对等网络中基于位置信息和文件流行度的自适应复本管理算法

陈 康 余宏亮 张 堃
(清华大学计算机科学与技术系 北京 100084)

摘 要 文件共享服务是对等网络中的一个重要应用,数据传输速率逐渐取代响应延迟成为影响用户体验的首要因素.文中研究了对等网络中的副本管理算法,这对于提高对等网络应用的可靠性,降低带宽消耗具有重要的意义.为了在广域网络存储系统中加速文件共享并降低网络带宽消耗,文中提出了 PLAR(Popularity and Locality-based Adaptive Replication)算法.PLAR 采用了基于位置信息和流行度的复本管理算法,该算法还同时引入了混合式的服务器选择策略以及远程增强策略.PLAR 算法在文中的 Granary 对等广域网存储系统中得到了实现.实验表明,通过 PLAR 算法下载速率平均能提高 60%以上,有效提高了共享速度并减少带宽消耗.

关键词 网络存储;对等网络;文件复本;带宽消耗;流行度
中图法分类号 TP393 **DOI 号**: 10.3724/SP.J.1016.2009.01927

Adaptive Replication Management Algorithm Based on Location and File Popularity for Peer-to-Peer Network

CHEN Kang YU Hong-Liang ZHANG Kun
(*Department of Computer Science and Technology, Tsinghua University, Beijing* 100084)

Abstract File sharing is one of the most important applications based on the peer-to-peer network structure. In such application, the data transferring speed is more important than the network latency. The authors have done some work on the replication algorithms based on the peer-to-peer network. Such algorithms are meaningful for improving the reliability of network applications as well as reducing the network traffic. For improving the performance of file sharing with reduced network traffic, this paper proposes a method called PLAR (Popularity and Locality-based Adaptive Replication). PLAR is based on the locality and popularity of objects as well as applying the mixed server selection method and remote boosting schemas. PLAR is used in the authors' Granary system which is data storage system for wide area network. The experiment results show that PLAR can improve the download speed over 60% on average which means that it can improve the sharing as well as reducing the network traffic.

Keywords network storage; Peer-to-Peer network; file replication; bandwidth consumption; popularity

收稿日期:2009-07-15;最终修改稿收到日期:2009-08-25.本课题得到国家自然科学基金(90718040,60603071)、国家"九七三"重点基础研究发展规划项目基金(2007CB310900)和国家"八六三"高技术研究发展计划项目基金(2008AA01Z112)资助.**陈 康**,男,1976 年生,博士,助理研究员,主要研究方向为分布式系统、存储系统.E-mail: ck99@mails.tsinghua.edu.cn; kang.chen.tsinghua@gmail.com. **余宏亮**,男,1976 年生,博士,副教授,主要研究方向为分布式系统、对等计算. **张 堃**,男,1983 年生,硕士研究生,主要研究方向为对等网络.

1 引 言

对等计算技术(Peer-to-Peer,P2P)是无中心的结构,主机同时作为信息的提供者与信息的消费者.存储是对等计算的重要应用,为用户提供数据长时间保存、文件共享、数据缓存等服务.复本(replication)是指将一份数据复制为内容相同的多份,放置在不同的存储设备或者节点上.复本技术能够提高网络中数据对象的可靠性,提高客户端访问数据的效率等.

在使用对等计算的广域网的存储系统中,复本的位置需要在网络上分布到不同的节点上,避免因局部的网络或者电力问题造成数据不可用.在基于DHT的对等存储中,一般是将数据放置在节点编号与数据对象的对象编号相近的几个节点上.由于编号使用Hash函数计算,编号相近节点实际上地理位置分布广泛,这就保证了复本的分散性,提高了数据的可用性.通过物理节点分割成更小的虚拟节点,能够避免Hash函数与数据热门不均匀性带来的负责均衡问题.另外,通过复本以及缓存的方式,在系统中保留一个数据对象的多个复本,能够提高对象的访问效率.

很多互联网应用使用响应延迟来评价用户体验,亦有不少研究关注于数据的一致性等.随着文件数据规模的增大,网络文件的传输速率变得更加重要,成为应用用户体验的重要因素.另外,不少系统为了简化设计,只允许用户访问自己的文件,而不提供数据共享的支持.但是实际上用户将自己的文件存放在P2P存储系统中,共享是一个很大的动机,而对于在分布式存储系统中共享文件的性能的研究存在着空缺.本文提出了一种在广域网络存储系统中加速文件共享并降低网络带宽消耗的算法——PLAR. PLAR采用了基于位置信息和流行度的复本管理算法.该算法还同时引入了混合式的服务器选择策略以及远程增强策略.算法也在一个具体的对等网络的实际系统Granary中获得实现.Granary是一个针对广域网环境设计的大规模分布式存储系统.系统由上百个服务器节点组成.节点在地理上是完全分散的,之间由互联网相连.

本文在第2节中讨论相关工作,并分析本文工作的特点;在第3节中论述位置信息的探测问题,讨论在网络环境下距离的相关问题;第4节主要讨论文件流行度的估算方法与预测方法,这对于复本放置是非常有意义的;第5节讨论基于位置与文件流行度的复本管理算法,是本文的重点算法;第6节是实验与评价;最后一节为总结.

2 相关工作

传统的分布式文件存储系统通常用响应延迟的大小来评价用户体验.随着多媒体文件共享的流行,文件的传输速率变得更加重要,已经取代响应延迟成为了影响用户体验的首要因素.另一方面,因为文件体积越来越大,在网络上传输文件就会消耗更多的网络带宽.已有的分布式存储系统,如CFS[1]、PAST[2]、Freenet[3]和OceanStore[4]等,都把精力集中在提供数据的可用性.也有研究集中于保证数据复本的一致性[4],对于性能的研究比较少.另外,不少系统为了简化设计,只允许用户访问自己的文件,而不提供数据共享的支持[2].

与复本相关的研究包括文件系统与广域存储系统中的缓存.复本(replication)和缓存(caching)是提高分布式系统性能的两种常用方法[2-4].但是基于以下3个方面的考虑,在Granary存储系统中使用复本,而不是缓存来提高文件下载的速度.

(1)复本是主动的,所以我们可以提前估计需要的复本数并创建复本;

(2)即使利用缓存来提高访问速度,我们仍然需要复本来保证数据的可用性.而同时保存缓存和复本会占用更多的存储空间;

(3)缓存并不是所有情况都能使用的.比如采用多跳路由时,数据从源节点,通过一个或多个中间节点,才能到达客户端,这时我们可以在中间节点上建立缓存.但是因为Granary采用的单步路由,数据从服务器直接到达客户端,所以没有合适的位置来利用缓存.但是由于多跳路由本身比单跳路由会有更多的网络开销,因此虽然缓存的网络开销比复本小,但是综合起来看,在单跳路由的系统中使用复本,并不一定比多跳路由的系统中使用缓存产生的网络流量多.

3 位置信息的探测

本节开始将介绍本文提出的PLAR(Popularity and Locality-based Adaptive Replication)算法,能够在广域网存储系统中加速文件共享并降低网络带宽消耗.此算法基于两个因素,一个是位置信息,另

外一个是文件的流行度.关于位置信息的测量将在本节进行讨论,而下节将研究文件流行度的衡量,在第 5 节中则根据这两个因素详细讨论 PLAR 算法的具体设计.

3.1　位置信息的客户端聚类算法

为了优化复本的放置,我们需要知道网络中两个主机之间的距离,即网络距离.网络距离通常是用往返延迟(Round-Trip Time,RTT),traceroute 跳数、DNS 域名和 IP 地址来衡量的.Andrews[5] 提出一种将主机聚类的算法,可以把大量的主机划分成数量较小的几个聚类,相同聚类中的主机之间的 RTT 较小,可以认为他们的网络距离较近,并有可能处于同一个自治系统(AS).图 1 是 Andrews 算法构成的 IP 树,IP 树里的每个节点表示一个 IP 前缀,比如 166.111.0.0/16.每个节点的前缀是它的父节点的前缀加上一位.比如根节点是 0.0.0.0/0,而根节点的两个子节点分别就是 0.0.0.0/1 和 128.0.0.1/1.而 Andrews 算法则在这个 IP 树的基础上构成若干个聚类,每个聚类节点中的 IP 之间的网络速度都比较快,可以认为是属于同一个自治网络.

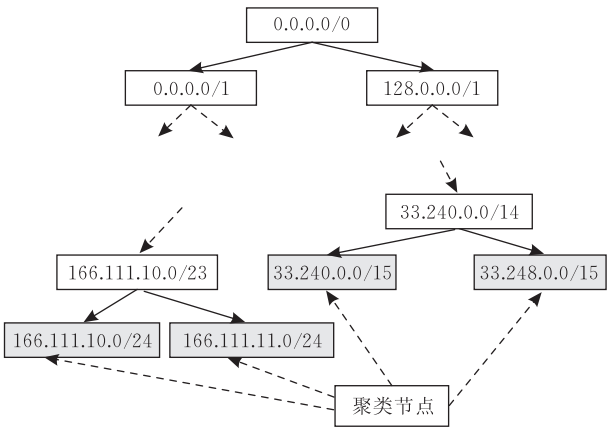


图 1　Andrews 算法中的 IP 树(其中带阴影的是聚类节点)

Andrews 的算法在实际应用中还存在一个问题,就是它只考虑了 IP 地址,这在一个自治网络拥有不止一个 IP 前缀的时候会失效.比如清华大学的校园网络拥有 2 个 IP 前缀:59.66.0.0/16 和 166.111.0.0/16,按照上面的算法它们是永远不会聚类到一起.虽然 IP 前缀不同,但是校园里的所有的主机都拥有同样的 DNS 域名后缀(.ip.tsinghua.edu.cn).所以我们引入另一个数据结构域名树,来记录每一个客户机的域名.所以,对域名树的聚类,在 Andrews 的算法的基础上,我们作了以下改进:

(1)顶级域名不能被合并.比如.tsinghua.edu.cn 就不能再合并到.edu.cn 上了.因为顶级域名的范围很大,合并顶级域名不能帮助客户端的聚类.

(2)聚类合并时,只是把原来的父节点标记成聚类节点,原聚类并不删除.如果原聚类有子节点,删除其子节点.也就是说,除了叶子节点之外,叶子节点的父节点也有可能是聚类节点(称为非叶子聚类节点),可见图 2.在合并聚类时,也要检查非叶子的聚类节点的子节点是否符合合并标准,如果不符合,就撤销该聚类.也就是说,聚类是可以再拆散的.因为和 IP 地址不同,域名只能通过对客户端的地址进行 DNS 反查得到,所以获知域名的过程具有不确定性.假设域名.x.y.z 下面有.a.x.y.z,.b.x.y.z 和.c.x.y.z 3 个域名,分别代表不同的自治网络,但是.a.x.y.z 和.c.x.y.z 的网络状况比较相近.如果一开始只有.a.x.y.z 和.c.x.y.z 的用户来访问,就会使系统误以为.x.y.z 只有这两个域名,于是就把它们合并成.x.y.z 了.但是后来.b.x.y.z 也来访问了,发现 Ping 值和其它两个域名的相差比较大,实际上它们不应该合并在一起.这样就会拆散.x.y.z 这个聚类,重新变成 3 个较小的聚类.

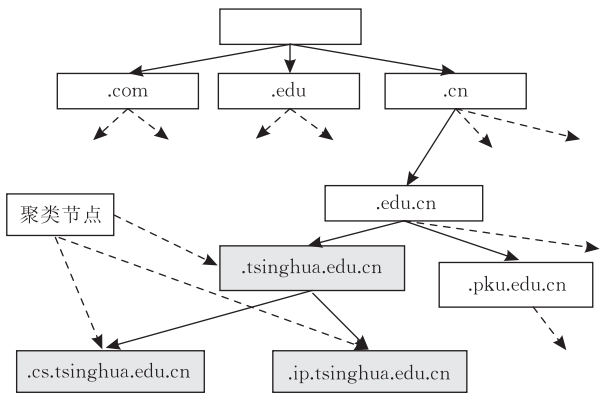


图 2　PLAR 客户端聚类算法中的 DNS 树
(其中带阴影的是聚类节点)

3.2　网络距离

有了 IP 树和域名树中的客户端聚类以后,下面定义网络上的两个主机 H_1 和 H_2 之间的 IP 距离:

(1)找到 IP 树上 H_1 和 H_2 所属的聚类叶子节点 s_1 和 s_2.

(2)找出同时包含 s_1 和 s_2 的最小子树,也就找到一个节点 p,使 p 是 s_1 和 s_2 的共同祖先,并且由 p 作为根节点的子树高度最小,设高度为 $\hat{h}(\hat{h}\geq 0)$.

(3)设 $h=\hat{h}/24$,那么 H_1 和 H_2 之间的 IP 距离 $d_{\mathrm{ip}}^{1,2}=h$.

(4)如果 H_1 和 H_2 都能通过 DNS 反查(DNS reverse look-up)查询出域名,那么它们之间的域名距离 $d_{\mathrm{domain}}^{1,2}$ 也可以在 DNS 域名树中用同样的方法求

得,否则取值为 1.

(5) 根据 $d_{ip}^{1,2}$ 和 $d_{domain}^{1,2}$,我们定义 H_1 和 H_2 之间的网络距离为 $d^{1,2}=d_{ip}^{1,2}\times d_{domain}^{1,2}$.

网络距离的取值范围为 $0\leq d^{1,2}\leq 1$. 从公式可以看出,不论是在 IP 树还是在域名树中,只要 H_1 和 H_2 处于同一个聚类,则有 $d^{1,2}=0$. 因为服务器之间也会传输文件,所以我们不仅可以计算客户机之间的网络距离,也可以计算服务器之间的网络距离. 本小节定义的网络距离将在复本放置算法和客户端的节点选择算法中有所体现.

4 文件流行度

4.1 文件流行度的估计

文件流行度就是一个文件被访问的频度. 流行度高的文件,即热门的文件会有较大的访问量,承载该文件的节点的负荷就会比较大,并且连接该节点的网络链路会变得拥塞. 所以热门的文件需要更多的复本以分散负载.

PLAR 使用一个叫做流行度表(Popularity Table,PT)的数据结构来计算文件的流行度. 系统中的每个节点都单独维护一个 PT,只记录本节点接受的数据访问请求. PT 的结构如图 3 所示. 表中每一项代表一个文件,记录了这个文件的总的请求频率. 另外,每个文件还有一个列表(source cluster list)记录这个文件被每个客户端聚类访问的频率. 客户端聚类就是图 1 中 IP 树的叶子节点,由 IP 前缀表示. PT 中的总访问频率每隔一定时间(比如 1 小时)清空一次,并把原来的数据放入一个历史队列中. 历史队列保存了最近一段时间淘汰下来的 PT,超时间范围的 PT 将被丢弃.

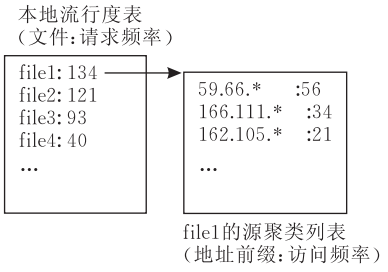


图 3 Popularity Table 的数据结构

PT 只有某个节点的文件访问记录,所以其反映的文件流行度可以成为“局部流行度“(Local Popularity, LP). 相对的,一个文件在整个系统中的访问频率称为“全局流行度”(Global Popularity, GP). 在具有中央控制的分布式存储系统中,比如集群中的存储系统,通常会使用 GP. 但是在 P2P 存储系统中,由于没有中心控制节点,广域网中节点之间消息传递的代价又比集群中高. 而且,由于 P2P 存储系统具有分布范围广、异构性强的特点,LP 和 GP 会表现出不同的特点. GP 仅代表了文件受欢迎程度的总分布,而 LP 还受数据放置位置的影响. 考虑下面几种情况,会发现 LP 比 GP 更适合作为 P2P 存储系统的复本策略的参考值:

(1) 如果对每个文件感兴趣的用户都是均匀分布在节点周围的,那么 LP 和 GP 的分布就可以是一致的. 即使某一个节点没有热门文件的复本,以致于该节点上的访问频率比其它节点低,但是也因为这个节点负载过轻,其它节点就会在适当的时机把热门文件的复本移动到这个节点上,这样这个节点的访问频率也会逐渐向全局的访问频率靠拢.

(2) 如果用户的兴趣在空间上不是均匀分布的,这时不同节点的 LP 就会反映出这个分布差异,这对针对用户的位置来优化复本的放置是非常有利的. 比如,如果一个文件 F 在客户端聚类 C 中很受欢迎,因为基于速度的考虑,客户端会优先选择附近的节点来访问,这样 F 在距离 C 最近的节点(或者从 C 访问速度最快的节点)上的 LP 就比较高. 我们设这个节点为 A. 这时 F 的 GP 可能不是很突出,因为在其它客户端聚类中有可能会是其它文件比较受欢迎(考虑客户端聚类可能代表一个校园网络、一个城市网络等等,不同的学校或者城市流行的东西会不一样,所以这情况是存在的). 节点 A 可以根据 F 较高的 LP 数值来为 F 创建更多的复本,从而加快聚类 C 中的客户端访问 F 的速度.

(3) 当一个节点处于过载状态时,最有效的办法是将这个节点上访问频率最高的几个文件多复制几份到其它节点上,来分散访问带来的负载. 所依据“访问频率”应该是根据 LP 来决定,而不是 GP.

4.2 文件流行度的预测

从文件访问记录我们只能知道一个文件在过去一段时间的流行度即访问频率. 但是如果想要更好地对流行度作出反应,我们就需要对未来作一些预测. 对文件下载日志[6]和流媒体访问日志[7-8]的分析表明,多数流行度较高的文件都会在发布之后的若干小时内达到流行度的高峰,从而为提供这些文件的节点带来巨大的流量负载. 如果这些热门的文件的复本数比较少,蜂拥而至的访问将耗尽这些复本所在节点的带宽和 I/O 资源,并可能在连接这些节点的关键网络链路上形成拥塞. 如果我们能够在热

门文件的流行度达到最高峰之前就提前作出反应,为热门文件多复制几份,就可以有效地缓解高峰到来时系统的负载不平衡状态.

我们分析了来自两个实际系统的访问日志.一个是清华大学校内的一个多媒体文件共享 FTP 服务器,另一个是 PowerInfo VoD 视频点播系统[9].从这两个日志中文件的流行度的时间序列,我们可以分析得到两种流行度变化的模式(图 4).

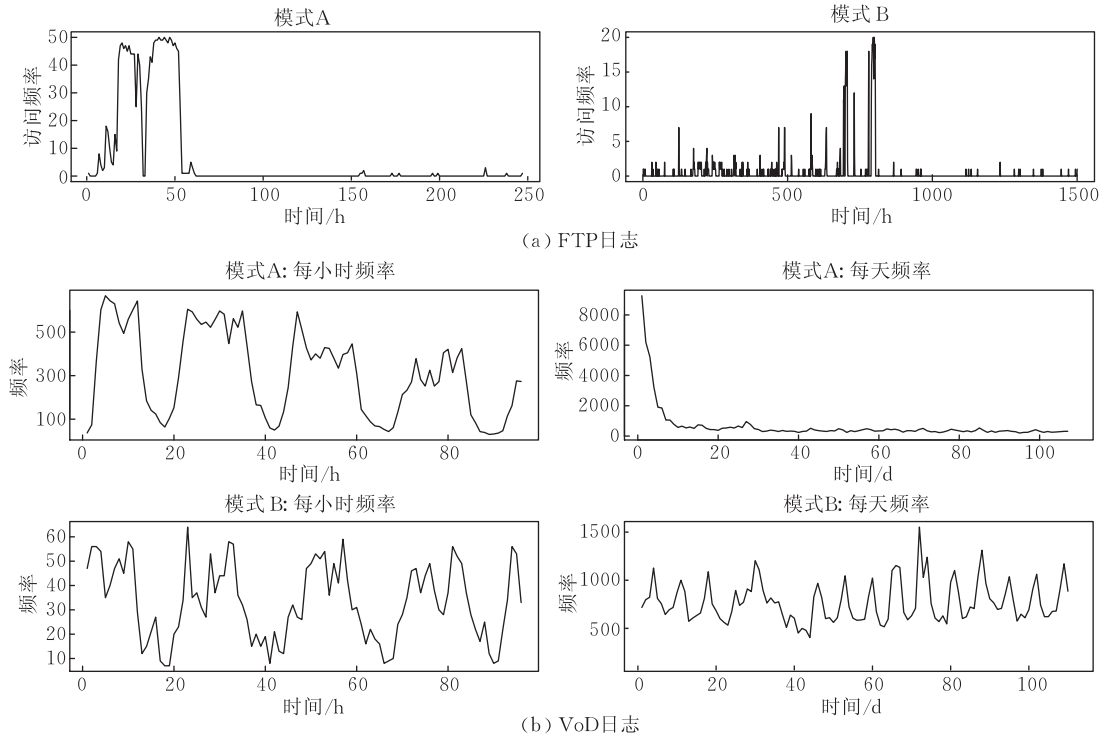


图 4　日志中的两种典型文件访问频率模式(模式 A,模式 B)

(1) 模式 A(Pattern A). 在文件发布后的很短时间内,由于突发的访问量,访问频率达到最高点.在此之后,每小时的访问频率以 1 天为周期进行振动,振动的幅度随着时间减小.每天的访问频率呈类似指数下降的趋势.日访问频率降到一个很低的水平后,就保持稳定.模式 A 代表了一类在发布初期就变得很流行的文件,比如新的电影或音乐专辑,但在发布初期的一阵高峰后会慢慢趋于平静.这个负荷高峰会对节点带来一定冲击.这个模式在文献[6]和文献[8]中也有提及.如果能够在访问频率到达高峰之前提前判断出这个模式,就可以采取多复制复本的办法缓解节点的压力.

(2) 模式 B(Pattern B). 和模式 A 不同,模式 B 在发布的初期并没有突发的访问量高峰.它只是以 1 天为周期稳定地振动.模式 B 的曲线和模式 A 稳定后曲线的一致,它代表了不热门的文件.

在对数据访问频率的预测的研究中,差分自回归滑动平均模型(ARIMA)是一个常用的时间序列的预测方法[10].在此基础上,我们使用一个"潜在流行度"(Potential Popularity,PP)的概念来代表一个文件在近期估计将达到的流行度水平.在 PLAR 中,我们为一个文件(确切地说是一个复本)定义 3 种流行度状态:未激活状态(I)、初始上升状态(R)和稳定状态(S).设 x_t 为第 t 个小时观察到的流行度,$\bar{x}_t$ 为第(t-23)到第 t 小时这 24 小时内的流行度的平均值.流行度状态之间的转换条件为

(1) 每个文件(即复本)的初始状态为 I.每一个新的流行度观察值 x_t 都和一个阈值 P_h 进行比较,如果 $x_t > P_h$,这个文件的状态变为 R.

(2) 当一个文件处于状态 R 已达 24 小时,将这 24 小时的流行度平均值 $\bar{x}_t$ 和 P_h 进行比较,如果 $\bar{x}_t \geq P_h$,文件的状态转为 S.

一个文件的潜在流行度值 p 由它当前的流行度状态和过去的访问频率(即流行度的观察值)决定.设 t 是当前的小时序号,则

$$\rho = \begin{cases} x_t & \text{(I)} \\ \max(x_t, 2x_{t-1} - x_{t-2})/2 & \text{(R)} \\ (x_{t-15} + x_{t-14} + \cdots + x_{t-10})/6 & \text{(S)} \end{cases}$$

当文件处于状态 I 时,它的潜在流行度值就是前一个小时的访问频率.当处于状态 R 时,潜在流行度根据前 3 个小时的访问频率变化趋势来决定,

这就对应了我们判断热门文件的第一个标准初始斜率.当文件处于状态 S 时,潜在流行度就是 12 个小时以前那个时刻左右 3 个小时的平均访问频率.因为文件的访问频率以 24 小时为周期进行振动,所以 12 个小时以前附近的访问频率正好就是 12 个小时以后附近的访问频率.此外,状态 I 和 R 的潜在流行度值和当前的访问频率相一致,而状态 S 的潜在流行度值和实际的访问频率有 12 个小时的位移,所以对处于状态 S 的文件,访问稀疏的时候正好是潜在流行度值较高的时候,这时候如果进行复本的复制,既可以使复制的工作避开访问的高峰,也可为即将到来的高峰做好准备.所以,我们计算状态 I 和 R 的潜在访问度的时候除以了 2,这样对于不是特别热门的文件,PLAR 将趋向于在状态 S 而不是状态 I 或 R 进行复制工作.

5 基于位置信息与文件流行度的复本管理算法

5.1 复本的生成

为了提高分布式存储系统中数据共享的速度,准确地估计每个文件需要的复本数量是非常重要的.在一个系统中,把对象按照请求的频率从高到低排序,如果第 i 个对象的请求频率 q_i 和 $1/i^{\alpha}$ 成正比(α 是一个常量),那么说这个请求分布符合类 Zipf 分布.对 Web 缓存的研究发现,数据对象的访问频率符合类 Zipf 分布[11].对视频点播系统(VoD)的研究显示数据复本的数量分布应该符合类 Zipf 分布[12].所以 PLAR 利用类 Zipf 分布来估计复本的个数.我们规定访问频率排第 i 位的文件所需复本数 q_i 为 $\frac{\beta M}{i^{\alpha}}$.其中 M 是整个系统的请求频率,β 是一个可配置的常量.我们可以假设用户请求在所有节点上是均匀分布的,这样 M 可以用一个节点上的请求频率 m 和节点的数量 N 来估计:$M=N\times m$.PLAR 维护一个按访问频率(或者潜在流行度)排序的文件列表.设 r_i 为排在第 i 位的文件的复本个数,如果 $r_i<q_i$,系统将会为该文件生成一个新复本.

但是,比较分布式系统中一个节点上的文件访问和单节点系统中的文件访问会发现,两者的模式并不是完全相同的.在分布式系统中,客户端往往可以从多个节点中选择一个来获取文件,而一个好的客户端应该会选择距离较近或者较空闲的节点,因此客户所在的位置会对某一个节点上的文件访问模式产生影响.对于一个特定的客户机,我们将网络距离在一定范围之内的节点定义为“本地节点”,其它的节点称为“远程节点”.发给本地(远程)节点的请求被称为本地(远程)请求.PLAR 的节点选择策略会优先选择本地节点来为客户提供服务,但是仍然会有 2 种情况导致选择了远程节点:

(1)本地节点都因超载而无法提供服务.

(2)没有任何一个本地节点拥有用户所请求的文件.

所以,远程请求一般反映了被请求的文件需要在请求的客户附近放置更多复本,而本地请求的组成则会和这个节点附近的文件的流行度组成一致.PLAR 提出一个叫做“远程增强”(Remote Boosting,RB)的策略来解决上面的问题.节点会比较本地和远程请求的频率,如果一个文件的远程请求比本地请求多,那么不管是不是已经达到,都会额外创建一个复本,而这个新复本将会放置在请求这个文件最多的客户机附近的节点上,以满足这些用户的需求.

5.2 复本的放置

因为广域网上不同主机之间,由于所处的服务提供商不同,地理位置不同,网络链路条件不同,数据传输的速度差别很大.前面描述的网络距离可以用来衡量这种差别.网络距离根据 IP 地址、DNS 域名等参数可以区分出不同的服务提供商和地理位置,而网络延迟(如 RTT)则可以区分出网络链路的状况.所以,网络距离小的节点之间传输速率较快,而网络距离大的节点之间可以认为传输距离较慢.为了让用户能够较快地访问到自己想要的数据,就一定要把数据放在据用户网络距离较短的节点上.

文献[13]介绍一种在 P2P 的多媒体文件发布网络中的复本放置方法,这篇文章将节点和客户端分成一些组(group),可以认为同一组的主机距离比较近,且网络速度较快.提出的优化目标就是所有访问所需的网络距离最小.

虽然 PLAR 中没有组的概念,但是因为我们为每个节点和客户端之间都可以估计出网络距离.我们定义节点 A 上的一个文件 f 的平均传输距离 $\overline{D}(A,f)$ 为

$$\overline{D}(A,f)=\frac{\sum_j d^{j,A}\times q_j}{\sum_j q_j},$$

其中 $d^{j,A}$ 是客户机 j 到节点 A 的网络距离,q_j 表示客户机 j 访问文件 f 的频率.我们选择 $\overline{D}$ 值最小的节点来放置文件 f 的新复本,被选中的节点称为目

标节点.

新复本的数据是目标节点主动从其它节点上下载的.假设节点 A 决定为文件 f 新建一个复本,目标节点为 B.首先 A 将 B 的 ID 添加到 f 的复本表中.在 Granary 中,复本表是存放在 DHT 中的,这样可以保证任何节点都可以访问到.接着 A 向 B 发送一个消息,告诉 B 需要下载 f 的数据.B 从一个存放有 f 的节点(称为源节点)下载数据.源节点不一定是 A,而是由本文下面会介绍的节点选择策略来决定的,这保证新复本的数据以最快的方式复制到目标节点.而且,因为复本表已经更新了,即使 B 只下载了一部分数据,如果用户从 B 下载比从其他节点快的话,就可以直接从 B 开始下载.实际上,因为 Granary 中两个节点之间的网络通常比节点到客户机之间的快,所以新复本的传输通常在用户下载完之前结束,这样用户就可以完全从 B 下载这个文件.

5.3　复本的替换

由于用户随时会上传文件到 P2P 存储系统中,同时系统根据复本生成算法也会随时创建新复本.不论是新上传的文件还是新生成的复本都需要占用磁盘空间.当磁盘空间不足以满足新文件和复本的要求时,节点需要删除一些数据来腾出空间.PLAR 采用最久未访问策略(Least Recently Used,LRU)来选择需要删除的数据.

每个节点自己维护一个 LRU 队列,队列中包含了节点上的所有文件.新产生的文件和复本被加到队列的尾部.当一个文件被客户端访问时,它也被从 LRU 队列中抽取出来放到队列的尾部.当需要删除一个文件时,从队列头部的文件开始,逐个删除,直到磁盘剩余空间达到新文件的存储要求.

上面给出的是一个朴素的 LRU 复本替换策略.但是在 P2P 存储系统中,因为广域网中网络不稳定,速度较慢,生成一个复本是代价比较高的一件事,所以要尽量避免无谓地删除复本.因此,PLAR 在删除复本时,不仅要依据 LRU,还要比较被删除的复本和新复本的重要程度,我们用"复本需求度" w 来量化这一"重要程度".

设一个节点在考察文件 g 的一个复本是否应该被删除时,它的复本需求度为 $w(g)$;当一个节点在考察文件 f 是否应该新建一个复本时,它的复本需求度为 $w(f)$,则 $w(g)=\frac{q_g}{r_g-1}$ 以及 $w(f)=\frac{q_f}{r_f}$.其中,q_g 和 q_f 分别为文件 g 和文件 f 需要的复本数,根据 Zipf-like 模型计算出.而 r_g 和 r_f 分别是文件 g 和文件 f 当前的复本数.从公式可以得出,当 $r_g=1$ 时,$w(g)=\infty$,也就是说如果 g 只剩下一个复本了,那么它就是不能被删除的.

在定义了复本需求度 w 之后,PLAR 复本的替换过程就可以这样描述:节点 A 在决定在节点 B 上为文件 f 新建一个复本时,首先把 B 加入到 f 的复本列表中,然后发消息通知 B 开始进行数据复制.这个消息中包括了 A 为 f 计算的复本需求度 $w(f)$.节点 B 收到消息后,从自己的 LRU 队列从头向尾依次检查每一个文件 g,计算 $w(g)$,如果 $w(g)<w(f)$,把 g 标记为可以删除.最后当标记删除的文件大小总和达到 f 的文件大小时,才删除标记的文件,并开始复制 f.否则 B 将自己从 f 的复本列表中删除,取消 f 的这次复制过程.

5.4　客户端节点选择策略

要使用 Granary 存储系统,用户首先连接到一个他知道的节点(称为入口节点).客户端会保存一个入口节点的列表,并且在每次登录后更新,这样客户端可以一直拥有最新的入口节点供选择.

登录后,入口节点会发送给客户端一个可用节点的列表.对于上传操作,可用节点列表包含所有剩余空间足够的节点;对下载操作,该列表就是被下载文件的复本列表(从 DHT 中获得).接着,客户端给所有的可用节点广播一个消息.节点接到广播的消息后返回给客户端一个确认消息,其中包括加权平均传输速率($\bar{R}$)以及该节点和客户端之间的网络距离.

节点根据以前的文件传输来计算($\bar{R}$).首先在 IP 树或 DNS 树中找到客户端所属的聚类节点.聚类节点中记录了属于该聚类的所有客户机的文件传输日志.将日志中的传输速率按照文件的大小和距现在的时间加权平均,就得到 $\bar{R}$:

$$\bar{R}=\frac{\sum_j size_j\times 0.5^{age_j}}{\sum_j time_j\times 0.5^{age_j}},$$

其中,j 是日志的序号,$size_j$ 是文件的大小,age_j 是传输完成距现在的时间(小时数),$time_j$ 是传输所花的时间.

因为 $\bar{R}$ 代表了该客户端以及其附近的客户端到一个节点的实际传输速率,我们可以用它来防止客户请求集中到一两个节点上.研究发现,如果客户端都只是简单地选择最近的节点(即最近策略),有可能给一两个节点带来很大负载,而其他节点却空

闲着[14]. 为了避免这种情况,我们使用混合式的节点选择策略. 这个策略分两步进行:

(1) 选择可用节点中 $\bar{R}$ 较高的一半节点;

(2) 在其中选择距客户端网络距离最近的节点.

负荷较大的节点,单个用户的传输速率也会比较低,因此 $\bar{R}$ 会比较低,这样在第一步选择中就会被淘汰.

6 实验与评价

6.1 实验环境概述

我们做了两个实验来检验我们的 PLAR 算法对于数据共享速度的改善效果. 第 1 个实验是利用 RONS 系统进行的网络模拟,利用了两个负载较重的数据共享系统的日志作为输入;第 2 个实验是在 PlanetLab(http://www.planet-lab.org/)平台上进行的仿真实验. 其中 RONS 是我们开发的面向复本的网络模拟器(Replication-Oriented Networking Simulator),特点是尽量简化网络底层的协议模型,但是保留了带宽的限制,并将重点放在复本管理算法对带宽需求、用户下载速率和对节点负载的影响上. 表 1 给出了在 RONS 模拟中使用的日志,而表 2 给出了模拟节点间的带宽配置.

表 1 模拟使用的日志

	时间长度/d	涉及文件个数	文件的平均长度/MB	文件访问次数	每小时访问次数	下载数据总量/GB
FTP	136	7525	56	5093123	30	12573
VoD	50	8540	125	97239	4244	610793

表 2 RONS 模拟中节点之间的带宽配置

链路类型	带宽
端节点—端节点	100Mbps
端节点—中转节点	1Gbps
中转节点—中转节点(同一个域内)	500Mbps
中转节点—中转节点(不同域之间)	1Gbps
客户端接入带宽	10Mbps

6.2 模拟实验

GT-ITM(http://www-static.cc.gatech.edu/projects/gtitm/)是一个研究用的网络拓扑生成工具,可以根据输入的参数生成符合要求的 transit-stub 模型的网络拓扑. 我们的模拟在 GT-ITM 生成的一个网络拓扑上运行. 这个网络包含 9 个中转域(transit domain),每个中转域有 12 个中转节点(transit node). 每个中转节点和 1 个端域(stub domain)相连,每一个端域平均含有 10 个端节点(stub node). 中转节点代表了路由器,而端节点代表了网络上的主机. 整个网络含有 108 个端域,108 个中转节点和 1080 个端节点. 节点之间的网络带宽设置在表 2 中列出. 我们随机选择了 40 个端节点作为 Granary 系统的存储节点,其它的端节点则作为客户机.

我们使用平均下载速度 $\bar{r}$ 来量化所有用户的下载速度. 设实验中第 i 个下载事件所下载的数据量是 s_i,花费的时间是 t_i,则 $r_i=\frac{\sum_i s_i}{\sum_i t_i}$. 表 3 列出了模拟中不同的复本策略下,用户的平均下载速度(表 3 中标记的解释:无复本(No replication)表示所有文件都只有一份,没有复本;完全镜像(Full mirror)表示每个文件在每个节点上都有复本;最近选择(Closest)表示客户端总是选择网络距离最近的节点;混合选择(Hybrid)表示首先在可用节点中选择具有较高的加权平均传输速率($\bar{R}$)的半数节点,然后在这些节点里面选择网络距离最近的;远程增强(RB)表示如果一个文件收到的远程请求比本地请求多,那么即使这个文件的总访问频率没有提高,也要为这个文件创建额外的复本). 可以看出,在使用 VoD 日志的实验中,复本的引入使用户的平均下载速率提高 300% 以上. 而使用 FTP 日志的实验中,这个提高并不是很明显. 主要原因是 FTP 日志中每个文件的访问频率相当低,对一个文件的访问比较分散,很少出现短时间内用户蜂拥而至的情况. 而在低访问频率的情况下,缓存和复本很难发挥作用.

表 3 基于 GT-ITM 的网络拓扑的模拟实验中,用户的平均下载速度

复本策略	FTP 日志	VoD 日志
No replication	679780	166747
Closest	680027	789661
Closest+RB	692055	819183
Hybrid	680031	801889
Hybrid+RB	692674	838081
Full mirror	906887	910695

从结果中,我们可以看到混合节点访问策略(Hybrid)的平均下载速率比最近选择要好,而远程增强策略(RB)可以更近一步提高下载速率. 在 VoD 日志的实验中,Hybrid 和 RB 的组合可以使平均下载速率提高 6%. 通过图 5,我们可以发现 Hybrid 和 RB 减少了低速下载的数量,这明显地改善了用户体验. 我们还跟踪了所有的下载事件中最慢的 20%,发现它们的平均速度在使用了 Hybrid 和 RB 之后提高了 14%.

为了分析网络带宽的占用情况,我们记录了连

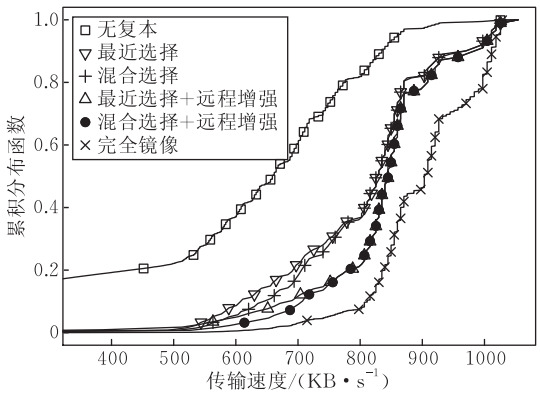


图 5　VoD 日志的模拟实验中,用户下载速率的 CDF 图

接主机的网络链路上通过的数据量.我们将那些数据流量最大的链路称为关键链路,因为这些链路成为了用户下载速度进一步提高的瓶颈.从图 6 可以看出,复本的引入明显地降低了关键链路上数据流量.而且,RB 使所有链路上的平均数据流量降低了 22%,关键链路上的平均数据流量降低了 36%.但是,我们也可以发现,Hybrid 策略在节省网络带宽方面,和 Closest 策略相比并没有什么优势,因为 Hybrid 虽然对总体的性能改善有帮助,但它的侧重点在于防止访问集中于少数节点上,而不是降低网络流量,所以没有明显地降低网络流量.

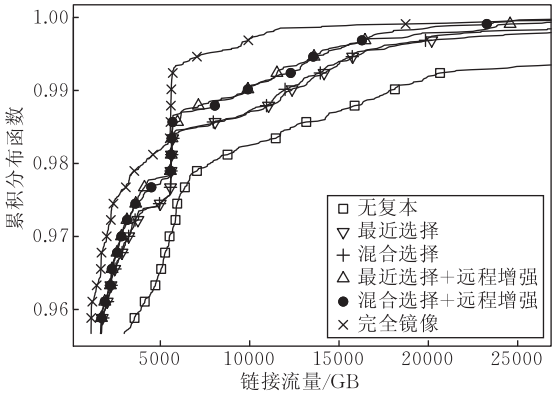


图 6　VoD 日志的模拟实验中,网络的关键链路上的数据流量的 CDF 图

从图 7 可以看出,PLAR 明显减少了关键链路上的网络拥塞,网络拥塞情况已经相当接近性能最优的全镜像(full mirror)的情况.另外,PLAR 复本算法对于网络流量的影响也可以通过传输数据所通过的网络跳数(也就是经过的路由器的个数)来反映.因为 PLAR 算法的一个复本放置的原则就是"将数据放置在距离需要的用户最近的节点上",所以算法的效果应该是普遍降低数据传输所需要的跳数.图 8 说明了这个效果.图中,PLAR 对应的曲线,跳数<4 的数据量增加了,而跳数>4 的数据量减小了,正说明 PLAR 算法普遍减小了数据传输所需要的跳数.这也就是为什么 PLAR 可以提高用户访问的速度并降低网络带宽的消耗.

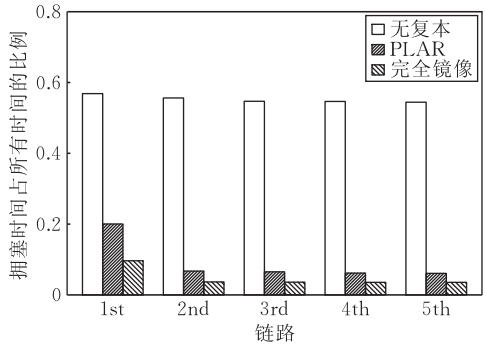


图 7　VoD 日志的模拟实验中,关键链路上的网络拥塞情况

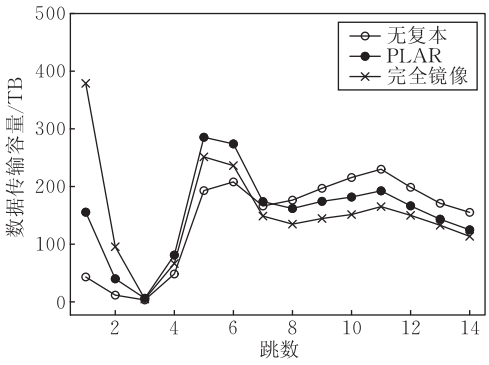


图 8　VoD 日志的模拟实验中,数据经过的网络跳数的分布图

6.3　PlanetLab 上的仿真实验

PlanetLab 是一个部署在互联网上的大规模分布式系统测试平台.它由分布在世界各地的各大高校和研究机构共同组建,用上千个节点构成一个广域网测试平台,适用于 P2P 等大规模广域网系统在实际环境中的测试.我们在 Granary 系统上实现了 PLAR 复本管理算法.为了测试 PLAR 算法在真实的广域网环境中的表现,我们在 PlanetLab 的 40 个节点上部署了 Granary 服务器,进行了两项仿真实验:日志回放实验(REPLAY)和共享实验(SHARE).

(1) REPLAY 实验重放了 Granary 系统在清华大学中的几个月公开测试运营中的日志.我们把日志中文件的访问分配到安装在 120 个 PlanetLab 节点上的客户端仿真器上.这些客户端仿真器将重放这些访问并记录下载速率.

(2) SHARE 实验模拟了一个典型的共享场景:一个用户上传了一个文件并共享给他在世界各地的朋友.我们在 193 个随机选择的 PlanetLab 节点上安装了客户端仿真器,让它们按一定次序来下载一个 5MB 大小的文件.

在仿真中我们测试了 4 种算法配置.

配置 1：naïve，每个文件固定有 2 个复本，复本的位置是随机选择的.

配置 2：near，将复本放在靠近这个文件的请求者的服务器上.

配置 3：popular，为热门的文件创建更多的复本.

配置 4：near and popular，即本文提出的加速算法，为热门的文件创建更多的复本，且复本放置在靠近请求者的服务器上.

另外，我们还测量了每个客户端与所有节点之间的最大可用带宽(max bw)(配置 5)，作为 PLAR 算法结果的参考.

图 9 显示了实验中记录的客户端所有下载操作的平均下载速率. Near and popular 比 naïve 的下载速率，在 REPLAY 中快 60%，在 SHARE 中快 100%. 我们发现，SHARE 的下载速率普遍比 REPLAY 快，这是因为 Granary 目前还处于测试运行阶段，REPLAY 所使用的日志的访问频率比较小，而在 SHARE 中一个文件被更多的用户下载，访问频率较高. 一般来说，一个文件的访问频率越高，越有利于优化复本的放置，平均下载速度也就越快. 从 SHARE 中下载速率的 CDF 分布图(图 10)可以很容易地看出

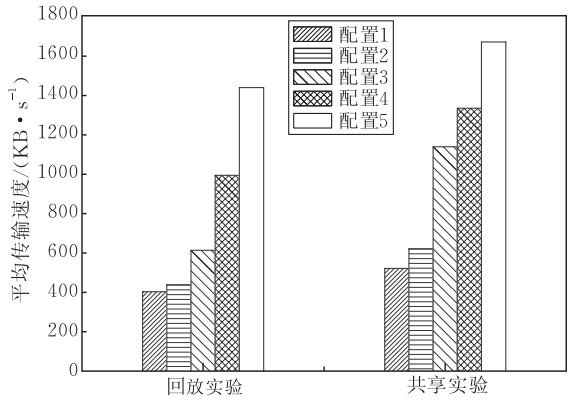


图 9 PlanetLab 上仿真实验中记录的用户平均下载速率

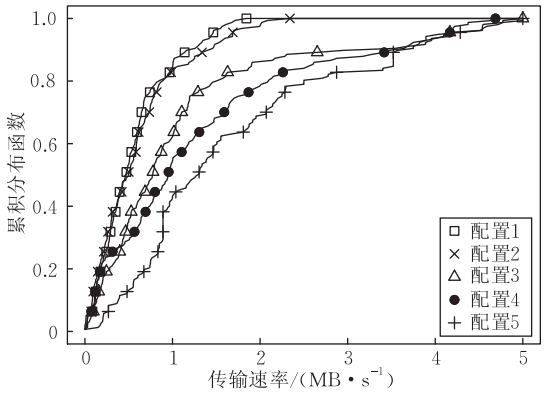


图 10 PlanetLab 上的共享仿真实验(SHARE)中客户端下载速率的 CDF 图

不同算法的性能差别. Near and popular 的下载速率的 CDF 曲线非常接近客户端与服务器之间最大带宽(max bw)的 CDF 曲线，说明算法的效果十分接近于最优值.

7 结 论

越来越多的用户在互联网上共享他们的文件，包括音频和视频文件. 针对专门的文件共享系统比如 BitTorrent，已经有不少的研究者试图改进，但是如何在一个 P2P 存储系统中提高文件共享的速度，研究的人并不多. 本文提出的基于流行度和位置信息的自适应复本算法 PLAR 试图解决这个问题. PLAR 将热门的文件复制很多份，并把这些复本放置到距离最需要的用户附近，这样可以降低网络带宽的消耗并提高文件下载的速度. 模拟和仿真的实验结果证明了 PLAR 具有比较好的效果.

参 考 文 献

[1] Dabek F, Kaashoek M F, Karger D et al. Wide-area cooperative storage with CFS//Proceedings of the 18th ACM Symposium on Operating Systems Principles(SOSP'01). Chateau Lake Louise, Banff, Canada, 2001: 202-215

[2] Rowstron A, Druschel P. Storage management and caching in PAST, a large-scale, persistent peer-to-peer storage utility//Proceedings of 18th ACM Symposium on Operating Systems Principles(SOSP'01). Chateau Lake Louise, Banff, Canada, 2001: 188-201

[3] Clarke I, Sandberg O, Wiley B et al. Freenet: A distributed anonymous information storage and retrieval system//Proceedings of the Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability. Berkeley, CA, USA, 2000. Berlin/Heidelberg: Springer-Verlag, 2001: 46-66

[4] Kubiatowicz J, Bindel D, Chen Y et al. OceanStore: An architecture for global-scale persistent storage//Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-IX). Cambridge, MA, USA, 2000: 190-201

[5] Andrews M, Shepherd B, Srinivasan A et al. Clustering and server selection using passive monitoring//Proceedings of the 21th Annual IEEE Conference on Computer Communications (INFOCOM'02). New York, USA, 2002: 1717-1725

[6] Park K, Pai V S. Scale and performance in the CoBlitz large-file distribution service//Proceedings of the 3rd Symposium on Networked Systems Design and Implementation (NSDI 2006). San Jose, CA, 2006: 3-3

[7] Sripanidkulchai K, Ganjam A, Maggs B et al. The feasibility of supporting large-scale live streaming applications with

dynamic application end-points//Proceedings of SIGCOMM04. Partland, Oregon, USA, 2004: 107-120

[8] Tang W, Fu Y, Cherkasova L et al. MediSyn: A synthetic streaming media service workload generator//Proceedings of the 13th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV03). Monterey, CA, USA, 2003: 12-21.

[9] Yu H, Zheng D, Zhao B Y et al. Understanding user behavior in large scale video-on-demand systems//Proceedings of the 1st EuroSys Conference (EuroSys'06). Leuven, Belgium, 2006: 333-344

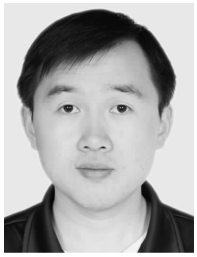
[10] Loeser C, Schomaker G, Brinkmann A et al. Content distribution in heterogeneous video-on-demand P2P networks with ARIMA forecasts//Proceedings of 4th International Conference on Networking(ICN 2005). Reunion Island, France, 2005: 800-809

[11] Breslau L, Cao P, Fan L et al. Web caching and Zipf-like distributions: Evidence and implications//Proceedings of the 18th Annual IEEE Conference on Computer Communications (INFOCOM'99). New York, USA, 1999: 126-134

[12] Chervenak A L, Patterson D A, Katz R H. Choosing the best storage system for video service//Proceedings of the 3rd ACM International Conference on Multimedia (Multimedia'95). San Francisco, CA, USA, 1995: 109-119

[13] Xiang Z, Zhang Q, Zhu W et al. Peer-to-peer based multimedia distribution service. IEEE Transactions on Multimedia, 2004, 6(2): 343-355

[14] Mogul J C. Emergent (Mis)behavior vs. Complex Software Systems//Proceedings of the 1st EuroSys Conference (EuroSys'06). Leuven, Belgium, 2006: 293-304



CHEN Kang, born in 1976, Ph. D.. He is currently working on distributed systems, storage systems.

YU Hong-Liang, born in 1976, Ph. D., associate professor. His research interests include distributed systems, peer-to-peer computing.

ZHANG Kun, born in 1983, master. His research interests focus on peer-to-peer networking.

Background

This paper belongs to the research filed of peer-to-peer computing. Peer-to-peer computing is one form of distributed computing that does not rely on any central nodes to provide services. The applications of peer-to-peer computing include the file sharing, data streaming as well as data backup. The researchers have done several works on the infrastructure as well as the applications of peer-to-peer computing. The research of peer-to-peer infrastructure is to make a model of underlying network on how to provide the basic interconnection network. Currently, there are two kinds of network infrastructure; one is structured peer-to-peer network which constructs the network with some structures such as distributed hash tables. The other is un-structured peer-to-peer network which does not contain any structure of the network and based on some technologies like message flooding or broadcasting. Structured peer-to-peer is more interesting and researchers have built and used various structured. The authors Granary system is also based on the structured peer-to-peer network for data storage and sharing. Some other researches of peer-to-peer computing focus on the applications of such computational model. One important of application is the data storage to wide area network. The authors' work is also about the data storage and sharing. Other applications like content delivery network, data backup as well as media streaming. Current work focuses on some aspects of storage like data availability, data consistency as well as data location. The authors' work is on another aspect as to improve the download capability of the file sharing system. This is quite important while the sharing of objects is practically used. User experiences are getting better with fast downloading capability. The method is called PLAR. PLAR is based on the locality and popularity of objects as well as applying the mixed server selection method and remote boosting schemas. PLAR is used in our Granary system which is data storage system for wide area network. The experiment results show that PLAR can improve the download speed over 60% on average which means that it can improve the sharing as well as reducing the network traffic. This work was supported by the National Natural Science Foundation of China under grant Nos. 90718040 and 60603071, the National Basic Research Program of China under grant No. 2007CB310900, and the National High Technology Research and Development Program (863 Program) of China under grant No. 2008AA01Z112. Those algorithms are meaningful for improving the reliability of network applications as well as reducing the network traffic. The team has done several works on the field of peer-to-peer computing. Granary was built to show the capability of storing the data reliably on the wide area network. The internal structure as well as some other issues such as data distribution, object location has been discussed in some other papers. This paper focuses on the features for improving the user experience with improved downloading capability.