解决这些主题的子系统本身就是有趣的系统，也是管理复杂性的案例研究。通常，这些子系统在内部构建为客户/服务系统，以递归方式应用本章的概念。接下来的两节提供了两个现实世界客户/服务系统的案例研究，并说明了后续章节中讨论的主题的必要性。

4.4 案例研究：互联网域名系统 (DNS)

互联网域名系统 (DNS) 是客户端/服务应用程序和命名方案成功实施的绝佳案例研究，在本例中用于命名互联网计算机和服务。尽管 DNS 是为该特定应用程序设计的，但它实际上是一个通用的名称管理和名称解析系统，它以层次结构将名称管理权分配给不同的命名机构，并以层次结构将名称解析工作分配给不同的名称服务器。它的设计使其能够快速响应名称解析请求，并可扩展到存储记录数量和请求数量非常大。它还具有很强的弹性，在面对多种网络和服务器故障时，它能够提供持续、准确的响应。

DNS 的主要用途是将用户友好的字符串名称（称为域名）与面向机器的网络连接点二进制标识符（称为 Internet 地址）关联起来。域名是分层结构的，术语“域”在 DNS 中以一般方式使用：它只是一组具有相同分层祖先的一个或多个名称。这种约定意味着分层区域可以是域，但它也意味着您桌上的个人计算机是一个只有一个成员的域。因此，尽管“域名”一词表示分层区域的名称，但 DNS 解析的每个名称都称为域名，无论它是分层区域的名称还是单个连接点的名称。由于域通常对应于管理组织，因此它们也是名称分配的委托单位，使用第 3.1.4 节中描述的分层命名方案。

就我们的目的而言，DNS 的基本接口非常简单：

```
$value <- DNS_RESOLVE (domain_name) $
```

该接口省略了 2.2.1 节命名模型的标准名称解析接口中的上下文参数，因为只有一个用于解析所有 Internet 域名的单一、通用、默认上下文，并且对该上下文的引用内置于 _as 配置参数中。

在通常的 DNS 实现中，绑定不是通过调用和过程（如我们的命名模型所建议的那样）来完成的，而是通过使用文本编辑器或数据库生成器来创建和管理绑定表。然后，这些表会通过某种幕后方法加载到 DNS 服务器中，只要它们的管理员认为有必要。这种设计的一个后果是，对 DNS 的更改

绑定通常不会在您提出请求后的几秒钟内发生；相反，它们通常需要几个小时。

域名是路径名，其组成部分由句点（称为点，尤其是在大声朗读域名时）分隔，并且最不重要的组成部分放在最前面。三种典型的域名是

ginger.cse.pedantic.edu　　ginger.scholarly.edu　　ginger.com

DNS 允许使用相对和绝对路径名。绝对路径名应该通过尾随点来区分。在人机界面中，尾随点很少出现；相反，_应用一种简单的多重查找形式。当呈现相对路径名时，_首先尝试附加由本地设置的配置参数提供的默认上下文。如果生成的扩展名称无法解析，_将再次尝试，这次只在最初呈现的名称后附加一个尾随点。因此，例如，如果呈现

DNS _具有明显相对路径名“ginger.com”，且默认上下文为“pedantic.edu.”_将首先尝试解析绝对路径名“ginger.com.pedantic.edu.”。如果该尝试导致结果，它将尝试解析绝对路径名“ginger.com”。

4.4.1 DNS 中的名称解析

DNS 名称解析至少可以按以下三种方式设计：

1. 电话簿模型：向每个网络用户提供一份文件副本，其中包含每个域名及其相关互联网地址的列表。此方案存在一个严重的问题：要覆盖整个互联网，文件的大小将与网络用户数量成正比，而更新文件需要向每个用户提供一份新副本。由于更新频率往往与文件中列出的域名数量成正比，因此保持更新所需的网络流量将随着域名数量的立方而增长。此方案在互联网中使用了近 20 年，后来被发现存在不足，并于 20 世纪 80 年代末被 DNS 取代。

2. 中央目录服务模型：将文件放在网络中某台连接良好的服务器上，并提供协议要求其解析名称。此方案可使更新变得简单，但随着用户数量的增长，其设计者必须采用越来越复杂的策略，以防止其成为性能瓶颈和潜在的大规模故障源。还有一个问题：控制中央服务器的人默认负责所有名称分配。这种设计不适合域名分配中的责任委托。

3. 分布式目录服务模型。其理念是拥有许多服务器，
每个子集负责解析域名的某个子集，还有一个协议来查找可以解析任何特定名称的服务器。正如我们将在以下描述中看到的那样，该模型可以提供委托和响应

以在保持可靠性和性能的同时扩大规模。出于这些原因，DNS 采用了这种模型。

在分布式目录服务模型中，每个名称服务器的运行方式相同：服务器维护一组名称记录，每条记录将一个域名绑定到一个互联网地址。当客户端发送名称解析请求时，名称服务器会查找其负责的域名集合，如果找到一条名称记录，则会返回该记录作为响应。如果未找到请求的名称，则会查找另一组引荐记录。每条引荐记录将 DNS 名称空间的某个层级区域绑定到其他名称服务器，以便帮助解析该层级区域中的名称。服务器从请求域名的最重要组成部分开始，在引荐记录中搜索与最多组成部分匹配的记录，并返回该引荐记录。如果没有匹配项，DNS 无法解析原始名称，因此会返回“无此域”的响应。

DNS 的引用架构虽然在概念上很简单，但有许多细节可以增强其性能、可扩展性和稳健性。我们先从一个简单的操作示例开始，然后再添加一些增强功能。图 4.10 中的虚线说明了当左下角名为 ginger.cse.pedantic.edu 的客户端计算机尝试解析域名 ginger.Scholarly.edu 时 DNS 的操作。第一步（如请求 #1 所示）是 _将该域名发送到根名称服务器，它以某种方式知道该服务器的 Internet 地址。第 4.4.4 节解释了 _如何发现该地址。

根名称服务器将请求中的名称与其所知道的域名子集进行匹配，从所请求域名的最重要部分开始（在本例中为 edu）。在本例中，根名称服务器发现它具有域 edu 的引荐记录，因此它以引荐做出响应，在本例中表示：“有一个名为 edu 的域的名称服务器。该名称服务器的名称记录将名称 names.edu. 绑定到 Internet 地址 192.14.71.191。”此响应说明，名称服务器与任何其他服务器一样，既有域名，也有 Internet 地址。通常，名称服务器的域名会提供一些有关它所服务的域名的线索，但没有必要的对应关系。使用完整的名称记录进行响应提供的信息比客户端实际需要的要多（客户端通常不关心名称服务器的名称），但它允许名称服务器的所有响应保持一致。由于名称服务器的域名并不重要，为了减少图 4.10 中的混乱，该图在所示的响应中省略了它。

当客户端收到此响应时，它会立即重新发送相同的名称解析请求，但这次它会将请求（图中的请求 2）定向到位于响应 1 中提到的 Internet 地址的名称服务器。该名称服务器将请求的路径名称与其所知道的域名集进行匹配，同样从最重要的部分开始。在本例中，它在引荐记录中找到了名称 Scholarly.edu. 的匹配项。因此，它会发回一个响应，说“有一个名为 Scholarly.edu 的域的名称服务器。

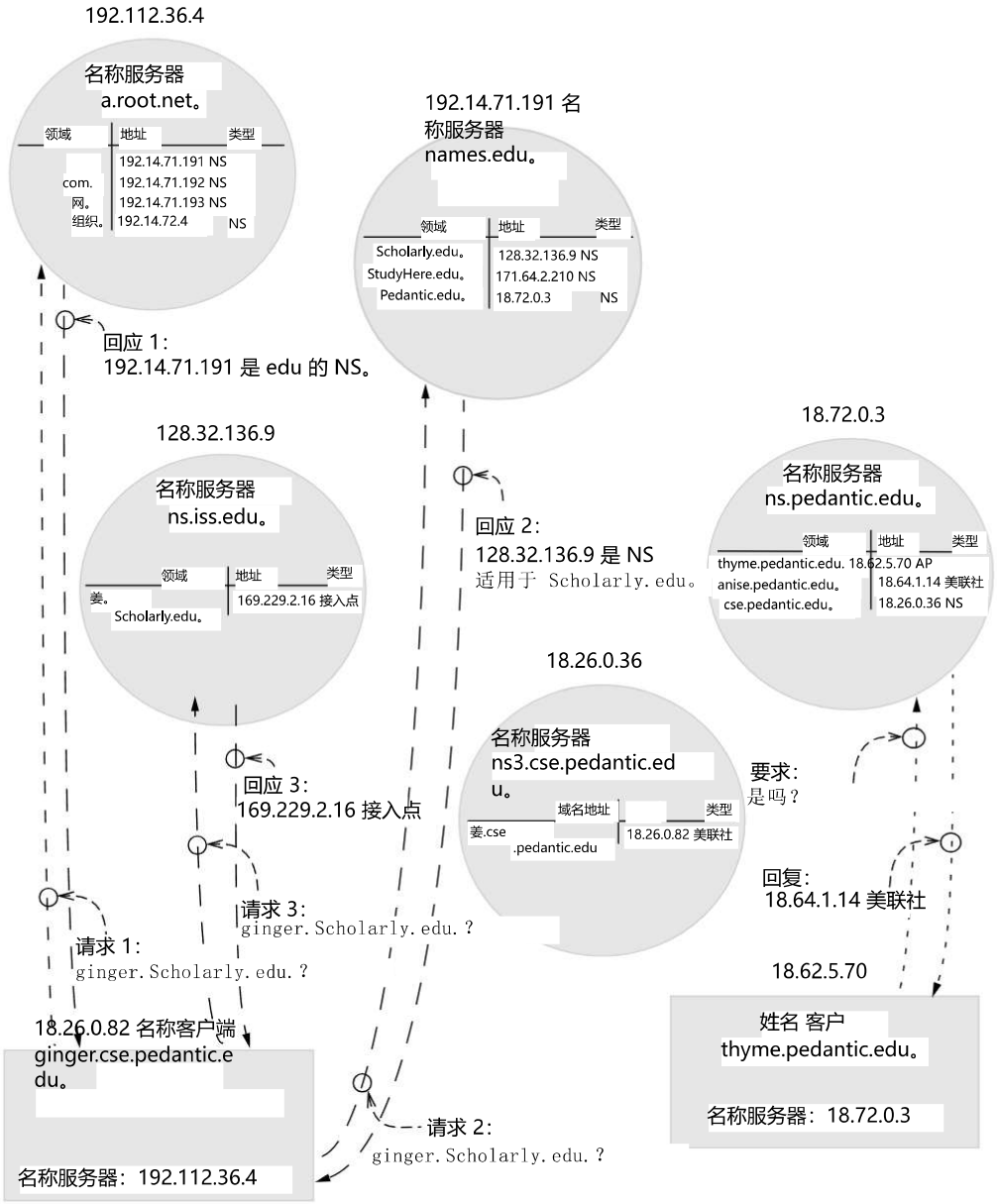

图 4.10

互联网域名系统的结构和操作。在该图中，每个圆圈代表一个名称服务器，每个矩形代表一个名称客户端。表或响应中的 NS 类型表示这是对另一个名称服务器的引用，而表或响应中的 AP 类型表示这是一个互联网地址。虚线显示了左下角的名称客户端发出的三个请求的路径，这些请求用于解析名称 ginger.Scholarly.edu，从根名称服务器开始。点线显示了右下角的名称客户端发出的请求的解析，这些请求用于解析 anise.pedantic.edu，从本地名称服务器开始。

该名称服务器的名称记录将名称 ns.iss.edu. 绑定到互联网地址 128.32.136.9。” 该图再次省略了名称服务器的域名。

对原始路径名的每个部分重复此序列，直到

DNS 最终到达一个拥有 ginger.Scholarly.edu 域名记录的域名服务器。该域名服务器返回一条响应：“ginger.Scholarly.edu 的域名记录将该域名绑定到互联网地址 169.229.2.16。” 这是对原始查询的答复，它将结果返回给调用者，后者可以继续与目标服务器发起消息交换。

持有域名名称记录或引荐记录的服务器被称为该域名的权威域名服务器。在我们的示例中，ns3.cse.pedantic.edu 是 ginger.cse 的权威域名服务器。

pedantic.edu。 域，以及所有其他以 cse 结尾的域名。

pedantic.edu. 和 ns.iss.edu. 是 Scholarly.edu. 域名的权威域名。

由于名称服务器不保存其自身名称的名称记录，因此名称服务器不能成为其自身名称的权威名称服务器。例如，根名称服务器对域名 edu. 具有权威性，而 names.edu.

名称服务器对所有以 edu 结尾的域名具有权威性。

这就是 DNS 运行的基本模型。下面对其运行进行一些详细说明，每个详细说明都有助于使系统响应速度快、健壮，并能够大规模扩展。

1. 实际上没有必要将初始请求发送到根名称服务器。

   DNS 可以将请求发送到任何它知道其互联网地址的方便的域名服务器。域名服务器并不关心请求来自何处；它只需将请求的域名与其负责的域名列表进行比较，以查看是否持有可用的记录。如果持有，它会响应请求。如果没有，它会通过返回指向根域名服务器的链接来响应。能够将任何请求发送到本地域名服务器意味着，客户端、域名服务器和目标域名三者都位于同一域（例如 pedantic.edu）的常见情况可以通过单个请求/响应交互快速处理。（图 4.10 右下角的虚线显示了一个示例，其中 thyme.pedantic.edu. 向 pedantic.edu 域名的域名服务器请求 anise.pedantic.edu 的地址。）此功能还简化了域名发现，因为客户端只需要知道任何附近域名服务器的互联网地址。第一次向附近的服务器发出远程名称请求（在当前示例中为 ginger.scholarly.edu）将返回对根名称服务器的互联网地址的引用。

2. 一些域名服务器提供所谓的递归名称服务（可能具有误导性）。如果名称服务器没有所请求名称的记录，则名称服务器将负责解析名称本身，而不是发送引用响应。它将初始请求转发到根名称服务器，然后继续遵循响应链来解析完整路径名称，最后将所需的名称记录返回给其客户端。通过

就其本身而言，该特性似乎仅仅简化了客户端的生活，但与下一个特性结合，它提供了主要的性 能增强。

3. 除了其权威记录外，每个域名服务器都应维护一个缓存，其中包含它从其他域名服务器获取的所有域名记录。因此，提供递归域名服务的服务器会收集记录，从而大大加快未来的域名解析请求。例如，如果 cse.pedantic.edu 的域名服务器提供递归服务，并被要求解析 flower.cs.scholarly.edu 域名，则在此过程中（假设它不会再次请求递归服务），其缓存可能会获取以下记录：

   教育　　请参阅 names.edu，网址为 198.41.0.4
   Scholarly.edu 指的是 ns.iss.edu，地址为 128.32.25.19 cs.Scholarly.edu 指的是 cs.Scholarly.edu，地址为 128.32.247.24 flower.cs.Scholarly.edu 互联网地址为 128.32.247.29

   现在，当该名称服务器收到例如解析名称 psych.Scholarly.edu 的请求时，它将在缓存中发现域 Scholarly.edu 的记录，并且能够通过将初始请求直接转发到相应的名称服务器来快速解析该名称。

   缓存中保存着一份副本，如果有人更改权威名称记录，该副本可能会过期。由于域名系统中现有名称绑定的更改相对较少，而且很难跟踪域名记录可能传播到的所有缓存，因此 DNS 设计不要求显式使更改的条目失效。相反，它使用过期时间。也就是说，DNS 记录的命名机构会为其发出的每条记录标记一个过期时间，该时间范围可能从几秒到几个月不等。DNS 缓存管理器应丢弃已超过过期时间的条目。DNS 缓存管理器提供了一种称为最终一致性的内存模型，该主题在第 10 章 [在线] 中讨论。

4.4.2 名称分级管理

域名形成一个层次结构，上述名称服务器的排列与该层次结构相匹配，从而分散了名称解析的工作。同一层次结构还通过分散操作名称服务器的责任来分散管理名称分配的工作。分散责任是分布式目录服务模型的主要优点之一。

其工作原理其实很简单：任何运营域名服务器的人都可以成为域名权威机构，这意味着他或她可以向该域名服务器添加权威记录。因此，在互联网发展初期的某个时候，某位 Pedantic 大学的网络管理员为域名 pedantic.edu 部署了一个域名服务器，并说服 edu 域名的管理员为该域名 pedantic.edu 安装一个绑定，将该域名与互联网关联起来。

在 pedantic.edu 域名服务器互联网地址演变的早期阶段。现在，如果 Pedantic 大学想要添加一条记录，例如，将其希望命名为 archimedes.pedantic.edu 的互联网地址，其管理员无需征求任何其他人的许可即可执行此操作。解析 archimedes.pedantic.edu 域名的请求可以到达互联网上的任何域名服务器；该请求最终将到达 pedantic.edu 域名的域名服务器，并在那里得到正确的响应。同样，学术研究所的网络管理员可以自行为名为 archimedes.Scholarly.edu 的互联网地址安装一条名称记录。尽管两所机构都为其其中一台计算机选择了 archimedes 这个名称，但由于两个域名的路径名不同，因此它们的管理员无需协调各自的名称分配。换句话说，它们的命名机构可以独立运作。

继续这种分散化方法，任何管理名称服务器的组织都可以创建较低级别的命名域。例如，Pedantic 大学的计算机科学与工程系可能拥有如此多的计算机，以至于该部门可以方便地自行管理这些计算机的名称。该部门只需为较低级别的域（例如名为 cse.pedantic.edu）部署一个名称服务器，并说服 pedantic.edu 域的管理员在其名称服务器中安装该名称的引荐记录。

4.4.3 DNS 的其他功能

为了确保名称服务的高可用性，DNS 规范要求每个运行名称服务的组织安排至少两个相同的副本服务器。此规范非常重要，尤其是在域名层次结构的较高级别，因为大多数 Internet 活动都使用域名，无法解析名称组件会阻止访问该名称组件下的所有站点。许多组织都有其名称服务器的三到四个副本，截至 2008 年，根名称服务器的副本约为 80 个。理想情况下，副本应连接到相距较远的网络，以便对本地网络和电力中断提供一些保护。同样，在命名层次结构的较高级别，独立连接的重要性会增加。因此，根名称服务器的 80 个副本分散在世界各地，但典型组织的名称服务器的三到四个副本更可能位于该组织的校园内。这种安排确保即使校园与外界断开连接，组织内部的名称通信仍可正常工作。另一方面，在这种断网期间，组织外部的通信者甚至无法验证名称是否存在，例如，无法验证电子邮件地址。因此，更好的安排可能是将组织的多个副本名称服务器中的至少一个附加到互联网的另一部分。

由于名称服务器需要复制，许多网络服务也需要复制，因此 DNS 允许将同一个名称绑定到多个 Internet 地址。因此，_ 的返回值可以是（假定）等效 Internet 地址的列表。客户端可以选择哪个

猜测到附着点的距离，或任何其他可能可用的标准。

DNS 的设计使名称服务非常强大。原则上，DNS 服务器的工作非常简单：接受请求数据包、搜索表并发送响应数据包。其接口规范不要求它维护任何连接状态或任何其他持久、可变的状态；其唯一的公共接口是幂等的。结果是，一台小型廉价的个人计算机可以为大型组织提供名称服务，这鼓励人们将计算机专用于此服务。反过来，专用计算机往往比提供多种不同且不相关的网络服务的计算机更强大。此外，可以设计具有小型只读表的服务器，以便在发生电源故障等情况时，它可以快速甚至自动恢复服务。（第 8 章［在线］和

§在线］讨论如何设计这样的系统。）DNS 还允许使用同义词，即间接名称。同义词通常用来解决两个不同的问题。以第一个问题为例，假设 Pedantic 大学计算机科学与工程系有一台计算机，其 Internet 地址为 minehaha.cse.pedantic.edu。这是一台较旧且速度较慢的机器，但以非常可靠而闻名。该系在这台计算机上运行万维网服务器，但随着服务器负载的增加，系里知道有一天需要将 Web 服务器移到一台速度更快的名为 mississippi.cse.pedantic.edu 的机器上。如果没有同义词，当服务器移动时，就需要通知所有人该系的万维网服务有了一个新名称。

使用同义词，实验室可以将间接名称 www.cse.pedantic.edu 绑定到 minehaha.cse.pedantic.edu，并将该间接名称作为其网站的名称公开。当 mississippi.cse.pedantic.edu 接管服务时，只需让 cse.pedantic.edu 域的管理员更改间接名称的绑定即可。所有一直使用名称 www.cse.pedantic.edu 访问该网站的客户都会发现该名称仍然正常工作；他们并不介意现在由另一台计算机处理这项工作。一般来说，服务名称的寿命预计会超过其与特定互联网地址的绑定，而同义词正好迎合了这种生命周期的差异。

同义词可以处理的第二个问题是允许一台计算机出现在两个截然不同的命名域中。例如，假设学术研究所的一个地球物理小组开发了一种预测火山爆发的服务，但该组织实际上没有适合运行该服务的计算机。它可以与商业供应商安排在名为 service-bureau.com 的机器上运行该服务，然后要求研究所名称服务器的管理员将间接名称 volcano.iss.edu 绑定到 servicebureau.com。然后，研究所可以用间接名称宣传其服务。如果商业供应商提高价格，只需重新绑定间接名称即可将服务转移给其他供应商。

由于解析同义词需要通过 DNS 进行一次额外的往返，并且 DNS 的基本名称到 Internet 地址绑定已经提供了一定程度的间接性，因此一些网络专家建议仅操作名称到 Internet 地址绑定即可获得同义词的效果。

4.4.4 DNS 中的名称发现

名称发现在域名系统中至少出现在三个地方：客户端必须发现附近的名称服务器的名称，用户必须发现所需服务的域名，解析系统必须发现不合格域名的扩展名。

首先，为了向名称服务器发送请求，它需要知道该名称服务器的 Internet 地址。 _在配置表中查找此地址。真正的名称发现问题是此地址如何进入配置表。原则上，此地址将是根服务器的地址，但正如我们所见，它可以是任何现有名称服务器的地址。最广泛使用的方法是，当计算机首次连接到网络时，它会执行名称发现广播，Internet 服务提供商 (ISP) 会对此做出响应，为连接者分配一个 Internet 地址，并告知连接者由 ISP 运营或为其运营的一个或多个名称服务器的 Internet 地址。终止名称发现的另一种方法是直接与本地网络管理员通信，以获取合适名称服务器的地址，然后将答案配置为

DNS_.

第二种名称发现形式涉及域名本身。如果您希望使用学术研究所的火山预测服务，您需要知道它的名称。必须发生一些以直接通信开始的事件链。通常，人们通过其他网络服务了解域名，例如通过电子邮件、查询搜索引擎、阅读新闻组中的帖子或浏览网页，因此最初的直接通信可能早已被遗忘。但使用其中每一项服务都需要知道域名，因此在更早的时候一定有过直接通信。个人电脑的购买者可能会发现它附带的 Web 浏览器已预先配置了制造商建议的万维网查询和目录服务的域名（以及制造商的支持站点和其他广告商的域名）。同样，互联网服务提供商的新客户通常可以在注册服务时被告知该 ISP 网站的域名，然后可以使用该域名发现许多其他服务的名称。

名称发现的第三个实例涉及用于非限定域名的扩展。回想一下，域名系统使用绝对路径名，因此如果 _ 呈现非限定名称（例如 library），则必须以某种方式将其扩展，例如扩展为 library.pedantic.edu。用于扩展的默认上下文通常是 _ 的配置参数。此参数的值通常由人类用户在最初设置计算机时选择，目的是尽量减少对最常用域名的输入。

4.4.5 DNS 响应的可信度

DNS 的一个缺点是，尽管它声称在其响应中提供权威的名称解析，但它不使用允许验证这些响应的协议。因此，入侵者有可能（不幸的是，相对容易）伪装成 DNS 服务器并向名称解析请求发送恶意或恶意的响应。

目前，处理此问题的主要方法是让 DNS 用户将其所有响应视为潜在的不可靠提示，并独立验证（使用第 7 章 [在线] 和第 11 章 [在线] 中的术语，我们称之为“执行端到端身份验证”）与该用户通信的任何系统的身份。另一种方法是让 DNS 服务器在与其客户端通信时使用身份验证协议。但是，即使 DNS 响应确实真实，它仍然可能不准确（例如，DNS 缓存可能包含过时的信息，或者 DNS 管理员可能配置了不正确的名称到地址绑定），因此细心的用户仍然希望独立验证其通信者的身份。

第 11 章 [在线] 描述了可用于身份验证的协议；网络专家们一直在争论是否或如何升级 DNS 以使用此类协议。

有兴趣了解更多有关 DNS 的读者应该浏览 DNS 阅读材料中的文档[进一步阅读建议 4.3]。

4.5 案例研究：网络文件系统 (NFS)

网络文件系统 (NFS) 由 Sun Microsystems, Inc. 于 20 世纪 80 年代设计，是一种客户端/服务应用程序，可通过网络为客户端提供共享文件存储。NFS 客户端将远程文件系统嫁接到客户端的本地文件系统名称空间上，并使其表现得像本地文件系统（参见第 2.5 节）。多个客户端可以挂载同一个远程文件系统，以便用户可以共享文件。

由于技术进步，人们开始需要 NFS。20 世纪 80 年代之前，计算机价格昂贵，每台计算机都必须由多个用户共享，并且每台计算机都只有一个文件系统。但经济压力的一个好处是，它可以轻松协作，因为用户可以轻松共享文件。20 世纪 80 年代初，建造工作站在经济上是可行的，这使得每个工程师都可以拥有一台私人计算机。但用户仍然希望拥有一个共享文件系统，以便于协作。NFS 正是提供了这一点：它允许任何工作站上的用户使用存储在共享服务器上的文件，共享服务器是一个功能强大的工作站，具有本地磁盘，但通常没有图形显示。

NFS 还简化了工作站集合的管理。如果没有 NFS，系统管理员必须管理每个工作站，例如，安排备份每个工作站的本地磁盘。NFS 允许集中管理；例如，系统管理员只需备份服务器的磁盘即可存档文件系统。在 20 世纪 80 年代，这种设置还具有成本效益：NFS 允许组织