

基于自聚簇的三层结构 P2P 网络模型

李江峰¹ 周兴铭² 张晨曦²

(同济大学电子与信息工程学院 上海 200092)¹ (同济大学软件学院 上海 200092)²

摘要 在传统的两层混合式 P2P 结构中,超级节点不仅要负责管理查询消息路由,而且还要对节点文件进行管理,这样很容易使超级节点的网络负载过大。同时,超级节点的简单指定方式容易引起虚拟拓扑网络与实际物理网络不匹配,更不能利用节点的兴趣局部性。针对传统两层混合式 P2P 模型的不足,提出了新的三层结构 P2P 网络模型。这种模型将超级节点的文件管理和消息路由功能分别交给两种不同的专属节点来处理,同时考虑节点的实际物理位置,并采用基于节点属性特征的自组织管理机制使节点自聚成簇,自适应地改进自身的搜索性能。实验结果表明,三层结构 P2P 模型能更有效地进行资源组织,提供高效的资源搜索。

关键词 簇,自组织,对等网,三层结构,属性特征

Peer-to-Peer Network Model with Three-tier Topology Based on Auto Clustering

LI Jiang-feng¹ ZHOU Xing-ming² ZHANG Chen-xi²

(School of Electronics and Information Engineering, Tongji University, Shanghai 200092, China)¹

(School of Software Engineering, Tongji University, Shanghai 200092, China)²

Abstract Super nodes of conventional hybrid P2P structure, which is two-tier topology, may overload because they manage both query routing and file management. The oversimplified way of generating a super node may bring mismatch between overlay topology and geography locality, and it also can not display the interests in neighborhoods. A P2P network model with three-tier topology based on auto clustering was proposed. In the model, two functions of super nodes in two-tier topology, file management and query routing, were disposed by two kinds of nodes respectively, and geography localities of nodes were considered. Nodes in P2P networks are able to self-organize based on their characters. Auto clustering was constructed to improve the performance. The experiment implies that the P2P network model with three-tier topology based on auto clustering is able to organize resources effectively and provide efficient search.

Keywords Cluster, Self-organized, P2P, Three-tier topology, Character

1 引言

计算机对等网(Peer-to-Peer network, P2P)技术是目前流行于国际计算机网络技术研究领域的一个热点。然而,对于大多数分布式无结构化 P2P 系统^[1-3],一般采用随机的节点组织方式和简单洪泛(flooding)的查询方式。虽然这种结构具有灵活性好、支持复杂的查询等优点,但是查询时会产生大量冗余的转发信息,对网络带宽的消耗非常大。为了克服这些缺陷,一些研究工作将分布式无结构化 P2P 与集中式 P2P 相结合,提出了混合式 P2P 结构模型。混合式类型^[4]将性能较高的节点指定为超级节点,按集中式的模式管理其下面的普通节点,超级节点之间按分布式非结构化方式互连。但是混合式 P2P 的简单的固定层次结构(超级节点与普通节点构成两层)和超级节点指定方式(仅根据节点的处理能力、带宽等机器性能)具有较大的局限性。

近年来,P2P 系统主要用于信息共享领域,查询检索操作是其最基本的操作。在提高检索性能方面,目前已有不少相

关研究。文献[5,6]考虑了虚拟拓扑与物理拓扑的匹配程度,在虚拟拓扑中,节点之间的连接与它们的传输延迟或距离有关;文献[7,8]根据节点的共同兴趣提高检索性能,但节点之间的“共同兴趣”仅仅简单地定义为含有相同的一组文件或属于某一种固定兴趣分类,这种判断过于简单。

针对上述问题,本文提出基于自聚簇的三层结构 P2P 网络模型(TAC),主要思想如下:(1)将传统两层混合式 P2P 中超级节点的节点管理和消息路由功能分别用中心节点和路由节点进行处理,以此形成三层结构的混合 P2P 网络模型;(2)采用基于节点属性特征的自组织管理机制,使得同一个簇内各节点的兴趣是相近的,从而访问兴趣相近的内容概率也比不同簇的节点大;(3)在 P2P 虚拟逻辑拓扑中,考虑节点的实际物理位置,减少虚拟网络与物理网络的不匹配;(4)考虑节点频繁加入和退出的动态变化,根据簇内节点数量的增加和减少,自适应地进行簇的自聚,控制簇的规模和网络负载。

本文第 2 节给出模型结构,第 3 节描述相关协议,第 4 节介绍动态环境下簇的自聚,包括网络规模控制和网络负载控

到稿日期:2008-04-29

李江峰(1983—),男,博士研究生,主要研究方向为分布式计算与并行处理、对等网络;周兴铭(1938—),男,中国科学院院士,教授,博士生导师,CCF 高级会员,主要研究方向为高性能体系结构、并行与分布处理;张晨曦(1960—),男,教授,博士生导师,主要研究方向为计算机体系结构、分布式计算与并行处理、对等网络。

制,第5节用模拟实验对TAC进行验证,最后总结全文并讨论下一步工作。

2 基于自聚簇的三层结构P2P网络模型TAC的体系结构

2.1 基本概念

定义1(中心节点) 中心节点(CentralNode, CN)由簇内性能好、在线时间长的节点充当,而且它逻辑上位于簇的中心,与簇内节点距离最短。

定义2(路由节点) 路由节点(RoutingNode, RN)类似于网络中的路由,是簇间的接口。

定义3(叶子节点) 叶子节点(LeafNode, LN)是自治系统中的终端用户节点。

由以上定义可知,中心节点的作用是维护所在簇的叶子节点的节点信息列表、负责簇内叶子节点的管理;路由节点的作用是存储其它簇的服务状态和信息列表,转发来自其它簇的消息,负责引导节点加入和消息路由;叶子节点的作用是定期向中心节点发送节点信息,提供简单的访问服务。

定义4(服务节点) 服务节点(ServerNode, SN)是中心节点和路由节点的总称。

定义5(簇) 簇(Cluster)是路由节点、中心节点集合和叶子节点结合组成的三元组。

$$\text{Cluster} = \langle \text{RoutingNode}, \cup \text{CentralNode}, \cup \text{LeafNode} \rangle$$

簇是按节点对资源需求、共享目的和属性的相似性进行划分的基本逻辑管理单位。一个簇内只有一个路由节点,可有多个中心节点和叶子节点。簇内节点有很大的相似系数,簇内文件能在很大程度上代表簇内部提供的服务信息。

2.2 TAC模型的体系结构

在研究P2P网络体系结构的过程中,发现P2P网络拓扑节点的分布具有小世界特征。我们考虑节点的固有特性,按属性特征将节点分放到不同的簇中。每个簇内只有一个路由节点,中心节点有多个,每个中心节点分管不同的叶子节点。簇与簇之间由路由节点连接。图1是基于簇的网络模型体系结构图。

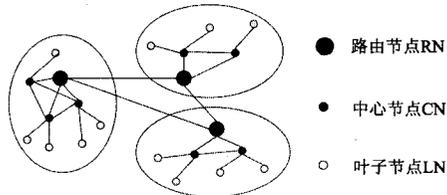


图1 基于簇的网络模型体系结构图

将基于簇的网络模型体系结构图进行抽象,可以得到基于簇的三层网络拓扑图。网络拓扑由路由节点层、中心节点层和叶子节点层构成,如图2所示。

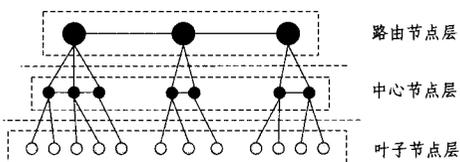


图2 基于簇的三层网络拓扑图

由以上模型描述可知,网络模型具有小世界特性,簇是由

有相似特征属性的节点组成。簇内节点有很大的相似系数,簇内文件能在很大程度上代表簇内部提供的服务信息。因此,在簇内找到满足服务请求的概率较大,节点出现跨簇请求服务的概率很小。

路由节点与簇内中心节点连接,并定期从其它簇的路由节点获取信息,维持一张一定跳数内的邻居簇的簇文件特征列表。中心节点向簇内其它中心节点转发其管辖的叶子节点的搜索请求,簇内节点也可以通过路由节点访问簇外节点的文件信息。

3 协议描述

3.1 节点的加入

节点P请求加入网络中,P找到一个初始路由节点作为P的引导节点(GuidingNode, GN),P将本地属性和特征状态信息传给GN,GN根据P的属性,将P引导到一个和其属性最相似的簇C中,簇C的路由节点RN找到一个地理位置与P最近的中心节点CN,P与CN建立连接,然后将本地信息发送给CN,CN更新索引列表信息。节点的加入算法描述如图3所示。

```

Let C represent the set of cluster, Ci represent the cluster in C
x → GN
if (sim(key(x), key(Ci)) ≥ sim(key(x), key(Cj)), Cj ∈ C)
then
    getconn(x, RNi)
end if
if (dist(x, CNij) ≤ dist(x, CNik), CNik ∈ Ci)
and (Capacity_CNij + 1 ≤ Max)
then
    Join(x, CNij)
end if
    
```

图3 节点加入算法

3.2 节点的退出

节点的退出分为叶子节点的退出和服务节点的退出。

叶子节点退出时向其所属的中心节点发送退出请求,中心节点断开与该叶子节点的连接,并更新索引列表信息。网络上的叶子节点可能由于掉线、机器故障等原因非正常退出,为了及时发现叶子节点退出,采用定期心跳法进行错误探测。

服务节点正常离开网络时,它首先定位并通知退出以后将代替它的服务节点,新的服务节点建立系统服务、资源、状态信息表和连接,在新的服务节点取代原有服务节点进入正常工作之后,原有服务节点断开同其它节点的连接并退出系统。如果服务节点由于掉线、机器故障等原因非正常退出,当系统探测到服务节点失效时,系统选取一个准服务节点,取代失效的服务节点。

为了增强系统的鲁棒性,在服务节点正常工作时,会选取一个节点作为候选服务节点,服务节点上所保存的其管辖范围内的节点信息也会保存在候选服务节点上。当服务节点失效后,系统选取候选服务节点作为准服务节点,取代原有服务节点。

如果服务节点和候选服务节点同时失效,则系统选取的准服务节点的方式为:若该服务节点为中心节点,则由该中心节点所在簇的路由节点暂时代理该中心节点,并在该中心节点下的叶子节点中选出新的中心节点和候选中心节点;若该

服务节点为路由节点,则该簇暂时离开网络,选出新的路由节点和候选路由节点,重新建立簇内网络连接。

3.3 路由协议

对于给定的资源搜索请求,叶子节点根据请求内容的语义相关性决定搜索请求的路由方向(簇内搜索或簇外搜索)。

如果搜索请求的文件内容与簇内共享文件的内容相似性超过一个特定的阈值,则采用簇内搜索,搜索请求消息在簇内各中心节点中传播。叶子节点首先将搜索请求发送给管辖它的中心节点,中心节点查询本地资源索引列表,若命中,则将共享此资源的节点信息返回给请求节点。如果没有命中,则采用一定的策略向其邻居节点转发该搜索请求消息。由于搜索请求只在中心节点之间转发,因此减少了叶子节点的带宽消耗。

如果搜索请求的文件内容与簇内共享文件的内容相似性低于阈值,则采用簇外搜索,搜索请求在簇外各簇的路由节点中传播。叶子节点通过管辖它的中心节点将搜索请求发送给簇内的路由节点,由于每个路由节点都维持一张一定跳数内的邻居簇的簇文件特征列表,路由节点将搜索请求消息发给文件内容相似性超过特定阈值的簇。

4 簇的自聚

为了提高网络性能,我们用簇分裂和簇合并对网络的规模进行控制和管理,控制簇的规模和网络负载。

4.1 簇分裂

当一个簇内的节点数量过多后,簇内服务节点负载会增大,系统使用效率降低。我们为每个簇设定一个节点数量的最大值,当簇内节点数量超过这个值,就要进行簇分裂。在进行簇分裂时,将簇中的中心节点分为两部分,一部分中心节点及其管辖的叶子节点继续放在原簇,另一部分中心节点及其管辖的叶子节点放入新簇,两个簇都重新组织簇内和簇外的连接。簇分裂算法描述如图4所示。

```

Let C represent the set of cluster
if (|C|>Max)
then
    RN_old=RN
    RN_new=RN_candidate
    getconn(RN_old,RN_new)
    m←num_CN/2+1
    for (i=0 ;i<m;i++)
        offconn(CNi,RN_old);
        getconn(CNi,RN_new);
    for (j=m;j<num_CN;j++)
        offconn(CNi,CNj);
    end for
end for
end if

```

图4 簇分裂算法

4.2 簇合并

当一个簇内的节点数量过少,簇的使用效率便较低。我们为每个簇设定一个节点数量的最小阈值,当簇内节点数量少于这个值时,就要进行簇合并。在进行簇合并时,选择一个属性特征与该簇最相似的簇进行合并,并在新簇中重新组织簇内部的连接。簇合并算法描述如图5所示。

```

Let C represent the set of cluster,Ci represent the cluster in C
if (|Ci|<Min)
then
    if (dist(RNi,CNij)≤dist(RNi,CNik),CNik∈C)
    then
        while (s≠j)
        do
            offconn(RNi,CNis)
        end do
    end if
    if (sim(key(Ci),key(Cs))≥sim(key(Ci),key(Ck)),Ck∈C)
    and (|Ci|+|Cs|<max)
    then
        m←num_CNi
        for (j=0 ;j<m;j++)
            getconn(CNij,RNs)
        n←num_CNs
        for (t=0;t<n;t++)
            getconn(CNij,CNst);
        end for
    end for
end if
end if

```

图5 簇合并算法

4.3 簇的规模定理

定理(簇的规模定理) 每个簇内所含有的节点数量与簇的性能相关, C 是簇集合, $\forall C_i \in C$,簇 C_i 的规模为:

$$k_i \leq Capacity(C_i) \leq 3k_i - 1$$

其中, k_i 是常数, k_i 值由簇 C_i 的路由节点的负载能力和带宽确定。

证明:假设簇规模的上界为 l_i ,即簇规模

$$k_i \leq Capacity(C_i) \leq l_i$$

1. 如果簇 C_i 的容量已经达到最大值,再加入一个新节点,即节点数量达到 $l_i + 1$ 时,将进行簇分裂, C_i 将近似等分成两个新簇,两个新簇中的节点数量大致相同,约为 $\frac{l_i + 1}{2}$ 。

簇 C_i 经过簇分裂后,如果簇中节点数量太少,则可能刚进行簇分裂又要进行簇合并,导致网络波动和管理开销增大。为了保证分裂产生的新簇再加入新节点后不会马上进行簇合并,簇中节点数量应大于簇规模的下界,因此

$$\frac{l_i + 1}{2} > k_i$$

即有

$$l_i > 2k_i - 1 \quad (1)$$

2. 如果簇 C_i 经过簇分裂后,要经过很长时间才进行簇合并,说明该簇的规模很大,虽然其分裂和合并的操作相对减少,但系统的管理和控制开销急剧增加,服务节点的负担会加重,容易引起信息瓶颈,甚至是服务节点失效,使系统的交互性和鲁棒性急剧下降。为了防止簇的规模很大,当簇分裂后形成的新簇中有 k_i 个节点离开后,该簇需要与其它某个簇进行簇合并,即此时簇中的节点数量小于簇规模的下界,因此

$$\frac{l_i + 1}{2} - k_i < k_i$$

即有

$$l_i < 4k_i - 1 \quad (2)$$

3. 由式(1)和式(2)可得

$$2k_i - 1 < l_i < 4k_i - 1$$

考虑 $l_i = 3k_i - 1$, 当簇 C_i 中的节点数量大于 $3k_i - 1$ 时, 簇被近似等分成两个新簇, 两个新簇中的节点数量大致相同, 约为 $3k_i/2$ 。任意时刻, 只有当此两个簇中任意一个簇中多于 $k_i/2$ 个节点离开或多于 $3k_i/2$ 个节点加入时, 系统才进行簇的合并和分裂操作, 使网络保持相对的稳定。因此, $l_i = 3k_i - 1$ 可以避免簇的频繁分裂和合并。

综上所述, 簇 C_i 的规模边界值取 $k_i \leq \text{Capacity}(C_i) \leq 3k_i - 1$ 是正确、合理的。 □

5 实验结果及分析

我们通过模拟实验来对上述使用三层结构的 P2P 模型 TAC 进行性能评估, 主要使用搜索成功率、搜索跳数、搜索延时三个指标来衡量系统的性能。搜索成功率定义为在单位时间内, 成功的搜索次数占总搜索次数的百分比。搜索跳数指的是单位时间内所有搜索请求的查询跳数的平均值。搜索延时是单位时间内成功搜索所需时间的平均值。搜索成功率越高, 搜索跳数越少, 搜索延时越小, 则网络性能越好。

我们采用 peersim^[9] 模拟工具进行大规模的仿真实验, 对协议的综合性能进行分析。实验中的基本配置如下: 节点总数为 1500 个, 根据节点性能(负载能力和带宽), 将网络中的节点分为 $\times 1, \times 10, \times 100$ 三个层次, 各层次的节点所占比例分别为 92%, 7.2%, 0.8%。网络连接延时分为 4 个数量级: $\times 10$ 微秒, $\times 100$ 微秒, $\times 1000$ 微秒, ∞ 。 ∞ 表示节点间连接超时, 节点之间的连接延时从这 4 个数量级中随机设定。

节点可随机加入或非正常离开网络, 网络状态信息更新频率为 5s, TTL(Time To Live)值设为 5。文档总数为 10000 个, 分为新闻、财经、音乐、体育、科技、教育 6 类。为了简化起见, 每个文档只有 1 个唯一的标识, 没有内容。

我们假定, 只要某个被搜索的节点存储了查询所针对的文档, 那么搜索成功。我们按上述配置模拟 TAC 和 KaZaA。在仿真中, 我们将单位时间设置为 20s。图 6 展示了 TAC 和 KaZaA 随拓扑变化性能的变化。

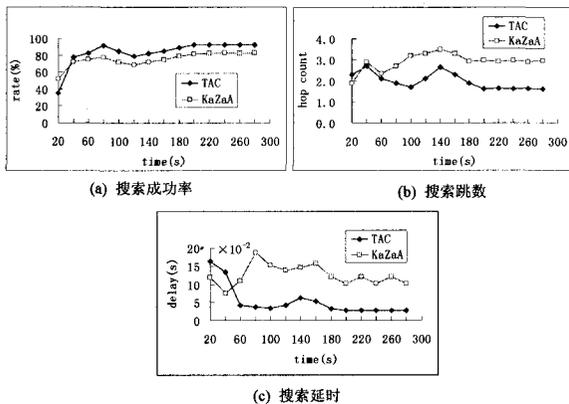


图 6 TAC 与 KaZaA 性能的比较

实验初始阶段, TAC 和 KaZaA 的搜索成功率、搜索跳数和搜索延时均发生了较大幅度的波动, 但在随后的过程中,

TAC 性能逐渐优化, 这是由于 TAC 中节点根据属性自聚, 查询消息由初始阶段的簇内簇外共同搜索逐渐转化为在簇内就能完成搜索, 提高了搜索效率, 减小了搜索跳数。同时, 物理网络拓扑与虚拟网络拓扑更加匹配, 减少了搜索延时。之后, TAC 和 KaZaA 的性能又发生了小幅度的降低, 主要是因为网络中不断有节点随机加入, 与此同时, 部分服务节点同时非正常离开网络。节点数量变化时, TAC 自适应地通过簇分裂和簇自聚调整了簇的规模, 虽然波动幅度比 KaZaA 大, 但是各方面性能参数仍然优于 KaZaA。随后, 网络状态趋于稳定, TAC 和 KaZaA 性能平稳增长, TAC 性能优于 KaZaA。通过实验结果我们可以发现, 基于自聚簇的三层结构 P2P 网络模型 TAC 整体性能优于 KaZaA。TAC 能有效提高系统的整体性能, 在动态自适应方面更具有优势。

结束语 我们提出了一个自聚簇的概念, 构建了一个基于自聚簇的三层结构 P2P 网络模型 TAC, 用以解决以往 P2P 网络兴趣局部性不明显、物理网络拓扑与虚拟网络拓扑不匹配、容易出现瓶颈影响网络性能等问题, 并讨论了其核心算法及其相关支撑技术。TAC 模型的主要工作包括: 在层次结构上, 将传统两层结构的混合式 P2P 中超级节点的节点管理和消息路由功能分别交由中心节点和路由节点进行处理, 减少了超级节点的网络负载; 在拓扑匹配上, 物理拓扑与虚拟拓扑匹配更一致, 减少了查询响应时间; 在网络自组织构造上, 一方面将具有相似兴趣的节点聚合在一起, 增强搜索请求传播的针对性, 减少消息转发次数; 另一方面, 通过簇分裂和簇合并控制簇的规模, 防止服务节点成为瓶颈, 提高了搜索性能。

参考文献

- [1] Zhuang L, Pan CJ, Guo YQ, et al. Connection management based on gnutella network. *Journal of Software*, 2005, 16(1): 158-164
- [2] The Gnutella Protocol Specification v0. 4. http://www9.lime-wire.com/developer/gnutella_protocol_0.4.pdf. 2000
- [3] Lv Q, Cao P, Cohen E, et al. Search and replication in unstructured peer-to-peer networks // 16th ACM International Conference on Supercomputing. New York, NY, June 2002: 84-95
- [4] Leibowitz N, Ripeanu M, Wierzbicki A. Deconstructing the ka-zaa network // 3rd IEEE Workshop on Internet Applications (WIAPP'03). Santa Clara, CA, USA, June 2003
- [5] Xiao Li, Liu Yunhao, Ni L M. Improving unstructured peer-to-peer systems by adaptive connection establishment. *IEEE Transactions on Computers (TC)*, 2005, 54(9): 1091-1103
- [6] Nakao A, Peterson L, Bavier A. A routing underlay for overlay networks // ACM SIGCOMM 2003. Karlsruhe, Germany, 2003: 11-18
- [7] Fast A, Jensen D, Levine B N. Creating social networks to improve peer-to-peer networking // 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining. Chicago, Illinois, USA, Aug. 2005: 568-573
- [8] Sripanidkulchai K, Maggs B, Zhang Hui. Efficient content location using interest-based locality in peer-to-peer systems // INFOCOM 2003. San Francisco, CA USA, 2003
- [9] <http://peersim.sourceforge.net>