

面向动态环境下兴趣转移的 P2P 自适应搜索^{*})

唐九阳 汤大权 肖卫东 张维明

(国防科技大学信息系统与管理学院 长沙 410073)

摘要 P2P 环境的高度动态特性导致了搜索性能的不稳定。针对节点的兴趣变化对搜索性能的影响,利用节点的自治和系统自组织特性,提出了一种动态自适应搜索机制 SAS。基于资源的合理组织以及查询历史统计信息,通过消息转发的智能性,及时反映资源分布以及搜索内容的动态变化,从而迅速定位资源提供节点,提高了搜索的性能和稳定性。实验分析表明,SAS 可以自动优化搜索性能,具有很好的自适应性。

关键词 对等网,兴趣,自适应,搜索

Self-adaptive Search towards Interests Transformation in Dynamic Peer-to-Peer Network

TANG Jiu-yang TANG Da-quan XIAO Wei-dong ZHANG Wei-ming

(School of Information System and Management, National University of Defense Technology, Changsha 410073, China)

Abstract High dynamism in P2P network consults in unstable search performance. Aiming at dealing with the influence of interests transformation on peer's search performance, a self-adaptive search mechanism, named SAS, was proposed. Based on the reasonable resource organization and historic search statistics information, SAS can immediately reflect the dynamic change of resource distribution and search content by intelligent message route, which improves the search efficiency. Experiment shows that the SAS can automatically improve search performance, and meanwhile guarantee the search stableness.

Keywords P2P, Interest, Self-adaptive, Search

1 引言

同任何大规模的分布式系统一样, P2P 系统的成功与否在很大程度上取决于其资源搜索机制的灵活性和可扩展性, 分布式资源搜索机制一直是 P2P 系统研究最活跃的领域之一^[1,2]。

高度动态是 P2P 环境的一个显著特点, 其中动态性包含两个层面的内容: 一是网络节点进入网络系统的动态性, 即节点可以自由地加入和退出网络; 二是网络节点兴趣的动态性, 由于节点的高度自治, 节点自身共享的资源以及节点的资源需求(通过节点的资源搜索反映)可能经常变化。虽然已经存在大量 P2P 环境下的搜索技术, 但是这些方法很少考虑动态环境对搜索性能造成的影响。

通过对查询的内容进行聚类, 文献[3]提出基于内容的智能搜索方法, 该方法利用历史信息, 借助查询反馈机制进行性能优化, 虽然该方法在保证查询效果的前提下提高了搜索的效率, 但系统的拓扑结构未发生改变。文献[4]提出了一种基于兴趣域的搜索机制, 通过将拥有相同元数据的节点划分为一个兴趣域, 搜索请求首先在兴趣域中传播, 随着搜索过程的进行, 节点通过对兴趣域内其它节点了解信息的增多来提高搜索效率。文献[5-8]中节点基于查询反馈, 维护对搜索有帮助的文件存放信息, 作为对后来的路由过程的启发。值得注意的是, 上述方法中节点的逻辑拓扑不随搜索过程发生改变, 难以从根本上提高系统搜索性能的动态适应性。

文献[9]提出了在 P2P 系统中使用基于节点兴趣的搜索

机制, 基于节点兴趣局部性原理, 在具有相同或相似兴趣的节点间建立捷径(shortcut), 搜索首先利用曾经提供过搜索结果的快捷节点, 失败后采取广播搜索方式。这一方案通过把搜索中的第一跳指向拥有相同兴趣的节点来提高搜索效率。文献[10]将邻居分为直接邻居和间接邻居(邻居的邻居), 选择以往表现好的节点作为直接邻居。经过一段时期的调整后, 兴趣相近的节点之间便会建立起直接连接。文献[11]从历史搜索结果中学习节点之间兴趣的相似度, 将节点按兴趣分类, 在具有相似兴趣的节点之间建立朋友关系, 搜索请求首先转发给朋友。该方法存在的问题是倘若节点的查询不是自身的兴趣, 那么建立的朋友关系对搜索帮助不大, 仍然类似 Gnutella 网络。

针对节点的兴趣变化对搜索性能造成的影响, 本文提出了一种动态自适应搜索机制, 详细描述了搜索过程和冗余消息的处理策略, 最后通过仿真实验对自适应搜索机制进行了分析。

2 动态环境对搜索性能的影响分析

本文的 P2P 网络拓扑采用文献[12]提出的 REC 模型。模型中的节点根据能力异构分为服务节点和普通节点, 普通节点和服务节点间的拓扑连接基于节点参与资源共享的不同动机, 分为资源连接和需求连接。具体见文献[12]。

本文主要关注 P2P 网络节点兴趣的动态性对搜索性能的影响, 因此, 从节点服务他人和服务自身的双重视角看, 节点的兴趣变化分为两类:

^{*}) 基金项目: 国家自然科学基金资助(60172012), 湖南省自然科学基金资助(03JJY3110)。唐九阳 博士, 讲师, 主要研究方向为对等计算、信息集成、知识管理。

(1)共享资源的变化。一般节点的共享资源较稳定,共享资源一旦确定后,很长时间内都不会变化。如学习钢琴演奏的用户初期钟爱理查德的作品,在相当长的时期内搜集了大量理查德的演奏曲,并将它们发布。但随着用户对钢琴曲理解程度的加深,逐渐对莫扎特的作品产生浓厚的兴趣,加入相应莫扎特作品讨论的论坛,共享莫扎特钢琴曲并交流感受。因此当节点的共享资源发生变化时,要求节点能够自动、及时地加入新资源所属的社区。

(2)资源需求的变化。节点的资源需求则通过节点的查询主题反映,类似互联网上的搜索,节点的查询主题可能是任意的,而且快速多变、时序性和重复性不强,譬如节点的查询主题由某段时期对数码相机的关注转移为对高保真音响的爱好。针对节点的这种短期不确定行为,要求节点能够即使不在需求资源所在的社区中仍然可以快速地查找到所需资源,同时保持自身资源所属的社区不变化。

下面通过一个例子进行相应说明。

如图1所示,节点 Peer_{2a} 原来属于社区 A,由于节点的自主性,节点提供与社区 D 主题相关的共享资源(设为 r)。假设根据拓扑进化,节点 Peer_{2b} 建立了与服务节点 D 的需求连接,而且社区 D 内已经完成了资源 r 的聚集,那么在节点 Peer_{2a} 的共享资源变化前,Peer_{2b} 针对资源 r 的查询只需通过需求连接在 1 跳内即可完成。由于 Peer_{2a} 共享资源的变化,如果缺少动态自适应机制引导节点 Peer_{2a} 加入社区 D,节点 Peer_{2b} 需要遍历整个网络直至定位到服务节点 A 才能搜索到网络上所有关于 r 的资源。显然,节点的搜索性能将有所下降。

基于图1,我们进一步考虑节点资源需求动态变化的情况。假设图中社区 B 为资源 s 的聚集,社区 D 仍为资源 r 的聚集。当节点长期重复的查询主题与共享资源不一致,如 Peer_{2b} 在很长一段时间内反复查询资源 r ,应用传统的兴趣转移调整方法^[9-11],Peer_{2b} 将脱离目前所属的社区 B,逐步加入到资源 r 的聚集社区 D。这种行为虽然满足了 Peer_{2b} 自身的查询需求,却是以破坏节点共享资源 s 的聚集为代价,相应导致其它节点针对共享资源 s 的搜索变得困难(原有针对 s 的搜索在社区 B 内即可完成,而今分布在两个社区 B、D 中,社区之间可能分隔很远),自然降低了其它节点的搜索性能。

因此,节点兴趣的动态变化对搜索性能的稳定造成了很大影响,需要设计相应的动态自适应搜索机制。

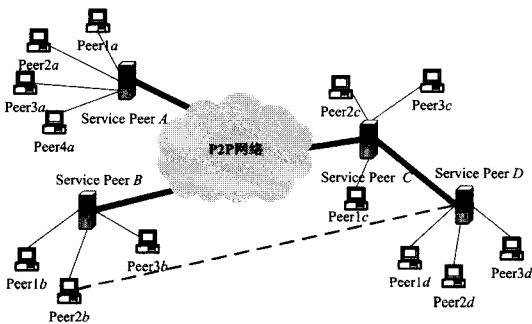


图1 动态环境分析

3 自适应搜索机制

面向动态环境下兴趣转移的自适应搜索的一个基本假设是所有的节点在搜索过程中都处在合适的位置上(即隶属于合适的社区),那么搜索请求可以以较高的概率被拥有相同兴

趣的节点满足。实际上根据节点周期性地运行拓扑自组织算法 SOMA^[13],该假设是成立的。由于节点兴趣由节点的共享资源显式表示和资源需求隐含表示,那么假设相应包含两层含义:如果相似资源的节点聚集在一个社区内,那么它被发现的概率较高,只需定位到该社区的服务节点就可找到所有满足查询请求的资源,相比资源散布在整个网络上的情况,相似资源节点的聚集缩小了查询范围,查询处理效率高;从资源需求的角度,如果一个社区可以满足前一个搜索请求,那么它满足当前搜索请求的概率也比较高,进一步,如果某个社区在过去的一段时间内比其它社区能够更好地满足当前节点的搜索请求,那么它满足当前搜索请求的概率也比较高。基于这个假设,节点只需通过需求连接建立与搜索请求对应的目标资源社区的捷径,就可缩短资源搜索跳数,减少社区间路由转发的次数。

因此,对于一个给定的查询,它的答案集很大程度上就在对应的社区(目标资源的聚集)而不是散落在整个系统,这样,资源搜索问题就转化成如何快速定位对应社区的问题。给出搜索的基本过程如下:

(1)确定资源所在的社区。当一个节点发起查询时,如果该节点是服务节点,跳(2);否则,节点首先计算查询项 q 与该节点维护的资源 r 的相似度 $\text{Sim}(q, r)$ ^[3],计算结果和预先设定的阈值(threshold)进行比较,这里分两种情况:如果计算结果大于或等于设定的阈值,则认为成功找到了匹配的资源,将与查询项 q 相似度大于等于阈值的资源作为查询结果返回,同时该节点将查询消息发送给资源连接的服务节点;如果查询项与该节点维护的资源的相似度小于设定的阈值,查询消息则传递给需求连接对应的服务节点。

(2)确定查询内容所在的具体位置。当查询消息到达后,接收到查询消息的服务节点先查看查询内容是否在本机上,然后根据它维护的资源索引确定内容是否在社区内的叶节点上,同时将查询按照一定策略发送给它的邻居服务节点。

直观上看,自适应搜索利用判断查询和当前节点所含资源的相似度,决定查询的转发路径:如果当前节点所含的资源和查询的相似度小于设定的阈值,那么该节点所属社区内的节点拥有和查询相关的资源的可能性也较小——因为根据节点面向兴趣转移的拓扑连接调整,同一社区内维护相似资源的节点。因此,借助历史反馈信息,将查询路由给需求连接的服务节点。反之,查询可能已经被发送到一个由一组查询主题相关的资源所在节点构成的社区中,当前社区内包含与查询相关的大部分资源对象。因此根据上述自适应搜索策略,对于给定的查询,目标社区的定位基本上可以在一个跳数内完成,有效控制了消息数量和搜索路径长度,从而提升了系统的整体搜索性能。

我们注意到,自适应搜索虽然增强了消息传播的针对性,但服务节点间的消息传播仍采用泛洪的方式,即服务节点向所有邻居服务节点广播查询消息,邻居服务节点再向自己的邻居服务节点广播,这个过程通过 TTL 的设置进行限定。该方法控制简单,但由于网络中存在环,即回路,P2P 搜索中会产生冗余搜索包。

为了防止查询消息陷入回路而被多次转发,许多系统中对所有查询消息都设定了唯一的 ID,用以辨别相同的消息。每个节点在转发消息前都将记录下消息的 ID,每当有新的消息到来,都会将它的 ID 与本地存储过的 ID 进行比较,如果发现相同,则说明节点先前已经收到该查询消息,那么直接丢弃

该消息,不再向其它节点转发,从而保证了任何节点不会多次转发相同的信息。

上述处理虽然筛选掉了冗余消息,但相应缩小了搜索范围。在图 2 中,A 为查询发起节点,由于 TTL_2 路径在某段时间内带宽负载重,查询消息通过 TTL_1 路径优先路由至节点 B, TTL_1 减为 0, 停止路由。对于随后 TTL_2 路径的同一查询,节点 B 通过判断发现已经接收过该查询消息,则不再转发。倘若查询所需的资源恰恰就在节点 C 上,那么将导致搜索的失败。实际上,倘若查询消息首先经过 TTL_2 路径路由至节点 B, 则查询消息可以路由到节点 C, 从而找到相应的资源。

在兼顾减少冗余消息以及扩大搜索范围的条件下, 本文给出以下策略: 倘若节点接收到当前查询的 TTL_n 大于先前接收到同一查询的 TTL_m , 则继续传播查询, 同时将节点上记录该查询的 TTL 值置为 TTL_n 。该处理可能少量增加网络上的消息数量, 但扩大了搜索范围, 提高了搜索成功率。那么对于上例节点 A 的 TTL 设置为 3 的查询, 不管在何种情况下, 查询消息都可路由到节点 C。

下面给出自适应搜索算法 SAS(Self-adaptive Search) 的伪码描述。

算法 1 SAS 算法伪码

```

procedure SAS(q){
  TTL=TTL-1
  if (TTL≤0)
    return null
  //节点类型属于叶节点
  if (NodeType==Leaf Peer)
    SearchLP(q)
  //节点类型为服务节点
  else
    SearchSP(q)
}
procedure SearchLP(q){
  if (Sim(q,r)≥threshold)
    // r 表示本地的匹配资源, Pi 为当前节点, LR(Pi) 代表 Pi 资源连接的服务节点
    return (r and LR(Pi), SAS(q))
  else
    //LD(Pi) 代表 Pi 需求连接的服务节点
    return LD(Pi), SAS(q)
}
procedure SearchSP(q){
  //查询消息第一次路由至该节点
  if (! hasRoute(q)){
    //将查询消息插入路由表
    insertRoute(q)
    //记录当前查询消息的 TTL 值
    ttlTable.put(q,q.getTTL())
    //本地搜索处理
    ProcessQuery(q)
    //处理服务节点的邻居服务节点
    for all sp∈Pi. ServicePeerNeighbors()
      return sp.SAS(q)
  }
  //冗余消息的处理
  else{
    //当前消息的 TTL 值大于历史记录 TTL 值
    if (q.getTTL() > ttlTable.get(q)){

```

```

//更新记录的 TTL 值
ttlTable.put(q,q.getTTL())
ProcessQuery(q)
for all sp∈Pi. ServicePeerNeighbors()
  return sp.SAS(q)
}
}
}

```

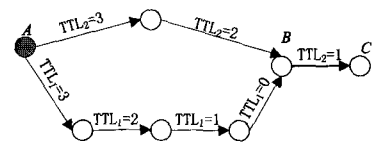


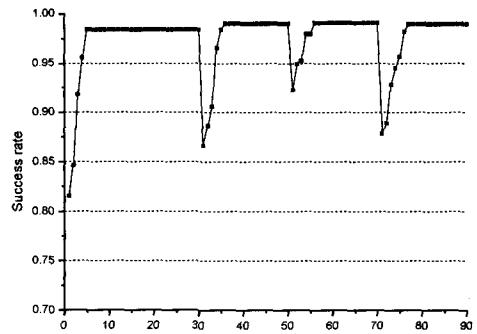
图 2 消息回路的解决

4 实验分析

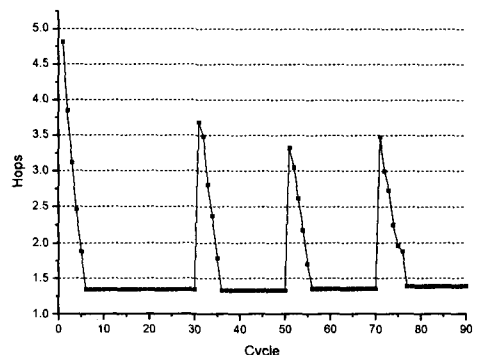
下面通过实验验证 SAS 的有效性。

所有实验都在一台 PC 机上完成, PC 机的配置为 CPU P4 2.0GHz, 内存 512MB, 操作系统为 Windows XP。实验中, 设定网络规模为 1500 个节点, 由 15 个服务节点和 1485 个叶节点构成, 服务节点之间的连接满足幂规律 ($\tau=3$), 每个叶节点随机连接到一个服务节点。每个节点提供的文件数量按 Zipf-like ($\alpha=1$) 分布。

以查询周期作为基本的采样点, 并采用搜索成功率和搜索路径长度两个指标作为度量的标准。在每个周期中, 各节点依据 Zipf-like ($\alpha=1$) 的概率发布一次查询。本文假定只要某个被搜索的节点存储了查询所针对的文件, 那么搜索成功。实验中, TTL 取值为 5, 图 3 展示了 SAS 搜索成功率和平均搜索路径长度随兴趣变化的情况。



(a) 搜索成功率



(b) 平均搜索路径长度

图 3 兴趣转移时 SAS 的性能变化

计算和通信开销, n 越大, 优势越明显。由表 1, CH-DMDH 协议与 GDH 协议相比, 则显著降低了最后一个组成员 U_n 的计算开销以及接收数据的通信开销。此外, GDH 协议中, 尽管组成员 U_n 只发送了一条消息, 但该消息中包含 $n-1$ 个 1024 位的密钥信息。

由以上分析, CH-DMDH 协议能有效地降低移动 ad hoc 网络中节点在进行安全的组密钥协商时所需要的资源开销。

结束语 随着移动 Ad Hoc 网络的应用与发展, 其密钥管理问题日益成为研究热点。本文提出的基于环状分层结构的组密钥协商协议, 在环状分层结构上基于多线性映射在移动 ad hoc 网络中实现了组密钥的协商和分配。与 GDH 和 TGDH 组密钥协商协议相比, 有效地减少了密钥协商过程中节点的资源消耗, 适用于移动 ad hoc 网络。但是 CH-DMDH 协议在节点频繁加入的情况下, 可能会需要重新运行协议来完成组密钥的协商和分配。因此, 如何采用批量更新的方法来减少协议重新运行的次数将是下一步研究的重点。

参 考 文 献

[1] Ateniese G, Steiner M, Tsudik G. New multi-party authentication services and key agreement protocols. *IEEE Journal on Selected Areas in Communications*[J], 2000, 18(4): 628-640
 [2] Ingemarsson I, Tang D T, Wong C K. A conference key distribution system. *IEEE Transactions on Information Theory* [J], 1982, 28(5): 714-720
 [3] Steiner M, Tsudik G, Waidner M. Differ-Hellman key distribution extended to group communication [C] // Conference on Computer and Communications Security. Usenix; ACM Press,

1996, 31-37
 [4] Steiner M, Tsudik G, Waidner M. DLIQUES: A New Approach to Group Key Agreement [C] // Proceeding of the 18th International Conference on Distributed Computing Systems. 1998; 380-387
 [5] Steiner M, Tsudik G, Waidner M. Key Agreement in Dynamic Peer Groups [J]. *IEEE Transactions on Parallel and Distributed Systems*, 2000, 11(8): 769-780
 [6] Perrig A. Efficient collaborative key management protocols for secure autonomous group communication [C] // International Workshop on Cryptographic Techniques and Electronic Commerce. 1999; 192-202
 [7] Kim Y, Perrig A, Tsudik G. Tree-based Group Key Agreement [J]. *ACM Transaction on Information and System Security*, 2004, 7(1): 60-96
 [8] Li J H, Renato Levy, Miao Yu. A Scalable Key Management and Clustering Scheme for Ad Hoc Networks // INFOSCALE'06. 2006; 1-10
 [9] Lee Seok-Lae, Jeun In-Kyung, Song Joo-Seok. Mixed Key Management Using Hamming Distance for Mobile Ad-Hoc Networks [C] // ICCS 2007, LNCS4488. 2007; 665-672
 [10] Ming T J C, How T C. Energy-Efficient and Scalable Group Key Agreement for Large Ad Hoc Networks [C] // PE-WASUN'05. 2005; 114-121
 [11] Wang Wei, Ma Jianfeng, SangJae M. Efficient Group Key Management for Dynamic Peer Networks [C] // MSN 2005, LNCS3794. 2005; 753-762
 [12] Carman D W, Kruss P S, Matt B J. Constraints and approaches for distributed sensor network security [R]. NAI Labs Technical Report. 2000
 [13] Trappe W, Wang Y, Liu K J. Resource-aware conference key establishment for heterogeneous networks [J]. *IEEE/ACM Transactions on Networking*, 2005, 13(1): 134-146

(上接第 60 页)

我们观察到, 初始随机生成的网络中, 由于文件的无序组织, SAS 的搜索成功率仅为 79.3%, 平均搜索路径长度为 4.8。而 SAS 通过网络自组织调整基本达到收敛状态以后 (第 6 个周期), 系统搜索成功率提高到 98.4%, 平均搜索路径长度降低到 1.34 (表明大部分查询在 1 个跳数内完成)。这是因为 SAS 在资源搜索时, 消息在社区间扩散而不是盲目地针对随机分布在整个网络上的资源进行泛洪, 增强了资源发现的针对性, 因此能够在保证高搜索成功率的同时提高系统的搜索效率。

从第 30 个周期开始, 我们每隔 3 万次查询随机选取 40% 的节点, 对它们的兴趣, 即共享文件和搜索请求进行随机互换。可以看出, SAS 在兴趣发生变化的瞬间搜索性能恶化, 搜索成功率下降, 平均搜索路径长度增长, 但随着搜索的进行, SAS 的查询效果越来越好, 并且能够很快恢复到收敛状态, (基本上在 5 个周期内完成)。由于 SAS 可以通过选择路由方向决定搜索的路径, 基于资源连接和需求连接的路由机制随着资源分布以及搜索内容的变化做动态自适应的调整, 因此能快速地自动优化搜索性能, 相对减轻了兴趣转移带来的搜索性能恶化的程度。

结束语 针对节点的兴趣变化对搜索性能造成的影响, 提出了一种动态自适应的搜索机制 SAS, 通过消息转发的智能性 (进行节点路由方向决策), 及时反映资源分布以及搜索内容的动态变化, 从而能够在节点兴趣发生转移的情况下迅速定位资源提供节点。实验分析表明 SAS 能自动优化搜索性能, 搜索性能好, 在动态环境下具有很好的自适应性。

参 考 文 献

[1] Stephanos A T, Diomidis S. A survey of Peer-to-Peer content

distribution technologies. *ACM Computing Surveys*, 2004, 36(4): 335-371
 [2] Lua E K, Crowcroft J, Pias M, et al. A survey and comparison of Peer-to-Peer overlay network schemes. *Journal of IEEE Communications Survey and Tutorial*, 2005, 7(2)
 [3] 何盈捷, 冯月利, 王珊. Peer-to-Peer 环境下基于内容的智能搜索. *计算机研究与发展*, 2004, 41 (增刊): 112-118
 [4] 杨舰, 吕智慧, 钟亦平, 等. 一种基于兴趣域的高效对等网络搜索方案. *计算机研究与发展*, 2005, 42(5): 804-809
 [5] Upadrashta Y, Vassileva J, Grassmann W. Social networks in Peer-to-Peer systems // Proceedings of the 38th Annual Hawaii International Conference (HICSS'05). Kona, Hawaii, 2005
 [6] Tsoumakos D, Roussopoulos N. Adaptive probabilistic search for Peer-to-Peer networks // Proceedings of 3rd Intl Conf. Peer-to-Peer Computing (P2P 2003). Linkoping, Sweden; IEEE Computer Society, 2003; 102-110
 [7] Zeinalipour-Yazdi D, Kalogeraki V, Gunopulos D. Exploiting locality for scalable information retrieval in Peer-to-Peer networks. *Journal of Information Systems*, 2005, 30(4): 277-298
 [8] Menasce D, Kanchanapalli L. Probabilistic scalable P2P resource location services. *ACM SIGMETRICS Performance Evaluation Review*, 2002, 30(2): 48-58
 [9] Sripanidkulchai K, Maggs B, Zhang H. Efficient content location using interest-based locality in Peer-to-Peer systems. *IEEE Infocom 2003*. San Francisco, USA, 2003
 [10] Ramanathan M K, Kalogeraki V, Pruyne J. Finding good peers in Peer-to-Peer networks // Proceedings of International Parallel and Distributed and Computing Symposium (IPDPS' 02). Fort Lauderdale, FL, 2002
 [11] 陈海涛, 龚正虎, 黄遵国. 一种基于学习的 P2P 搜索算法. *计算机研究与发展*, 2005, 42(9): 1600-1604
 [12] 肖卫东, 唐九阳, 汤大权, 等. 基于社会学原理的 P2P 网络模型 REC. *计算机科学*, 2007, 34(6): 38-40
 [13] 唐九阳, 张维明, 肖卫东, 等. 类人类社会基于社区的对等网自组织构造. *计算机研究与发展*, 2006, 43(8): 1383-1390