作站模型的应用
    - 简单编辑
    - 电子邮件等
  - 适用于处理机池模型的应用
    - make
    - 大型模拟
    - 人工智能程序

1. 分布式系统模型

1.4 混合模型

- 为每个用户提供一个专用工作站并附加处理机池
  - 交互式工作在工作站上运行
  - 非交互式进程在处理池中运行
- 优点：
  - 快速交互响应
  - 有效资源利用
  - 设计简单

1. 分布式系统模型

1.5 体系结构

① 体系结构的样式

② 系统体系结构

1. 分布式系统模型
1.5.1 体系结构的样式

- 大型系统能否成功开发的关键：
  - 体系结构的设计和采用
- 表示法：体系结构样式 （Architecture Style）
  - **组件** (Component)
    - 一个模块单元，可以提供良好定义的接口
  - **链接器** (Connector)
    - 组件之间相互的连接方式
    - 组件之间的数据交换
  - 这些元素如何集成到一个系统中

1. 分布式系统模型
1.5.1 体系结构的样式

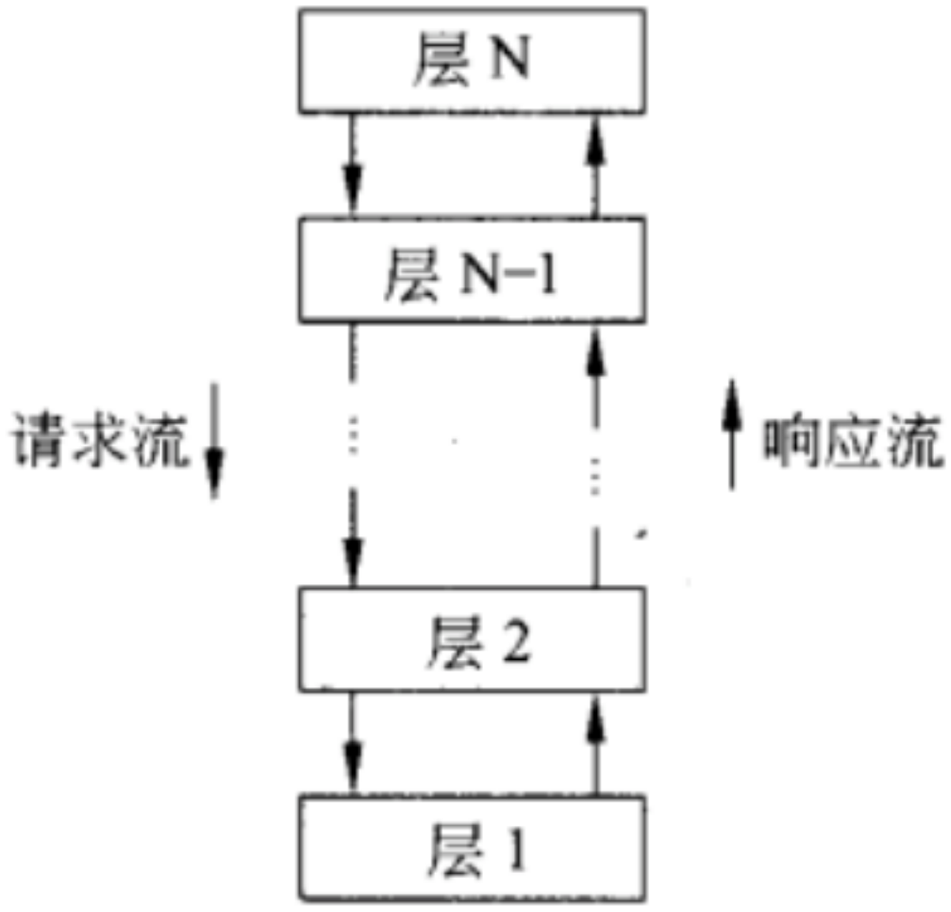
- 分布式体系结构的样式
  - 分层体系结构
  - 基于对象的体系结构
  - 以数据为中心的体系结构
  - 基于事件的体系结构

大型软件系统最重要的样式

1. 分布式系统模型

1.5.1 体系结构的样式

- 分层体系结构

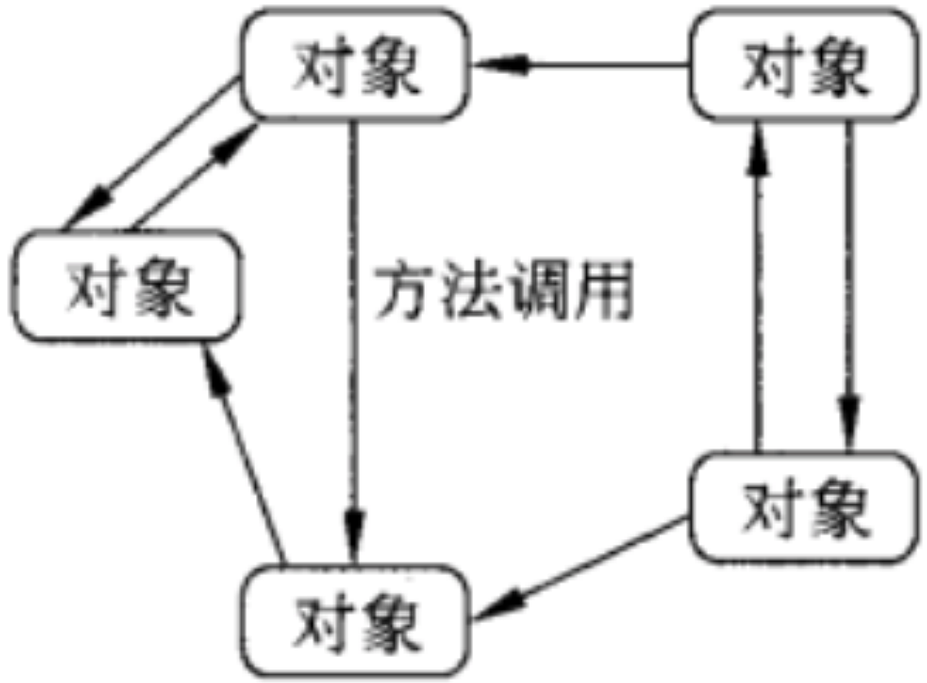


被网络通信广泛采用

1. 分布式系统模型

1.5.1 体系结构的样式

- 基于对象的体系结构

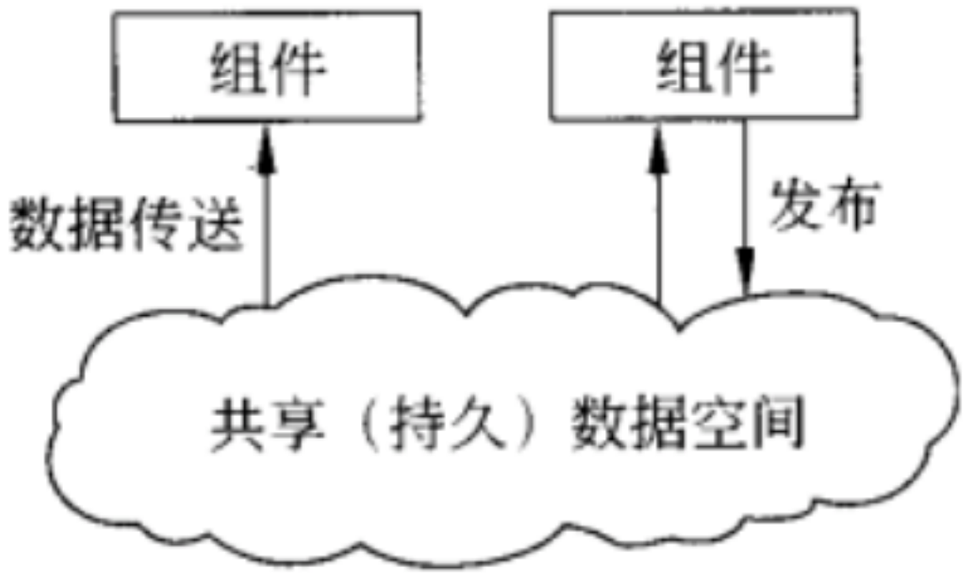


通过远程调用机制来连接
如：客户-服务器系统体系结构

1. 分布式系统模型

1.5.1 体系结构的样式

- 以数据为中心的体系结构

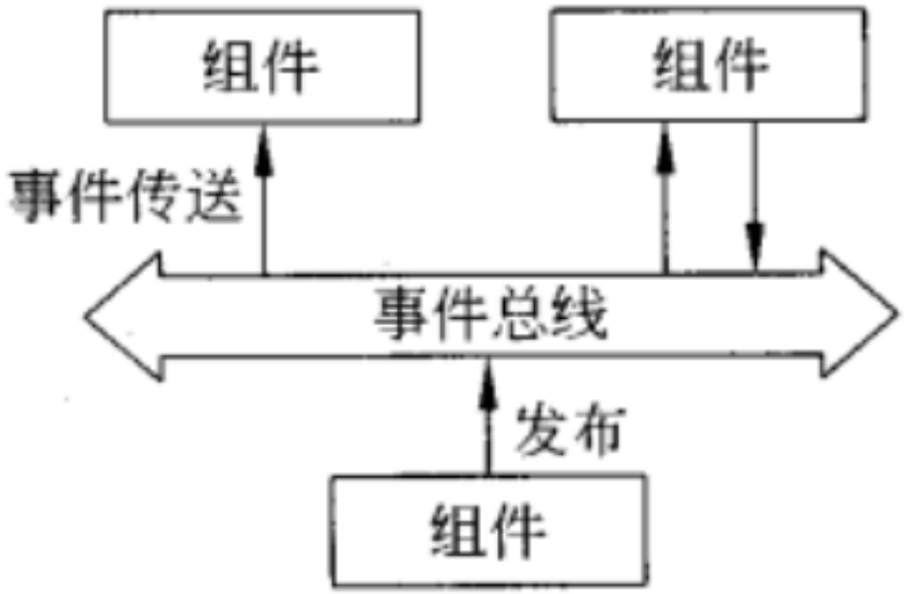


如：基于Web的分布式系统

1. 分布式系统模型

1.5.1 体系结构的样式

- 基于事件的体系结构
  - 通过事件的传播来通信



如：发布 / 订阅系统

1. 分布式系统模型
1.5.2 系统体系结构

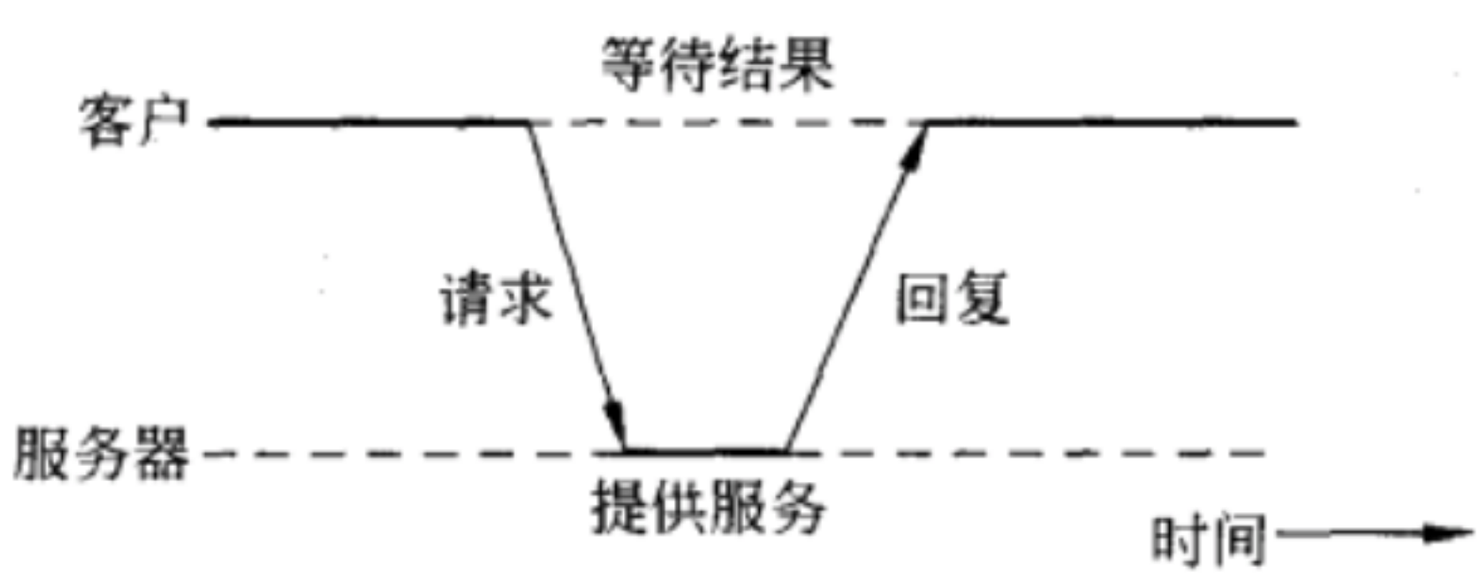
- 系统体系结构
- 组件
- 组件的交互
- 组件的**位置**

1. 分布式系统模型
1.5.2 系统体系结构

- 系统体系结构
  - 集中式体系结构
  - 非集中式体系结构
  - 混合体系结构

1. 分布式系统模型
1.5.2.1 集中式体系结构

- 基本的客户-服务器模型



1. 分布式系统模型
1.5.2.1 集中式体系结构

- 应用分层
  - 用户接口层
  - 处理层
  - 数据层
- 例:
  - Internet搜索引擎
  - 金融决策支持系统

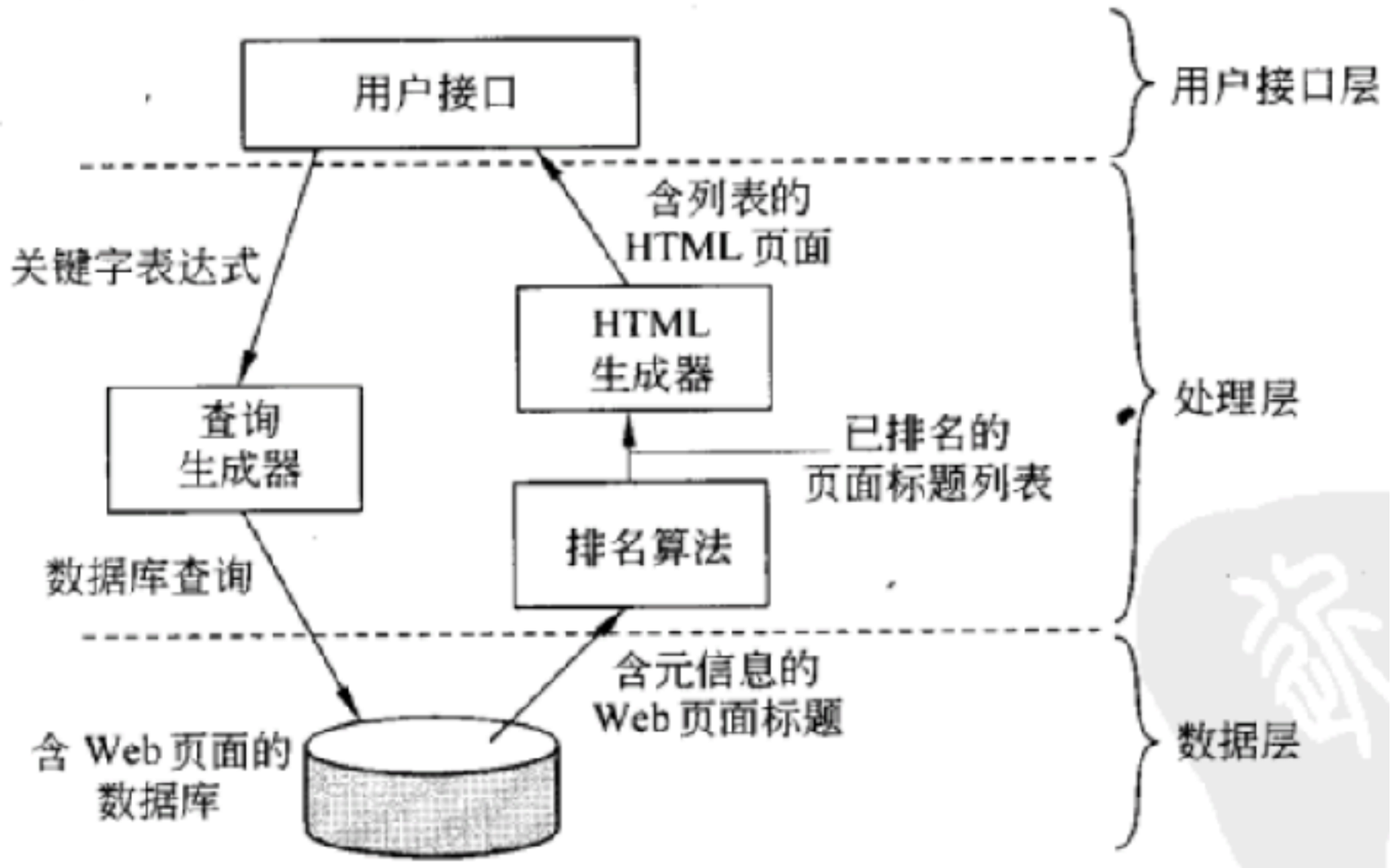


图 2.4　因特网搜索引擎简化成三个不同的层

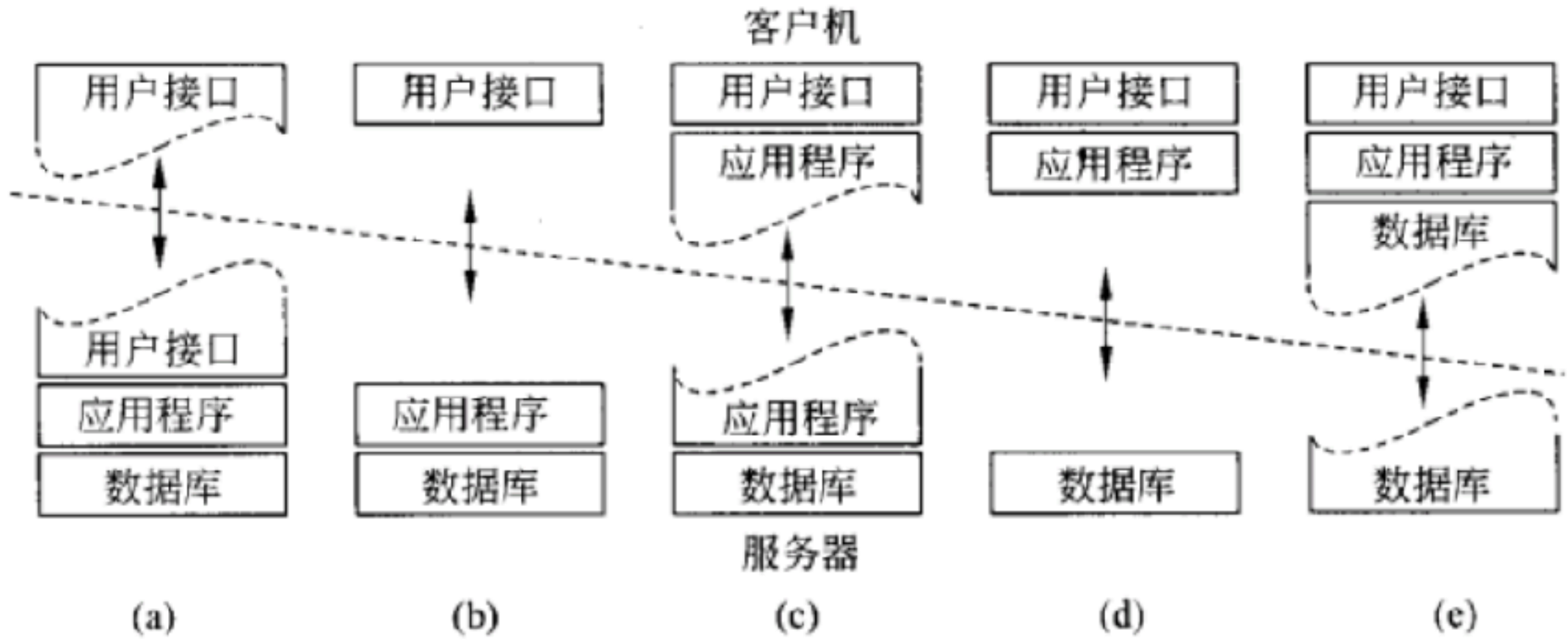
1. 分布式系统模型
1.5.2.1 集中式体系结构

- 多层体系结构
  - 两种类型的机器：
    - 客户机
    - 服务器
- 然后，将程序分布在不同的机器中

1. 分布式系统模型

1.5.2.1 集中式体系结构

- 多层体系结构



文字处理软件

银行应用程序

Web浏览

1. 分布式系统模型

1.5.2.2 非集中式体系结构

- 客户-服务器组成的体系结构——**垂直分布性**
- **水平分布性**
  - 客户或服务器可能在物理上被分割成**逻辑上相等**的几个部分
  - 但每个部分都操作在整个数据集中自己共享的部分，从而实现负载的平衡
  - 如：点对点系统 (peer-peer system)
    - 结构化、非结构化

1. 分布式系统模型
1.5.2.2 非集中式体系结构

- 结构化的点对点体系结构
    - 覆盖网络是用一个确定性的过程来构成的，如
        - 分布式哈希表 (distributed hash table, DHT)
            - 作用
                - 在不需要服务器的情况下，每个客户端负责一个小范围的路由，并负责存储一小部分数据，从而实现整个DHT网络的寻址和存储
            - 实现
                - 从一个大的标识符空间中选取一个**随机关键值**赋给**数据项**
                - 从这个标识符空间中选取一个**随机数**赋给该系统中的**节点**
                - 当**查找该数据项**时，返回对应**节点**的**网络地址**

1. 分布式系统模型
1.5.2.2 非集中式体系结构

- Chord系统
  - 主流DHT协议之一
  - 问题：如何在P2P网络中找到存有特定数据的节点?
  - 节点按逻辑组成一个环
  - 关键值k的数据项映射到最新标识符id>=k的节点

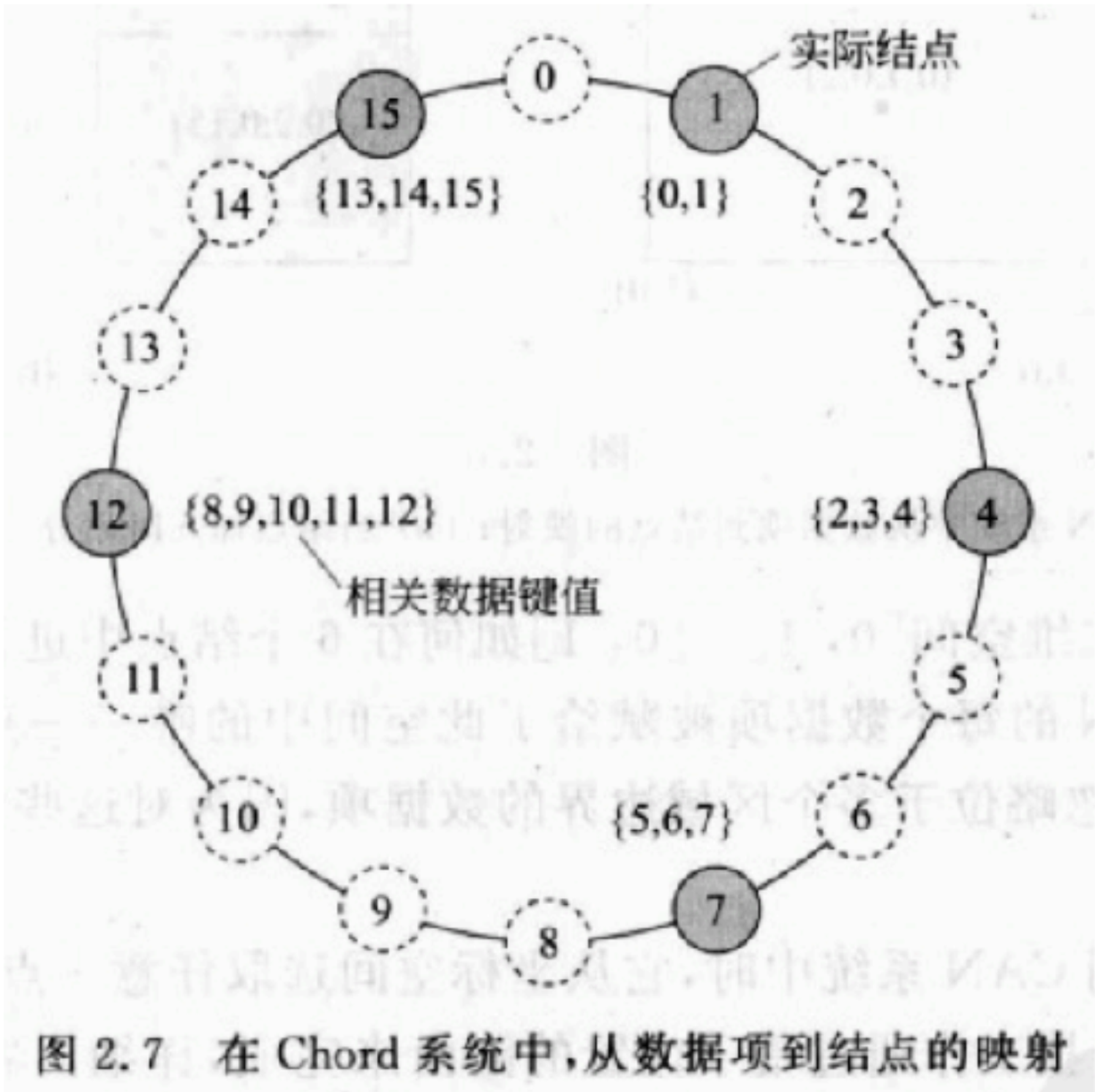
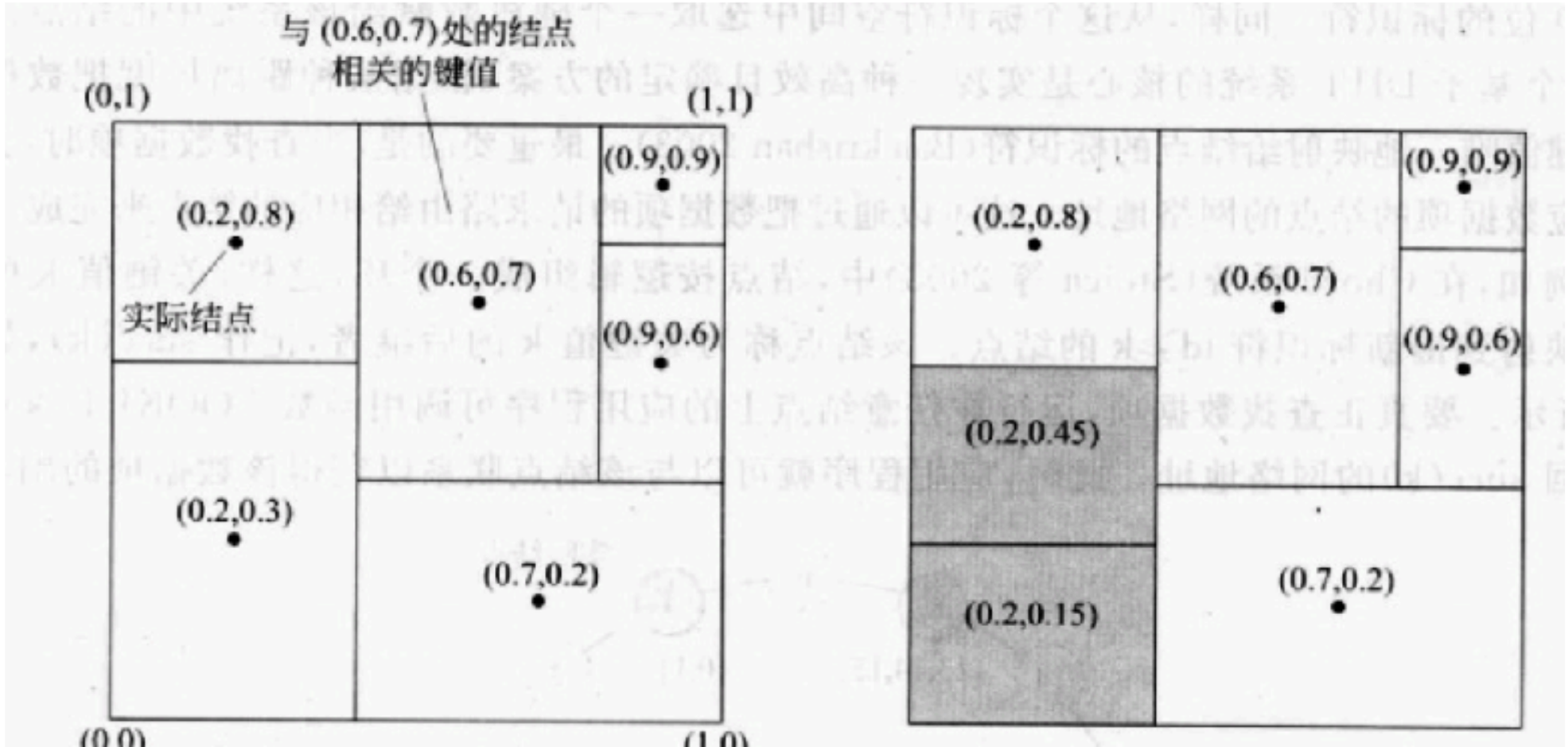


图 2.7　在 Chord 系统中，从数据项到结点的映射

1. 分布式系统模型
1.5.2.2 非集中式体系结构

- CAN系统
  - 另一种DHT系统，上下文可编址网络，(Content Addressable Network)
  - 当节点P要加入到CAN系统中时，P从坐标空间中任取一点，然后在它所位于的区域中查找Q；将区域分为两等份，给P和Q



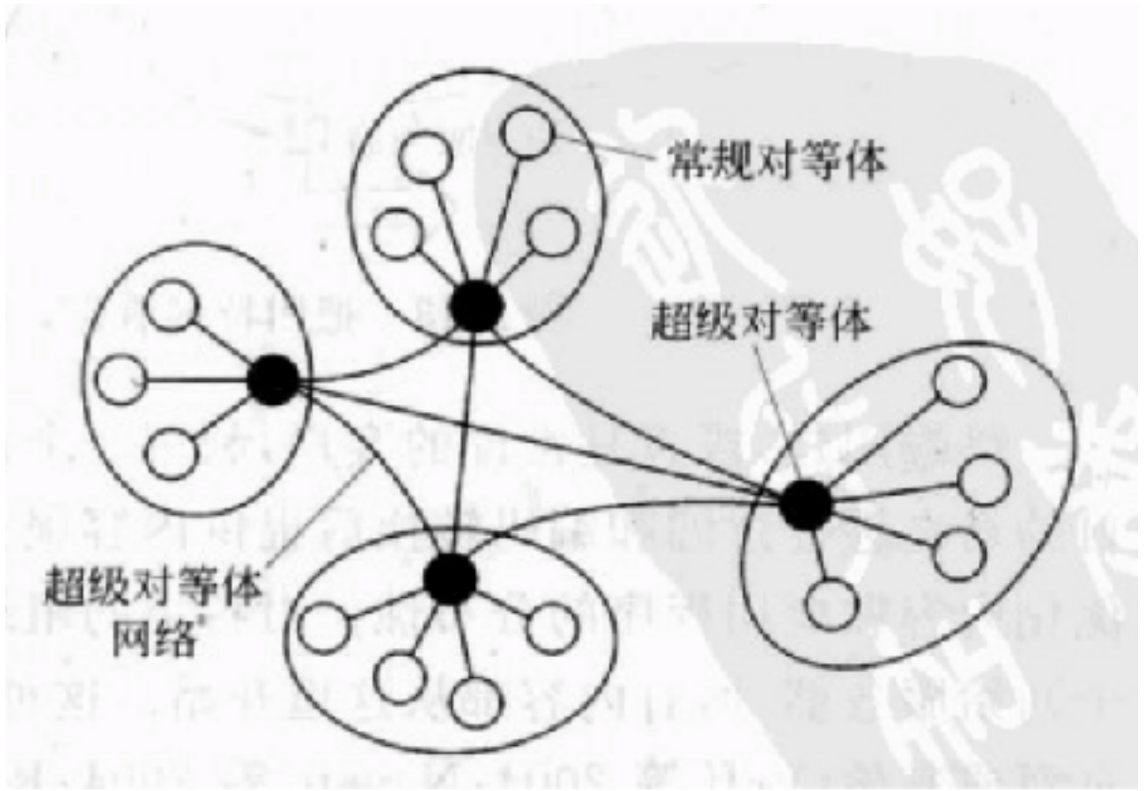
1. 分布式系统模型

1.5.2.2 非集中式体系结构

- 非结构化的点对点体系结构
  - 主要依靠随机化的算法来构造覆盖网络
  - 主要思想：
    - 每个节点维护一个邻节点列表
    - 但这个**列表**用**随机**的方法来构造
    - 同时，数据项随机分布于网络节点中
- 因此，节点通过泛洪该网络来查询数据项

1. 分布式系统模型
1.5.2.2 非集中式体系结构

- 超级对等体
  - 在非结构化点对点系统中，随着网络增大，相关数据项的点位变得更加困难
  - 解决方案：放弃点对点系统的对称性，如
    - **上下文传送网络**（content delivery network, CDN)
      - 节点提供缓存空间，让Web用户快速访问
      - 而**代理程序**（Broker）可以快速查询这些节点， 而这种节点通常称为**超级对等体**（superpeer）



1. 分布式系统模型
1.5.2.3 混合体系结构

- 边界服务器系统 (edge-server system)

  - Internet服务提供商 (Internet Service Provider, ISP)

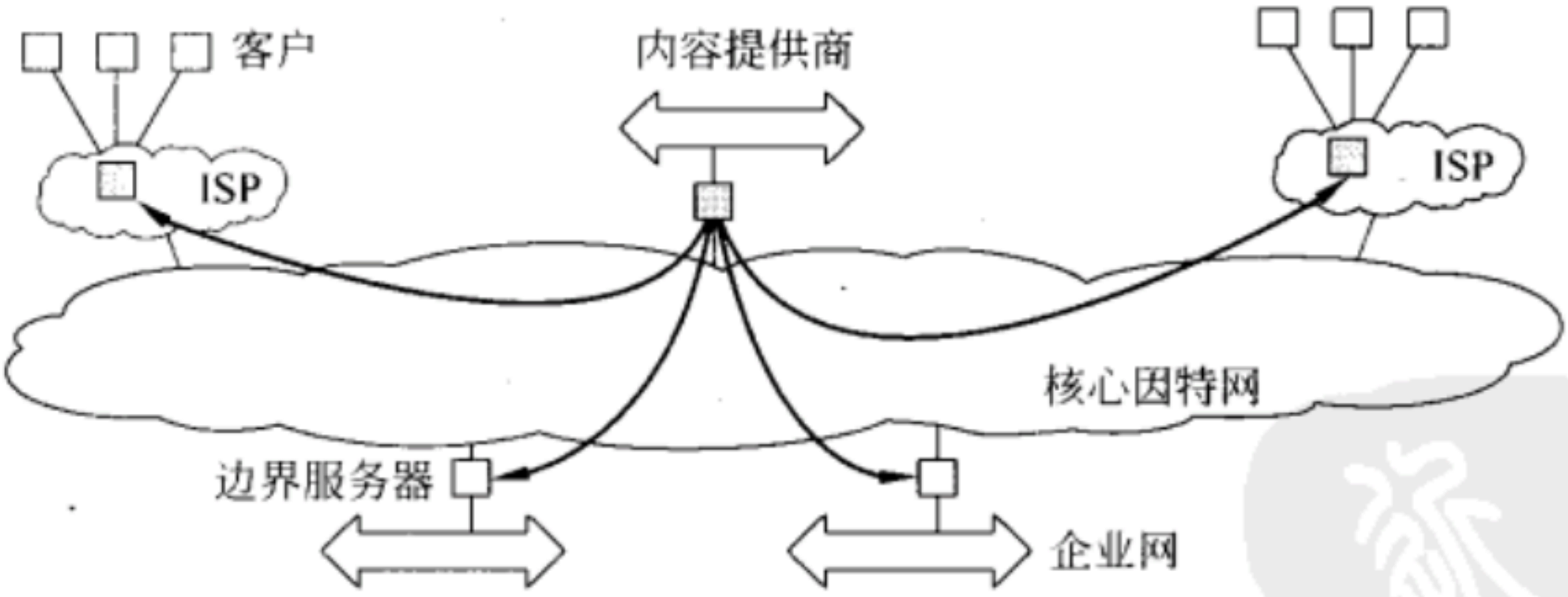
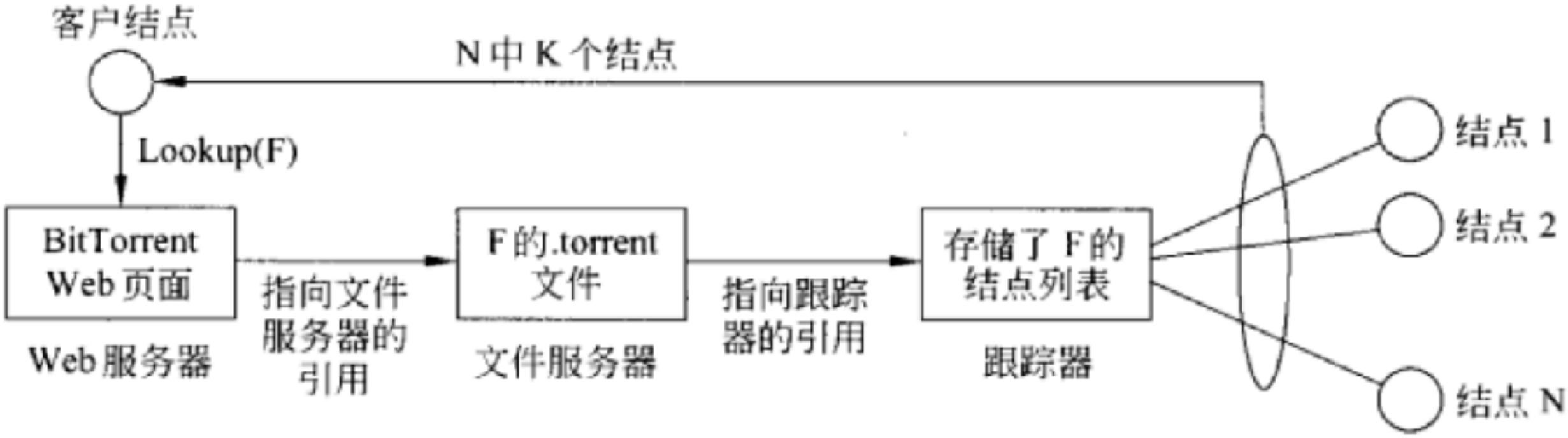


图 2.13　把因特网看作是由一系列边界服务器组成的

1. 分布式系统模型
1.5.2.3 混合体系结构

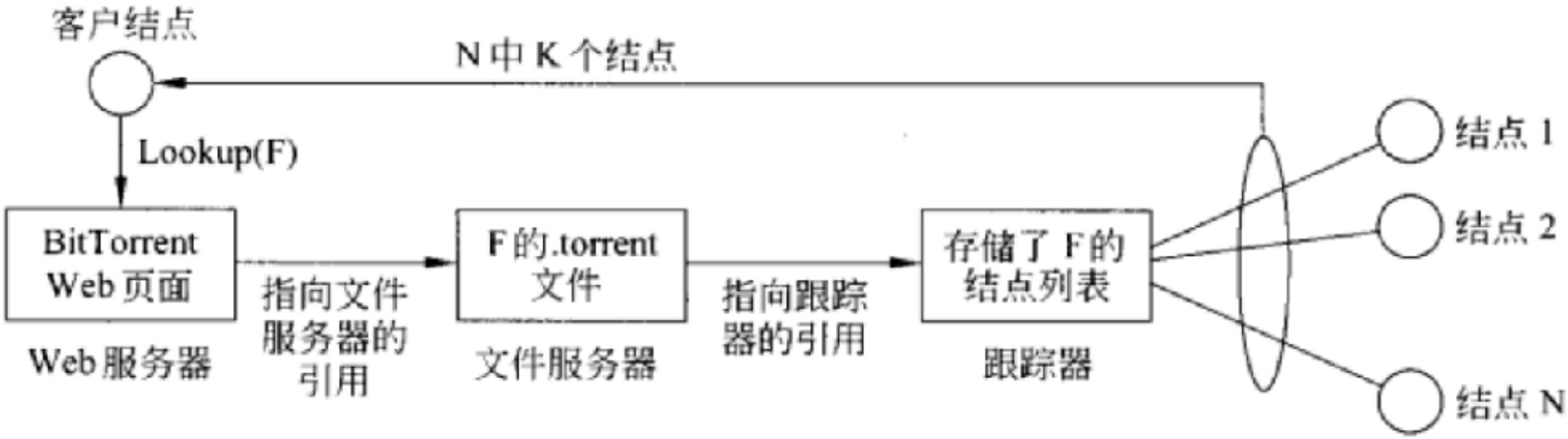
- 协作分布式系统
  - BitTorrent (2003)
    - 基本思想：当一个终端用户要查找某个文件时，他可以从其他用户哪里下载文件块，直到所下载的文件块能够组装成完整的文件为止

Bram Cohen



1. 分布式系统模型
1.5.2.3 混合体系结构

- BitTorrent的跟踪器（tracker）
  - 维护活动节点的记录
  - 通常，每个文件（或文件集）只有一个跟踪器
  - 特性：节点被强制为其他节点提供帮助！
  - 结果表明，系统的瓶颈由跟踪器导致



1. 分布式系统模型
1.5.2.3 混合体系结构

- 协作分布式系统
  - Globule协作内容分布式网络 (CDN) 2006
    - 终端用户（和组织结构）自愿提供增强的Web服务器
    - 每个服务器有如下组件
      - （1）可以吧客户的请求重定向到其他服务器的组件
      - （2）用于分析访问模式的组件
      - （3）用于管理Web页面复制的组件

1. 分布式系统模型
1.5.2.3 混合体系结构

- 协作分布式系统
  - Globule协作内容分布式网络 (一种CDN实现) 2006
    - 非集中式的特征
      - 初始服务器(origin server)与其他服务器的协作
    - 集中式的特征
      - 代理程序(Broker)负责注册服务器