SmartCache：基于兴趣的协作式 Web 缓存*)

陈海涛　卢宇彤　黄遵国
（国防科技大学计算机学院　长沙 410073）

摘　要　发现并验证了 Web 访问的局部性原理，在其基础上提出一种全新的集中分类学习但分布协作运行的基于 P2P 的客户端缓存模型——SmartCache。SmartCache 采取集中方法学习节点访问兴趣并按照访问兴趣对节点分类，节点之间按照兴趣分类组成非结构化的对等缓存网络，缓存网络的运行是完全分布的。仿真实验表明，SmartCache 以低代价实现了高缓存命中率，能够有效提高缓存容量和访问速度。

关键词　Web 缓存，分类，协作，对等网络，兴趣社团

SmartCache: Interest-based Cooperative Web Cache

CHEN Hai-tao　LU Yu-tong　HUANG Zun-guo
(School of Computer, National University of Defense Technology, Changsha 410073, China)

Abstract　Discoveried and verified a locality principle in Web access pattern. Then presented a new client-based cache model—SmartCache, in which clients collaborate into unstructured peer-to-peer cache network according to access interests. SmartCache gains high hit rate with low consumption.

Keywords　Web cache, Classification, Cooperation, Peer-to-peer network, Interest community

1　引言

随着 Internet 的发展，特别是以 HTTP 协议为基础的 WWW 的发展，使得 Web 信息量和访问量剧增，从而导致 Web 服务面临以下问题：1）Web 服务器常成为访问瓶颈，面临严重的过载问题。2）局域网内的客户端和广域网上服务器之间的通信延迟比较大，导致用户满意度降低。3）局域网的出口带宽受限，往往成为用户的访问瓶颈。很多网络技术致力于解决这 3 个问题，例如 Web 集群、镜像服务器、CDN、Web 缓存等。其中 Web 缓存机制是一种广泛使用的能够同时减少网络服务瓶颈，降低网络流量，减少服务延迟的有效方法。现有的 Web 缓存机制主要集中于广域网上的 cache 服务器集群和局域网内的缓存服务器的调度机制研究。

P2P 技术作为一种革命性的技术，致力于合理高效地组织利用 Internet 边缘上的大量分布的计算、存储、信息等资源，充分释放互联网的蕴含的巨大边缘资源。本文发现并验证了 Web 访问的局部性原理，在其基础上提出一种全新的集中学习分类但分布协作运行的基于 P2P 技术的客户端缓存模型——SmartCache。

2　相关工作

传统的 Web 缓存技术按照参与节点可分为 3 种。第一种是位于广域网上的缓存服务器集群，包括 cache 阵列和 CDN 技术等，该领域的研究侧重缓存服务器之间的协作模型和协议[1,2]；第二种是局域网内的缓存服务器，该领域研究侧重于缓存服务器本地的缓存调度机制和缓存本地查找算法[3]；第三种基于客户端的 Web 缓存，研究重点在于缓存的分布查找算法[4-6]。前两种技术研究比较成熟，第三种技术还有较大的发展空间。

局域网内集中式的 cache 服务器虽然管理简单，但是由于其存储能力、计算能力、网络带宽的限制，容易成为性能瓶颈，尤其是在现在 Web 访问中包含大量多媒体信息的情况下。相比之下，局域网内客户节点之间的带宽是各节点能分到的广域网带宽的百倍甚至更高，访问延迟可以是广域网访问延迟的十分之一左右，而且客户节点的聚合存储能力、计算能力、网络带宽与客户节点的数量是线性增长关系，也就是说客户节点的聚合能力与客户节点的聚合 Web 服务需求是同比增长的，于是采用客户节点本身作为缓存集群，成为一种有前景的研究方向。当前该方向的研究主要包括以下成果。

CoopNet[4] 要求 Web 服务器登记愿意作为 cache 的客户节点，对于以后的访问节点返回一个重定向节点列表。该客户节点对可用节点进行测试选择最近客户节点请求服务，请求失效后转向服务器节点。该方法最大的局限性在于该缓存只对单一站点访问有效，并且对于服务器和客户端都不透明，要求用户为特定站点的访问安装 IE 插件。

BuddyWeb[5] 将 Web 客户端组织为非结构化的对等网络。节点的 Web 访问请求首先通过移动 Agent 在本地局域网内部节点的缓存中搜索，本地搜索成功率可以达到 15%～35%。BuddyWeb 采取基于节点缓存内容相似度的路由策略，基于客户节点的兴趣词列表计算节点兴趣之间的相似度，查询首先被路由到和该节点具有最高相似度的相邻节点。该方法需要一组服务器收集节点的兴趣词列表，并且要集中计算各客户节点的访问相似度，其基于向量空间的相似度计算方法比较复杂，计算开销大。此外该方法搜索跳数不固定，有时会比较高。

文献[6]提出将客户端组织为 DHT 网络作为缓存，客户节点加入 DHT 网络，承担 DHT 网络维护带来的较大开销，与减少带宽消耗的设计目的相矛盾，Web 缓存的数量巨大，

*)863 项目(2006AA01Z401)。**陈海涛**　博士，助理研究员，研究方向为分布资源管理、网络安全、网络计算；**卢宇彤**　研究员；**黄遵国**　副研究员。

更新频繁，在 DHT 网络中的发布要消耗大量的计算和网络开销；Intranet 中节点的加入和退出是比较频繁的，造成 DHT 网络的结构维护开销大。

3 SmartCache

3.1 设计思路

要设计有效的基于客户端的 Web 缓存，需要解决两个挑战：A）如何组织客户端节点，在客户端之间共享缓存？ B）如何实现分布 Web 缓存的高效搜索？

现有的基于客户端的分布式缓存设计方案或者只能适应特定网站[4]，或者需要复杂的计算和配置[5]，或者参与节点的开销太大[6]，在实际应用中面临困难。本文试图给出一种保持集中式 Web 缓存服务器的简单管理特性，但是更健壮并且无访问瓶颈，同时又是简单有效的基于客户端的分布式缓存设计方案。

对于挑战 A 的解决方案：将客户端组织为非结构化的 P2P 网络，以 P2P 的模式共享缓存。有集中服务器的类似 Napster 的结构存在性能瓶颈和单一失效点。结构化的 P2P 网络将导致大的节点维护开销和内容发布开销，尤其不适合频繁而且文件数量巨大的 Web 内容。

对于挑战 B 的解决方法：学习节点的兴趣相似度将节点按照兴趣分类，搜索限制在兴趣类似节点之间。本文在 Web 访问领域发现一个局部性定律：如果节点 A 和 B 曾经访问过许多相同 Web 内容，那么很可能节点 A 和 B 即将访问的内容也有很多重合之处。后续部分的测试将充分验证该局部性定律。从另外一个角度看，访问过相同网站可能代表两个节点的访问兴趣有一定的相似度。也就是说，该定律提供了一种计算节点之间兴趣相似度的简单有效的新方法。该定律是 SmartCache 的基本设计依据，说明曾经访问过相同的文件是缓存搜索可利用的重要信息，可以根据不同节点访问 Web 的历史信息构造朋友关系，朋友很少的节点的 Web 缓存能够以高概率满足 Web 访问请求。利用该局部性定律需要解决两个问题：节点间兴趣相似度的计算；基于相似度的节点缓存共享算法。

3.2 体系结构设计

SmartCache 采取集中和分布相结合的混合体系结构，示意图如图 1。节点兴趣相似度的计算是集中进行的，是最经济和便于管理的方式。缓存管理服务器记录网络内节点的历史搜索请求，定期离线计算客户节点之间的兴趣关联，在具有相似访问兴趣的节点间建立朋友关系，并将朋友关系分发到普通客户节点。当前绝大多数 Intranet 都有防火墙或者 Web 缓存代理记录 Web 访问日志，所以缓冲管理服务器定期下载 Web 访问日志即可。节点的缓存互助采取分布结构，消除了集中式结构的瓶颈效应。客户节点按照朋友关系组成无结构的对等网络，并在对等网络中共享自己的本地 Web 缓存，其 Web 搜索请求首先在对等缓存网络内搜索，失效情况下才到广域网内查找。

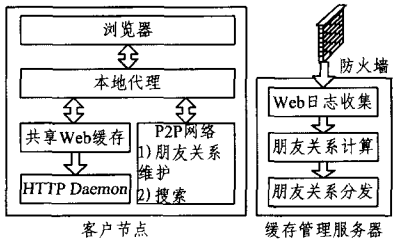

图 1 SmartCache 的体系结构

3.3 离线朋友关系计算

朋友关系构造基本算法流程如下：1）计算客户节点之间的 Web 请求的重叠次数；2）对于每个客户节点，按照其它客户节点历史搜索请求与其重合次数从高到低排序；3）选择排名前 K 个节点作为自己的朋友节点。修改朋友节点的选择标准，可以有一些变种算法。例如考虑 URL 访问的稀有度，优先选择于自己共同访问比较稀有文档的节点作为朋友节点，这样可以提高稀有 URL 访问的本地搜索成功率。

缓存服务器运行离线朋友关系计算程序分为两个时机：或者根据预配置的信息定期运行；或者当客户节点发现自己的朋友关系的有效程度太低，主动向缓存服务器申请重新计算自己的朋友节点。

3.4 分布缓存搜索算法

客户节点的访问流程如下：

1）客户节点将本地缓存中所有内容组织成哈希链表，确保在 1－2 跳内能完成本地搜索；

2）节点将自己的访问请求哈希成定长的二进制字符串；

3）节点将哈希后的搜索请求采用 UDP 协议发送到自己的 m 个朋友节点。m 是一个可调参数，模拟测试表明通常 m 为 3－6 就足够了；

4）客户节点的朋友节点请求收到其请求后进行本地搜索，然后将搜索结果返回给搜索节点；

5）如果客户节点收到成功返回，则向内部客户节点请求所需内容，若有多个返回结果，则向时间最新的节点请求内容；

6）若局域网搜索失效，则向 Web 缓存服务器发出请求，如果没有 Web 缓存服务器，则直接向原始 Web 服务器请求；

7）客户节点判断请求内容，如果是动态产生内容，则不缓存，静态内容则存储在本地缓存，并登记访问时间。

4 理论分析和功能扩展

缓存的容量分析。单一缓存服务器的容量为固定值 C_s，每个客户端缓存的容量为 C_c，客户端的数量为 N，则客户端分布式缓存和集中缓存的大小比例为 NC_c/C_s。当 N 足够大，客户端分布缓存容量将远大于集中缓存服务器的缓存容量，相应地其命中率将更高。而且客户端分布缓存的容量随节点数量 N 增加而线性增长，具有很好的可伸缩性。

响应速度分析。这里只讨论缓存命中页面的响应速度。响应速度＝硬盘命中率×硬盘响应速度＋内存命中率×内存响应速度。假设缓存服务器和客户端需要相同时间在本地找到所需内容，所有内容的索引都置于内存中，命中位置分为内存命中和硬盘命中（缓存服务都会将部分常用内容置于内存中）。硬盘命中的响应速度要比内存命中慢百倍以上，故响应速度主要由内存命中率和硬盘命中率决定。对于集中缓存服务器，内存的容量是固定的，必然将大部分内容置于硬盘上。对于 Web 客户端组成的分布缓存，各客户端缓存内容有限，其本地缓存内容可以完全置于内存中。如果集中缓存内存命中率为 30%，对等缓存网络内存命中率为 90%，硬盘和内存响应速度比为 100，则估算集中缓存服务器与对等缓存网络响应时间比为（30%×1＋70%×100）/（90%×1＋10%×100）≈7。显然，对等缓存网络命中内容的响应速度要远高于集中缓存服务器。

Flash Crowd 的抵御是 Web 缓存服务面临的一个挑战。当发生 Flash Crowd 时，大量用户密集用户访问特定范围内

容，尤其是用户下载较大文件时，传统的集中缓存服务器将面临严重的服务过载。在 SmartCache 模型中，负载是均分在各客户节点，该问题可以得到有效缓解。为了更好地解决 Flash Crowd 问题，我们提出一种基于小世界网络的方法[7]。当前按照朋友关系组成的对等缓存网络的聚集系数高，但是网络直径长，不符合小世界网络的特征[7]。通过在每个节点的朋友关系中保留一个随机节点，则该随机连接使对等缓存网络转换为小世界网络。当 Flash Crowd 发生的时候，由于改造后的对等缓存网络直径很小，引起雪崩访问的内容能够在很短的步长内扩散到整个局域网，大大减少对外流量的消耗。访问内容在对等缓存网络中扩散时间为 $T_C=T_0+T_1\times D$，其中 D 为网络直径，通常为 5～8；T_0 为第一个节点从广域网原始服务器下载该内容到局域网所需时间，T_1 为局域网内客户节点之间单次下载时间。因为对等缓存网络的网络直径 D 很小，而局域网内的下载时间 T_1 也比较短，所以 T 也很小。该 Flash Crowd 过程对广域网特定内容的总体下载流量需求最小可以为单次下载。对于较大文件，允许客户节点从多个朋友节点分块下载，扩散时间会更短。对于集中缓存，访问内容扩散到所有节点时间为 $T_S=T_0+N\times C/B$，其中 C 为热点内容的大小，B 为缓存服务器的带宽，可以认为 $T_s\approx T_0+N\times T_1$。这里对集中缓存服务器的扩散时间计算已经是高估，因为大量节点的集中访问将使缓存服务器过载甚至不能对外提高服务，则扩散时间比为 $T_s/T_C\approx(T_0+N\times T_1)/(T_0+D\times T_1)/\approx N/D$。如果 N 为 500，D 为 10，则集中缓存的扩散时间为对等缓存网络的扩散时间慢 50 倍。

SmartCache 对于几个缓存设计的传统问题提供以下解决方案：

• 敏感内容的保护。客户对于密码等相关内容不加入缓存；对于通过安全加密链接获取的内容不加入缓存。

• 动态网页的处理。对于动态网页的框架不缓存，但对于动态网页内部包含的静态内容如图片、flash 广告等进行缓存，据统计这样能减少 80%的流量。

• 缓存的时间问题。客户节点对所有的缓存内容登记其访问时间，当向其他客户节点请求时选择最新的版本。此外，客户节点向原始服务器发送时间请求，可以判断内容是否修改。

• 节点在线互助实现实时流媒体的缓存。朋友节点可以直接将自己在观看的媒体流转发给请求的客户节点。

5 仿真测试

本文采用了 3 组得到广泛应用、分别代表不同规模网络访问情况的 Web 访问日志来测试 SmartCache 算法。Boston[8]是波士顿大学的 Web 日志数据，包括 558261 条访问条目，节点有 538 个。Berkeley[8]是伯克利大学的 Web 日志数据，包括 1703836 条访问条目，节点共 5222 个。Boeing[8]是波音公司的 Web 日志数据，包括 4421526 访问条目，节点共 28895 个。将 Web 访问日志按时间分为多块，缓存服务器根据前面的时间块计算出节点间的朋友关系，然后将后面的访问日志回放测试命中率，其中每个节点有 5 个朋友节点。

图 2 显示 SmartCache 算法的命中率可以达到 66%～72%。同时算法在三组不同规模的网络测试数据中获得相似性能，说明了 SmartCache 算法的稳定性。相比于一般的集中 Web 缓存服务器，通常只能获得低于 40%的命中率，SmartCache 将所有客户节点的缓存聚集为一个巨大容量的缓存，从而较大地提高了缓存命中率。该对等缓存网络的缓存能力能够随着客户节点的数量而增加，所以具有良好的可伸缩性，能够在不同规模的局域网中获得相似的性能。集中的缓存服务器的命中率却对客户节点数量、缓存服务器的配置很敏感。

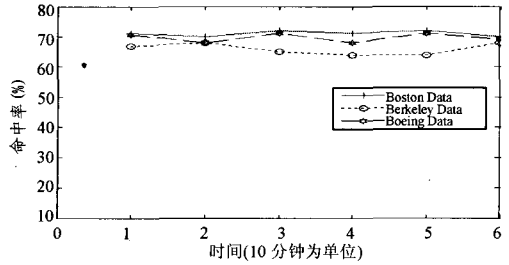

图 2 SmartCache 命中率

既然单跳的朋友关系可以获得如此的命中率，那么猜想提高朋友关系的层数是否可以进一步提高命中率？表 1 比较了多跳的 SmartCache 算法和 BFSFLood[9]算法在 Boeing 数据中的测试性能。BFSFlood 随机选择一组固定数量的邻居节点转发搜索请求，是 Gnutella 的一个升级版本。

表 1 SmartCache 与 BFSFlood 的比较

		第一跳	第二跳	第三跳	第四跳
SmartCache (Boeing 数据)	命中率(%)	70.60	7.6	3.58	1.45
	搜索开销	5	18.4	69.05	212.46
	搜索效率	14.12	0.41	0.05	0.007
BFSFlood (Boeing 数据)	命中率(%)	5.82	13.29	19.11	20.40
	搜索开销	5	24.89	113.39	548.06
	搜索效率	1.164	0.53	0.17	0.04

搜索开销表示搜索平均涉及的节点数。搜索效率表示平均每个被搜索节点可以获得的搜索成功率，等于搜索命中率除以搜索开销。从表 1 可以看到，在第一跳，SmartCache 能够获得比广播方法高 14 倍的命中率，搜索效率高 12 倍，充分说明节点的兴趣相似度的集中学习是十分成功的。此外，随搜索跳数的增加，SmartCache 算法的搜索成功率增长非常缓慢，但是搜索开销却迅速增长，相应的搜索效率也迅速降低。对于 BFSFlood 算法，其搜索成功率相对随搜索跳数增长较快，但是搜索开销增长迅速，搜索效率也随之减低。除了第一跳，SmartCache 算法的搜索效率要低于 BFSFlood。综合表 1 的结果，可以知道单跳的 SmartCache 算法能够以很低的代价获得很高的缓存命中率，增加其搜索跳数不能显著提高命中率，反而会带来巨大的搜索开销。

结束语 SmartCache 是一种全新的基于客户端的 Web 缓存方法，与相关研究成果相比具有以下特点：基于节点兴趣分类的分布式缓存服务实现非常简单，能够以很低的网络和计算开销获得很高的命中率，并且没有性能瓶颈，有良好的应用前景；提出了集中和分布相结合的新型体系结构，分布是消除性能瓶颈的关键，而对节点兴趣分类进行集中学习和计算从网络开销和计算代价评价都是最经济的方式；采用激励相容的缓存管理策略，用户只需缓存自己访问过的内容；可伸缩性好，缓存能力与客户节点数成正比；与现存的缓存技术兼容；对于 Flash-Crowd 有很好的抵御能力。未来的研究问题包括：1)更有效的兴趣分类学习算法，例如考虑文件稀有度等。2)在朋友关系上增加语义描述，进一步提高系统的可扩

(下转第 118 页)

Checksum:校验和。

AID:要查询的映射关系中的接入标识。

RID:要查询的映射关系中的交换路由标识。

3.2 域内移动通信

终端 MN1 由 AS-1 移动 AS-1',具体的通信流程如图 4 所示。

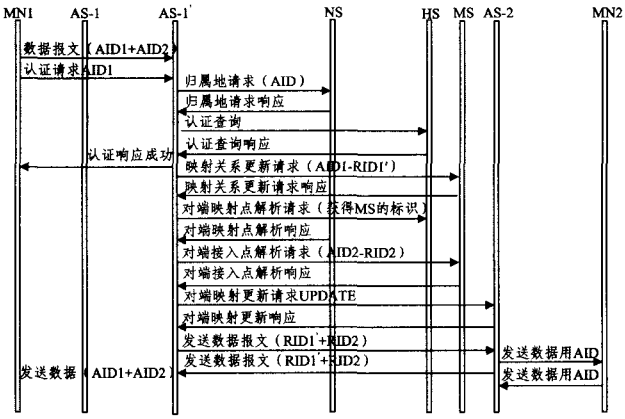

图 4 域内移动通信流程

1)端 MN1 由 AS-1 移到 AS-1'接入网络。2)终端 MN1 以自己的接入标识 AID1 为源标识,以通信对端 MN2 的接入标识 AID2 为目的标识,向 AS-1'发送数据包。3) AS-1'发现终端 MN1 不是自己管辖的用户,强制触发其发送认证请求;通过 NS 查询获得用户归属的服务器 HS 标识。4)终端 MN1 向 AS-1'发送认证请求消息;AS-1'向 HS 发送认证查询消息,HS 返回查询。5)AS-1'返回终端,MN1 认证请求消息。如果通过认证,就能允许其接入网络;反之,则认定终端 MN1 是非法用户,不允许其接入网络。6)如认证通过,AS-1'为终端 MN1 的接入标识 AID1 分配交换路由标识 RID1',形成映射关系 AID1-RID1',并存入本地映射表中。7)AS-1'向映射服务器 MS 报告这个最新的映射关系。8)如果 AS-1'不知道对端 MN2 的映射关系,则要去映射服务器查询。9)映射服务器返回查询消息,AS-1'发送将查询得到的终端 MN2 的映射关系 AID2-RID2 存入对端映射表中。10)AS-1'向 AS-2 发送 UPDATE 消息,AS-2 将 UPDATE 消息中携带的终端 MN1 的映射关系存入自己的对端映射表中。11)AS-2 向 AS-1'响应 UPDATE 消息。12)AS-1'将数据包的源和目的接入标识替换为交换路由标识,并转发出去。13)AS-2 收到数据包后也进行交换路由标识到接入标识的替换,并将数据包向接入网中的终端 MN2 转发出去,最后终端 MN2 收到终端 MN1 的数据包。

结束语 针对下一代通信网络的移动性、安全性,基于终端的身份与位置分离的设计思想,本文从全新的角度考虑问题,提出一种基于一体化网络的移动通信解决方案,对于我们构建基于身份标识和路由标识分离机制的新型一体化网络与路由交换机制也将具有重要的借鉴意义。

但本方案只是通过对现有移动通信技术和新体系结构研究的基础提出的一种设想,对于该方案的具体实现还有待进一步深入研究。尽管如此,通过对这些新的设计技术的研究,我们相信这些非常新颖的设计思路将会影响人们解决类似问题的方法,对于移动通信问题的解决也有着非常重要的现实意义。

参 考 文 献

[1] Stewart R, Xie Q. Stream Control Transmission Protocol[S]. IETF, RFC 2960, October 2000

[2] Johnson D, Perkins C, Arkko J. Mobility support in IPv6[EB/OL]. http://www.ietf.org/rfc/rfc3775, 2004

[3] Teraoka F, Ishiyama M, Kunishii M, et al. LIN6: A Solution to Mobility and Multi-homing in IPv6[R]. internet draft-teraokaipng-lin6-01.txt (work in progress), 2001

[4] Ishiyama M, Kunishii M, Teraoka F. An analysis of mobility handling in LIN6[EB/OL]. http://www.lin6.net/papers/wpmc.pdf, 2001

[5] Moskowitz R, Nikander P. Host identity protocol architecture [EB/OL]. http://www.ietf.org/ internet-drafts/draft-ietf-hip-arch-02.txt, January 2004

[6] Nikander P, Arkko J, Henderson T. End-host mobility and multi-homing with the host identity protocol[EB/OL]. http://www.ietf.org/ internet-drafts/draft-ietf-hip-mm-01.txt, February 2005

[7] Nikander P, Ylitalo J, Wall J. Integrating Security, Mobility, and Multi-homing in a HIP Way[C]// Proceedings of Network and Distributed Systems Security Symposium (NDSS'03). San Diego, CA, US: Internet Society, February 2003: 87-99

[8] Balakrishnan H, Lakshminarayanan K, Ratnasamy S. A layered naming architecture for the Internet[C]// Proc. of the ACM SIGCOMM 2004. 2004: 343-352

[9] Clark D, Braden R. FARA: Reorganizing the Addressing Architecture[A]. ACM SIGCOMM[C], Germany, 2003: 313-321

[10] Yumiba H, Imai K, Yabusaki M. IP-based IMT Network Platform[J]. IEEE Pers. Commun., 2001, 8: 18-23

(上接第 89 页)

展性。3)基于收藏夹的缓存分类技术。

参 考 文 献

[1] Rodriguez P, Spanner C, Biersack E W. Analysis of Web Caching Architectures: Hierarchical and Distributed Caching. IEEE/ACM Transactions on Networking, 2001, 9(4): 404-419

[2] Li Wen-syan, Hsiung Wang-pin, Po O, et al. Challenges and Practices in Deploying Web Acceleration Solutions for Distributed Enterprise Systems// WWW2004. New York, 2004

[3] 贺琛,陈肇雄,黄河燕. Web 缓存技术综述. 小型微型计算机系统,2004,25(5):36-842

[4] Padmanabhan V N, Sripanidkulchai K. The Case for Cooperative Networking// IPTPS 2002. Cambridge, 2002

[5] Wang Xiao-yu, Ng Wee-siong, Ooi Beng-Chin, et al. Buddyweb: A P2P-based Collaborative Web Caching System// Networking 2002 Workshops. Pisa, Italy, 2002

[6] 黄桂敏,杨明福,王学光. 分散型Web缓存模型. 计算机工程,2004,29(17):29-30

[7] Watts D, Strogatz S. Collective dynamics of small-world networks. Nature, 1998: 393-440

[8] Webtraces. http://www.web-caching.com/traces-logs.html

[9] Kalogeraki V, Gunopulos D, Zeinalipour-Yazti D. A Local Search Mechanism for Peer-to-Peer Networks// CIKM. McLean USA, 2002