DNS 缓存一致性机制

杨子凡

Nov 03, 2026

DNS 解析的性能与数据时效性始终构成一对矛盾。解析器为了降低查询延迟而引入各级缓存，但缓存中的记录一旦过期或上游发生变更，旧数据就会造成服务中断或访问异常。缓存一致性在 DNS 语境下被定义为：确保各级缓存中的资源记录与权威数据在可接受的时间窗口内保持一致。文章目标是梳理 TTL、权威刷新、CDN、DoH 及递归多级缓存协同策略，帮助读者在实际工程中做出权衡。

1　DNS 缓存层级与角色

DNS 解析路径上存在四类缓存角色。客户端 stub resolver 缓存位于操作系统内核或应用程序进程内，缓存时间通常仅数十秒到数分钟。本地或企业递归服务器如 dnsmasq、Unbound 或 Windows DNS 承担内部网络的解析任务，其缓存容量更大、TTL 策略更灵活。公共递归如 Google Public DNS 或 Cloudflare 1.1.1.1 面向全球用户，缓存命中率极高，但也放大了任何不一致的影响。权威服务器是数据源头，通过 NS 记录和区域传输向下游发布变更。各级缓存的 TTL 取值存在差异，导致同一域名在不同节点可能呈现不同结果。

2　TTL 机制与过期策略

RFC 1034 与 RFC 1035 规定，TTL 表示资源记录可被缓存的最大秒数。权威区文件中既可为整个区域设置最小 TTL，也可为单条记录单独指定 TTL。SOA 记录中的最小 TTL 字段同时充当消极缓存的默认值。RFC 2308 引入消极缓存机制，NXDOMAIN 与 NODATA 响应也会被缓存，避免重复查询不存在的名称。零 TTL 意味着记录不可缓存，解析器每次都必须向上游发起查询。递归服务器常在 TTL 剩余时间低于阈值时启动预取，从而平滑缓存更新过程。

3　缓存更新与失效触发

缓存失效分为被动与主动两种方式。被动失效依赖 TTL 自然衰减，记录在过期后被丢弃。主动失效则通过多种协议实现。DNS NOTIFY 配合 IXFR 或 AXFR 可在主从服务器间快速同步增量变更。EDNS0 的 EXPIRE 选项允许权威服务器告知递归服务器记录的绝对过期时间。RFC 8767 定义的 Served Stale 策略允许递归服务器在权威不可达时继续提供过期数据，并异步尝试刷新。权威服务器宕机或区域签名更新时，缓存中的旧签名或旧地址仍可能被使用，造成解析异常。

4 多级缓存一致性挑战与对策

公共递归与权威之间的最长 TTL 偏差是常见问题。权威将 A 记录 TTL 设置为 60 秒，而公共递归可能将其上限截断为 300 秒，导致变更传播延迟。CDN 边缘节点缓存与源站 DNS 的漂移表现为：边缘节点依据本地缓存的 IP 提供服务，而源站已切换至新地址。DoH 与 DoT 加密隧道隐藏了原始 DNS 报文，中间设备无法观察或调整 TTL。拆分 DNS 环境下，内部与外部权威返回不同结果，递归服务器需同时维护两套缓存。staleness ratio 与 update propagation latency 可作为量化指标，前者衡量过期记录占比，后者统计变更从权威到达终端用户的时间。

5 工程实践与案例

大型网站进行 DNS 切换时通常采用零停机方案。首先将 TTL 降低至 60 秒，等待全球缓存自然过期，再将 A 记录指向新 IP，待流量完全切换后恢复原 TTL。Kubernetes 环境中，CoreDNS 配合 NodeLocal DNSCache 可在节点本地缓存上游响应，降低延迟并减少对 CoreDNS 的并发压力。使用 EDNS Client Subnet 时，缓存键需包含客户端子网前缀，否则不同子网用户可能共享错误结果。GitHub 全球 DNS 劫持事件中，攻击者篡改公共递归缓存，导致数小时内大量用户被导向恶意地址，凸显了缓存残留的危害。

现有 DNS 缓存机制缺乏强一致性协议，变更传播时间取决于各级 TTL 叠加。未来方向包括 DNSSEC 签名时间戳与自动刷新机制，可在签名过期前触发主动更新；基于 QUIC 或 HTTP3 的 DNS 快速刷新通道能在毫秒级内推送变更；集中式配置中心与 DNS 联动可在发布新版本时同时更新配置与 DNS 记录。运维与开发者可遵循以下 checklist：确认各级缓存的 TTL 策略是否一致；为关键域名设置合理的消极缓存 TTL；监控 staleness ratio 与 propagation latency；定期演练 DNS 切换流程；评估 DoH/DoT 对内部策略的影响。