Mapper:一种基于组播的 Peer-to-Peer 文件匿名访问协议

叶保留,顾铁成,吴敏强,陆桑璐,陈道蓄
(南京大学计算机软件新技术国家重点实验室,江苏南京 210093)

摘 要: Peer-to-Peer (P2P) 文件系统的一个基本问题是如何在保护节点隐私的基础上为数据访问提供高效服务. Mapper 将 IP 组播技术和多级代理转发技术相结合,解决了 P2P 文件访问的相互匿名问题. 协议还通过 MRFC 算法将组成员筛选和缓存位置选择有机统一,在减少组播开销的同时保证了文件布局对用户访问模式的动态自适应性. 实验表明,Mapper 能有效缓解网络负载,提高数据的易获取性,具有良好的伸缩性和自适应性.

关键词: P2P; 匿名; IP 组播; 缓存
中图分类号: TP393 **文献标识码**: A **文章编号**: 0372-2112 (2004) 05-0754-05

Mapper:A Multicast-Based Peer-to-Peer File Anonymous Retrieval Protocol

YE Bao-liu, GU Tie-cheng, WU Min-qiang, LU Sang-lu, CHEN Dao-xu
(*State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, jiangsu* 210093, *China*)

Abstract: A fundamental problem of Peer-to-Peer file sharing system is how to protect the privacy and anonymity of nodes while providing efficient data access service. We presented a Multicast-based Peer-to-Peer file anonymous retrieval protocol called Mapper. By the combination of IP multicast technology and Multi-proxy forwarding technology, Mapper satisfies the requirements of mutual anonymity during the file access process. The protocol also integrates group member selection with cache position selection via MRFC (Most Requested Frequency Caching) algorithm, thus file distribution can be adjusted adaptively with respect to dynamic usage patterns and multicast overhead can be controlled. The experimental results show that Mapper can alleviate network traffic, reduce access delay efficiently, and has the merits of scalability, reliability and adaptability.

Key words: Peer-to-Peer; anonymity; IP Multicast; cache

1 引言

Napster[1]的流行将 P2P 研究推向了分布式计算领域的中心. P2P 通常指在非集中式控制环境下聚合 Internet 上分布资源(计算能力、存储空间等),完成一些关键功能(分布式计算、数据/目录共享、协作服务等)的应用系统. 典型的 P2P 系统是一种大规模、动态变化、无集中控制的自组织网络,系统内所有节点都是完全对等体,同时扮演服务器和客户机角色,节点间可直接通信、共享信息. P2P 计算模型具有低代价、高可用等特点,被认为是改善 Internet 现状、提升 Internet 计算能力的有效途径.

P2P 网络的开放特征和对等通信模式要求在提供高效数据访问服务的同时,提供匿名访问机制,保护节点隐私,避免恶意跟踪和攻击. TCP/IP 协议设计虽充分考虑了 Internet 上与网络性能相关的技术问题,但 IP 报文中的地址信息使节点隐私丧失,网络窃听者可轻易获取通信双方的实体信息. 文献[4~8]针对 IP 网络提出了 Internet 匿名保护方案,但相关工作旨在解决请求发起者匿名(Initiator anonymity)问题,未考虑应答者匿名(Responder anonymity). IP 组播的一个重要特征是可通过一个 D 类 IP 地址来标识所有组内成员,屏蔽特定实体信息. 此外,IP 组播可在传输路径的分支节点自动复制报文、实现多点并发投递. 因此,将 IP 组播用于 P2P 文件共享系统能有效解决相互匿名问题,且有很好的实现效率.

本文针对分布式无结构 P2P 文件共享系统提出了一种基于 IP 组播的数据访问协议——Mapper. Mapper 的主要特点在于:首先,将多级代理转发技术和 IP 组播技术结合,使发送者匿名和应答者匿名在文件访问中同时得到满足,实现了通信实体间的相互匿名(Mutual anonymity);其次,通过 MRFC 算法将组成员筛选方法和数据缓存位置选择策略相结合,建立自适应缓存机制,降低了数据传输延迟、缓解了网络负载.

收稿日期:2003-05-22;修回日期:2003-11-05
基金项目:国家重点基础研究发展规划项目(No. 2002CB312002);国家 863 高科技项目(No. 2001AA113050)

2 背景

2.1 匿名性

匿名访问在 IP 报文传输中就被提出,但早期网络中服务节点相对固定且公开,协议设计侧重于对请求者 IP 地址的匿名保护. P2P 系统的资源开放性和通信对等性对匿名保护提出新的要求. 文献[2]根据 P2P 系统的应用特征,定义出作者(author)匿名、发布者(publisher)匿名、读者(reader)匿名、服务器(server)匿名、文档(document)匿名、查询(query)匿名等六种匿名需求. 实际通信中通常通过发送者匿名、应答者匿名、相互匿名等三种匿名方式保护节点隐私. 发送者匿名要求消息的接收者无法识别消息的发起者,应答者匿名则要求请求发送者无法判断信息服务的提供者. 发送者匿名和应答者匿名都只考虑了通信的一方,而相互匿名要求使发送者匿名和应答者匿名同时得到满足. 在 P2P 文件共享系统,发送者通常指文件的读者,而应答者常指回送目标文件的写者,理想的 P2P 系统应支持相互匿名.

2.2 相关工作

为实现 Internet 报文的发送者匿名,Anonymizer[3]、LPWA[4]在请求者与服务者之间引入单级代理(Single proxy)转发报文,隐藏请求者地址. 该方法的缺点是不能保证发起者对代理的匿名. Mixes[5]通过报文重序提供对发送者匿名的支持. Onion Routing[6],Crowds[7],Horders[8]等协议采用串行化多级代理技术,在请求者与服务者之间建立报文转发路径实现发送者匿名. 这些协议的设计差异主要体现在路径选取方法、报文加密方式等方面. 多级代理技术虽加强了发送者匿名度,但增加了系统开销[7]和访问延迟,而且上述方法的基本假设都是无需支持应答者匿名.

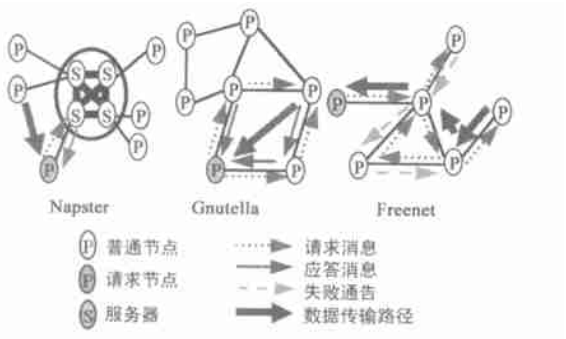

图 1　典型 P2P 系统的文件定位过程

近两年来,研究人员也针对 P2P 文件共享系统的匿名需求提出了一些解决方法. Napster 常被认为是最早的 P2P 应用系统,所有节点信息都通过集中式目录来维护和发布,未对系统安全和节点隐私提供任何保护. Gnutella[9]针对全分布无结构 P2P 网络,采用泛洪方式定位服务节点,通过请求者与服务者的直接通信实现文件传输. Gnutella 能在搜索阶段保证相互匿名,但在文件传输方式使相互匿名丧失. Freenet[10]是分布式结构化 P2P 文件共享存储系统,在文件的读、写阶段都采用多级代理转发机制维护相互匿名. 但这种全链路通信方式和全程缓存技术使系统具有较大访问延迟和资源开销,且文件传输存在单点失败隐患. 图 1 给出了上述三种 P2P 系统文件访问过程示意图. APFS[11]是第一个引入 IP 组播技术的 P2P 匿名文件共享系统,但与本文的实现机制不同. APFS 通过 Onion Routing 解决相互匿名问题,IP 组播主要用于会话管理. 与本文工作建立的动态服务相比,APFS 基于集中式目录服务结构,没有考虑数据的动态部署,缺乏灵活性和自适应性,且匿名度会随时间推移而下降[11].

3 Mapper 设计

Mapper 针对全分布无结构 P2P 文件共享系统,采用基于多级代理动态转发机制维护目标文件定位过程中的相互匿名. 借助 IP 组播技术维护传输阶段的相互匿名. 基于 Mapper 的文件访问过程实际上也是一个组播会话创建、活动、消亡的动态过程.

3.1 访问流程

为获取目标文件,请求者首先查询本地系统. 如果没有目标文件,则申请一个组播地址 IP_m,随机产生一个 64 位长的组会话标志 grp_id,根据目标文件键值 $file\_id$ 和预设的搜索范围 TTL 创建查询消息四元组 $msg\langle grp\_id, file\_id, TTL, IP_m\rangle$. 随后启动组播会话,进入消息路由. 中间节点收到查询消息后首先判断本地系统有无目标文件,若存在,则结束消息路由,通过 IP_m 发送目标文件;否则,采用基于键值相似度的回溯算法查询本地路由信息表,确定下一转发节点. 转发前,节点将根据组成员筛选算法判断是否将当前节点加入组播组 grp_id. 路由终止的条件是找到目标文件或超出搜索范围. 文件访问流程如下:

(1) 初始化:请求者获取文件健值 $file\_id$,设置 TTL 值,生成一个随机组播会话标识 grp_id,申请并加入组播地址 IP_m,创建消息四元组 $msg\langle grp\_id, file\_id, TTL, IP_m\rangle$,进入消息路由阶段.

(2) 消息路由:(a) 执行本地查找,若发现目标文件,则更新该文件的最近访问时间记录,转(4);(b) 查询路由表,确定下一转发节点 N,若转发长度超出 TTL 许可范围或转发节点 N 为空,则转 5;(c) 执行组成员筛选算法决定是否将当前节点加入组播组 IP_m,转发请求到节点 N.

(3) 重复执行(2).

(4) 文件传输:(a) 当前节点加入组播地址 IP_m,通过 IP_m 向组成员发送目标文件;(b) 文件接收者在请求转发频率表删除相应记录项,并路由表中添加到数据源的路由信息,转(6).

(5) 查询失败处理:当前节点加入组播地址 IP_m 并向组成员发送查询失败消息.

(6) 会话结束:当前节点通过 IP_m 发送会话结束通知.

3.2 路由查找

Mapper 中的每个节点都维护一张由文件名键值和转发节点两字段组成的路由信息表,记录经由该节点转发的请求消息. 消息路由实际上是对一个动态有向图的遍历过程. Mapper 根据目标文件与本地路由表文件名键值相似度优先关系确定转发节点的选择顺序,并结合深度优先算法提出基于键值相

似度的回溯算法,算法 1 描述了算法流程.

算法 1 基于键值相似度的回溯算法

```
Node keyRouting(grp_id, file_id, TTL, IP_m, P)
{
  if(TTL < 0)
  {
    output("请求失败");
    joinGroup(P, IP_m, grp_id); '当前节点加入组播
    return(P); '返回节点 P,结束路由
  }
  if(发现目标文件)
  {
    output("请求成功");
    joinGroup(P, IP_m, grp_id);
    return(P); '返回节点 P,结束路由
  }
  Visited[P, grp_id] = true; '设节点已被访问标志
  memSelect(file_id, P, IP_m, grp_id); '组成员筛选
  routeSort(P, file_id); '按相似度排序路由记录
  N = firstRT(P); '获取第一个转发节点
  While(N <> null && visited[N, grp_id] = = false)
  {
    keyRouting(grp_id, file_id, TTL-1, IP_m, N);
    N = nextRT(P); '获取下一转发节点
  }
}
```

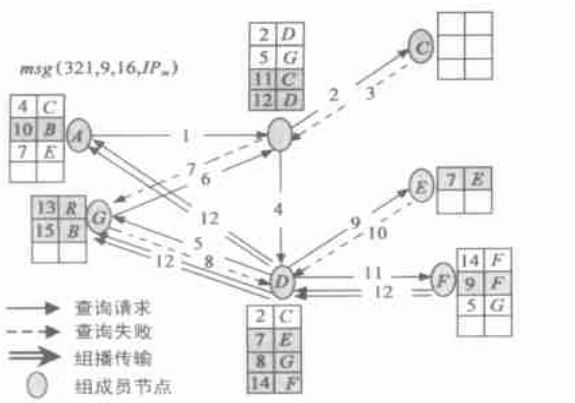

图 2 数据获取过程

以图 2 为例,节点 A 发出对键值为 9 的文件服务请求.根据算法 1,A 将请求消息转发到节点 B;B 不存在目标文件,继续转发请求至节点 C;C 中没有任何路由信息,返回请求失败消息给 B;B 据键值相似度优先关系将请求信息转发给第二个侯选节点 D;D 根据深度优先原则将请求转发到节点 G;G 继续转发信息至 B;B 检测到环路径,返回请求失败消息至 G.随后,请求消息依路由查找算法顺次遍历图 2 中节点 D、E、D,最后在节点 F 处找到目标文件,结束路由查找.

3.3 组成员筛选

研究表明,一定时段内用户对热门文件的访问往往呈现一定分布规律(如对 Web 文件和流媒体文件的访问都符合 Zipf 分布).因此,根据用户请求模式,为热门文件合理设置缓存,可避免节点状态变化产生的请求失败,并满足负载均衡需要.Mapper 将缓存位置的选择和组成员筛选相结合,通过 MRFC 算法在目标文件拥有最大访问频率的当前节点设置缓存,并使其加入组播组.算法目标是尽可能在客户区域为热门文件设置缓存,提高访问的可达性.

为准确反映访问模式的变化趋势,协议综合当前节点对文件请求转发的历史信息和近期行为,建立了基于低通过滤的转发频率计算公式:

$$\lambda(file\_id) = \alpha\lambda_{avg}(file\_id) + (1-\alpha)\lambda_{new}(file\_id) \quad (1)$$

其中 $0 \le \alpha \le 1$,$\lambda(file\_id)$ 为过滤后对 $file\_id$ 的转发频率,$\lambda_{avg}(file\_id)$ 指上一周期的平均转发频率,$\lambda_{new}(file\_id)$ 是最近一个周期内文件请求的转发频率.常量 α 是权重系数.系统内每个节点都维护一张由 $file\_id$、$\lambda_{avg}(file\_id)$、$\eta(file\_id)$(当前周期内转发频次)、$\lambda(file\_id)$ 四个域组成的请求转发频率表.各节点每隔周期 T 按公式(2)、(3)更新转发频率表的所有记录,并开始一个新的统计周期.

$$\lambda_{avg}(file\_id) = \eta(file\_id)/T \quad (2)$$

$$\eta(file\_id) = 0 \quad (3)$$

算法 2 MRFC 算法

```
memSelect(file_id, P, IP_m, grp_id)
{
if (extFwdRec(file_id) = = null) '转发记录存在否
  {
    addFwdRec(file_id); '增加记录项
  }
  else
  {
    λ_new(file_id) = (η(file_id)/(△t)
    λ(file_id) = compFreq(file_id);
    updateFwdRec(file_id); '更新 file_id 的记录项
    if(max(λ(file_id))) '对最高转发频率情形处理
    {
      getCacheSpc(); '用 LRU 获取自由空间
      joinGroup(P, IP_m, grp_id); '加入组播组
    }
  }
}
```

4 匿名度验证

C Shields 等人在文[8]中对匿名度的量化判断作出如下定义:设 x 是某条通信链路发起者的概率为 $Pr_e(x)$,其中 $x \in S$,$S \neq \Phi$,且 $\sum_{y \in S} Pr_e(y) = 1$,则对匿名协议 A 而言,节点 x 相对其他实体 e 的匿名度 $d_{x,e}(A)$ 可由式(4)表示.

$$d_{x,e}(A) = \sum_{y \in S \neq x} Pr_e(y) = 1 - Pr_e(x) \quad (4)$$

协议 A 相对协作实体集 S 的总体匿名度可由下式定义:

$$d(A) = \min\{d_{x,e}(A)\},\ \forall e \in E,\ \forall x \in S \quad (5)$$

其中 S 指由协议 A 维护匿名性的实体集,E 是网络中所有成员.

根据上述定义和 Reiter 和 Rubin 对匿名度的分类[7],我们对 P2P 系统的匿名度等级作下述形式化描述:

完全暴露(Provably exposed):攻击者可证实 x 是发送者(或接收者). $d_{x,e}(A)=0$.

暴露(Exposed):存在 x 不是发送者(或接收者)的可能. $0<d_{x,e}(A)<0.5$.

可能清白(Probable innocence):x 是否为发送者(或接收者)的可能相当,但与其他实体相比,有更高的概率可能. $0.5\leq d_{x,e}(A)<d_{y,e}(A)$ 且 $d_{x,e}(A)<1-1/|S|$, $\forall y\neq x\in S$.

超出怀疑(Beyond Suspicion):x 不比系统中其他实体具有更高的概率可能是发送者(或接收者). $|S|>1$, $1-1/|S|\leq d_{x,e}(A)$, $d_{y,e}(A)\leq d_{x,e}(A)$, $\forall y\neq x\in S$.

完全隐私(Absolute privacy):攻击者不能发现当前通信. 此时有 $|S|\equiv\infty$, $d_{x,e}(A)=1$.

表 1 对 Mapper 在不同阶段的匿名度进行了总结. 在搜索阶段,对协作式攻击节点而言,由于任何中间节点既不能推断前趋节点是否是发送者,也不能判断后继节点是否为应答者,每个节点作为通信链路起止点的概率相等. 所以,对任何节点 x, y 都有 $Pr_e(x)\leq 1/|S|$, $Pr_e(y)\leq 1/|S|$,且 $d_{y,e}(A)\leq d_{x,e}(A)$ 成立. 据定义知,发送者和接收者的匿名度都为"超出怀疑". 对于本地窃听者攻击情形,由于接收转发消息的第一个节点充当了本地窃听者,因此,存在节点 x 使得 $Pr_e(x)>0.5$ 成立,故发送者匿名度将降至"暴露"级别. 在文件传输阶段,共享组地址特征使得对任何组成员 x, $d_{x,e}(A)=1-1/|S|$ 都成立,因而发送者和接收者对两种攻击者都能保持"超出怀疑"匿名度.

表 1　Mapper 的匿名度

	攻击者	发送者匿名度	应答者匿名度
查询阶段	本地窃听	暴露	超出怀疑
	协作节点	超出怀疑	超出怀疑
传输阶段	本地窃听	超出怀疑	超出怀疑
	协作节点	超出怀疑	超出怀疑

5　性能分析

5.1　网络收敛性

我们采用文献[10]的实验环境对 Mapper 的收敛性进行了验证. 图 3 给出了请求路径长度随时间的变化趋势曲线. 运行初期,文件分布的单一性限制了请求的成功率,访问步长较大. 但随时间推移,缓存技术使文件布局据用户请求情况自适应调整,网络逐步收敛,并最终趋于平稳. 图 3 表明,Mapper 和 Freenet 收敛状态相似,但由于 Mapper 采用了选择性局部缓存策略,因此收敛速度相对较慢. 而 Freenet 的全程缓存策略虽赢得了收敛速度,但牺牲了网络带宽,增加了传输延迟.

5.2　伸缩性

Mapper 的可伸缩性主要受缓存算法和消息路由机制的影响,可伸缩能力主要表现为平均请求路径长度随系统规模的变化关系. 以 5.1 节实验为基础,通过周期性向系统注入新节点,图 4 给出了稳定状态下平均请求路径长度与网络大小的关系曲线. 该图表明,Mapper 的平均搜索路径长度受网络规模影响较小,平均请求路径长度近似于 $O(\log N)$(N 为网络节点数).

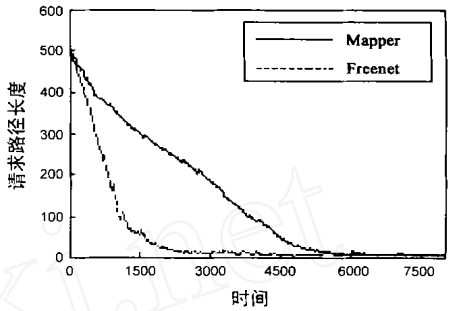

图 3　请求路径长度变化曲线

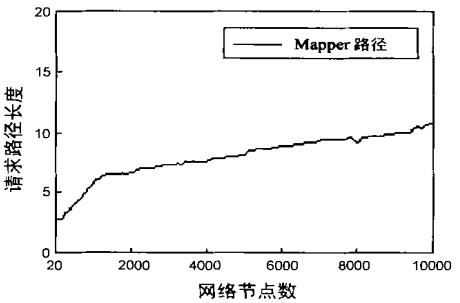

图 4　请求路径平均长度与网络大小关系曲线

平均请求路径长度决定了组会话的大小. Mapper 的收敛性和可伸缩性表明,稳定状态下组播规模不会很大,而自适应缓存策略又进一步限制了组播范围. 因此,Mapper 能有效抑制组播开销,且有较好的协议效率.

5.3　网络连接特征

根据对稳定状态下路由表信息的统计和分析,系统中大多数节点拥有较少的路由信息,同时也有少数节点的路由信息会呈现出"Hub"状态. 该现象表明网络中大多数节点只拥有少量邻居节点,而少数节点具有较大连接度. 因此,Mapper 满足界定"小世界(Small-world)"网络拓扑的两个关键特征[12]:具有较小的平均访问长度和独立于网络大小的较大的聚合系数.

5.4　可靠性

Mapper 的可靠性可从三个方面得到保证. 首先是协议设计的网络结构特征. Mapper 节符合"小世界"网络拓扑特征,随机失败大多发生于弱度节点,不会造成网络分区现象. 其次,缓存机制. 缓存为文件提供了多点备份,一方面能有效避免单点失效引起查询失败,另一方面还能实现负载均衡. 最后,IP 组播的引入. 避免了链式返回方式中单点失败引起的传输失败问题,使系统具有很强的稳健性. 假设每个节点都具有相同的失败概率 P_r,则对链式传输而言,传输失败的概率为 $1-(1-P_r)^s$,失败概率随 s 增加而增加,而 IP 组播方式只与请求

者相关,失败率恒为 P_r.

6 结束语

与已有工作相比,Mapper 设计的主要特点体现在以下两个方面:首先,将多级代理转发技术和 IP 组播技术相结合,实现了 P2P 文件访问的相互匿名.运行时通过动态路由和动态组管理机制有效避免了恶意跟踪,使得匿名度不受时间影响,进一步增强了系统安全.其次,在组成员筛选过程中集成缓存策略,建立了基于访问频次的组成员筛选算法,使得系统能根据用户访问模式的变化对文件布局作动态自适应调整,透明地迁移和复制数据文件,保证了协议效率.实验表明,Mapper 具有良好的收敛性、可靠性和可伸缩性.

参考文献:

[1] Oram A. Peer-to-Peer: Harnessing the Benefits of a Disruptive Technology[M]. California: O'Reilly and Associates, Inc, Mar 2001.

[2] Dingledine R, Freedman M J, Molnar D. The free haven project: Distributed anonymous storage service[A]. In Proceedings of International Workshop on Designing Privacy Enhancing Technologies: Design Issues in Anonymity and Unobservability[C]. Berkeley. CA: Springer-Verlag. 2001. 67 - 95.

[3] http://www.anonymizer.com.

[4] Gabber E, Gibbons P, Kristol D, et al. Consistent, yet anonymous, web access with LPWA[J]. Communications of the ACM, 1999, 42(2): 42 - 47.

[5] Chaum D. Untraceable electronic mail, return addresses, and digital pseudonyms[J]. Communications of the ACM, 1981, 24(2): 84 - 88.

[6] Reed M, Syverson P, Goldschlag D. Proxies for anonymous routing[A]. In Proceedings of 12th Annual Computer Security Applications Conference[C]. San Diego, CA: IEEE CS Press, December 1996. 95 - 104.

[7] Reiter M K, Rubin A D. Crowds: Anonymity for web transactions[J]. ACM Transactions on Information and System Security, 1998, 1(1): 66 - 92.

[8] Shields C, Levine B N. A protocol for anonymous communication over the Internet[A]. In proceedings of 7th ACM Conference on Computer and Communications Security[C]. Athens, Greece: ACM Press, 2000. 33 - 42.

[9] Clips. The Gnutella Protocol Specifications v0.4[EB/OL]. http://dss.clip2.com/GnutellaProtocol04.pdf.

[10] Clarke I, Sandberg O, Wiley B, et al. Freenet: A distributed anonymous information storage and retrieval system[A]. In Proceedings of International Workshop on Designing Privacy Enhancing Technologies: Design Issues in Anonymity and Unobservability[C]. Berkeley. CA: Springer-Verlag, 2001. 46 - 66.

[11] Searlata V, Levine B N, Shields C. Responder anonymity and anonymous peer-to-peer file sharing[A]. Proceedings of the 9th International Conference on Network Protocols (ICNP 2001)[C]. Riverside, CA: IEEE Computer Society. November 2001. 272 - 280.

[12] Iamntichi A, Ripeanu M, Foster I. Locating data in (small-world?) peer-to-peer scientific collaborations[A]. In the Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS'02)[C]. Cambridge, MA: Springer-Verlag. 2002. LNCS(2429). 232 - 241.

作者简介:

叶保留 男,1976 年生于江苏如东,现为南京大学计算机科学与技术系软件理论专业博士研究生,主要研究方向为分布式计算与并行处理.

顾铁成 男,1965 年生于江苏南京,现为南京大学计算机科学与技术系软件理论专业博士研究生,主要研究方向为分布式计算与并行处理.