

内容分发网络中的代理服务器放置算法

肖磊, 黄传河, 贾小华, 熊大红, 李 桓, 蔡 莉

(武汉大学计算机学院, 武汉 430072)

摘 要: 内容分发网络是通过把内容分发至网络边缘, 减轻源服务器负载, 减小用户延迟。该文在代理服务器容量有限以及内容发布者预算有限的情况下, 提出了一种基于贪婪算法的代理服务器放置算法, 目标在于最小化用户访问延迟。

关键词: 内容分发网络; 代理放置; 内容放置

A Placement Algorithm of Content Distribution Network

XIAO Lei, HUANG Chuanhe, JIA Xiaohua, XIONG Dahong, LI Huan, CAI Li

(Computer School, Wuhan University, Wuhan 430072)

[Abstract] Content distribution network is a network of machines which distribute content to the edge of the Internet. The goal is to balance the origin server load and to reduce the average delay for all clients to download objects in the network. A greedy placement algorithm is given in this paper, in the situation which proxy is capacitated and budget is limited, to minimize the the average delay.

[Key words] Content distribution network (CDN); Proxy placement; Content placement

1 内容分发网络简介

内容分发网络提供了一种智能化的解决方案, 通过复制源服务器内容(动态或者静态的内容)到地理位置不同的代理服务器上, 同时根据用户请求的特定内容, 自动把该请求转发到含有请求内容副本且距离用户最近的代理服务器上去, 从而可以避免连接到提供该内容的源服务器。重要的是该过程对用户是透明的, 网络镜像服务需要用户自己选择, 无法自动判断最佳站点。常用的请求重定向技术有HTTP重定向, IP重定向和DNS重定向。通常, DNS重定向是最有效的技术。

内容分发网络具有以下几点显著优点^[1]:

- (1) 由于在地理上或者网络拓扑结构中更接近所需内容, 客户将获得更快的响应速度。
- (2) 对各种类型内容的支持, 如: HTML、图像、动态内容、经验证内容以及流媒体等。
- (3) 高效传递互联网上内容和服务。
- (4) 安全传输内容。
- (5) 通过集中管理降低管理成本。

内容分发网络服务商可以和网络服务提供商(ISP)合作, 在ISP的设施中配置它们基于软件的内容分发服务器。这种方法可以让ISP将自己的网络内容快速地发送给目标用户, 而又不需要自己创建一个专门的内容分发网络, 所以这种方法已经获得了市场的普遍欢迎。目前, 提供CDN服务的公司有Akamai、Digital Island、Inktomi等公司。

2 内容分发网络中的代理放置问题

内容分发网络的目标是减少用户访问延迟、均衡服务负载、降低网络带宽消耗。为达到这个目标, 一般有两种方法: (1) 有效重定位用户请求到离用户最近的代理服务器上; (2) 有效放置代理服务器。对于方法(1), 已经有很多文章做了大量的讨论。对于方法(2), 该问题一般可以简单描述为一个k-中心问题, 即在网络中寻找k个中心节点, 使其他节点到这个k个节点的距离最小。该问题已被证明是一个NP复杂问题。文献[6]中对k-中心问题进行了详细的讨论,

并给出了一个得到目标函数可行解的 $6\frac{2}{3}$ 近似算法。

代理服务器的放置一般有两类算法: 一种是基于某种网络拓扑结构的算法, 通常假设为树状结构; 另外一种贪婪算法, 得到一个近似解。文献[2]中对这两类算法进行了详尽的比较, 结果显示, 不管使用那种放置方法, 在某些限制条件下, 当增加代理服务器数目到一定程度后, 对减小客户到服务器的往返时间RTT和均衡服务负载的效果将不再明显; 比较而言, 文中所用贪婪算法效果较好。文献[3]中对于CDN中几种常用代理服务器放置算法进行了详细的讨论, 该文重点在于考虑用户所产生的通信负载问题, 而忽略用户实际上的下载内容, 从而存在一个限制, 即一个客户只可以访问一个代理服务器, 也就是说源服务器上的所有内容将完整复制在该代理服务器上。但是, 这个限制是不现实的, 除非代理服务器的容量代价减小到可以忽略不计, 而且, 将所有内容完整复制在代理服务器上也是浪费的, 因为对于某些区域的用户, 可能主要访问量在于某些特定内容, 而很少访问到其他内容。

用户的下载时间可能受到多方面因素的影响, 如服务器过载, 网络瓶颈, 网络延迟等。从理论上来说, 由于光速的限制, 网络延迟是最难得到改进的, 而不像网络瓶颈和服务过载可能由于硬件设备的更新得到改进。下面将讨论在内容发布者预算和候选代理服务器存储容量有限的条件下, 如何选择代理服务器以及在其上放置的内容, 使用户的访问延迟最小化。

3 减小用户访问延迟的代理放置算法

Internet的拓扑结构用连通图 $G(V, E)$ 表示, 其中 V 代表节点的集合, E 代表网络链路的集合。对于链路 $(u, v) \in E$, $d(u,$

基金项目: 国家自然科学基金资助项目(60273071)

作者简介: 肖磊(1978—), 男, 硕士生, 主要研究: 分布并行处理; 黄传河, 教授; 贾小华, 博导; 熊大红、李桓、蔡莉, 硕士生

收稿日期: 2003-05-22 **E-mail:** whu-lei@163.com

v)表示链路的距离。令 S 表示分发内容的Web服务器, S 的内容为数据对象, 用 $O = \{o_1, o_2, \dots, o_m\}$ 表示; 数据对象可以是图像, 视频、音乐文件或其他大型数据文件等。用 z_{oi} 表示数据对象 o_i 的大小, 用 w_{oi} 表示数据对象 o_i 的更新频率。客户节点 $v \in V$ 访问 o_i 的频率为 $r(v, o_i)$, $1 \leq i \leq m$ 。

假定可供 S 选择的服务器有 t 个, 记为 $C = \{p_1, p_2, \dots, p_t\}$, $C \in V$ 。每个候选服务器 p_i 可供 S 使用的存储容量为 Z_{pi} , 费用为 F_{pi} 。 S 的问题是:

(1) 选取一组候选服务器 $P, P \subseteq C$

(2) 确定放置在每个候选服务器 $p_i \in P$ 上的数据对象 $O', O' \subseteq O$

CDN的一个重要目标就是减少用户从 S 上下载内容的等待时间。把从用户到目标对象的距离作为延迟的度量, 令 $d(v, o_i)$ 表示从 v 到数据对象 o_i 的距离, 一般情况下, 它是从 v 到存放 o_i 的候选服务器的距离; 但是, 如果 v 离 S 最近, 则 $d(v, o_i)$ 是 v 到 S 的距离。可以得到用户 v 下载数据对象 o_i 的延迟为 $r(v, o_i) \times d(v, o_i)$ 。那么, 所有客户 V 下载数据对象 O 的平均延迟为

$$Delay = \sum_{v \in V} \sum_{i=1}^m r(v, o_i) \times d(v, o_i) \quad (1)$$

在每个代理服务器上能够存放的数据对象受到该服务器的存储容量的限制, 即为

$$\forall p_i \in P: \sum_{o_i \in p_j} z_{oi} \leq Z_{pi} \quad (2)$$

假设内容发布者的预算为 B , 所选取的代理服务器应满足

$$\sum_{p_j \in P} F_{p_j} \leq B \quad (3)$$

我们的问题是找出 P 以及复制到 P 上的数据对象, 使得按公式(1)定义的用户平均延迟在满足式(2), 式(3)的条件下最小。即

$$\text{Min} \sum_{v \in V} \sum_{i=1}^m r(v, o_i) \times d(v, o_i) \quad (4)$$

$$\text{s.t.} \forall p_i \in P: \sum_{o_i \in p_j} z_{oi} \leq Z_{pi}$$

$$\sum_{p_j \in P} F_{p_j} \leq B$$

该问题是一个NP复杂问题, 考虑用贪婪法求解, 算法描述如下:

$D[m, t]$ 表示一个 $m \times t$ 矩阵, 假设 $D[i, j]$ 表示内容 O_i 放在代理服务器 P_j 上时, 用户对内容 O_i 访问产生的延迟, 即 $D[i, j] = \sum_{v \in V} r(v, o_i) \times d(v, o_i)$, 这里 $d(v, o_i)$ 是用Dijkstra算法得到的 v 到 o_i 所在的代理服务器的最短距离。假设MAX表示计算机精度允许范围内的最大值。在 $D[i, j](1 \leq i \leq m, 1 \leq j \leq t)$ 中选取最小值 $D[k, l]$, 即 o_k 放置在 P_l 上, 并把其值重新赋为MAX, 即 $D[k, l] = \text{MAX}$ 。这样选取的目的是为了使产生延迟较大的放置能够在后来的放置中由于代理服务器的增加得到改进。由于费用只和代理服务器有关, 因此把 P_l 上剩余的存

储空间也放置完。当然, 同样的原因, 我们也选取在 P_l 上放置产生延迟较小的内容放置。同时, 更新这些内容在其他服务器上假设放置所产生的延迟。在预算没有使用完的前提下, 对于 $D[i, j](1 \leq i \leq m, 1 \leq j \leq t-1, 1 \leq j \leq t)$ 重复上述选取。

最后, 得到 $D[i, j]$ 中值为MAX的行(i)列(j)号, 即表示内容 o_k 放置在 P_l 上。

step 1 对于 $1 \leq i \leq m, 1 \leq j \leq t$, 找出 $D[i, j]$ 中最小的元素, 假设为 $D[k, l]$, 并对其赋新值MAX

step 2 计算 $Z_{pi} = Z_{pi} - z_{oi}$
while($Z_{pi} > 0$)

{

对于 $1 \leq i \leq m$

找出 $D[i, j]$ 中最小的元素, 假设为 $D[k, l]$ 并对其赋新值为MAX

对于 $1 \leq j \leq t, j \neq l$

重新计算 $D[k, j]$

setp 3 计算 $B = B - F_{pi}$, 如果 $B > 0$, 且 $D[m, t]$ 存在非MAX值, 转到setp 1; 否则, 转到setp 4

step 4 $D[i, j]$ 中所有值为MAX的元素即为所求

算法的时间复杂度为 $O(am(1+b)mt)$, a 大约为 B 除以所有代理服务器平均费用的值, $0 < b < 1$, 空间复杂度为 $O(mt)$ 。这里为了算法描述方便, 所以假设了一个极大值MAX, 实际计算中, 可以采用把每次找出的最小值重新赋为其相反值, 即 $D[i, j] = -D[i, j]$ 的方法, 然后选取延迟最小且非负的元素。

该算法不一定能得到最优解, 但可以得到一个较优解。

因为, 算法中对某个内容, 每次选取使所有用户对其访问延迟最小的代理服务器放置, 这种贪婪原则可能导致另外一个问题, 即如果该服务器的费用较高, 但是另外某些服务器的费用和却较小, 如果把该内容分放在这些服务器上, 由于对于这两种放置情况, 用户对该内容的访问频率是相同的, 但是后者放置中, 用户到服务器的距离更小, 因此用户对该内容的访问延迟也就更小。此外, 算法并没有考虑代理服务器内容更新的代价问题, 同时, 算法中所使用的用户和服务器的距离是在知道网络拓扑的情况下使用Dijkstra最短路径算法计算的, 实际中不可能完全知道Internet的拓扑结构。

4 结束语

本文讨论了在内容分发网络中代理服务器的放置问题, 并提出了一种贪婪放置算法, 目标在于减少用户的访问延迟。该问题可能还有其他比较好的解法, 如假设某内容是否放置在某服务器上, 可以引进0-1变量, 从而转化为一个0-1整数规划问题, 进而使用分支定界法, 割平面等方法求解。

参考文献

- 1 Content Delivery Networks: An Introduction (White Paper). <http://cdn.hcltech.com>
- 2 Cronin E, Jamin S, Cheng Jin. Constrained Mirror Placement on the Internet. IEEE, 2002-02
- 3 Qiu Lili, Padmanabhan V N, Voelker G M. On the Placement of Web Server Replicas. IEEE, 2001-01
- 4 Guo Changjie, Xiang Zhe. A Novel Greedy Heuristic Placement Algorithm in Distributed Cooperative Proxy Systems. IEEE, 2001-01
- 5 Jia Xiaohua, Li Deying. Placement of Read-write Web Proxies in the Internet. IEEE, 2001-01
- 6 Charikar M, Guha S. A Constant-factor Approximation Algorithm for the k-median problem. Proceedings of the 31st Annual ACM Symposium on Theory of Computing, 1999