

P2P 网络的实时性问题*

姚 静¹, 黎忠文², 郭 兵¹

(1. 四川大学 计算机学院, 成都 610064; 2. 成都大学 电子科学与技术学院, 成都 610106)

摘 要: 由于 P2P 网络的动态性、分散性和半可信性,其实时性研究和应用成为公认的难题。当前 P2P 多媒体应用系统和实时计算系统或多或少采用了实时保障措施,但均没有对 P2P 实时性进行系统研究。在总结现有 P2P 网络系统采用的实时技术的基础上,分析了 P2P 网络结构、搜索协议(算法)和任务调度对实时性的影响,提出 P2P 网络实时性研究存在的问题,总结出 P2P 实时性研究需要解决的关键技术。

关键词: 对等网络; 实时性; 网络结构; 搜索算法; 任务调度

中图分类号: TP393 **文献标志码:** A **文章编号:** 1001-3695(2011)01-0020-05

doi:10.3969/j.issn.1001-3695.2011.01.005

Real-time performance of peer-to-peer network

YAO Jing¹, LI Zhong-wen², GUO Bing¹

(1. Dept. of Computer Science, Sichuan University, Chengdu 610064, China; 2. Dept. of Information Science & Technology, Chengdu University, Chengdu 610106, China)

Abstract: However, due to the instability, decentralization and semi-trust of the peer-to-peer network, real-time research is a well-known difficult problem. Though many measures had been applied to deal with this problem, there was no systematic research of real-time performance. This paper summarized the real-time technology of P2P network in the past few years, then analyzed the relationship between real-time performance and P2P network architecture, searching protocol (algorithm), task scheduling. In the end, put forward the remaining difficulties and key technologies of real-time performance.

Key words: peer-to-peer (P2P); real-time performance; network architecture; searching; scheduling

随着 Internet 的广泛普及,曾经在某些领域作为 C/S 结构的补充而存在的 P2P 技术,将传统的“内容位于中心”存储模式转变为“内容位于边缘”存储模式^[1],在分布式信息或资源共享、在线流媒体服务、协同工作、大规模并行计算等领域显示出独特优势。

P2P 发展初期主要应用于大规模文件共享和下载系统,如 Napster、eMule、BT 等。利用 P2P“人人为我,我为人人”的思想,参与服务的对等节点越多时,下载速度越快。这种思想也用于近年来发展迅猛的 P2P 实时流媒体技术,如 PPLive、QQLive、UUSee 等。P2P 实时流媒体技术提供边下载、边观看的服务,要求已下载的资源必须能够满足播放速度,否则将导致视频出现“卡壳”现象,影响观影感受。P2P 网络将数以万计的计算机相连,这些计算机大多处于空闲状态,闲置着巨大的计算能力;与此同时,越来越多的研究领域需要计算机协助,处理超大规模的数据计算。传统的共享计算能力系统如 SETI@home^[2],未能解决服务器性能瓶颈和单调失效的问题,不适用于对实时性要求严格的应用领域,如军事作战领域。战场上的战术系统由传感器网络、卫星终端和作战系统组成。该系统可能由成千上万通信节点和计算节点通过不稳定的网络相连,并要求能够在严格的时间约束下识别、追踪、攻击多个目标对象。

综上所述,P2P 实时性要求在动态网络环境下及时响应外部事件的请求,并在严格的时间约束条件下完成外部事件的处理,

如节点通信和数据传输,保证各节点和实时任务协调一致地工作。P2P 实时性研究如何在 P2P 网络构建实时应用和服务,包括 P2P 网络拓扑结构、P2P 实时搜索协议和算法、P2P 实时任务调度算法和策略。本文从这三方面着手,研究了现有 P2P 网络为满足实时性采用的方法和策略。

1 P2P 网络拓扑结构

P2P 网络拓扑结构是构建 P2P 实时应用的基础。目前,P2P 网络体系结构分为集中目录式 P2P、纯分布式 P2P、混合 P2P。纯分布式 P2P 又可划分为非结构化 P2P 和结构化 P2P 两种。

集中目录式 P2P 有中央服务器,服务器只保存资源的索引信息,资源仍保存在各节点。节点通过服务器获得资源位置信息,然后直接与资源所在节点相连。全分布式非结构 P2P,没有中央服务器,链状的节点之间构成分散式网络。全分布式结构化 P2P 在全分布式非结构 P2P 的基础上构建确定的网络拓扑结构,从根本上改变了 P2P 网络无结构的状态。目前受到广泛好评的是使用 DHT(distributed hash table)构建的结构化 P2P 网络。混合 P2P 引入超级节点(super peer)的概念,按节点能力(可用带宽、处理速度、存储空间)划分为普通节点和超级节点。超级节点可分为搜索节点和索引节点两种,搜索节点和邻近若干普通节点组成一个簇,簇间通过搜索节点相连。

收稿日期: 2010-07-03; **修回日期:** 2010-08-27 **基金项目:** 福建省自然科学基金资助项目(2008J0034);成都大学人才引进基金资助项目(201001)

作者简介: 姚静(1988-),女,硕士研究生,主要研究方向为计算机网络、实时计算(leehomandyj@163.com);黎忠文(1970-),女,教授,博士,主要研究方向为计算机网络、实时计算;郭兵(1970-),男,教授,博导,主要研究方向为嵌入式实时系统。

索引节点是从搜索节点中选取或是从外部进入的高性能节点,维护簇内节点资源索引信息。

表 1 从可拓展性、可靠性、可维护性、发现算法效率和是否支持复杂查询等方面对这四种拓扑结构作了比较。

表 1 四种网络拓扑结构属性比较					
拓扑结构/属性	拓展性	可靠性	可维护性	发现算法效率	是否支持复杂查询
集中目录式 P2P	差	差	最好	最高	支持
全分布式非结构化 P2P	差	好	最好	中	支持
全分布式结构化 P2P	好	好	好	高	不支持
混合 P2P	中	中	中	中	支持

鉴于 P2P 网络动态和不稳定性以及实时应用和服务的需求,采用相对稳定、拓展性强和查询速度快的网络拓扑结构,有助于提高实时系统的整体性能。

2 搜索协议和算法

P2P 搜索技术是构建 P2P 实时应用的关键,对等节点的反应时间是衡量实时性的重要指标。在 P2P 系统中,对等节点的反应时间多是由网络通信和传输时间决定的,好的搜索技术不仅可以大大降低资源定位和消息传播时间,还能减少网络的消息开销。

按照搜索算法适用的网络结构,可将搜索技术划分为集中式搜索、分布式搜索和半结构化搜索三类。分布式搜索又划分为分布式非结构化搜索和分布式结构化搜索。

2.1 集中式搜索

集中式搜索适用于集中目录式 P2P 网络结构。所有 P2P 节点与已知地址的 P2P 目录服务器相连,服务器负责对 P2P 网络中的共享文件和资源进行索引和查询。典型的实例是 Napster^[4]。集中式搜索的优点是:服务器的引入,提高了资源搜索速度,查询范围能覆盖所有节点;节点可以动态地与服务器交互信息,索引信息更新速度快;搜索过程中的消息量小,节省了网络带宽。集中式查询的缺点是:中央服务器的能力限制了查询能力和网络的可拓展性;一旦服务器失败,整个系统则瘫痪,容错性不高。

2.2 分布式非结构搜索

分布式非结构搜索适用于纯分布式非结构 P2P 网络结构。

1) 泛算法及其改进

a) 标准洪泛算法^[5]。原理是节点对所有邻居节点广播消息,邻居节点再对自己所有邻居节点广播消息。为了限制广播消息范围,引入 TTL (time-to-live),节点每转发一次消息则将 TTL 减 1,当 TTL = 1 时,停止转发。洪泛算法的优点是:随着跳数增加,被“通知”的节点个数呈指数级增长,网络覆盖率高,查询速度快;缺点是:广播风暴引起的巨大网络开销,网络中的低带宽节点会因资源过载而失效,导致查询速度慢和网络的可拓展性降低。

b) 规范化洪泛^[6]。在标准洪泛的基础上,按照网络中最小的节点度数 d_{\min} 洪泛消息。若节点度数为 d_{\min} ,则向所有的邻居节点转发消息(发起搜索的节点除外);若节点度数大于 d_{\min} ,则从邻居节点中任选出 d_{\min} 个,洪泛消息(发起搜索的节点除外)。纯分布式非结构网络结构中,某些节点度数很高,规范化洪泛比标准洪泛具有更好的性能。

c) 迭代洪泛^[7]。一般要进行多次洪泛,深度递增。每一

次洪泛结束后,节点需等待源节点反馈信息。搜索结果若满足要求或到达最大深度,则停止洪泛。平均情况下,该算法有效地减少了网络中的消息开销,但在最坏情况下,算法引起的消息开销比标准洪泛算法更糟糕。

d) 启发式洪泛。指每个节点统计邻居节点的历史记录,随机选取表现优秀的邻居节点洪泛消息。评定节点是否优秀,可以根据邻居节点在历史查询中返回的满意结果个数多少查询消耗时间长短等,也可以综合考虑多个因数选取优秀节点。启发式洪泛通常只需要反问较少节点即可达到满意结果,减少了网络的信息开销,查询速度也相当可观。但是,P2P 网络的动态和不稳定性,很难保证过去表现优秀的节点今后也表现优秀。当曾经优秀的节点不再优秀时,算法性能将大大降低。

2) 随机漫步算法^[6,8]和基于流言协议算法^[9,10]

随机漫步和基于流言协议两种算法非常相似。发起查询的节点从所有邻居节点中随机选择 k 个节点,发送查询消息;接到消息的节点再从自己所有邻居节点中随机选出 k 个节点,发送查询消息。

随机漫步^[8]算法是通过 TTL 和每隔一段时间就询问一次发起查询的节点是否继续转发的轮询机制来控制转发。轮询机制要求当前节点和发起查询的节点交换信息,随着跳数的增加,被“通知”的节点呈指数级增长,响应时间也呈指数级增长。文献[6]引入泛化随机漫步。在发起查询之前,给出期望被“通知”节点预算个数 s 。假设发起查询的节点的邻居节点个数为 k ,分别分配 s_1, s_2, \dots, s_k 个任务。其中 $s_1 + s_2 + \dots + s_k = s (s_i \geq 0, 1 \leq i \leq d)$ 。在向邻居节点 i 发送查询消息时,需要带上指定的预算 s_i 。在每个被“通知”的节点处重复该过程,直到预算为 1,查找满意的结果。该方法具有很高的查询速度,同时能够很好地利用具有较多邻居的节点转发查询消息,避免了个别能力较低的节点因负载过重而消失。

基于流言协议算法在流言开始前,指定流言回合 n 。当查找到满意节点或流言次数达到 n ,则停止流言消息。文献[9]引入 RTG-L 算法,要求已向外流言过消息的节点不再向外流言消息,通过该方式控制网络中的消息开销。与 RTG 标准流言算法不同的是,RTG-L 算法中的 k 值可根据需求改变。文献[10]引入 RTG-G 算法,在 RTG-L 的基础上稍作修改。发起查询的节点首先在最近法定人数集中流言消息,然后该集的某些节点再向外流言消息。法定人数集系统的交叉属性,使只访问任意一个法定人数集即可知道全局情况。

3) 基于节点兴趣的搜索算法

基于节点兴趣搜索算法是根据节点历史搜索结果中学习节点之间的兴趣相似度,并根据节点兴趣相似度来转发查询消息。文献[11]提出基于学习的搜索方法 SmartSearch。SmartSearch 采用分布式的被动学习方法,从历史搜索结果中学习节点之间的兴趣相似度,将节点按照兴趣分类,在具有相似兴趣节点之间建立朋友关系,搜索请求首先转发到朋友节点,当其失效后利用广播进行搜索。SmartSearch 稳定高效,比传统算法在低开销情况下性能有数量级的提高。文献[12]利用位置对节点分群,通过历史查询的搜索反馈结果动态选择转发与实际位置相邻节点的算法(FP 算法)。算法通过计算邻居节点的兴趣相关度定期调整邻居节点。与洪泛算法相比较,算法在搜索时间上改进约 10% ~ 40%,能很好地控制总的消息数和重复访问节点的比例,提高了查询效率。文献[13]利用向量空

间模型将文件内容向量化,再对向量空间进行聚类,得到节点的兴趣度。计算节点之间的兴趣相似度,在本地建立兴趣索引表;在搜索时,根据兴趣索引表直接将查询请求转发到有相似兴趣的节点,能在不影响查询结果的前提下,又能较少地访问节点的数量,提高查询效率。基于节点相似度的搜索算法查询速度快,具有很高的命中率,但该算法不适用于网络波动现象严重的 P2P 网络。随着节点的频繁加入和离开,势必导致相似节点的不可用,降低查询效率和命中率。

4) 基于蚁群算法的搜索机制

将人工智能领域的蚁群算法引入到 P2P 搜索技术的研究。基于蚁群算法的 P2P 网络模型^[14]可以理解为:将整个 P2P 网络看做是由很多蚁巢以及连接蚁巢的通路构成,每个蚁巢看做是 P2P 网络中的对等节点。因此,P2P 网络的搜索过程可以理解为蚂蚁搜索食物的过程。通过信息素浓度来选择搜索节点。信息素实际上是一种与本节点相关的历史搜索结果的统计数据。文献[15]利用蚁群算法信息素的正反馈机制指导查询请求消息和转发,将查询请求消息尽量发往资源可能存在的节点上。

采用的信息素挥发式为 $\tau_i = (1 - \rho) \times \tau_i$, 信息素的更新式为

$$\tau(mj) = (1 - \rho)\tau(mj) + \Delta\tau(mj)$$

$$\Delta\tau(mj) = \begin{cases} \delta \frac{S_{\text{find}}}{S_{\text{max}}} + (1 - \delta) \frac{\text{TTL}}{\text{TTL}_{\text{max}}} & \text{蚂蚁经过的路径 } j \\ 0 & \text{else} \end{cases}$$

其中: ρ 为信息素的挥发系数,则 $1 - \rho$ 表示信息素的残留因子, ρ 的取值为 $\rho \in [0, 1]$; $\Delta\tau(mj)$ 是本次循环中节点上的关于本次查询的信息素增量,初始时刻每个节点的 $\Delta\tau(mj) = 0$; S_{find} 表示搜索蚂蚁搜索到的资源数量;TTL 表示搜索蚂蚁搜索到资源时的生存时间; δ 表示匹配资源的数量与查询时间的权重。蚂蚁在越短的时间内搜索到越多的资源,则该条路径上的信息素增量就越大,后来的蚂蚁搜索该条路径的概率越大,时间越短。

文献[16]提出基于多种信息素调控机制的改进蚁群算法,不需要全局性的文件分类规则就可以根据被搜索文件的类型智能地进行路由,并拥有较好的搜索效率。信息表更新规则如下: $\alpha = (1 - \rho)\alpha + \Delta\alpha$ 。其中: ρ 是挥发系数,通常设置 $\rho < 1$,用来避免信息素的无限增加; $\Delta\alpha$ 为信息素的增量,这里根据 $\Delta\alpha = Q \times (\mu^{S_{it}})$ 。 Q 是增加的信息素原始总量,为常数; μ 为节点浓度差,是一个介于 0 和 1 的调节因子; S_{it} 为目标节点 t 到 i 的跳数,这样就保证了在每次增加信息素浓度时,离目标节点越近的节点增量越多,其增量随着当前节点与目标节点距离的增大而递减。每隔一段时间,信息素会自动更新,形成信息素的挥发现象,这时 $\Delta\alpha$ 取值为 0。

2.3 分布式结构化搜索

分布式结构化搜索利用分布式结构化网络结构,利用内容索引(key, value)中的 key 与节点标志符之间的关系,提高查找效率。

1) Chord^[17] 算法及其改进

Chord 基于一维环形空间,每个关键字和节点都分配一个 m 位的标志符。所有节点按照标志符从小到大的顺序按顺时针方向连接成逻辑环。 $\langle \text{key}, \text{value} \rangle$ 对存储在标志符等于 key 的节点上;若该节点不存在,则存储在该节点的后继节点上。

Chord 算法引入路由指针表,节点 n 的指针表中第 i 项记录标志符大于或等于 $n + 2^{(i-1)}$ 的节点,查找时间为 $O(\log N)$ 。文献[18]在已有的 Chord 算法的基础上,引入了最近访问节点表和邻居节点表。最近访问节点表加速热门资源的查找,邻居节点表是备份物理位置上相近的节点的部分资源。该改进的 Chord 算法具有更好的负载均衡和系统健壮性。Chord 算法简单、性能好,但随着节点规模的增大和节点间性能的差异,将严重影响系统性能,还存在拓扑适配问题。

2) Pastry^[19]

Pastry 把节点和关键字标志符表示为一串以 2^b 为基的数,Pastry 与 Chord 算法类似,节点按标志符大小顺序连接成逻辑环, $\langle \text{key}, \text{value} \rangle$ 对存储在最接近节点标志符的节点上。节点维护三张表:路由表、邻居节点表、叶子节点表。查询时,首先查询叶子节点表,若查询失败,再依次查询路由表和邻居节点表。该路由过程是收敛的,经过有限次步骤即可找到目的节点。其查询效率为 $O(\log_2^b N)$ 。

3) SWAD(SWTM-Kademlia)^[20] 算法

SWAD 算法将 SWTM 和 Kademlia 相结合,即利用 SWTM 算法采用局部贪心算法代替传统洪泛,有效降低资源分散引起的搜索开销,并引入 Kademlia 算法抗扰动机制。SWAD 网络中,每个文件(K-roots)维护常数个指向相似文件的连接,形成 Gs 网络。所有的文件根据 ID 值大小排列成一个环形拓扑,称为 Gw 网络。该算法在高强度的网络波动下依旧能保持良好的稳定性和高性能。

分布式结构化搜索算法具有很高的查询效率,但是利用二进制关键值查找算法不支持模糊查找,也不支持多关键字查找。

2.4 半结构搜索算法

半结构化搜索应用于混合 P2P 网络,将节点按能力划分为超级节点和普通节点,在超级节点所管理的自治区内部和超级节点之间采用不同的搜索算法。

文献[21]提出基于 LBFS(breadth-first-search)的混合搜索,将模型中的超级节点负荷限制在较低层次上,因此超级节点称为低量级超级节点;物理位置相近的节点通过选举超级节点构成一个簇,以使邻近节点间的查询尽量在簇内进行;由于簇内超级节点的性能存在差异,超级节点失效后,其“代言”的某一客户节点将成为新的超级节点;同时,使用滑动窗口的节点能力评估机制动态地设定超级节点的负载上限。该算法的搜索机制为:簇内采用简单、高效的集中式索引方式,同时引入数据项备份机制;簇间采用局部广度优先搜索。簇内搜索时,按照重复率大小,超级节点将在缓存之前查询返回的数据项值。LBFS 是一种轻量级洪泛搜索,建立局部广度优先生成树,在 2-跳的局部范围内进行 BFS,减少冗余消息。该算法很好地克服了查询绕路问题,有效地减少了消息冗余,具有较高的搜索效率和较好的扩展性。

文献[22]引入移动 agent,根据网络环境、节点服务的状态以及用户请求,选择合适的节点进行访问来完成用户的要求,提高应用程序的效率;生成一定数量的移动 agent,并预先为每个移动 agent 分配迁移路线进行资源搜索;超级节点处驻扎有静态 agent 代理,负责生成响应数量的 query agent 和预定路线,根据执行结果修改超级节点的历史查询记录信息,并以此与节点查询的 query agent 交互。该算法的查询机制为:节点首先向

其所在的超级节点查询资源,若没有找到,超级节点则生成移动 query agent 和预定查询路线,与其他超级节点交互查询信息。该算法提高了资源搜索效率,节约了网络带宽,能自主调节响应时间。

超级节点利用流言协议查询消息^[3],节约了查询时间,并具有较高的消息开销。源超级节点首先检查本地路由表,若自治区中某个普通节点能够执行子任务,则将子任务信息发送给该节点执行;否则,在超级节点邻居中流言消息。获得流言消息的超级节点执行源超级节点查询步骤,直到找到目的点或耗尽查询时间。其中,每个超级节点只向外流言一次消息。构造超级节点的邻居节点时,只有当两个节点间的消息丢失率不大于 m_{lr} 时,才能够成为邻居节点。为了进一步减少系统的消息开销,超级节点 A 在向超级节点 B 流言消息时,若发现 B 的邻居节点 C 的节点 ID 小于 A 的节点 ID,则节点 A 不向 B 流言消息。

3 任务调度策略

任务调度是影响 P2P 实时性的重要因素。好的调度算法不仅可以提高任务执行的成功率,还能够避免因节点离开或失效导致的任务重调度。P2P 环境下任务调度的特点是:面向异构平台、大规模和非集中式、不干涉 P2P 网络中节点内部的调度策略,必须具有可拓展性、能够动态自适应。这些特点使应用于网格计算的调度算法无法直接适用于 P2P 网络环境。目前应用于 P2P 环境的任务调度策略有四种:

a) 基于相似度的任务调度策略^[23]。计算现行任务与历史任务之间的相似度,以确定能够直接调度的节点,从而不但能够减少请求信息的发送次数,而且有可能提高任务调度的成功率。当通过任务相似性所获取的节点个数不能满足任务调度需求时,通过计算任务需求资源与节点可提供资源的相似度并利用传输时间因素对其进行修正,以获取满足任务需求的有序节点集合。任务相似度计算公式为

$$\text{sim}(p, p') = \frac{\alpha \times \text{timeSim}(p, p') + \beta \times \text{sourSim}(p, p')}{\alpha + \beta}$$

其中: p 表示现行任务; p' 表示历史任务; α 、 β 表示权重; $\text{timeSim}(p, p')$ 表示时间相似度; $\text{sourSim}(p, p')$ 表示资源相似度。

$$\text{sourSim}(p, p') = \frac{\sum_{k=1}^n W_{ik} \times W_{jk}}{\sqrt{(\sum_{k=1}^n W_{ik}^2) \times (\sum_{k=1}^n W_{jk}^2)}}$$

$$\text{timeSim}(p, p') = \begin{cases} 0 & |T_p - T_{p'}| > \varepsilon \\ e^{-|T_p - T_{p'}|} & \text{else} \end{cases}$$

资源权重表示执行某任务所需要的实际资源量。 W_{ik} 为 p 的第 i 个资源的权值; W_{jk} 为 p' 的第 j 个资源的权值; n 为资源种类; T_p 和 $T_{p'}$ 分别代表 p 和 p' 的估计执行时间; ε 可根据任务对时间的敏感程度取不同的值。

资源相似度计算公式为

$$\text{sim}(p, s) = \beta_1 \times \text{sourSim}(p, s) + \beta_2 \times f(L_{ps}), \beta_1 + \beta_2 = 1$$

$$\text{sourSim}(p, s) = \frac{\sum_{k=1}^n W_{ik} \times W_{jk}}{\sqrt{(\sum_{k=1}^n W_{ik}^2) \times (\sum_{k=1}^n W_{jk}^2)}}$$

其中: p 表示现行任务; s 表示历史任务; W_{ik} 为 p 的第 i 个资源的权值; W_{jk} 为 s 的第 j 个资源的权值; n 为资源种类; L_{ps} 表示 p 所在节点与 s 之间的传输时间; f 是随 L_{ps} 的增加单调递减的映

射函数, L_{ps} 映射为 $[0, 1]$ 的某一数值; β_1 和 β_2 代表不同的权值,分别控制资源和传输时间对任务执行的影响程度。

b) 基于多目标的并行任务调度策略^[24]。考虑了任务执行时间、节点间通信时间和调度费用等多个目标,通过对多个目标不同参数的控制,使用线性规划算法将多目标转换为单目标任务调度模型,在此基础上用匈牙利算法^[21]完成任务调度,达到任务调度中多目标的平衡。以三个目标为例,任务调度模型如下:

$$\min(z) = \sum_{i=1}^n \sum_{j=1}^m (\lambda_1 c_{ij} + \lambda_2 b_{ij} + \lambda_3 d_{ij}) x_{ij}$$

其中: λ_1 、 λ_2 、 λ_3 是根据最终目标得到的各个分目标的系数; c_{ij} 、 b_{ij} 、 d_{ij} 表示各个分目标的具体数据,如任务执行时间或通信时间等; x_{ij} 为决策变量。

c) 基于节点在线时间预测的调度策略^[25]。系统中位于监控节点的监控模块可维护网络中的节点信息,通过统计各个节点历史在线时间得出节点在线时间规律,预测当前节点在线时间,并用分布式双向跳链^[17]按序存储。监控节点记录节点上次加入时间和上次离开时间,估算出本次节点的预测在线时间 T_{exp} 。节点的剩余在线时间 $T_{left} = T_{exp} - T_{ela}$ 。其中, T_{ela} 表示节点已在线时间。系统将任务调度至在线时间较长的节点处执行,以避免节点失效导致任务重调度和网络开销。

d) 利用闲散时间的调度策略^[26]。分布式实时任务可划分为子任务的集合,子任务之间具有因果和时序依赖关系。文献[12]为每个子任务分配本地闲散时间,用于寻找下一个远程调度节点。通过最大化满足点到点的时间约束来提高整个任务的实时性约束。对任意节点 Nr_i ,该节点处的子任务所需执行时间为 $Er_{i,j}$,则本地闲散时间 $LS_{i,j} = Sr_i / (Nr_i - 1)$; 若 $Nr_i > 1$, $LS_{i,j} = Sr_i$; 若 $0 \leq Nr_i \leq 1$, Nr_i 为全局剩余闲散时间。本地终止时间 $LX_{i,j} = t + Er_{i,j} + LS_{i,j}$, t 为子任务在 Nr_i 处开始执行的时间。

4 进一步研究的问题

随着 P2P 多媒体技术和实时计算等方面的优势日益明显, P2P 实时应用也将成为一大热门研究领域。目前, P2P 网络节点的不稳定性和分散性等特点, 导致构建于 P2P 网络上的实时应用并不多。根据 P2P 网络自身特点, 并借鉴已有的研究成果, 提出几点在接下来的研究中应注意的问题:

a) P2P 网络结构的选择。网络结构是构建 P2P 应用的基础, 好的结构可以充分发挥 P2P 网络自身优势, 并能在一定程度上减弱 P2P 网络的动态性和不稳定性。目前, 构建于分布式非结构 P2P 网络拓扑和混合 P2P 网络拓扑的实时应用较多。分布式非结构 P2P 没有确定的网络拓扑结构, 导致网络的拓展性不高, 限制了网络的规模; 混合 P2P 网络拓扑汲取了集中式目录 P2P 的快速查找和分布式非结构 P2P 去中心化的优点, 具有很高的优势。同时, 分布式结构化 P2P 在网络拓展性和稳定性方面也具有自身独特优势。在构建实时应用时, 可根据具体的应用选择合适的网络拓扑结构。

b) 节点查询和路由协议。P2P 网络节点规模巨大、分布范围广, 如何采用有效的节点信息查询协议, 在众多对等节点中快速搜索到合适节点, 成为实时性约束的关键。目前, 在分布式非结构 P2P 中应用最广泛的是基于洪泛的搜索算法, 如随机漫步算法和基于流言协议算法。相较于标准洪泛, 这些算法

在不降低搜索速度的前提下,消息开销有了很大削减。应用于纯分布式非结构 P2P 的搜索算法具有很高的搜索速度,但基本上都不支持模糊查找和多关键字查找。应用于混合 P2P 的搜索算法基本上在簇内使用集中式搜索,簇间使用分布式搜索算法。在今后的研究中,搜索算法应着重于查询速度和消息开销两方面。

c) 调度算法和策略。P2P 网络节点处于对等地位,节点的异质性、能力的差异、匿名性、在线时间长短等都成为影响实时性的关键。任务调度算法应充分考虑这些因素,选择合适的节点,避免恶意节点攻击以及节点失效导致的任务重调度等。在之后的研究中,可以将目前 P2P 网络信誉机制以及将非实时任务的调度策略灵活应用于实时任务调度,构建高效的实时调度算法和策略。

d) 容错性。P2P 网络最大的特点是节点随意加入和离开,这导致网络结构的动态和不稳定性。虽然现有的网络结构在一定程度上改善了这种不稳定性,但网络波动造成的拓扑不稳定性无法避免。节点加入时,需要通知网络中其他节点自己能提供的资源服务;节点离开时,不仅要通知其他节点自己要离开,若节点处有任务运行,还需要发送任务被迫中断的信号,释放网络中锁定的资源。为保证实时性,同一个实时任务可维护多个执行副本,提高执行成功的概率。

5 结束语

目前,构建于 P2P 网络的实时应用并不多,但随着人们对网络服务质量的要求不断提高和 P2P 技术的发展,实时应用必然成为未来的一大研究热点。本文提出了 P2P 实时性的要求和研究内容,重点从 P2P 网络结构、搜索算法和实时调度策略三方面对当前研究工作进行了总结和比较,最后提出在 P2P 实时应用时要注意的几点问题。

参考文献:

- [1] RATNASAMY S, FRANCIS P, HANDLEY M. A scalable content-addressable network [C]//Proc of Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications. New York: ACM Press, 2001: 161-172.
- [2] ANDERSON D P, COBB J, KORPELA E, *et al.* SETI@home: an experiment in public-resource computing [J]. *Communications of the ACM*, 2009, 45(11): 56-61.
- [3] HUANG Fai, RAVINDRAN B, JENSEN E D. RT-P2P: a scalable real-time peer-to-peer system with probabilistic timing assurances [C]//Proc of IEEE/IFIP International Conference on Embedded and Ubiquitous Computing. Washington DC: IEEE Computer Society, 2008: 97-103.
- [4] Napster Inc. The napster homepage [EB/OL]. (2001) [2010-06-21]. <http://www.napster.com/>.
- [5] Gnutella Inc. The Gnutella homepage [EB/OL]. (2001) [2010-06-21]. <http://gnutella.wego.com/>.
- [6] GKANTSIDIS C, MIHAIL M, SABERI A. Hybrid search schemes for unstructured peer-to-peer networks [C]//Proc of the 24th Annual Joint Conference of IEEE Computer and Communications Societies. Washington DC: IEEE Communications Society, 2005: 1526-1537.
- [7] YANG B, GARCIA-MOLINA H. Improving search in peer-to-peer networks [C]//Proc of the 22nd International Conference on Distributed Computing Systems. Washington DC: IEEE Computer Society, 2002: 5-14.
- [8] LV Qin, CAO Pei, COHEN E, *et al.* Search and replication in unstructured peer-to-peer networks [C]//Proc of the 16th International Conference on Supercomputing. New York: ACM Press, 2002: 84-95.
- [9] HAN Kai, RAVINDRAN B, JENSEN E D. RTG-L: dependably scheduling real-time distributable threads in large-scale, unreliable networks [C]//Proc of the 13th IEEE Pacific Rim International Symposium on Dependable Computing. Washington DC: IEEE Computer Society, 2007: 314-321.
- [10] ZHANG Bo, HAN Kai, RAVINDRAN B, *et al.* RTQG: real-time quorum-based gossip protocol for unreliable networks [C]//Proc of the 3rd International Conference on Availability, Reliability and Security. Washington DC: IEEE Computer Society, 2008: 564-571.
- [11] 陈海涛, 龚正虎, 黄遵国. 一种基于学习的 P2P 搜索算法 [J]. *计算机研究与发展*, 2005, 42(9): 1600-1640.
- [12] 孙华志, 侯洁. 基于资源为止与节点反馈的 P2P 搜索算法 [J]. *计算机科学*, 2008, 35(6): 43-45.
- [13] 谭义红, 陈治平, 林亚平. 基于兴趣挖掘的非结构化 P2P 搜索机制研究与实现 [J]. *计算机应用*, 2006, 26(5): 1164-1166.
- [14] MICHLMAYR E, PANY A, KAPPEL G. Using taxonomies for content-based routing with ants [J]. *Computer Networks*, 2007, 51(6): 4514-4528.
- [15] 赵宏, 谢伟志, 张晨曦. 基于蚁群算法的非结构化 P2P 搜索研究 [J]. *计算机科学与技术*, 2009, 19(2): 31-34.
- [16] 苏玉, 毛力. 基于蚁群算法的非结构化 P2P 搜索机制的研究 [J]. *计算机工程与设计*, 2010, 31(5): 939-941.
- [17] MORRIS S R, KARGER D, KAASHOEK M F. Chord: a scalable peer-to-peer lookup service for Internet applications [J]. *IEEE/ACM Trans on Networking*, 2003, 11(1): 17-32.
- [18] 李士宁, 夏贻勇, 倪红波, 等. 一种改进的 Chord 搜索算法 [J]. *计算机工程与应用*, 2008, 44(22): 139-142.
- [19] ROWSTRON A, DRUSCHEL P. Pastry: scalable, decentralized object location and routing for large-scale peer-to-peer systems [C]//Proc of the 18th IFIP/ACM International Conference on Distributed Systems Platforms. New York: ACM Press, 2001: 329-350.
- [20] 方, 何锐邦, 刘新, 等. 基于多媒体特征的抗扰动 P2P 搜索 [J]. *计算机工程*, 2010, 36(8): 111-116.
- [21] 徐强, 孙乐昌, 单红, 等. 基于 BFS 的轻量级 P2P 搜索模型 [J]. *计算机工程*, 2009, 35(3): 106-111.
- [22] 王龙. 基于移动 agent 的 P2P 搜索方法研究 [J]. *微计算机信息*, 2010, 26(3): 152-153.
- [23] 孟宪福, 张晓燕. 对等网络环境下基于相似度的任务调度策略研究 [J]. *计算机集成制造系统*, 2007, 13(12): 2446-2451.
- [24] 孟宪福, 张晓燕. 对等网络环境下多目标约束的并行任务调度策略研究 [J]. *计算机集成制造系统*, 2008, 14(4): 761-766.
- [25] ZHANG Hao, JIN Hai, ZHANG Qin. Scheduling strategy of P2P based high performance computing platform base on session time prediction [C]//Proc of the 4th International Conference on Advances in Grid and Pervasive Computing. Berlin, Heidelberg: Springer-Verlag, 2009: 364-379.
- [26] HAN Kai, RAVINDRAN B, JENSEN E D. Exploiting slack for scheduling dependent, distributable real-time threads in mobile Ad hoc networks [C]//Proc of the 15th International Conference on Real-time and Network Systems. France: Institut National Polytechnique de Lorraine, 2007: 225-234.