

PXE 启动技术详解

马浩琨

Nov 19, 2025

1 导言

想象一下，当机房新到一百台服务器时，如果逐一使用 U 盘安装操作系统，不仅耗时耗力，还容易出错。PXE (Preboot eXecution Environment，预启动执行环境) 技术正是为解决这一问题而生。PXE 由 Intel 设计，作为 BIOS 或 UEFI 的扩展，允许计算机在未安装操作系统或硬盘的情况下，直接从网络服务器启动并加载程序。其核心价值在于实现无盘启动和批量网络部署，成为自动化运维和云计算环境中的关键技术之一。本文将深入解析 PXE 的工作流程、核心组件，并探讨其在实际应用中的配置方法，帮助读者从零开始全面掌握这一技术。

2 PXE 概述

PXE 是一种预启动执行环境，旨在让计算机通过网络启动，而无需依赖本地存储设备。它的设计目标在于扩展传统 BIOS 或 UEFI 的功能，使得设备在开机时能够从远程服务器获取启动文件。这种机制不仅提升了部署效率，还降低了硬件依赖，特别适用于大规模 IT 环境。

PXE 的优势主要体现在高效批量部署、无盘工作站支持、系统恢复与维护以及灵活性上。例如，在企业 IT 运维中，管理员可以同时为数十台服务器安装操作系统，而无需手动操作；在数据中心，新上架的服务器可以通过 PXE 快速初始化；网吧和学校机房则能利用它实现统一管理；云计算平台如 OpenStack 也依赖 PXE 进行镜像部署。这些应用场景突显了 PXE 在现代计算中的重要性，它不仅简化了运维流程，还为自动化奠定了基础。

3 PXE 启动的底层原理与流程

PXE 启动依赖于几个关键条件：客户端网卡必须支持 PXE 并在 BIOS 或 UEFI 中启用网络启动；局域网内需部署 DHCP 服务器和 TFTP 服务器；服务器端则需提供必要的启动文件，如引导程序和操作系统镜像。这些组件协同工作，确保客户端能够顺利从网络启动。

核心组件包括 DHCP 服务器、TFTP 服务器、PXE 引导程序以及网络存储服务。DHCP 服务器负责分配 IP 地址，并通过选项 66 (Next-Server) 指定 TFTP 服务器的地址，以及选项 67 (Bootfile-Name) 指定初始引导文件名。TFTP 服务器则用于传输小型启动文件，如引导加载器，它基于简单的 UDP 协议，适合在启动初期使用。PXE 引导程序通常是 pxelinux.0 或 GRUB2 的网络模块，作为客户端获取的第一个智能程序，负责后续的启动流程。网络存储如 NFS、HTTP 或 SMB 则提供大型操作系统安装文件，确保完整系统的加载。

PXE 启动过程可以分为多个步骤。首先，客户端开机后，PXE ROM 会广播一个 DHCP Discover 包，其中包含选项 60 标识自身为 PXE 客户端。接着，DHCP 服务器响应并提供 IP 地址、TFTP 服务器地址和引导文件名。

然后，客户端连接到 TFTP 服务器下载引导文件如 `pxelinux.0`。执行该文件后，它会通过 TFTP 下载配置文件，通常位于 `pxelinux.cfg/` 目录下，根据 MAC 地址或 IP 地址查找。配置文件指示加载操作系统内核和初始 RAM 磁盘，例如 `vmlinuz` 和 `initrd.img`。最后，内核启动并移交控制权，使用更高效的协议如 HTTP 挂载安装源，进入系统安装或启动阶段。整个过程体现了网络协议的高效协作，其中每个步骤都依赖于前一步的成功执行。

4 实战配置示例

以下以 Linux 系统结合 Kickstart 无人值守安装为例，演示 PXE 的配置过程。假设环境包括一台 CentOS 服务器作为服务端，以及一台支持 PXE 的客户端虚拟机。

首先，安装和配置 DHCP 服务。在服务器上，使用 `yum install dhcp` 命令安装 DHCP 软件包，然后编辑 `/etc/dhcp/dhcpd.conf` 配置文件。该文件中需设置 `next-server` 指向 TFTP 服务器的 IP 地址，并使用 `filename` 指定引导文件名为 `pxelinux.0`。例如，一个典型的配置段可能如下所示：

```
1 subnet 192.168.1.0 netmask 255.255.255.0 {
2   range 192.168.1.100 192.168.1.200;
3   option routers 192.168.1.1;
4   option domain-name-servers 8.8.8.8;
5   next-server 192.168.1.10;
6   filename "pxelinux.0";
7 }
```

这段代码定义了子网范围、网关和 DNS 服务器，同时通过 `next-server` 和 `filename` 确保客户端能定位到 TFTP 服务器和引导文件。解读时需注意，`next-server` 必须指向正确的 TFTP 服务器 IP，否则客户端无法获取启动文件。

接下来，安装 TFTP 服务。使用 `yum install tftp-server` 命令安装，并设置 TFTP 根目录为 `/var/lib/tftpboot`。然后，将必要的 PXE 引导文件如 `pxelinux.0` 和 `menu.c32` 复制到该目录。这些文件可从 SYSLINUX 项目获取，它们负责提供启动菜单和加载功能。配置 PXE 菜单时，在 `/var/lib/tftpboot/pxelinux.cfg/default` 文件中定义启动项，例如：

```
1 default menu.c32
2 prompt 0
3 timeout 300

5 label linux
6   menu label ^Install CentOS 7
7   kernel vmlinuz
8   append initrd=initrd.img inst.repo=http://192.168.1.10/centos7 ks=http
9     ↳ :/192.168.1.10/ks.cfg
```

这段代码指定了默认启动菜单，使用 `menu.c32` 提供图形界面，并设置超时时间。`label linux` 部分定义了启动项，其中 `kernel` 和 `append` 行分别指定内核文件和初始 RAM 磁盘，以及安装源和 Kickstart 文件路径。解

读时需注意，`inst.repo` 参数必须指向正确的 HTTP 共享目录，否则系统无法加载安装文件。

提供安装源时，将 CentOS ISO 镜像挂载或解压到 HTTP 共享目录，例如使用 `mount -o loop /path/to/CentOS-7-x86_64-DVD.iso /var/www/html/centos7` 命令。同时，创建 Kickstart 文件 `ks.cfg` 实现无人值守安装，该文件包含分区、用户设置等自动化配置。最后，启动客户端进行测试，观察其从获取 IP 到开始安装的完整流程，确保所有服务正常运行。

5 常见问题与故障排查

在 PXE 启动过程中，常见问题包括客户端无法获取 IP 地址、TFTP 连接超时或文件未找到，以及内核加载失败等。例如，如果客户端获取不到 IP 地址，可能原因是 DHCP 服务未运行、网络连接问题或防火墙阻挡。解决方法是检查 DHCP 服务状态、网络配置和防火墙规则，确保端口 67 和 68 开放。

当客户端提示「TFTP Open timeout」或「File not found」时，通常与 TFTP 服务相关。需验证 TFTP 服务是否启动、目录权限是否正确，以及文件路径是否匹配。例如，使用 `systemctl status tftp` 命令检查服务状态，并确认 `/var/lib/tftpboot` 目录中的文件完整。同时，防火墙需放行 UDP 69 端口，否则客户端无法连接 TFTP 服务器。

内核加载失败可能由于 `initrd.img` 或 `vmlinuz` 文件损坏或不匹配，或 PXE 配置文件中参数错误。例如，检查 `pxelinux.cfg/default` 文件中的 `kernel` 和 `append` 行，确保 `inst.repo` 路径正确。调试时，可以在配置文件中添加 `debug` 参数，并查看 DHCP 和 TFTP 服务器日志，以获取详细错误信息。这些排查步骤有助于快速定位问题，提高 PXE 部署的可靠性。

PXE 技术通过无盘启动和网络批量部署，极大地提升了 IT 运维的效率，成为现代数据中心和自动化环境的核心。本文回顾了其工作流程和核心组件，强调了在实际应用中的重要性。展望未来，相关技术如 UEFI HTTP Boot 使用 HTTP 协议替代 TFTP，提升了传输性能；iSCSI Boot 则通过远程磁盘挂载实现另一种无盘启动方式。这些技术与 PXE 共同构成了灵活、自动化的系统部署生态，推动着计算环境的持续演进。