

语义对等网中资源元数据混合索引策略研究*

刘震,邓苏,罗雪山,黄宏斌

(国防科技大学 信息系统与管理学院,湖南 长沙 410073)

摘要 面向对等网络信息资源语义共享问题,首先提出了一种基于多本体的资源元数据描述框架,将资源元数据分为核心和扩展两部分,在此基础上重点对资源元数据索引策略进行了研究,提出了一种混合索引机制,针对资源核心元数据采用一种社区策略来进行组织索引,针对扩展元数据采取 DHT 和语义关联相结合的方式组织索引,其中结合了本地索引和全局索引的特点,并在此基础上对资源发现流程和负载均衡方法进行了设计。与现有相关系统相比,本索引体系有效地提高了对等网络环境下资源定位的效率和系统的可扩展能力。

关键词 对等网;本体;元数据;资源索引

中图分类号 TP311 **文献标识码** A

Research on Hybrid Indexing Strategy of Resource Metadata in Semantic Peer-to-peer Network

LIU Zhen, DENG su, LUO Xue-shan, HUANG Hong-bin

(College of Information System and Management, National Univ. of Defense Technology, Changsha 410073, China)

Abstract For the problem of semantic content sharing of information resources in P2P networks, this paper first proposes a multiple-ontology based resource metadata description framework, which divides resource metadata into core metadata and extensional metadata. Then, based on the metadata framework, this paper places emphasis upon the research of indexing strategy of resource metadata. A hybrid indexing architecture is provided. It uses community strategy to organize core metadata element and uses DHT and semantic association to index extensional metadata element, which combines local indexing and global indexing. Based on this indexing strategy, the work flow of resource discovery and the load balance strategy are designed. Compared with the related current systems, this indexing architecture will greatly improve the efficiency of resource discovery and the system's scalability.

Key words peer-to-peer; ontology; metadata; resource indexing

在一些大型企事业团体和虚拟组织环境(如教育系统、军事组织)中,存在着大量的电子信息资源,在地理上具有分布性,而且在资源类型上具有多样性的特点。组织内的各个单位依据专业知识或特定任务分别管理着一组信息资源集合。从技术角度说,不同的单位可以看成是信息网络中多个相对独立工作的结点,这些结点可以随时加入和离开网络,而不用关注其他结点的行为状态,具有对等网络(Peer-to-Peer Network)系统管理的特点。那么,如何有效地索引网络环境中这些信息资源,来保证资源的快速定位和共享是一个亟待解决的问题。现有的 P2P 系统,如第一代的 Gnutella^[1]、KaZaA 等已经比较普及,第二代的 PAST^[2]、CFS^[3]等基于 DHT 的 P2P 系统也得到了很快的发展。而这些系统主要面向文档共享,其中提供的信息检索能力还比较局限,往往局限于关键字或全局标识匹配的层次上。对面向语义的、多样化的信息资源管理和共享支持还不够,而且不能支持模糊查询。在这些系统中如第一代 P2P 系统主要采用一种本地索引策略,针对非结构化的网络拓扑,本地索引策略中,元数据基于文档空间进行分割,每个文档的所有术语列表都存储在一个结点上。在这种索引策略下,资源查询被广播给所有结点,其缺点是每个结点都涉及到查询过程中,导致系统的可扩展性较差。对于第二代 P2P 系统主

* 收稿日期:2006-05-18
基金项目:国家 863 高技术基金资助项目(2002AA131010)
作者简介:刘震(1976-),男,博士生。

要采取一种全局索引策略,针对结构化网络拓扑,在全局索引策略中,元数据基于术语进行分布。每个结点中管理着一组术语的逆向列表,即包含特定术语的所有文档列表。包含多个术语的查询请求被发送到负责这些术语的结点,识别同时包含这多个术语的文档的联接操作所需的通信开销随着术语的逆向列表的增大而相应增大,而且只能提供精确匹配。

对于现有的基于模式和本体的P2P系统,如EDUTELLA^[4],SWAP^[5]等主要采取一种本地索引策略和一种类Gnutella的松散无结构对等网络,路由还是具有一定的盲目性,系统的可扩展性较差。本文考虑通过对网络环境下各类信息资源进行基于本体的元数据描述的基础上,提供一种更强的资源共享能力和语义查询的能力。元数据描述模型和查询模型在文献[6]中进行了详细介绍,本文的重点是提供一套有效的索引体系,来支持这种对等网络环境下资源的快速高效定位。其中结合了本地索引和全局索引的特点,采取社区策略、DHT和语义路由相结合的方式,对系统中资源的核心元数据和扩展元数据的语义元素进行有效索引,从而提高系统中面向语义、支持复杂查询的资源发现能力,而且有效增强了系统的可扩展性。

1 相关研究工作

在对等网领域,如Gnutella^[1],Pastry^[7]等,往往采用关键字或全局标识(如文件名)来作为资源元数据描述,查询主要采用字符串匹配的方式,大大局限了资源匹配的灵活性和面向内容的资源定位能力。在语义对等网研究中,采用基于语义本体和模式的方式来描述共享资源的元数据和相关语义内容已经成为当前研究的一个趋势。EDUTELLA^[5],SWAP^[6]是基于对等网体系结构的知识共享研究项目,它们都采用RDF来进行元数据描述。在EDUTELLA中采用基于Super-peer的资源管理机制,基于本体、规则或查询对数据资源进行聚类,并采用一种Hypercube的拓扑结构来保证对于每个查询每个结点只被访问一次,但其采用的是一种Super-peer间的广播策略。而SWAP中没有采用一种结构化的拓扑结构,相对拓扑维护开销较小,其提供一种基于REMINDIN的查询路由策略,主要采用一种基于学习的路由转发机制。在PeerDB^[8]中,采用一种基于关系模式的资源元数据描述框架,每个结点维护一组本地可获取的元数据集合(Local Dictionary)和一组对外(其他结点)提供的元数据集合(Export Dictionary)。在以上这些P2P系统中采用的分布资源索引方法着重在语义互操作性上,系统主要采取本地索引机制,当结点规模不断扩大时,系统性能会急剧下降,因而可扩展性较差。本文考虑结合多种资源索引策略,提供一种面向语义共享、可扩展能力更强的资源索引服务。

2 元数据模型和查询模型

本文采用对等资源Agent网络来对多个组织的分布异构的各类资源进行管理。资源的组织管理结构层次如图1所示,每个资源Agent结点管理着注册上来的一个或多个资源对象。采用基于本体的方法来对资源的元数据进行描述,其中每个本体都是建立在对自己相关领域的理解基础上。

本系统中信息资源对象的元数据采用一种基于本体的模式进行描述。元数据包括两个层次:核心层(Core Layer)和扩展层(Extensional Layer)。核心层是系统应用环境内每一个资源描述都包含的部分,是对环境中所有资源的共性特征和语义本体的描述,其中包括资源管理元素、概要元素、物理特征元素、安全和信任元素等,扩展层则是根据具体应用领域和任务环境下对核心元数据的扩展,提供面向特定领域或特定任务的本体模型,提供粒度层次更高的、面向语义内容的资源描述,将有效地提高信息资源检索的效率和精度,如针对数据库资源的模式语义的描述,多媒体信息语义内容的抽取等。对于结点本体结构和元数据模型的结构详细描述可参见文献[6],下面根据资源对象分布索引说明的需要,对资源对象的相应元数据结构和资源发现请求模型进行了一个简化描述。

资源对象的元数据结构 $M_{object} = (C_{key}, C_{ext}, A, R, P, I)$, T 是术语集合, D_{basic} 是预定义的基本数据类型集合。其中: C_{key} 为核心层概念集合, C_{ext} 为扩展层概念集合。一个概念的可以形式化定义为 $c = (t_c, A_c)$ 其中 $c \in C_{key} \cup C_{ext}$, $t_c \in T$, $A_c \subseteq A$, 表示 c 的属性特征集合。 A 是所有属性的集合。每个属性可以形式化定义为 $a = (t_a, dt_a)$, $t_a \in T$, $dt_a \in D_{basic} \cup C_{ext}$ 。 R 定义了概念之间的二元语义关系集合, $\forall r \in R$ 可表示为 $r < c, c' >$, 本系统中描述的语义关系主要包括:等价关系 R^{same} 、继承关系

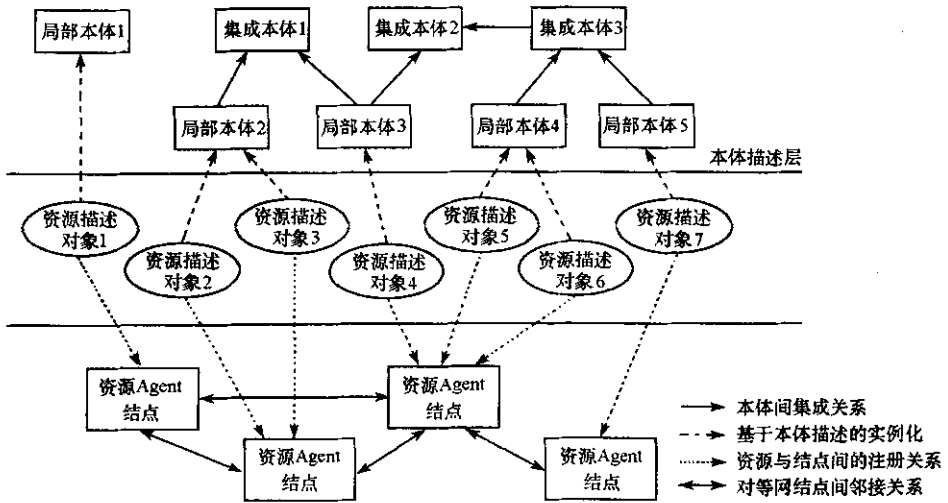


图1 基于复合多本体结构的资源组织结构

Fig. 1 Multi-ontology based resource organization architecture

H^{Inh} 、聚合关系 H^{Aggr} 、关联关系 R^P 。 P 是资源对象相关属性约束集合。 I 是资源针对 C_{key} 和 C_{ext} 的所有概念实例集合。

资源发现请求模型,即查询模型 $Q=(C_Q, R_Q, qc)$ 。 C_Q 表示查询中涉及的概念集合,包括核心和扩展层概念。 R_Q 表示查询中涉及的关系集合。 qc 定义了查询中概念实例必须满足的约束条件。

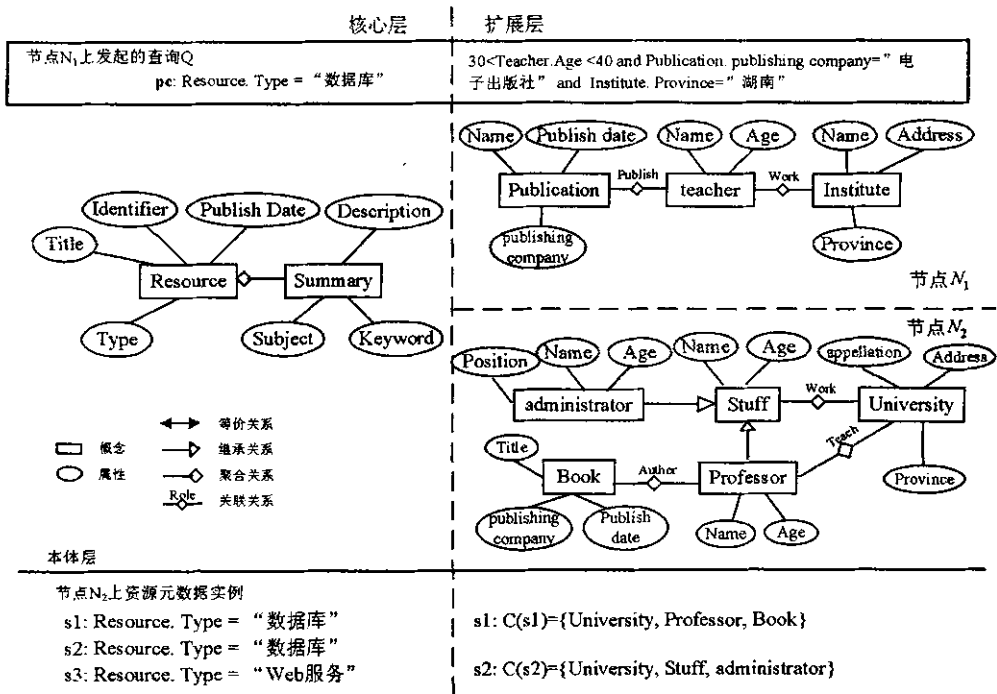


图2 一个简化资源元数据和查询描述实例

Fig. 2 A simple example of resource metadata and resource request

如图2所示,提供了一个教育领域的相关元数据描述和查询描述的简化实例。左边对核心元数据进行了描述,包括资源标题(Resource. Title)、资源类型(Resource. Type)等;右边对扩展元数据进行了描述,分别对不同节点 N_1, N_2 的资源语义内容进行了描述,包括概念和概念间关系描述,如节点 N_1 中

的概念 Publication、teacher 等, 结点 N_2 中的概念 University、Professor 等; 上面的方框内对一个查询实例进行了描述; 下面对结点 N_2 具体资源实例 s_1, s_2, s_3 的元数据进行了描述。

3 总体索引体系及其工作机制

系统总体索引体系结构如图 3, 总体思想: 各数据源依据第 2 节中描述的元数据结构对资源对象的元数据进行描述, 并将资源描述文档注册到其隶属的资源 Agent 结点 N_i 。 N_i 负责相应资源对象的管理和相关语义元素索引的任务。 本系统中根据资源核心元数据和扩展元数据分别进行相应的分布索引。 对于核心元数据元素是所有资源都必须包含的部分, 采取分组的方式, 将 N_i 加入到满足特定约束 $p_i(C_{key})$ 的社区 G_i 。 扩展层元数据元素不是所有资源对象都包含的内容, 当系统规模不断扩大的情况下, 采用本地索引的方式定位效率和命中率都较低, 为了在分布环境下高效定位, 采取一种全局索引和本地索引相结合的 DHT 方式对扩展层概念进行索引。 本系统中采用结构化重叠网络 Chord^[9] 协议提供相应的扩展层概念的检索服务。

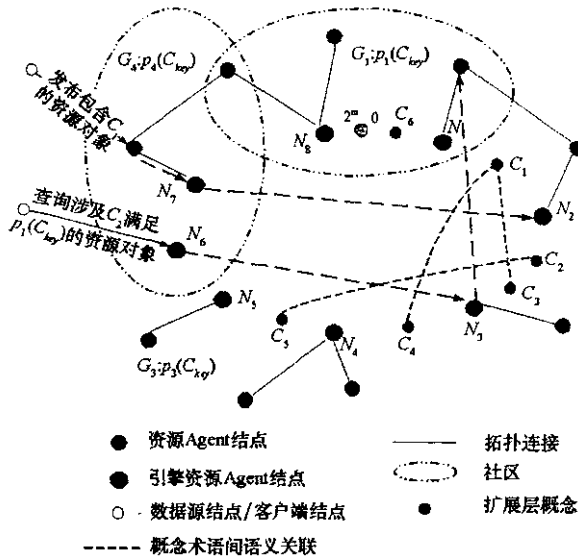


图 3 总体索引体系结构

Fig.3 General indexing architecture

3.1 基于社区的核心元数据索引

对于核心元数据采取一种基于社区重叠网络的方式对满足特定约束 $p_i(C_{key})$ 的资源 Agent 结点空间进行划分, 如根据核心元数据概念元素“Resource”来划分结点空间, $p_1(\{Resource\}) = \text{“Resource. Type=‘数据库’”}$, 则此社区内所有结点都包含数据库资源。对于每一个社区都存在相应的社区标识 GID 和相应的社区入口约束 $p_i(C_{key})$, 社区之间可能重叠和互相包含。当某一新的资源对象 r_i 加入到隶属的资源 Agent 结点 N_i , 首先 N_i 对 r_i 的元数据 M_{ri} 进行解析, 抽取其中的核心概念和扩展概念元素分别进行处理, 对扩展层概念的处理将在 3.2 节中进行介绍。对于核心层元数据 $M_{ri}^{key} = (C_{key}, I_{ri})$ 利用不同社区间的连通关系, 将 M_{ri}^{key} 传递给系统中的所有社区, 匹配相应社区 G_i 的 $p_i(C_{key})$, 当 I_{ri} 满足 p_i 则将 N_i 加入社区 G_i , 并建立与社区 G_i 之间的拓扑连接。

3.2 基于 Chord 的扩展元数据混合索引

扩展层元数据不是所有资源对象都包含的部分, 是根据具体领域对元数据的语义扩展, 所以能够更好的区分信息资源, 为了更为高效准确地定位资源, 采用 DHT 的方式对其进行索引。本系统中采用 Chord 协议^[9] 在结点间维护环状结构, 并使用 $SHA1$ ^[10] 作为一维环状空间中从扩展层概念到结点的映射方法。Chord 环形网络中的引擎结点是从资源 Agent 结点中选择出来, 一般引擎结点是具有较长的

在线时间和负载能力较大的结点,对于非引擎结点的资源 Agent 结点必然连接到一个引擎结点。每个引擎结点映射到一个 $0 \sim 2^m$ 的一个环状空间上,包含一个后继结点列表(Successor List)、查找路由表(Finger Table)和语义关联结点列表(Semantic List)。结点 ID 为 n 的后继结点列表中列出在标识空间上 ID 紧接在当前结点 n 之后的结点,结点查找路由表(Finger Table)中结点 ID 呈 2 的指数级间隔,最多包含 m 个条目,第 i 个条目中存储着结点 $s = \text{successor}(n + 2^{i-1})$ 。当发布一个资源对象 r_i 的扩展元数据的时候,首先抽取其中所有的扩展层概念 c_i ,通过 SHA1 哈希转换得到键值 k_i ($0 \leq k_i \leq 2^m$),由结点 ID 最接近 k_i 的后继结点负责相应概念 c_i 的管理,同时将扩展元数据中的概念列表通过组播分发到负责这些概念的所有结点。在引擎结点中信息资源的索引结构如图 4 所示。采用一种本地索引和全局索引相结合的方式,这样将避免涉及多概念查询时的通信开销,提高资源命中的效率。为了能够定位相关概念的近似概念,引擎结点会周期性触发相应的术语语义关联检查,当周期内存在概念术语增加,引擎结点会把增量术语与其他结点中术语语义关联度进行计算,建立不同引擎结点间的术语关联(同义、广义或狭义),即生成结点间拓扑连接并将相应术语关联度存入到语义关联结点列表。

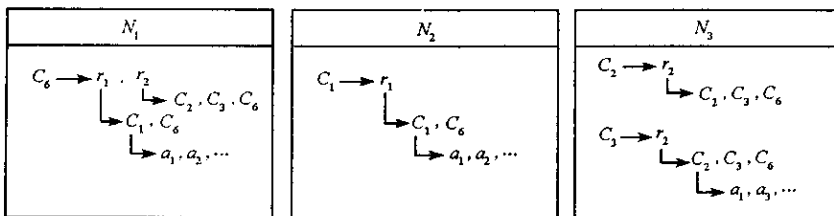


图 4 基于扩展层概念的混合索引

Fig. 4 Hybrid indexing of extensional concepts

3.3 资源发现流程

资源发现流程如下:

步骤 1:客户端 X 可以发起资源发现请求 Q 到任意资源 Agent 结点 N_i ,请求结构如第 2 节中描述的查询模型结构定义。

步骤 2: N_i 对查询进行解析,提取其中的核心概念集合 C_k ,核心概念约束 $\mu(C_k)$,以及扩展层概念集合 C_e 。

步骤 3:如果 $C_e = \phi$,基于社区进行资源定位,在不同社区 G_i 间转发查询 Q ,判断 G_i 对应社区约束 $q(C_k)$ 是否与 $\mu(C_k)$ 相容,即满足两个约束的实例是否存在交集,存在则遍历社区内所有结点,将满足 $\mu(C_k)$ 的资源对象 ID 返回给 N_i , N_i 将结果返回给客户端 X ;否则将查询传递给下一个社区。直到达到特定 TTL 值或遍历了所有社区。

步骤 4:如果 $C_e \neq \phi$, N_i 将查询 Q 发给邻近 Chord 网络引擎结点 N_j , N_j 基于 Chord 路由协议将 Q 传递到负责 C_e 中涉及概念术语的所有引擎结点 N_x 。

步骤 5: N_x 针对术语对应资源的本地索引进行一次本地匹配,并对其匹配度进行评分,选取其中满足特定阈值的资源对象,并将查询 Q 发送到负责这些资源对象的资源 Agent 结点,判断是否与相应结点社区约束相容,不相容则结束操作,否则进行包含核心、扩展概念和相应实例的完全匹配,匹配算法参见文献 [11]。最终将匹配度满足特定阈值的资源对象返回给 N_i ,并最终返回给客户端 X 。

步骤 6: N_x 根据概念术语语义关联将查询传递给相应引擎结点 N_y , N_y 执行操作与第 5 步中 N_x 执行的操作相同。

在涉及扩展层概念查询时,查询平均访问结点数近似为 $O(m \log_2 N)$,其中 m 为查询平均涉及扩展概念数, N 为引擎结点数。在只涉及核心概念查询时平均访问结点数近似为 $O(G + N_g G_m)$,其中 G 为社区数, G_m 为平均匹配查询约束的社区数, N_g 为 G_m 中每个社区平均包含结点数。

为了避免对相同查询的重复处理以及减轻对于一些普及查询的热点问题,查询结果可以在查询发

起资源 Agent 结点和查询传递路由结点上缓存一段指定时间,如果类似查询到达一个结点有现存的查询结果,则立即返回查询结果而不继续路由。

3.4 负载均衡

现在计算机的存储能力有了突飞猛进的增长,但是对一些特别普遍的扩展概念术语的维护,会极大地增加系统的存储开销和计算开销,因而可以在引擎结点中设计了一个普及度阈值(PT)。当引擎结点的某概念术语的逆向列表长度 $L > PT$ 时,该引擎结点将停止对此概念术语 t 的索引,并停止对涉及 t 的查询的处理。这一操作对系统的查询性能不会有太大影响,因为只用普及概念进行资源定位,无法有效地区分资源,这种查询往往意义不大。

即使对一些普及概念进行相应处理以后,各个引擎结点中索引的概念术语还是不可能完全均衡,有些引擎结点索引的资源数量可能较大。同时不同术语的访问量也存在不均衡性,这样某些引擎结点的计算负载可能过高,从而影响系统的总体性能。针对这一问题,本文采取一种自适应负载均衡机制。假定某一引擎结点 N_x 的索引键值范围为 $[K_b, K_e]$,当 N_x 的索引数据量或单位时间数据访问量大于某一阈值时,即触发引擎扩展事件。首先在 $[K_b, K_e]$ 间随机产生键值 K_t ,同时 N_x 采用随机步(random walk)路由的策略在非引擎结点中查找满足引擎结点条件(在线时间 and 负载能力满足特定条件)的结点 N_y ,并将 K_t 作为 N_y 的结点 ID,将 N_y 加入到引擎环状空间中来,对 N_x 的键值范围进行划分,从而 N_x, N_y 分别负责 $[K_b, K_t], [K_t + 1, K_e]$ 的资源索引。通过这种自适应的负载均衡策略将有效地均衡各引擎结点的负载。

4 模拟试验

本系统根据设计对 Chord 源码进行必要修改,生成相应仿真试验平台 RA_Chord。在实验环境中模拟 100 个资源 Agent 结点,并在其中选择 32 个结点作为 Chord 引擎结点,简单设计相应的核心元数据,包括一个“Resource”核心概念,并以“Resource.Type=数据库”、“Resource.Type=图像”、“Resource.Type=视频”、“Resource.Location=长沙”、“Resource.Location=其他”构建五个社区,并模拟 500 个资源。资源采用院校领域常用概念,基于本系统设计的元数据模型构建 500 个资源扩展元数据描述,平均每个资源扩展层元数据包括扩展层概念 20 个,将资源随机分布到 100 个资源 Agent 结点上,并将相应的扩展层概念索引到引擎结点。构建了一个所有概念的语义关联字典库,预先根据所辖术语间的语义关联建立 Chord 引擎结点间拓扑连接。同时,由于现有的基于语义模式的 P2P 系统主要采用一种无结构的拓扑网络,因而试验采用一种类 Gnutella 松散无结构对等网络来进行对比,路由策略采用 5 个随机步(Random Walker)的方案,TTL=8 来进行模拟。随机步方案就是随机地选择当前结点的一个邻结点转发查询请求,直达到达相应的 TTL 值。

为了体现本系统中资源索引方法的高效性,本文从搜索性能上来进行评价,采用以下指标来评价系统的性能。(1)搜索命中率表示成功搜索的比例,定义为成功搜索到的资源数目与实际匹配资源数目的比值。如图 5 所示,分别随机在 100 个结点中选取 20、40、60、80、100 个,在 RA_Chord 中并分别取 8、16、16、32、32 个结点作为引擎结点,得到相应查询命中率,可以看出在本系统随着规模扩大,对查询命

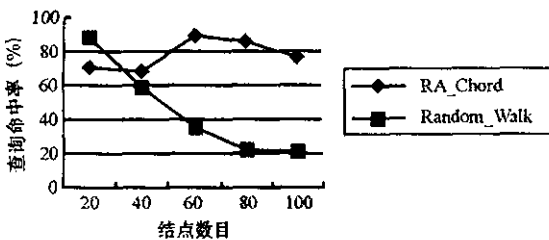


图 5 搜索命中率比较

Fig. 5 Contrast of hit ratio of search

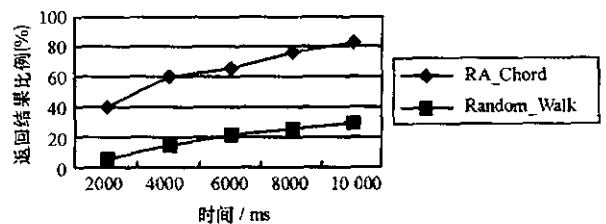


图 6 查询响应时间比较

Fig. 6 Contrast of reaction time of search

中率的影响不大,性能上明显优于 Random_walk 方法(2)搜索开销,采用搜索延迟时间来度量查询开销的大小,如图6所示,采用100个结点,32个 Chord 引擎结点,针对10个查询进行统计,看不同时间内返回命中结果的比例,从图中可以看出本系统的查询效率明显高于 Random_walk 方法。说明本系统在访问较少结点的基础上更快地定位到相应资源。

从仿真实验结果可以看出,本系统搜索效率明显优于其他系统,而且系统的可扩展性较好。但是当查询中不涉及扩展层概念时,查询主要采取一种基于社区的语义关联路由的方式,所以效率上与其他基于社区(兴趣域)的 P2P 系统相当。

5 总结

本文针对语义对等网络环境,采用一种基于本体的资源元数据描述机制,并在此基础上,提供一套有效的混合索引体系,面向资源语义概念元素的索引,其中结合了本地索引和全局索引的特点,采取社区策略、DHT 和语义路由相结合的方式,对系统中资源的核心元数据和扩展元数据的语义元素进行有效索引,有效地提高系统中面向语义、支持复杂查询的资源发现能力,而且通过模拟试验可看出,相对于现有基于语义和模式的 P2P 系统有着更高的查询命中率和更强的可扩展性。

参考文献:

- [1] Ripeanu M, Foster I, Iamnitchi A. Mapping the Gnutella Network: Properties of large-scale peer-to-peer systems and implications for system design[J]. Internet Computing, 2002, 6(1): 50-57.
- [2] Rowstron A, Druschel P. Storage Management and Caching in PAST, A Large-scale, Persistent Peer-to-peer Storage Utility[A]. In SP'01[C], 2001.
- [3] Dabek F, Kaashoek M. et al. Wide-area Cooperative Storage with CFS[A]. In SOSR'01[C], October 2001.
- [4] Nejdl W, Wolf B. et al. EDUTELLA: A P2P Networking Infrastructure Based on RDF[A]. In International World Wide Web Conference(WWW'02[C]), 2002.
- [5] Ehrig M, Haase P, et al. The SWAP Data and Metadata Model for Semantics-based Peer-to-peer Systems[A]. In Multiagent System Technologies(MATES'03[C]), 2003.
- [6] 刘震,邓苏,罗雪山,等.面向对等网信息语义共享的元数据模型框架研究[J].计算机科学,2006,31.
- [7] Rowstron A, Druschel P. Pastry: Scalable, Distributed Object Location and Routing for Large-scale Peer-to-peer Systems[A]. In: IFIP/ACM International Conference on Distributed Systems Platforms(Middleware'01[C]), Heidelberg, Germany, 2001: 329-350.
- [8] Ooi B C, Shu Y, Tan K L. Relational Data Sharing in Peer-based Data Management Systems[A]. ACM SIGMOD Record[C], 2003, 32(3).
- [9] Stoica I, Morris R, Karger D, et al. Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications[A]. In: Proceedings of ACM SIGCOMM[C] 2001.
- [10] National Institute of Standards and Technology. Publication 180-1: Secure Hash Standard[S], 1995.
- [11] 刘震,邓苏,罗雪山,等.基于多本体语义相似度计算的对等网资源动态匹配算法研究[J].计算机科学,2006,31.

