基于 k 中心点算法的 TOPO 服务器算法的研究

王 美,李晓峰,孟令军,张立军
(山东建筑大学 计算机科学与技术学院,山东 济南 250101)

摘 要:在一个城域网中,数字电视机顶盒在对节目进行下载的时候,拥有这个节目资源的机顶盒的数量也许有很多个,怎么才能找到最近的一个机顶盒进行节目的下载是网络负载均衡中比较重要的问题,也是文中的研究目的。文中需要建立一个网络拓扑结构,给对应的机顶盒分配相应的 IP 地址,将这些已知的信息存放到数据库中,使用 VS2010 软件进行编程,在具体实现过程中运用到了数据挖掘中的 k 中心点算法,最终找到距离最近的机顶盒的地址下载目标资源。

关键词:网络负载均衡;数据库;VS2010;k 中心点算法
中图分类号:TP301.6 **文献标识码**:A **文章编号**:1673-629X(2014)04-0122-04
doi:10.3969/j.issn.1673-629X.2014.04.031

Research of TOPO Server Algorithm Based on K-medoids Algorithm

WANG Mei, LI Xiao-feng, MENG Ling-jun, ZHANG Li-jun
(College of Computer Science and Technology, Shandong Jianzhu University, Jinan 250101, China)

Abstract: In a metropolitan area, when the set-top box in digital TV downloads program, there may be several set-top boxes which have the program resources. How to find the nearest one is one of the most important issues in the network load balancing, also it is the purpose in this study. First need to set up a network topology, set-top box should have the corresponding IP address. These known information should be stored in the database, using VS2010 software for programming. K-mediods algorithm is applied in the realization of the processing. At last, find the nearest address to download the target resource.

Key words: network load balancing; database; VS2010; k-mediods algorithm

0 引 言

在数字电视中,机顶盒的使用范围变得越来越广,数字电视的应用技术也越来越成熟[1]。但是在一个 DHCP 服务器中,利用机顶盒在下载节目资源的时候,怎么才能找到最近的那个拥有资源的节点进行资源的下载,这在网络负载均衡中是一个相当重要的问题。在一个既定的网络拓扑结构中,可以对网络中的拓扑结构进行深入的分析,运用某些算法,找到最适合下载资源的节点。可以对网络中拓扑结构进行具体化,给网络中的每一个节点都分配一定的 VLAN 号进行唯一的标识,并给网络节点分配特定的网络地址[2]。对于每个终端节点,按照一定标准对网络拓扑结构中的位置分配四位的路径标号。然后利用数据挖掘中的 k 中心点算法对数据库中存储的四维数据的操作及处理,对数据库中的节点的网络地址及路径信息进行分析,进而找到下载资源的最邻近的位置节点。

1 网络中拓扑结构分布

在城域网中部署一台 DHCP 服务器,服务器中的每个机顶盒都对应有分配标识的 VLAN 号[3]。在服务器中,根据每个机顶盒接入的 VLAN 号的不同,分配一个 IP 地址。在对应的子网范围内应该有许多空闲的 IP 地址,从中随机地选择一个进行动态分配。机顶盒使用 USB 技术扩展了移动硬盘并将其作为存储器,进一步集成了 P2P 客户端软件,而客户端软件之间进行视频节目的 P2P 下载,进而实现高清视频节目分发的目的[4]。

每一个机顶盒用户,如果下载节目资源,需要接入核心路由交换机集群,中间需要使用多级交换机。如图 1 所示,中间圆形 CR 标识节点为交换机的集群,图中的 S 节点代表交换机,A ~ V 代表终端节点,由交换机集群、交换机及终端构成树形的城域网树状结构。每个接入交换机下的用户组属于同一个子网,并且这

收稿日期:2013-07-11 **修回日期**:2013-10-19 **网络出版时间**:2014-01-28
基金项目:济南市科技成果转化项目(201211004)
作者简介:王 美(1986-),女,山东德州人,硕士研究生,研究方向为流媒体;李晓峰,博士后,教授,研究方向为流媒体。
网络出版地址:http://www.cnki.net/kcms/detail/61.1450.TP.20140128.1132.008.html

和城域网的 VLAN 号是一一对应的,这样就分别形成接入网 VLAN 以及子网 A ~ V。

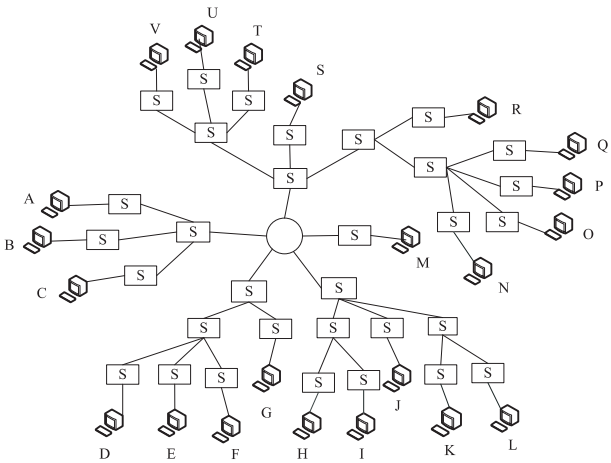

图 1　网络拓扑结构分布

资源分配基本原理:为了能够使得机顶盒找到所需要下载目标资源所在的位置节点,城域网研发了一个用于查询资源的索引服务器 INDEX。各个机顶盒开机后,把各自所拥有的节目情况以及目前的 IP 地址都发送至 INDEX 服务器。即使机顶盒下载了部分节目资源,也会随时向 INDEX 服务器进行报告,这就使得 INDEX 服务器对当前各个机顶盒所拥有节目资源的情况了如指掌。如果一个机顶盒需要进行节目的下载,它就会向 INDEX 服务器发送一个查询资源信息的请求,然后 INDEX 服务器就会查找出所有拥有该资源的机顶盒。

由子网 A ~ S 形成的接入网的网络地址设置如表 1 所示。

如图 1 所示,VLAN 就是网络拓扑结构中的终端,ID 是网络终端的标识,网络地址是在子网范围内随机分配的,掩码代表的是掩码的位数。其实想要下载资源,仅仅知道拥有资源的机顶盒是不够的,为了节省带宽,需要在这些拥有资源的机顶盒中找出与请求资源的机顶盒最邻近的节点。在此基础上,开发了拓扑服务器 TOPO,从而就可以掌握全局城域网的拓扑结构,如:交换机布局、IP 地址分布。这样的话,机顶盒在请求下载资源时,INDEX 服务器在找到所有拥有这个资源的机顶盒节点后,就会把请求下载机顶盒资源的 IP 地址及拥有资源的机顶盒的 IP 地址都同时发送给 TOPO 服务器,然后拓扑服务器找出与请求下载资源的机顶盒最近邻的机顶盒,把这个机顶盒 IP 地址返回给 INDEX 服务器,然后 INDEX 服务器就会返回给请求下载资源的机顶盒。最后,机顶盒从它最邻近的机顶盒进行资源下载。调用的接口采用的方式是文本交互,第一步输入请求下载资源机顶盒的 IP 地址,第二部顺序输入各个拥有目标资源的机顶盒的 IP 地址。通过进一步计算后,再输出最邻近的节点的 IP 地址。

表 1　网络地址表

VLAN	VLANID	网络地址	掩码
A	1025	10.5.4.0	24
B	1026	10.5.5.0	24
C	1027	10.5.8.0	24
D	1028	10.5.24.0	24
E	1029	10.5.26.0	24
F	1030	10.5.25.0	24
G	1031	192.168.36.0	22
H	1032	10.5.20.0	25
I	1033	10.5.20.128	25
J	1034	10.5.21.0	24
K	1035	10.5.22.0	24
L	1036	10.5.23.0	24
M	1037	10.5.23.0	24
N	1038	10.5.27.192	26
O	1039	10.5.27.64	26
P	1040	10.5.27.0	26
Q	1041	10.5.27.128	26
R	1025	10.5.29.0	24
S	1043	192.168.20.0	23

算法设计思想:首先需要输入请求资源的 IP 地址,然后系统根据 k 中心点算法计算出这个 IP 地址在网络拓扑结构中所在的网址;然后再分别输入拥有资源的节点的 IP 地址,分别计算出拥有资源的节点的 IP 网址,将拥有资源的 IP 网址分别与请求资源的 IP 网址的距离进行比较;首次比较时,将第一个差值赋给最优路径值,进一步计算下一个拥有资源的 IP 网址与请求资源的 IP 网址距离,与最初的最优路径值进行比较,如果距离值更小,则赋值给最优路径值,如果不是更小,则不输出,不赋值,循环进行比较,直到比较结束。输出路径距离值最小的那个下载资源节点的相关信息,包括是哪个节点,IP 地址是多少等等。

因为网络拓扑结构中的信息比较多,为了方便起见,选择使用 Microsoft SQL Server 2008 数据库,从而将已经知道的信息都存储在数据库中,为进一步的算法设计及实现做准备。

在 A ~ S 子网中,节点的分布已经是固定好了的,但是在算法实现过程中,节点的路径及编码信息的确定的方法需要进一步研究分析。分析如下:首先需要对网络拓扑结构进行处理,需要对 A ~ V 的子网进行编码,方便进行比较。如果要计算彼此之间的路径值,则需要对子网中的这些 A ~ S 网络节点进行详细的编码。编码规则如下所示:

编码位数为四位,最高位代表距离集群最近的路径值,在四位数中的权值最大;最低位,也就是最右位代表距离集群位置最远,权值最小。为了使用 k 中心点算法进行思维数据的比较,不满四位的编码值在低

位上自动补 0。进行编码的节点的位置不同,路径值也不同,编码值就不相同,这在算法的使用及其实现过程中是非常关键的。详细规则如下:距离集群的距离只有 1 的节点的路径的权值从左到右顺序编码为 0,1,2,3,4。对第一次节点下属的分支节点的编码规则都是如此。以 A、D、P 节点为例,A 节点路径长度为 2,第一条路径编码值为 0,第二条路径编码也是 0,这样编码路径长度是 2,那么 A 节点的路径编码值是 0000;D 节点的第一条路径的编码值是 1,第二条路径编码值是 0,第三条路径的编码值是 0,那么 D 节点的路径编码值是 1000;P 节点的第一条路径的编码值是 4,第二条路径编码值是 0,第三条路径的编码值是 0,第四条路径的编码值是 2,那么 P 节点的路径编码值是 1000。综上所述,A ~ S 的网络拓扑结构的编码结果如下所示:

A:0000	B:0100	C:0200	D:1000
E:1010	F:1020	G:1100	H:2000
I:2010	J:2100	K:2200	L:2210
M:3000	N:4000	O:4001	P:4002
Q:4003	R:4010	S:4100	

2 k 中心点算法

K-means 算法对孤立点是非常敏感的,一个具有极大值的对象会在一定程度上扭曲数据的分布情况[5]。想要修改算法进而消除这种情况的敏感性,可以不使用簇中所有对应的平均值当做参照点,可以选择一个在簇中最中心的位置对象,也就是 mediod[6]。这样的划分方法依旧是采用最小化所有对象和参照点之间的相异程度之和的原理来实现的。这也是 k-medoids 方法的理论基础[7]。

2.1 k 中心点算法的基本策略

首先在每个簇的所有对象中任意选择一个对象作为代表;剩下的对象根据它与代表对象之间的距离分配给距离最近的一个簇[8]。继而反复地使用非代表对象取代代表对象,从而达到改进聚类质量的目的。聚类结果的质量好坏用一个代价函数来进行估算,这个代价函数评估了对象与其参照对象之间的平均相异程度。为了可以准确地判定任意一个非代表对象 O_{random} 是否为当前的这个代表对象 O_j 的更好的替代,每一个非代表对象 p 都需要考虑下面这四种情况[9]:

第一种情况:对象 p 当前隶属于代表对象 O_j。假设 O_{random} 替代 O_j,且对象 p 与 O_i 的距离是最近的,且 i 与 j 不相等,那么对象 p 就会被重新分配给对象 O_i。

第二种情况:对象 p 当前隶属于代表对象 O_j。假设 O_{random} 替代 O_j,并且对象 p 与对象 O_{random} 的距离是最近的,那么对象 p 就会被重新分配给对象 O_{random}。

第三种情况:对象 p 当前隶属于代表对象 O_i,且 i 不等于 j。如果对象 O_j 被对象 O_{random} 代替,但是对象 p 还是和 O_i 的距离是最近的,这种情况下,对象的隶属关系是没有变化的。

第四种情况:对象 p 当前隶属于代表对象 O_i,且 i 不等于 j。如果 O_j 被 O_{random} 代替,且 p 离 O_{random} 最近,那么 p 被重新分配给 O_{random}。

每当重新分配发生时,square - error 所产生的差别对代价函数有影响。因此,如果一个当前的代表对象被非代表对象所代替,代价函数计算 square - error 值所产生的差别。替换的总代价是所有非代表对象所产生的代价之和。如果总代价是负的,那么实际的 square - error 将会减小,O_j 可以被 O_{random} 替代。如果总代价是正的,则当前的代表对象是可接受的[10]。

2.2 PAM

PAM(Partitioning Around Medoids,围绕 k-medoids 的划分)算法是最早提出的 k-medoids 算法之一[11]。PAM 的思想是对 n 个对象给出 n 个划分。算法开始时随机选出 k 个代表对象,然后该算法反复地查找,直到能够找到更好的代表对象替换。在此期间,所有可能的对象对都会被分析,在每个对中,其中的一个对象被当成代表对象,而另一个不是代表对象[12]。对所有可能存在的各种组合,都要对聚类结果的质量进行估算。对于任何一个对象 O_j 来说,它可以被 square-error 值减少最大的那个对象所替代。在上一次迭代中所产生的最佳对象的集合将会成为下一次迭代的代表对象[13]。如果 n 和 k 的值都比较大时,这种迭代方法的计算代价就会相当高。如果存在“噪音”或者孤立点数据的时候,k-medoids 方法要比 k-means 方法健壮一些,原因是 medoid 不会像平均值那样容易受到极端数据的影响。然而,从执行代价上来看,k-medoids 方法要比 k-means 算法高不少。除此之外,这两种方法都要求将结果簇的数目定为 k 个[14]。

算法:k-medoids。

输入内容:结果簇的数目 k,以及含有 n 个对象的数据库;

输出内容:k 个簇,并且要使所有对象与它最近代表对象的相异程度总和最小。

具体方法如下:

(1) 任意选择 k 个对象当作初始的代表对象;

(2) repeat;

(3) 将每个剩余的对象指派给距离它最近的代表对象所代表的簇中;

(4) 任意选择一个非代表对象 O_{random};

(5) 对用 O_{random} 代替 O_j 的总代价 S 进行计算;

(6) 假如 S 小于 0,则 O_j 被 O_{random} 替换,形成有 k 个代表对象的新的集合;

(7) 直到不发生任何变化。

在使用 k 中心点算法进行程序设计时,首先要找出请求资源的 IP 的网址,然后需要计算出有目标资源的 IP 的网址,拥有资源的 IP 网址也许有好几个,怎么找到最近的是一个非常重要的问题,利用 k 中心点算法对 IP 地址的四维数据进行比较,使得差值比较容易区分。

3　PAM 在 TOPO 服务器中的实现

因为在算法实现过程中,利用了 PAM 对 IP 地址及网络拓扑结构中的节点的四维数据进行处理,所以数据的处理过程需要使用到数据库。下面首先对算法实现的流程进行说明,然后对文中所用数据库的设计进行介绍。

3.1　算法设计流程

算法流程基本如下所示:

(1)首先需要在系统中输入需要请求资源的节点的 IP 地址,利用 PAM 算法求出这个节点在网络拓扑结构中对应位置的网络地址。

(2)然后需要分别输入拥有请求资源的目标节点的 IP 地址,分别计算出这些目标节点在网络拓扑结构中的网络地址。

(3)分别计算出请求资源的 IP 网络地址与拥有资源的 IP 网络地址的距离差值,第一个进行比较的自动赋给最优路径值,顺序比较中的差值如果比这个最优值小,则替代最优值,直到比较结束后,输出差值最小的那个拥有资源的节点的标识号、IP 地址等信息。

3.2　数据库设计

数据库中的表只有一个,数据库中的表命名为 Infor,数据库表结构中的属性有 VLANID(主键),VLAN,First,Second,Third,Fourth,$Source_1$,$Source_2$,$Source_3$,$Source_4$,maskcode。VLANID 是主键,定义类型是 nvarchar,VLAN 是已经分配好的节点的标志号,定义类型也是 nvarchar。First,Second,Third,Fourth 为获取的 IP 地址的字段值,定义类型都为 int。$Source_1$,$Source_2$,$Source_3$,$Source_4$ 是网络中 A ~ V 节点拓扑结构的编码位值,定义类型是 int。

3.3　算法实例分析

为了进一步验证算法实现过程中节点的处理情况,可以参照以下这两个实例进行详细的分析:

(1)请求下载资源的机顶盒的 IP 地址是 10.5.27.237,有 4 个机顶盒有目标资源,其 IP 是 10.5.29.7、10.5.5.238、192.168.39.42、10.5.23.53;在程序设计界面中"请输入请求资源的 IP 地址:"处输入机顶盒的 IP 地址,在"请输入有目标资源的 IP 地址:"处输入拥有资源的 IP 地址。

拥有资源的 IP 网址显示空间的完整内容如下:

拥有资源的 IP 地址是:10.5.29.7

IP 网址是:

10.5.29.0 VLAN:R VLANID:1042

拥有资源的 IP 地址是:10.5.5.238

IP 网址是:

10.5.5.0 VLAN:B VLANID:1026

拥有资源的 IP 地址是:192.168.39.42

IP 网址是:

192.168.36.0 VLAN:G VLANID:1031

拥有资源的 IP 地址是:10.5.23.53

IP 网址是:

VLAN:L VLANID:1036

(2)请求下载资源的机顶盒的 IP 地址是 10.5.22.163,有 3 个机顶盒拥有目标资源,其 IP 地址分别是 10.5.27.193、10.5.27.32、10.5.27.142。

拥有资源的 IP 网址显示空间的完整内容如下:

拥有资源的 IP 地址是:10.5.27.193

IP 网址是:

10.5.27.192 VLAN:N VLANID:1038

拥有资源的 IP 地址是:10.5.27.32

IP 网址是:

10.5.27.64 VLAN:O VLANID:1039

拥有资源的 IP 地址是:10.5.27.142

IP 网址是:

10.5.27.128 VLAN:Q VLANID:1041

从以上系统测试的结果可看出,系统计算的结果与人为在网络拓扑结构中看到的节点位置关系一致。

4　结束语

在数字电视的使用过程中,机顶盒所起的作用是至关重要的。机顶盒在下载资源的时候,网络负载均衡问题也是不容忽视的。文中对在 DHCP 服务器中,机顶盒在下载节目资源时怎么才能找到最近的拥有资源的终端节点下载资源这个网络负载均衡的问题进行分析与研究。文中采用的编程工具是 VS2010,使用的编程语言是. net,使用的数据库是 SQL Server2008,在实现过程中结合了数据挖掘中的 k 中心点算法,在机顶盒的应用中具有一定的理论意义和实践意义。

参考文献:

[1]　毕厚杰. 新一代视频压缩编码标准 H.264[M]. 北京:人民邮电出版社,2004.

(下转第 130 页)

上,文中的方法除了低于 SRC(l_1_l_s)方法外,都远优于其他的方法。而且 RTS-CR 方法提高了 CR 的识别率近 4% ~18%,几乎超过了优秀的 SRC(l_1_l_s)算法。而且 SRC(l_1_l_s)[4]在优化目标函数时需要计算 L_1 范数,所以计算复杂度高。而文中的方法只需要计算简单的线性加和欧几里得距离,故在计算复杂度上要远低于 SRC(l_1_l_s)。

总之,从上述实验中可以看出 RTS-CR 方法在识别率上或者计算复杂度上都远优于同类的方法,可以应用于一些实际的人脸识别系统。

5 结束语

文中提出了一种为协同表示(CR)构造最优字典的方法。实验验证了文中的最优字典构造方法不仅大幅度提高了 CR 方法的识别率,而且优于同类算法和优秀的 SRC(l_1_l_s)算法。在以后的研究中,需要考虑如何为测试样本自动地选择最适合的编码字典。

参考文献:

[1] Turk M, Pentland A. Eigenfaces for recognition[J]. Cognitive neuroscience, 1991, 3(1): 71-86.

[2] Belhumeur P N, Hespanha J P, Kriegman D J. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection [J]. IEEE transactions on pattern analysis and machine intelligence, 1997, 19(7): 711-720.

[3] Bartlett M S, Movellan J R, Sejnowski T J. Face recognition by independent component analysis [J]. IEEE transactions on neural networks, 2002, 13(6): 1450-1464.

[4] Wright J, Yang A Y, Ganesh A, et al. Robust face recognition via sparse representation[J]. IEEE PAMI, 2009, 31(2): 210-227.

[5] Yang A Y, Ganesh A, Zhou Z H, et al. Fast l1-minimization algorithms and application in robust face recognition [R]. Berkeley: UC Berkeley, 2010.

[6] Zhang Lei, Yang Meng, Feng Xiangchu. Sparse representation or collaborative representation: Which helps face recognition? [C]//Proc of ICCV. [s. l.]: [s. n.], 2011.

[7] Deng Weihong, Hu Jiani, Guo Jun. Extended SRC: Under sampled face recognition via infraclass variant dictionary [J]. IEEE transactions on pattern analysis and machine intelligence, 2012, 34(9): 1864-1870.

[8] Thian N P H, Marcel S, Bengio S. Improving face authentication using virtual samples[C]//Proceeding of the 2003 IEEE international conference on acoustics, speech, and signal processing. [s. l.]: [s. n.], 2003: 6-10.

[9] Yeon-Sik R, Se-Young O. Simple hybrid classifier for face recognition with adaptively generated virtual data[J]. Pattern recognition letters, 2002, 23: 833-841.

[10] Beymer D, Poggio T. Face recognition from one example view [C]//Proc of ICCV. [s. l.]: [s. n.], 1996.

[11] Vetter T. Synthesis of novel views from a single face image [J]. International journal of computer vision, 1998, 28(2): 103-116.

[12] Jung H C, Hwang B W, Lee S W. Authenticating corrupted face image based on noise model [C]//Proceedings of the sixth IEEE international conference on automatic face and gesture recognition. [s. l.]: [s. n.], 2004.

[13] Tang B, Luo S W, Huang H. High performances face recognition system by creating virtual sample [C]//Proceedings of the 2003 international conference on neural networks and signal processing. [s. l.]: [s. n.], 2003: 14-17.

[14] Tan Xiaoyang, Chen Songcan, Zhou Zhihua, et al. Face recognition from a single image per person: A survey [J]. Pattern recognition, 2012, 39(9): 1725-1745.

(上接第 125 页)

[2] 江 勇,林 闯,吴建平. 网络传输控制的综合性能评价标准[J]. 计算机学报,2002,25(8):869-877.

[3] 崔 勇,吴建平,徐 恪. 互联网络服务质量路由算法研究综述[J]. 软件学报,2002,13(11):2065-2075.

[4] Wang F, Gao L. Inferring and characterizing internet routing policies[C]//Proc of ACM SIGCOMM internet measurement conference. [s. l.]: [s. n.], 2003.

[5] Han Jiawei, Kamber M. 数据挖掘概念与技术[M]. 北京:机械工业出版社,2007.

[6] Yan Y, Han J. gSpan: Graph-based substructure pattern mining[C]//Proc of ICDM. [s. l.]: [s. n.], 2002.

[7] 陈京民. 数据仓库与数据挖掘技术[M]. 北京:电子工业出版社,2002.

[8] Karyp G, Han E H. A hierarchical clustering algorithm using dynamic modeling[J]. Compute, 1999, 32(8): 68-75.

[9] 肖海军,洪 帆,张昭理,等. 基于融合分类和支持向量机的入侵检测研究[J]. 计算机仿真,2008,25(4):130-132.

[10] 孙 庚,冯艳红,郭显久,等. K-means 聚类算法研究[J]. 长春师范学院学报(自然科学版),2011,30(1):1-4.

[11] 邵峰晶,于忠清. 数据挖掘原理与算法[M]. 北京:中国水利水电出版社,2003.

[12] Holder L B, Cook D J, Djoko S. Substructure discovery in the subdue system[C]//Proc of AAAI workshop on knowledge discovery in database. Seattle, WA: AAAI Press, 1994: 169-180.

[13] 毛国君. 数据挖掘原理与算法[M]. 北京:清华大学出版社,2005.

[14] 张春阳,周继恩,刘贵全,等. 基于数据仓库的决策支持系统的构建[J]. 计算机工程,2002,28(4):249-251.