

基于 P2P 的 RFID 编码解析网络结构与算法研究

刘学洋^{1,2}, 赵 文^{1,2}, 张世琨^{1,2}, 王立福^{1,2}

(1. 北京大学软件工程国家工程研究中心, 北京 100871;

2. 北京大学信息科学技术学院软件研究所高可信软件技术教育部重点实验室, 北京 100871)

摘 要: 本文结合 P2P 技术, 提出了分级 P2P RFID 编码解析网络结构. 并基于改进的 kademia 算法, 在该网络中实现了 RFID 编码解析服务. 实验数据表明, 分级 P2P RFID 编码解析网络相对于基于 DNS 技术的 EPCglobal ONS, 具有负载均衡、可扩展性好、能有效应对单点失效等优点. 最后, 本文给出分级 P2P RFID 编码解析网络需要继续深入考虑的问题.

关键词: RFID; 编码解析网络; 分级 P2P; Kademia

中图分类号: TP311 **文献标识码:** A **文章编号:** 0372-2112 (2008) 12A-102-07

Research on P2P-Based RFID Code Resolution Network Architecture and Algorithm

LIU Xue yang^{1,2}, ZHAO Wen^{1,2}, ZHANG Shi kun^{1,2}, WANG Li fu^{1,2}

(1. National Engineering Research Center for Software Engineering, Peking University, Beijing 100871, China;

2. Key Laboratory of High Confidence Software Technologies (Ministry of Education), School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China)

Abstract: In this paper, a hierarchical P2P RFID code resolution network architecture is proposed. Based on modified Kademia, RFID code resolution service is implemented. By simulating, this paper draws the conclusion that hierarchical P2P RFID code resolution network has such advantages as load balance, scalable, single node failure tolerance etc. Finally, future works are given.

Key words: radio frequency identification (RFID); code resolution network; hierarchical P2P; Kademia

1 引言

无线射频识别 (Radio Frequency Identification, RFID) 技术是一种非接触、多目标、移动目标识别的自动识别技术. RFID 跨地区、跨行业乃至全社会的规模化应用需要建立一套可靠高效的公共服务基础设施和信息共享机制, 而这种共享机制的核心是建立一个高效健壮的 RFID 编码解析网络, 但目前这一机制尚未成熟. 美国 EPCglobal 的技术体系^[1] 和日本泛在 ID 中心 (Ubiquitous ID Center) 的 RFID 标准体系^[2] 均涉及 RFID 编码解析服务的内容.

RFID 编码解析服务负责将 RFID 编码解析为存放物品静态信息的网络资源地址 (URI). EPCglobal 提出的 ONS^[3] 采用了域名解析服务 (DNS) 的实现原理, 实现分层管理、分级分配. EPC-ONS 现在全球部署了 14 台 Root ONS 服务器, 每台 Root ONS 服务器负责一个特定国家

或地区的 EPC^[1] (Electronic Product Code, 电子产品编码) 编码解析服务. 根据 EPC 编码标准和实现原理, 在一个国家或地区内, Local ONS 数量巨大, 这势必造成 Root ONS 服务器负载过重问题, 但是 EPCglobal 和 Verisign 尚未明确提出针对 Root ONS 服务器过载问题的解决方案.

目前, 在解决传统 DNS 的根节点负载过重问题, 实现域名解析网络的负载均衡和单点失效问题等方面, MIT、微软等大学和企业开始研究在域名解析中采用 P2P (Peer-to-Peer) 技术.

P2P 技术用于构建点对点网络, 是随着网络应用而兴起的一门新技术. 它主要指计算机之间弱化或取消了服务器的作用, 以对等方式形成网络连接, 各台计算机可以意识到周围其他计算机的存在, 允许任意一台计算机直接去访问其他计算机. P2P 网络具有完全对称、无服务器、可扩展性好等特点.

目前, 在 P2P 网络结构方面, 主要有中心化拓扑、全

分布式非结构化拓扑、全分布式结构化拓扑和半分布式拓扑等几种。

为了提高查询效率, L. Garcés-Erice 等人提出了一种层次式的 P2P 网络结构^[4]。这种层次式的 P2P 网络结构相对于扁平化的 P2P 网络而言, 可以有效地降低操作的延迟。同时, 这种结构中, 本地数据在异地存储、异地查询仍然普遍存在并且是不可避免的, 这在很大程度上抵消了层次式 P2P 网络降低操作延迟的效果。

利用 P2P 的优势, 麻省理工大学的 Cox^[5] 等人基于 DHash——一种建立在 Chord^[6] 算法基础上的分布式哈希表来实现域名解析服务。他们设计的系统 DDNS 不但减少了大量的 DNS 管理问题, 还继承了 Chord 单点失效和负载均衡的特性, 并具有较高的可恢复性等优点。另一方面, 相对于传统 DNS, DDNS 的延迟却高出很多。在 DDNS 中, 如果系统具有一百万个节点, 那么一次查询需要 20 次 RPC, 而传统 DNS 一般仅需要 2~4 次 RPC。

柏林洪堡大学的 Benjamin Fabian^[7] 等人将 P2P 技术应用于 EPCglobal ONS 的实现, 提出了一种基于分布式哈希表(DHT)的对象信息分布式体系结构 OIDA(Object Information Distribution Architecture)。他们分析了基于 OIDA 实现的非集中式 ONS 进行数据访问的方式, 以及对 RFID 编码解析服务私密性保护的途径。他们的研究结果显示, 相对于 EPCglobal ONS 而言, 基于 DHT 的非集中式 ONS 的私密性保护有所增强。但是文献[7]中, 并未考虑基于 OIDA 实现的 ONS 的查询效率问题。

当前 RFID 编码解析服务中, 主要存在的问题如下:

(1) 服务器过载

DNS 查询的域名空间大小和全球的计算机数量是同一量级的, 但是对 RFID 来说, 如果 RFID 标签得到普遍使用, 那么可能每件商品上都有一个以上的 RFID 标签, 从而 RFID 标签的编码空间大小将远远超出 DNS 的域名空间若干个数量级, 从而给 RFID 编码解析服务带来沉重的压力。

(2) 单点失效

在部分编码解析服务器失效的情况下, 编码解析网络应该具有继续提供正确服务的能力。

(3) 编码方案多种多样

目前不同国家、不同行业已经提出了多种 RFID 编码方案, 并且这些方案有些兼容有些不兼容。不同的编码方案, 其编码解析方式也各不相同。鉴于目前商品流通的全球化, 提供一种可以兼容各种编码方案的编码解析服务成为必需。

本文提出了一种分级 P2P RFID 编码解析网络结构, 以达到增强 RFID 编码解析网络负载均衡、容忍单点失效的目标, 并支持与不同编码解析网络的平等互联, 同时具有较高的 RFID 编码解析效率。

2 RFID 编码解析网络结构

本节介绍了 863 课题“RFID 公共服务体系架构设计及应用服务关键技术研究与开发”项目组结合我国国情提出的一种 RFID 编码方案 CID-V, 基于该编码结构本文提出了一种分级 P2P RFID 编码解析网络结构, 并给出在该网络中数据存储、编码查询等基本流程。

2.1 一种 RFID 编码方案—CID-V

在 RFID 编码方面, 目前存在多种方案, 包括 EPCglobal 的产品电子编码(EPC)^[1] 和日本泛在 ID 中心的 ur-code^[2]。在这些编码方案中, 均不包含地域信息。在目前我国行政管理、企业管理、产品质量检测与监督等方面, 实行的是分地域、分级管理。所以, 为了有效支持这种管理方式, 需要在编码中加入地域信息。在分析 EPCglobal 的标签数据标准和国内电子标签标准工作组提出的数据格式的基础上, 项目组提出了一种新的 RFID 编码规范—CID-V。编码方案如表 1 所示。

表 1 可变长 RFID 编码 CID-V 结构

码段	头部	有效位	分段标示位	国家、地区标识
长度(bits)	8	4	3	4-16
码段	城市区号	组织机构代码	产品分类码	序列号
长度(bits)	8-16	48	4-27	61-38

CID-V 中各编码段说明如下:

- 头部: 标识不同编码类型, 对一种编码类型该段取值是固定的;

- 有效位: 共 4bits, 前 2bits 表示国家地区标识码的长度, 后 2bits 表示城市区号的长度;

- 分段标示位: 共 3bits, 用来表示产品类别码和序列号的分割位置, 而产品类别码和序列号总长度是一定的;

- 国家、地区标识: 采用全球区号码, 长度 4-16bits;

- 城市区号: 采用电信网电话区号; 长度 8-16bits, 从头至尾每 4bits 表示一个十进制数字;

- 组织机构代码: 标识一个企业或组织, 由 8 位数字或大写字母表示, 每一位由 0~35 表示(A~Z 由 10~35 表示), 共需 48bits;

- 产品类别码: 在同一个企业或组织内部, 唯一标识一个产品类别; 长度可变;

- 序列号: 在同一个产品类别下, 唯一标识一个单品, 长度可变。

该编码方案的优点在于: (1) 适应我国企业管理制度地域性强的现状, 能有效满足按级别、按地域管理的要求; (2) 兼容其它国内外编码方案, 支持与其它编码的转换与相互查询。

2.2 分级 P2P RFID 编码解析网络结构

对于编码解析服务而言, 其主要功能是将 RFID 编

码解析为存放物品静态信息的网络资源地址 URI(我们称之为 RFID IS 地址). 在实际应用中, RFID IS(信息服务)一般由企业提供一个或多个 IS 服务器, 一个 RFID IS 地址对应企业的大量产品编码.

为解决 RFID 编码解析网络负载均衡、单点失效等问题, 并获得较好的查询效率, 本文引入 L. Garces Erice 等人提出的层次式的 P2P 网络结构^[4]以实现编码解析网络. 如果直接在 RFID 编码解析中应用层次式的 P2P 网络结构^[4], 对 RFID 编码解析服务而言, 在这种结构中存储大宗同类商品的 RFID 编码和 IS 地址的相关信息, 那么这些同类商品的信息将会分布到各个不同的节点组中, 从而存储和查询代价会急剧上升, 并且不利于缓冲机制的使用. 本文将对层次式的 P2P 网络结构进行改进以适应 RFID 编码解析服务的需求.

通常, 每个应用 RFID 编码并提供 IS 服务的企业或组织会建立其 RFID 编码解析服务器, 并和 RFID 编码的管理机构进行互联. 因此, 在 RFID 编码解析网络中, 最基本的节点是分布在各企业和组织的编码解析服务器. 另外, 根据 CID-V 编码中的国家地区标识和城市区号, 可以将同一地域的编码解析服务器组织在一起, 形成一个区域的编码解析服务器组, 从而充分利用组内服务器通信速度高的优势.

本文基于 CID-V 编码方案, 提出了一种适应 CID-V 编码结构的分级 P2P RFID 编码解析网络结构, 该结构支持多级的 P2P RFID 编码解析网络. 本文将以两级的 P2P RFID 编码解析网络为例, 说明分级 P2P RFID 编码解析网络的结构. 两级的 P2P RFID 编码解析网络结构示意图如图 1 所示.

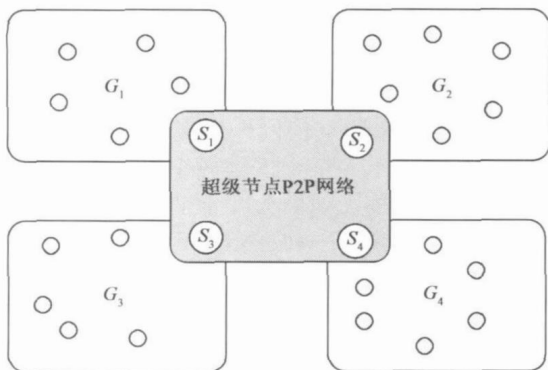


图1 两级P2P RFID编码解析网络结构示意图

对于一个两级的 P2P RFID 编码解析网络而言, 其上级网络用于解析编码对应的节点组, 下级网络解析编码对应的 IS 地址. 因此, RFID 编码格式可以归纳为 C_1, C_2 , 其中 C_1 段用于上级网络解析. 在 CID-V 中, C_1 段对应从头部到组织机构代码的前六个字段, C_2 段对应产品分类码和序列号两个字段.

在两级的 P2P RFID 编码解析网络中: (1) 所有节点

划分为节点组, 各节点组内组织为 P2P 网络, 每个节点组具有唯一的组标识 GID . 这一级网络存储所有的 $\langle \text{Hash}(C_1, C_2), \text{IS 地址} \rangle$ 的偶对. 节点组中每个非超级节点对应 CID-V 中的一个或多个组织机构代码, 每个组织机构代码最多可对应一个节点; (2) 每个节点组设置一个超级节点(一般由编码解析服务提供商设置), 所有的超级节点组织为一个上级 P2P 网络, 这一级网络负责确定编码与节点组的对应关系. 这一级网络中存储的信息为 $\langle \text{Hash}(C_1), GID \rangle$; (3) 各 P2P 网络均可独立采用任意的 P2P 算法.

设分级 P2P RFID 编码解析网络中所有节点的全集为 P , 每个节点具有唯一的标识 NodeID . 将全部节点划分而成的节点组的数目为 I , GID_i 为第 i 个节点组的组标识, G_i 为第 i 个节点组中所有节点的集合, S_i 为第 i 个节点组中的超级节点, 则全部超级节点的集合 $S = \{S_1, S_2, S_3, \dots, S_i\}$, 每个节点组中的普通节点集合为 $R_i = G_i - \{S_i\}$, r_{ij} 为第 i 个节点组中第 j 个普通节点, 全部的普通节点集合 $R = P - S = \bigcup_{i=1}^I R_i$. 对于普通节点, 如果该节点由某个企业或组织建立, 则该节点具有一个特征码 CharID , 特征码用于提高编码解析网络的查询速度(在第 3 节算法 3 中给出详细描述), 其取值为 $\text{Hash}(c_1)$ (c_1 为企业或组织在 CID-V 中的编码值. 后文中 C_1 表示编码中的字段, c_1 表示一个编码值).

在分级 P2P RFID 编码解析网络中, 超级节点存储、维护 C_1 字段与节点组的对应关系, 并在查询发生时, 将查询请求转发到相应节点组. 因此, 超级节点上需要存储如下信息:

- 上层超级节点网络路由表, 该路由表的内容由上层网络使用的 P2P 路由算法确定;

- $\text{Hash}(C_1)$ 与节点组标识 GID 之间的映射信息.

普通节点存储、维护 RFID 编码和 IS 地址之间的映射信息, 并负责发起查询和返回查询结果. 因此, 普通节点上需要存储如下信息:

- 该普通节点所在组的 GID , 以方便查询数据、插入数据、节点加入和节点退出等;

- 节点组内网络的路由表, 该路由表的内容由本组网络使用的 P2P 路由算法确定;

- $\text{Hash}(C_1, C_2)$ 与 IS 地址之间的映射信息.

2.3 分级存储服务

在两级的 P2P RFID 编码解析网络中, 数据存储服务流程如下:

(1) 某个普通节点 r_{ij} 接收到数据 $(c_1, c_2, \text{IS 地址})$ 的存储请求, r_{ij} 首先将数据存储在本地;

(2) r_{ij} 根据 G_i 所采用的 P2P 算法, 查找 NodeID “距离” $\text{Hash}(c_1, c_2)$ 最近的 k 个节点集合 $\{r_{ik1}, r_{ik2}, \dots, r_{ikk}\}$ (“距离”的定义和 k 值由 P2P 算法确定), 并在这些节

点上存储数据:

(3) r_j 向超级节点 S_i 发送存储 ($\text{Hash}(c_1)$, GID_i) 偶对的请求, S_i 在本地存储后, 根据上级网络采用的 P2P 算法, 在上级网络中其他节点保存偶对。

如果是多级的 P2P RFID 编码解析网络, 数据存储流程可以在上述流程基础上迭代扩展。

根据数据存储流程, 我们可以得出如下结论:

对任意 RFID 编码 (c_{11} , c_{21}) 和 (c_{12} , c_{22}), 如果 $c_{11} = c_{12}$, 则 ($\text{Hash}(c_{11}$, c_{21}), IS 地址 1) 和 ($\text{Hash}(c_{12}$, c_{22}), IS 地址 2) 存储于同一个节点组 G_i 中。

该性质为提高编码解析效率, 实现不同编码解析网络对等互联提供了支持:

(1) 节点组内产生的数据, 由该节点组负责查询。一般而言, 节点组的划分往往根据地域、行业等划分, 组内查询速度一般要高出组间查询速度很多, 从而可以有效提高 RFID 编码解析速度。

(2) 为不同编码解析网络互联提供了支持。支持将不同编码的编码解析网络视为节点, 实现各网络之间的对等互联。由此生成的编码解析网络, 可以确保同类编码仍然由原编码解析网络存储并解析, 对原有编码解析网络不需要进行修改。对不同类编码, 则通过上级网络转到对应的编码解析网络进行解析。

2.4 分级查询服务

在分级 P2P RFID 编码解析网络中, 其查询服务流程如下:

(1) 对某个 RFID 编码 (c_1 , c_2) 的查询, 首先由发起查询的普通节点 r_j 将查询转发给本组超级节点 S_i ;

(2) S_i 根据查询编码中 c_1 段在上级网络中查询得到对应的组标识 GID_k ;

(3) 如果 $\text{GID}_k = \text{GID}_i$, 则由 r_j 在 G_i 查询并返回查询结果; 否则 S_i 将查询转发给节点组 G_k 的超级节点 S_k , S_k 在 G_k 中查询并返回查询结果。

在查询流程中, 查询分为两步: 上级网络查询和本地网络查询。这种查询方式的优势在于:

(1) 单次查询得到响应经过节点间的“跳”数相对于不分级 P2P 网络显著减少, 从而显著降低延迟^[4];

(2) 对于每个节点组以及超级节点的上级网络而言, 在数据存储、查询上均采用 P2P 技术, 从而可以保证数据和查询在组内各节点上均匀分布, 从而有效实现负载均衡;

综上, 分级 P2P RFID 编码解析网络能够很好的应对 RFID 编码解析服务潜在并发查询量巨大的特点, 有效实现负载均衡。并且基于 P2P 技术的性质, 当分级 P2P RFID 编码解析网络中出现单点失效时, 编码解析服务仍然保持可用。

3 基于 kademlia 算法的实现

在分级 P2P RFID 编码解析网络中, 上级网络和各节点组采用的 P2P 算法可以基于现有的 Chord^[6]、CAN^[9]、Pastry^[10] 和 Kademlia^[8] 等算法实现, 并不特别限定所采用的算法。其中 Kademlia^[8] 更符合 RFID 编码解析服务的需求。本节将介绍基于 kademlia^[8] 算法实现的两级 P2P RFID 编码解析网络, 并在下节给出实验数据分析。

在该网络中, 相对于 Chord^[6]、CAN^[9]、Pastry^[10] 等其他 P2P 路由算法而言, 采用以 kademlia^[8] 算法为基础改进的路由算法具有更大的优势, 主要原因在于:

(1) 存储的信息具有多个副本, 不但有助于提高查询效率, 同时也增强了信息的可用性, 能够更有效的应对单点失效;

(2) Kademlia^[8] 算法中 K-buckets 易于附加更多的路由信息, 提高查询效率;

(3) 支持采用缓存技术, 进一步提高查询效率。

在该网络中, 每个节点具有一个 160 位的节点标识 NodeID, 该标识通过散列该节点的 MAC 地址和时间戳得到。由某个企业或组织建立的普通节点, 具有 160 位的特征码 CharID, 该特征码通过散列该企业或组织在 CID-V 中的编码段得到。在每个节点的 K-Buckets 中, 除了要保存其它节点的 NodeID 和 IP 地址等信息外, 对于具有特征码的节点, 还需要保存相应的特征码 CharID。对需要存入 RFID 编码解析网络的 (C_1 , C_2 , IS 地址) 的信息偶对, 将 C_1 , C_2 进行散列, 得到 160 位的 key。每个 key 对应的信息偶对, 存储到和该 key 距离最近的节点中。节点间距离 d 按照 kademlia 算法定义为节点 NodeID 的 XOR 值, 即, 假定两个节点的 NodeID 分别为 a 与 b , 则有: $d = a \text{ XOR } b$ 。

在基于 kademlia 实现的两级 P2P RFID 编码解析网络中, 主要涉及到如下算法: (1) 节点加入和节点退出; (2) 数据存储; (3) 编码查询; (4) 组内查询。

其中, 普通节点加入网络的算法和标准 kademlia 相似。对于超级节点加入网络的算法, 和普通节点仅有一点区别: 当网络中加入一个超级节点时, 表明在网络中新增加一个节点组。此时, 不仅需要根据 kademlia 算法来执行节点加入过程, 还需要在该超级节点上初始化节点组的组标识 GID。此外, 在节点退出处理和各节点 (包括超级节点和普通节点) 的 K-buckets 更新的处理上, 与标准 kademlia 网络处理方法相同。

在基于 kademlia 的两级 P2P RFID 编码解析网络中, 数据存储算法如算法 1 所示。

在该算法中, findnodes() 方法的功能是查找与第二个参数的散列值最近的 K 个节点。为了适应分级 P2P

的网络结构,并为查询提供支持,算法 1 中加入了在超级节点中保存 RFID 编码 c_1, c_2 中编码段 c_1 对应的信息偶对 $\langle \text{Hash}(c_1), \text{GID}_i \rangle$ 的过程,以帮助超级节点在查询时确定和待查询编码对应的节点组。

算法 1:

节点 r_i 存储信息偶对 $\langle c_1, c_2, \text{ISAddress} \rangle$ 的算法。(c_1, c_2 表示某个 RFID 标签的编码, ISAddress 表示该编码对应的 IS 地址)

输入: 节点 r_{ij} , 节点 r_{ij} 所在区域节点组 GID_i , 超级节点 S_i , 信息偶对 $\langle c_1, c_2, \text{ISAddress} \rangle$

输出: 保存了 $\langle \text{Hash}(c_1, c_2), \text{ISAddress} \rangle$ 和 $\langle \text{Hash}(c_1), \text{GID}_i \rangle$ 的编码解析网络。

```
dataStore( $r_{ij}, \text{GID}_i, c_1, c_2, \text{ISAddress}$ )
// 在接收存储要求的节点  $r_{ij}$  本地保存需存储的信息偶对
store( $r_{ij}, c_1, c_2, \text{ISAddress}$ );
// 从节点  $r_{ij}$  开始, 在  $r_{ij}$  所在节点组中查找距离
//  $\text{Hash}(c_1, c_2)$  最近的  $K$  个节点
contactlist = findnodes( $r_{ij}, c_1, c_2, K$ );
// 在 contactlist 中所包含的每个节点上保存信息
foreach Knode in contactlist do store(Knode,  $c_1, c_2, \text{ISAddress}$ );
// 如果  $S_i$  上没有保存  $c_1$  对应的节点组, 则在超级
// 节点  $S_i$  上保存  $\langle \text{Hash}(c_1), \text{GID}_i \rangle$ 
if getvalue( $S_i, \text{Hash}(c_1)$ ) = null
    then begin
        store( $S_i, c_1, \text{GID}_i$ );
// 在上层网络 contactlist 中所包含的每个节点上保存信息
contactlist = findnodes( $S_i, c_1, K$ );
foreach Knode in contactlist do
    store(Knode,  $c_1, \text{GID}_i$ );
    end
end.
```

在算法 1 成功执行后, 信息偶对 $\langle \text{Hash}(c_1, c_2), \text{ISAddress} \rangle$ 将存储于与 $\text{Hash}(c_1, c_2)$ 距离最近的 K 个节点以及接受信息偶对 $\langle c_1, c_2, \text{ISAddress} \rangle$ 存储请求的节点中。

在基于 kademlia 的两级 P2P RFID 编码解析网络中, 编码查询算法如算法 2 所示。

算法 2

节点 r_{ij} 查询编码 c_1, c_2 的 IS 地址的算法。

输入: 节点 r_{ij} , RFID 编码 c_1, c_2 , 超级节点 S_i

输出: IS 地址

```
query( $r_{ij}, c_1, c_2, S_i$ )
// 如果和编码  $c_1, c_2$  对应的节点组就是  $r_{ij}$  所在节点组,
// 则由  $r_{ij}$  在本组发起查询
if getvalue( $S_i, \text{Hash}(c_1)$ ) =  $\text{GID}_i$ 
    then return findvalue( $r_{ij}, c_1, c_2$ );
// 在上层网络查找和  $c_1$  对应节点组的 GID
 $\text{GID} = \text{findvalueinUpperNet}(S_i, c_1)$ ;
// 获取和  $c_1$  对应节点组的超级节点  $S_{\text{node}}$ 
 $S_{\text{node}} = \text{getNodeByGroupID}(\text{GID})$ ;
// 由  $S_{\text{node}}$  查找距离  $\text{Hash}(c_1, c_2)$  最近的  $K$  个节点
contactlist = findnodes( $S_{\text{node}}, \text{Hash}(c_1, c_2), K$ );
// 如果列表中有某个节点保存了待查信息,
```

// 则返回该信息和该节点的信息, 否则返回空

```
foreach Knode in contactlist do
    If (value = getvalue(Knode,  $\text{Hash}(c_1, c_2)$ ))  $\diamond$  null then return (Knode, value)
    else return (null)
end.
```

在算法 2 中, findvalue() 方法的功能是在 r_{ij} 所在节点组中查找与 $\text{Hash}(c_1, c_2)$ 对应的 IS 地址, 其实现方法见算法 3。在 contactlist 中, 按照和 $\text{Hash}(c_1, c_2)$ 距离的升序保存了与 $\text{Hash}(c_1, c_2)$ 距离最近的 K 个节点, 并将第一个存储 $\langle \text{Hash}(c_1, c_2), \text{ISAddress} \rangle$ 偶对的节点信息(同时该节点也是距离 $H(c_1, c_2)$ 最近的节点)和 IS 地址返回。这里返回节点信息, 是为了支持缓冲机制。当采用缓冲机制后, r_{ij} 再次查询包含代码段 c_1 的 RFID 编码时, 可以直接通过缓存访问该节点, 从而提高查询效率。

算法 3:

在节点组内查找与 $\text{Hash}(c_1, c_2)$ 对应的 IS 地址的算法。

输入: 节点 r_{ij} , 待查询的 c_1, c_2

输出: 保存信息的节点, IS 地址

```
findvalue( $r_{ij}, c_1, c_2$ )
// 如果在  $r_{ij}$  的  $K$ -buckets 中有节点特征码和  $\text{Hash}(c_1)$  相等,
// 则在该节点查询 IS 地址并返回该节点和 IS 地址。
foreach Kbucketnode in  $r_{ij}.Kbuckets$ 
    If Kbucketnode.CharID =  $\text{Hash}(c_1)$ 
    then
        {  $\text{ISAddress} = \text{getvalue}(r_{ij}, \text{Hash}(c_1, c_2))$ ;
          return(Kbucketnode,  $\text{ISAddress}$ );
        }
// 找出  $r_{ij}$   $K$ -buckets 中距离  $\text{Hash}(c_1, c_2)$  最近的  $K$  个节点,
contactlist = selectNearNode( $r_{ij}, \text{Hash}(c_1, c_2), K$ )
// 对 contactlist 中每个节点重复上述过程直到找到特征码和  $\text{Hash}(c_1)$ 
// 相等的节点或者 contactlist 不再变化为止。
foreach Node in contactlist
    { foreach Kbucketnode in Node.Kbuckets
      If Kbucketnode.CharID =  $\text{Hash}(c_1)$ 
      then
          {  $\text{ISAddress} = \text{getvalue}(r_{ij}, \text{Hash}(c_1, c_2))$ ;
            return(Kbucketnode,  $\text{ISAddress}$ );
          }
      contactlist = combine(contactlist,
          selectNearNode(Node,  $\text{Hash}(c_1, c_2), K$ ))
    }
// 如果列表中有某个节点保存了待查 IS 地址, 则
// 返回该地址和该节点的信息
foreach Node in contactlist do
    If (value = getvalue(Node,  $\text{Hash}(c_1, c_2)$ ))  $\diamond$  null then return (Node,  $\text{ISAddress}$ )
    else return (null)
end.
```

算法 3 和标准 kademlia 算法中 findvalue() 算法的区别在于: 算法 3 首先要是在 K -buckets 中查找有没有特征

码和待查询 CID-V 编码中组织机构代码部分 C_1 散列值相等的节点, 如果有, 根据算法 1 中的存储算法, 可以直接在该节点查询到对应的 IS 地址, 从而提高查询效率。

4 相关实验分析

分级 P2P RFID 编码解析网络是实现 RFID 编码解析服务的一种新的设计, 我们通过实验试图验证本文给出的这种编码解析网络能够较好地解决 ONS 所固有的负载均衡、单点失效的问题, 相对于基于 DNS 技术的 ONS, 其效率对用户而言也是可以接受的。

4.1 实验模拟环境

我们开发了一个在单台机器上模拟大量 P2P 网络节点的程序, 从而可以模拟分级 P2P RFID 编码解析网络的运行。在实验中使用 5 台机器, 一台模拟超级节点组成的上级 P2P 网络, 另外四台模拟四个下级 P2P 网络中大量的普通节点, 它们分别与第一台机器上的一个超级节点组成共四个下级 P2P 网络。网络协议是上述基于 kademia 改进的协议。RFID 编码采用 CID-V。用于对比的是基于 DNS 技术的 ONS 相关数据, 取得这些测试数据的硬件环境和数据量与分级 P2P RFID 编码解析网络相同。

每台机器的配置如表 2 所述。

表 2 实验环境的机器配置

配置项	配置
CPU	entium 4 3Ghz
内存	1G
Java 虚拟机	JVM 1.5.0
操作系统	Windows XP

我们事先生成 50 万条编码解析映射信息, 格式统一为 $\langle c_1, c_2, ISAddress \rangle$ 。所有普通节点以随机的顺序将被赋予的信息存储并发布到网络。

在模拟实验中我们主要考察以下指标:

(1) 查询命中率: 用于验证该编码解析网络的正确性。我们设定查询的超时时间为 0.2 秒。查询命中率=被发起后在超时时间内返回且结果正确的查询数目/所有查询。

(2) 查询平均响应时间: 用于评价该编码解析网络的服务性能。这也是编码解析网络的主要性能指标。

(3) 单点失效时的查询命中率: 用于评价该编码解析网络解决单点失效问题的效果。

4.2 实验一

每个普通节点以 0.2 秒的频率进行查询。实验分为两组, 第一组中普通节点选择查询的编码从所在的区域节点组被赋予的编码中随机挑选; 第二组则正好相反。每组实验进行 10 次, 使得普通节点在发出 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000 次查询后结束实验。

图 2 显示了实验的结果。在实验环境下, 组内查询的效率远比组间查询的效率要高得多。每组实验都在节点查询数较多的时候平均响应时间略有增长, 这主要是因为消息的处理采用监听队列 Buffer 的实现方法, 在消息较多的时候有一定的阻塞。

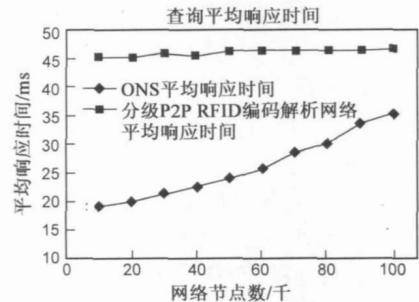


图 2 查询平均响应时间

图中未显示的数据是查询命中率平均高达 99.73%。将未能一次成功的查询收集起来再次提交时, 发现全部查询成功, 说明第一次未成功的主要原因可能在于超时或者消息丢失。这也间接证明编码解析网络能提供正确的编码解析查询服务。

4.3 实验二

由于基于 DNS 的 ONS 网络固有地存在单点失效问题, 因此我们对比两级 P2P RFID 编码解析网络中, 存在节点失效的下级 P2P 网络和节点全部在线的下级网络, 来验证本文的设计能够解决单点失效问题。

我们让两级 P2P RFID 编码解析网络的下级 P2P 节点组 1 中的节点保持全部在线, 组 2 中的节点部分失效。使另两个组的普通节点以 0.2 秒的频率分别针对组 1 和组 2 中的编码发起 1000 次查询。将组 2 中的节点失效效率从 5% 逐渐提高到 50%, 进行多次实验。

图 3 显示了实验的结果。在节点失效效率低于 10% 时, 查询命中率受到的影响较小, 最高降低 1% 的命中率。随着节点失效效率的提高, 命中率逐渐下降, 但下降的幅度并不大, 从数据上看, 当失效节点达到 50% 时, 命中率仍高达 87%。实验说明小规模单点失效时, 网络仍能提供正确的编码解析服务, 当出现大规模的节点失效时, 网络仍然能够提供大部分正确的编码解析服务。

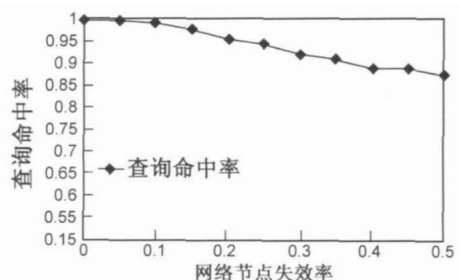


图 3 单点失效时的查询命中率

在图 3 中, 没有表现出来的数据是在节点失效时, 平均查询响应时间并没有大幅度地下降, 当 50% 节点失效时, 响应时间仅仅下降了不到 10%。

通过实验分析, 本文提出的分级 P2P RFID 编码解析网络能够正确地提供 RFID 编码解析服务, 即使对于少量的单点失效仍然保持较好的查询性能。

5 结论

本文提出了一种基于 P2P 技术的 RFID 编码解析网络结构。该结构将 RFID 编码解析网络中的节点划分为多个节点组, 每个节点组构成一个组内的 P2P 网络; 每个节点组中的超级节点组织为上级 P2P 网络。并以 kademlia^[8]算法为基础, 改进了 kademlia^[8]算法的路由表 K-buckets 以及存储、查询等关键流程, 给出了在分级 P2P RFID 编码解析网络中实现数据存储、数据查询的算法。实验数据表明, 分级 P2P 的 RFID 编码解析网络相对于基于 DNS 技术的 EPCglobal ONS, 具有负载均衡、可扩展性好、能有效应对单点失效等优点。

在分级 P2P RFID 编码解析网络中, 仍然存在一些重要的问题尚未解决。首先是安全性的问题, 包括在分级 P2P RFID 编码解析网络中如何防止中间人攻击、DoS 攻击、重定向攻击等安全威胁, 以及如何应对 RFID 应用所带来的私密性保护的问题。同时, 相对于基于 DNS 技术的 EPCglobal ONS, 当网络规模较小的时候, 分级 P2P RFID 编码解析网络的查询效率要低一些, 如何对查询进行优化以进一步提高查询效率, 也是需要研究的一个重要课题。

参考文献:

- [1] EPCglobal. The EPCglobal Architecture Framework[S]. The EPCglobal Standards Development Process, September 2007.
- [2] Ken Sakamura. Ubiquitous ID Technologies 2008[Z/OL]. Ubiquitous id center. http://www.uidcenter.org/pdf/UID910W001-080226_en.pdf, 2008.
- [3] EPCglobal. EPCglobal Object Name Service(ONS) 1.0.1[S]. The EPCglobal Standards Development Process, September, 2007.
- [4] L Garces Erice, E W Biersack, P A Felber, K W Ross, G Urvoy-Keller. Hierarchical peer-to-peer systems[A]. Proceedings of ACM/IFIP International Conference on Parallel and Distributed Computing (EuroPar) [C]. Klagenfurt, Austria: Springer Berlin, 2003. 1230-1239.

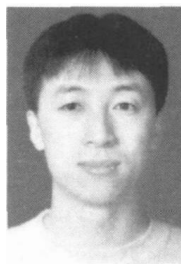
- [5] Russ Cox, Athicha Muthitacharoen, Robert T Morris. Serving DNS using a peer to peer lookup service[A]. Revised Papers from the First International Workshop on Peer to Peer Systems [C]. Cambridge, MA: Springer Berlin, 2002. 155-165.
- [6] Ion Stoica, Robert Morris, David Liber Nowell, David R Karger, M Frans Kaashoek, Frank Dabek, Hari Balakrishnan. Chord: A scalable peer to peer lookup protocol for internet applications [A]. Proceedings of the 2001 SIGCOMM conference [C]. San Diego, CA: IEEE Computer Press, October 2001. 149-160.
- [7] Benjamin Fabian, Oliver Gunther. Distributed ONS and its impact on privacy[A]. Communications, 2007. ICC '07. IEEE International Conference [C]. Glasgow, Scotland, UK: IEEE Computer Press, June 2007. 1223-1228.
- [8] Petar Maymounkov, David Mazieres. Kademlia: A peer to peer information system based on the XOR metric[A]. Revised Papers from the First International Workshop on Peer to Peer Systems [C]. Cambridge, MA: Springer Berlin, 2002. 53-65.
- [9] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, Scott Shenker. A scalable content addressable network [A]. Proceedings of the 2001 SIGCOMM conference [C]. San Diego, CA: IEEE Computer Press, October 2001. 161-172.
- [10] Antony Rowstron, Peter Druschel. Pastry: Scalable, decentralized object location and routing for large scale peer to peer systems [A]. Middleware 2001: IFIP/ACM International Conference on Distributed Systems Platforms [C]. Heidelberg, Germany: Springer Berlin, 2001. 329-348.

作者简介:



刘学洋 男, 1978 年出生, 湖北钟祥人, 博士研究生, 主要研究领域为软件工程和信息安全。

E-mail: liuxy@cs.pku.edu.cn



赵文 男, 1967 年出生, 辽宁大连人, 博士, 副研究员, 主要研究领域为软件工程和 workflow 技术。