CDN 中的代理服务器放置算法*

熊大红，黄传河，贾小华，肖 磊，蔡 莉，李 桓
（武汉大学 计算机学院，湖北 武汉 430072）

摘 要：CDN 作为解决网络堵塞、带宽瓶颈等问题的一种重要的技术，已为大家逐渐熟悉。目前众多的文章都集中在解决 CDN 服务器对于用户请求的有效重定向问题，以减少请求的延迟和平衡负载，而很少关注代理服务器镜像的放置策略。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，提出了一种基于动态规划和贪婪算法组合而成的代理服务器放置算法，目标在于最小化网络带宽消耗。

关键词：内容分发网络；代理放置；内容放置
中图法分类号：TP368.5 **文献标志码**：A **文章编号**：1001-3695(2004)05-0250-02

A Placement Algorithm of Content Distribution Network(CDN)

XIONG Da-hong, HUANG Chuan-he, JIA Xiao-hua, XIAO Lei, CAI Li, LI Huan
(*School of Computer, Wuhan University, Wuhan Hubei* 430072, *China*)

Abstract: As we all know, CDN became a technique for soluting the network traffic and bandwidth bottleneck. Existing work on CDNs has primarily focused on techniques for efficiently redirecting user requests to appropriate CDN servers to reduce requestlatency and balance load. In this paper, a dynamic programming approach and a greedy placement algorithm are given, in the situation which proxy is capacitated and budget is limited, and the goal is to minimize the network bandwidth consumption.
Key words: CDN(Content Distribution Network); Proxy Placement; Content Placement

1 引言

互联网的飞速发展拓宽了我们的视野，丰富了我们的经验。然而随着各项技术的发展，尤其是视频、音频的传输使大多数网站受到了网络堵塞的困扰。而用户在使用网络时对网站的浏览速度和效果却愈加重视，一般而言，网站访问响应速度取决于如下因素：网络的带宽瓶颈、传输途中的路由阻塞和延迟、网站服务器的处理能力等。多数情况下，网站响应速度和访问者与网站服务器之间的距离有密切的关系。如何才能让各地的用户都能够进行高质量的访问，并尽量减少由此而产生的费用和网站管理压力呢。内容分发网络(Content Distribution Network，CDN)这一概念的提出正好解决了以上的问题，从而为广大客户提供了内容丰富的高质量服务。

2 CDN 简介

CDN 是一个建立并覆盖在互联网之上，由分布在不同区域的节点服务器群组成的虚拟网络，它通过实现用户对网站的就近访问及网络流量智能分流，从技术上全面解决由于网络带宽小、用户访问量大、网站分布不均等对用户访问效果的影响，大大提高了网络的响应速度。CDN 的技术原理是在现有的互联网络中建立一个完善的中间层，将网站的内容分布到最接近用户的网络“边缘”，使用户能以最快的速度，从最接近用户的地方获得所需的信息。它提供了一种智能化的解决方案，通过复制源服务器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，区别于网络镜像服务：需要用户自己选择，无法自动判断最佳站点。

内容分发网络具有以下几点显著优点：

(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。

3 CDN 中的代理服务器放置问题

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。可分为两个方面：①有效重定位用户请求到离用户最近的代理服务器上去；②有效放置代理服务器。第一个问题已经有大量文章对此作出了讨论，在此不再赘述，现在我们来研究第二个问题：也即在众多的 ISP 网络服务商中选取一定数量服务器作为代理服务器。CDN 的一个重要目的是减少网络带宽消耗，本文主要是针对这个方面提出算法。

代理服务器的放置一般有以下几类算法：①基于某种假设

收稿日期：2003-06-01；**修返日期**：2003-07-02
基金项目：国家自然科学基金资助项目(60273071)

网络拓扑结构的算法,通常假设为树状结构;②贪婪算法,得到一个近似解;③随机算法,由内容所有者随机挑选代理服务器;④采用 Hot Spot 算法,计算以候选节点为圆心,半径为 r 的圆内的访问量,当半径 r 逐渐增大时,根据圆内的访问量选取一定数量的候选节点作为代理服务器。文献[1]中对这四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,对减少客户到服务器的往返时间 RTT 和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。

4　问题描述

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 $(u,v)\in E$,$d(u,v)$ 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 $O=\{o_1,o_2,\cdots,o_m\}$ 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} 表示数据对象的 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 $r(v,o_i)$,$1\leqslant i\leqslant m$。

假定可供 S 选择的服务器有 t 个,记为 $C=\{p_1,p_2,\cdots,p_t\}$,$C\subseteq V$。每个候选服务器 p_j 可供 S 使用的存储容量为 Z_{pj},费用为 F_{pj}。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 $p_j\in P$ 上的数据对象,$O'\subseteq O$

CDN 的一个重要目的就是减少网络带宽消耗。我们定义数据访问成本为数据量乘以传输的距离,则 v 下载 o_i 的成本为 $r(v,o_i)\times z_{oi}\times d(v,o_i)$。上载 o_i 到代理服务器或更新代理服务器上的数据对象也有成本。令 $P(o_i)$ 表示存放 o_i 的代理服务器的集合,其中 $P(o_i)\subseteq P$。假定上载或更新代理服务器的数据采用组播方式,组播机制使用最短路径树传送 o_i 到 $P(o_i)$。令 $SPT(S,P(o_i))$ 表示以 S 为树根、包含 $P(o_i)$ 中所以代理服务器的最短路径树。则访问数据对象 o_i 的平均成本可表示为:

$$AccessCost(o_i)=\sum_{v\in V} r(v,o_i)\cdot z_{oi}\cdot d(v,o_i)+w_{oi}\cdot z_{oi}\sum_{(x,y)\in SPT(S,P(o_i))} d(x,y)$$

上述公式的第一项为用户读取 o_i 的成本,第二项是更新 o_i 的成本。访问 S 的所有数据对象的成本可表示为:

$$AccessCost=\sum_{i=1}^{m} AccessCost(o_i) \tag{1}$$

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:

$$\forall p_j\in P:\sum_{o_i\in p_j} z_{oi}\leqslant Z_{pj} \tag{2}$$

假设内容发布者的预算为 B,所选取的代理服务器应满足:

$$\sum_{pj\in P} F_{pj}\leqslant B \tag{3}$$

我们的问题是找出 P,即式(1)以及复制到 P 上的数据对象即式(2),使得按式(1)定义的访问成本满足式(2),式(3)的条件下最小。即

$$\text{Min}\sum_{i=1}^{m} AccessCost(o_i) \tag{4}$$

$$\text{S.t.}\quad \forall p_j\in P:\sum_{o_i\in p_j} z_{oi}\leqslant Z_{pj}$$

$$\sum_{p_j\in P} F_{pj}\leqslant B$$

5　放置问题的解决方案

先对问题描述中的条件进行一点改进,Ts 表示以 s 为根的树,整个网络转换为树结构 Ts,那么对于上面描述的 C 中的元素 p_i 都为 s 的子孙节点,为了方便描述,我们用层次遍历来定 p_i 的下标,也从 s 开始层次遍历 Ts,依次遍历的候选服务器为 $p_1,p_2,\cdots,p_t$。对于 o_i,假定放置 o_i 的服务器集合为 C_i,我们有:当 C_i 中只有 s 一个元素时

$$AccessCost(o_i)=\sum_{v\in V} r(v,o_i)\times z_{oi}\times d(v,o_i) \tag{5}$$

定义 $L(p_j,C_i)$ 为沿 p_j 到 s 的路径上第一个 C_i 中的元素。我们假定 $v\in C_i-s$,并且 Tp_j 中无其他 C_i 中元素,那么对 o_i 的费用增加了:

$$AccessCost(o_i,v)=-\sum_{v\in Tpj} r(v,oi)d(p_j,L(p_j,C_i))z_{oi}+w_{oi}d(p_j,L(p_j,C_i))z_{oi}=-(\sum_{v\in Tpj} r(v,oi)-w_{oi})d(p_j,L(p_j,C_i))z_{oi} \tag{6}$$

上面式子的第一部分表示增加节点 p_j 到 C_i 中所增加的读费用,第二部分表示增加的写费用。那么此时放置 o_i 的总费用为式(5)和式(6)的和:

$$AccessCost(o_i)=\sum_{v\in V} r(v,o_i)\times z_{oi}\times d(v,o_i)-(\sum_{v\in Tpj} r(v,oi)-w_{oi})d(p_j,L(p_j,C_i))z_{oi} \tag{7}$$

由上式来看,第一部分是固定的,所以只要第二部分达到最大,那么式(6)就为最小,即费用最低。很显然只要保证 $r(v,o_i)-w_{oi}>0$ 即可。下面为具体的算法:

```
(1)For(i=1;i<=m;i++)
(2){
(3)For(j=1;j<=t;j++)
(4){ if(r(v,oi)>woi) then
(5)Rij=(Σ r(v,oi)-woi)d(pj,L(pj,Ct))zoi/zoi//Rij是为减少费用和数据大小的比值,作为后面选取节点的依据
(6)=(Σ r(v,oi)-woi)d(pj,L(pj,Ct))
(7)end if}
(8)}
//选取Rij中最大的,将pj放到对于的Ct中,对pj,选取Rkj中最大的,依次将pk放到对于的Ck中,对进行操作过的Rkj置0,直到Rkj都为0或者容量超出pj的最大容量Zpj为止。然后重新在Rij选取中最大的,直到Rij都为0或者费用超出总费用B限制为止。
(9)F=0//费用
(10)For(j=1;j<=t;j++){ Zj=0} // pj的容量
(11)do
(12){ 对所有Rij选取其中最大的一个
(13)k=Rij的第一个下标
(14)l=Rij的第二个下标
(15)F=F+Fpl
(16)Zl=Zl+zok
(17)Rkl=0
(18)If(F<=B) then
(19){while(Zl<=Zpl)
(20){将pl放入Ck中
(21)对于l选取Rjl为最大的一个
(22)If(Rjl!=0)将第一个下标赋给k
(23)Else 跳出此循环
```

(下转第 254 页)

(1)基于 Handle System 的域名服务框架

Handle System 的报文可以通过 UDP 或 TCP 传输,UDP 和 TCP 的协议端口号是 2641[3],可以将域名对应成为一个 Handle,设立一个 Proxy,将域名的请求转变为对 Handle 的请求,改造域名解析服务器,用 Handle 的管理模型来管理 DNS 的区文件(Zone File)。

DNS 服务器与 Handle 服务器组合在一起,可以称之为 Handle-DNS Server。DNS 服务器和 Handle 服务器之间的报文传输,通过一个驱动程序,将域名的请求转换为对 Handle 的请求,采用 Handle 协议。域名的区文件在 Handle Server 中存储为 Handle 数据文件,如图 4 所示。

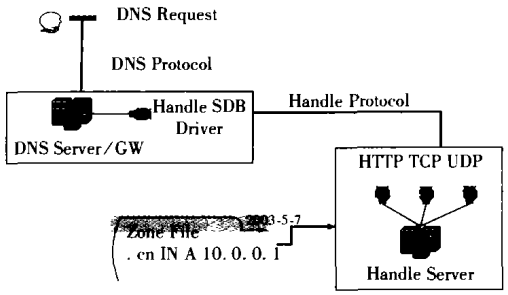

图 4 基于 Handle System 的域名服务框架

(2)域名与 Handle 的对应

每个域名都可以对应为一个 Handle,域名区文件中的所有记录都转换为 Handle,存储在 Handle 服务器中,这种对应关系如下:

DNS	Naming Authority Handle
"."	0. NA/dns {for DNS root}
us	0. NA/dns. us
va. us	0. NA/dns. us. va
cnri. reston. va. us	0. NA/dns. us. va. reston. cnri
cn	0. NA/dns. cn
cnnic. net. cn	0. NA/dns. cn. net. cnnic

4 结束语

域名是互联网寻址技术的基础,也是迄今为止互联网上应用最为广泛的技术之一,同时也发现域名技术有一些不足的地方。针对这些不足,有两种解决问题的思路:一种是继续以域名技术为基础,不断完善和改进,如 IETF 成立了 DNS Extensions 工作组,解决了 DNSSEC,Secure Domain Name System Dynamic Update 等域名安全方面的问题[4],成立了 Internationalized Domain Name 工作组来解决多语种域名的问题[5];另一种是建立一套新的互联网寻址体系,满足下一代互联网对寻址定位技术的需求,同时兼容域名系统。本文遵循后一种思路,做了一些研究工作。

参考文献:

[1] Sam X Sun, Larry Lannom. Handle System Overview [EB/OL]. http://hdl. handle. net/4263537/6001,2002-09.

[2] Sam X Sun, Larry Lannom. Handle System Namespace and Service Definition[EB/OL]. http://hdl. handle. net/4263537/6002, 2002-11.

[3] Sam X Sun, Larry Lannom. Handle System Protocol (ver 2.1) Specification[EB/OL]. http://hdl. handle. net/4263537/6003,2002-11.

[4] IETF DNS Extensions Working Group[EB/OL]. http://www. ietf. org/html. charters/dnsext-charter. html,2003-05.

[5] IETF Internationalized Domain Name Working Group[EB/OL]. http://www. ietf. org/html. charters/OLD/idn-charter. html,2003-05.

作者简介:

毛伟(1968-),男,主任,研究员,硕士生导师,硕士研究生,主要研究方向为计算机网络系统、互联网寻址技术;孙洵(1964-),男,双硕士研究生,美国国家研究推进机构 CNRI(The Corporation for National Research Initatives)研究员;王峰(1977-),男,博士研究生,主要研究方向为计算机网络系统、互联网寻址技术。

(上接第 251 页)

(24) $Z_l = Z_l + z_{ok}$

(25) $R_{kl} = 0$}

(26) For(i = 1; i < = m; i ++){ $R_{il} = 0$}

(27)}

(28) End if

(29)}while((存在 R_{ij}! = 0) and (F < B))

(30)输出 C_i

上述算法并不能求得最优解,但可得到一个较优解,这是因为在算法中获取某个候选代理节点为对象 o_i 的代理是通过以 R_{ij} 为贪婪准则,R_{ij} 为设置此节点为代理节点所减少的读写费用/o_i 的大小。首先选取 R_{ij} 最大的代理节点,然后随之选取对于此节点的数据对象,选取标准依然为取对于此代理节点 j 的 R_{ij} 为最大的数据对象。此算法可能导致数据对象的大小比较大的节点不能找到代理服务器,因为它可能使总费用减少得不是很多,当总放置费用 B 已经最大时,而此对象仍然没有放下去,但有些 R_{ij} 比较小,其代理节点的容量 Z_{pj} 很大,依然能被放置下去。而同样也可以以 $(\sum_{v \in Tpj} r(v, oi) - w_{oi})\ d(p_j, L(p_j, C_i))\ z_{oi}$ 为贪婪准则,选取合适的代理服务器以及在此服务器上的放置,具体算法与利用 R_{ij} 为贪婪准则类似,在这里就不必过多描述。

6 结束语

代理服务器的放置涉及到很多方面的要求,如网络延迟、用户下载服务器文件的等待时间、网络带宽等方面的问题。本文并没有对它们进行过多的阐述,而是着重于访问成本。本文先是在树的网络拓扑结构下利用动态规划法计算出所有可能使得成本降低的候选代理服务器,然后利用贪婪法筛选出满足代理服务器容量和整个预算限制的代理服务器。

参考文献:

[1] Lili Qiu, V N Padmanabhan, G M Voelker. On the Placement of Web Server Replicas[J]. IEEE INFOCOM,2001,(3):1587-1596.

[2] Jianliang Xu, Bo Li, D L Lee. Optimal Replica Placement On Transparent Replication Proxies for Read/Write Data[J]. IEEE International,2002,3-5,103-110.

[3] Xiaohua Jia, Deying Li. Placement of Read-write Web Proxy on the Internet[J]. IEEE International,2001,16-19,687-690.

[4] 严蔚敏,吴伟民. 数据结构(C 语言版)[M]. 北京:清华大学出版社,1997.

作者简介:

熊大红(1978-),男,硕士研究生,研究方向为计算机网络、分布并行计算、量子计算;黄传河(1963-),男,教授,研究方向为计算机网络、分布并行计算、量子计算;贾小华(1962-),男,博士生导师,研究方向为计算机网络、分布式系统。