

到邻近的“深入服务器”缓存中，从这里检索静态内容；同时将该静态内容提供给客户，邻近的缓存也经谷歌的专用网将请求转发给“大型数据中心”，从这里检索个性化的搜索结果。对于某 YouTube 视频，该视频本身可能来自一个“邀请坐客服务器”缓存，而围绕该视频的 Web 网页部分可能来自邻近的“深入服务器”缓存，围绕该视频的广告来自数据中心。总的来说，除了本地 ISP，谷歌云服务在很大程度上是由独立于公共因特网的网络基础设施提供的。

CDN 通常采用两种不同的服务器安置原则 [Huang 2008]：

- **深入。**第一个原则由 Akamai 首创，该原则是通过在遍及全球的接入 ISP 中部署服务器集群来深入到 ISP 的接入网中。（在 1.3 节中描述了接入网。）Akamai 在大约 1700 个位置采用这种方法部署集群。其目标是靠近端用户，通过减少端用户和 CDN 集群之间（内容从这里收到）链路和路由器的数量，从而改善了用户感受的时延和吞吐量。因为这种高度分布式设计，维护和管理集群的任务成为挑战。
- **邀请做客。**第二个设计原则由 Limelight 和许多其他 CDN 公司所采用，该原则是通过在少量（例如 10 个）关键位置建造大集群并使用专用高速网络连接这些集群来邀请到 ISP 做客。不是将集群放在接入 ISP 中，这些 CDN 通常将每个集群放置在这样的位置上，即同时靠近许多第一层 ISP 的 PoP 的位置（参见 1.3 节），例如，一个大城市中同时靠近 AT&T PoP 和 Verizon PoP 的几英里范围内。与深入设计原则相比，邀请做客设计通常产生较低的维护和管理开销，可能以对端用户的较高时延和较低吞吐量为代价。

一旦 CDN 的集群准备就绪，它就可以跨集群复制内容。CDN 可能不希望将每个视频的副本放置在每个集群中，因为某些视频很少观看或仅在某些国家中流行，事实上，许多 CDN 没有将视频推入它们的集群，而是使用一种简单的拉策略：如果客户向一个未存储该视频的集群请求某视频，则该集群检索该视频（从某中心仓库或者从另一个集群），向客户流式传输视频时的同时在本地存储一个副本。类似于因特网缓存（参见第 2 章），当某集群存储器变满时，它删除不经常请求的视频。

1. CDN 操作

在讨论过这两种部署 CDN 的重要方法后，我们现在深入看看 CDN 操作的细节。当用户主机中的一个浏览器指令检索一个特定的视频（由 URL 标识）时，CDN 必须截获该请求，以便能够：①确定此时适合于该客户的 CDN 服务器集群；②将客户的请求重定向到该集群的某台服务器。我们很快将讨论 CDN 是如何能够确定一个适当的集群的。但是我们首先考察截获和重定向请求所依赖的机制。

大多数 CDN 利用 DNS 来截获和重定向请求；这种使用 DNS 的一个有趣讨论见 [Vixie 2009]。我们考虑用一个简单的例子来说明通常是怎样涉及 DNS 的。假定一个内容提供商 NetCinema，雇佣了第三方 CDN 公司 KingCDN 来向其客户分发视频。在 NetCinema 的 Web 网页上，它的每个视频都被指派了一个 URL，该 URL 包括了字符串“video”以及该视频本身的独特标识符；例如，转换器 7 可以指派为 <http://video.netcinema.com/6Y7B23V>。接下来出现如图 7-4 所示的 6 个步骤：

- 1) 用户访问位于 NetCinema 的 Web 网页。