

蚁群算法在网格资源发现中的应用

徐韵洁, 冉晓旻

(信息工程大学 信息工程学院, 郑州 450002)

摘要: 利用 RDF(resource description framework, 资源描述框架) 屏蔽网格资源的异构性和地理分布性, 并采用布隆过滤器将资源描述转换为位向量以实现资源发现的高速匹配; 将转换后的资源索引根据分层组织模型在顶层节点进行整合。最后利用蚁群算法进行搜索, 高效可靠地找到满足用户服务质量(quality of service, QoS) 需求的资源。实验结果表明该方法可行且搜索效率较好, 查询的资源数和次数越多, ACA 越具有优势。

关键词: 网格资源发现; 资源描述框架; 布隆过滤器; 分层组织模型; 蚁群算法; 双重资源发现机制

中图分类号: TP393; TP301.6 **文献标志码:** A **文章编号:** 1001-3695(2013)05-1492-03

doi:10.3969/j.issn.1001-3695.2013.05.054

Ant colony algorithm in application of grid resource discovery

XU Yun-jie, RAN Xiao-min

(Institute of Information System Engineering, Information Engineering University, Zhengzhou 450002, China)

Abstract: This paper introduced ACA into grid resource discovery. Firstly, it shield the heterogeneity of grid resources using RDF, changed the description of resources to bit maps for high-speed matching simultaneously using Bloom filter. Secondly, it integrated the transformed resource index on the top-level nodes based on the model of the hierarchical organization. Finally, it met user's QoS requirements efficiently and reliably using ACA search resources. Experimental results demonstrate the effectiveness of the proposed approach, simultaneously with the increase in the number of resources and inquiries, ACA has more advantages.

Key words: grid resource discovery; RDF; Bloom filter; hierarchical organization model; ant colony algorithm(ACA); mixed resource discovery mechanism

网格在缺少中央控制、全局信息以及严格信任关系的情况下能够协同使用地理上分布的各种资源^[1]。资源发现是资源拥有者和资源请求者之间的纽带, 是很多应用的重要基石。资源发现功能的强弱直接决定着网格的使用效率和友好程度。目前, 众多学者和研究机构提出了许多网格资源发现机制, 如集中式、分布式、非结构化、结构化等, 各有其优缺点。蚁群算法是一种用来在图中寻找优化路径的几率型算法, 它由 Dorigo 等人^[2]提出, 其灵感来源于蚂蚁寻找食物过程中发现路径的行为。它具有较强的鲁棒性、优良的分布式计算机制、易与其他方法结合等优点。目前对其研究已渗透到多个应用领域, 成为交叉学科中一个非常活跃的前沿性研究问题。当前研究多利用蚁群算法对任务和资源进行分配调度, 较少利用蚁群算法来进行资源发现的研究。本文基于分层的网格资源组织模型, 在文献[3]的基础上进行进一步的探索: a) 介绍一种描述资源和查询请求的方法, 以提高查询匹配的效率和降低查询在节点间传播的开销; b) 将蚁群算法运用于资源发现, 使之能够在资源分层组织模型上快速准确地发现所需资源。

1 资源描述

为了消除网格资源的异构性, 本文采用基于 XML 的 RDF 描述方法, 结合元数据来描述资源和查询。同时采用 Bloom filter 技术对 RDF 信息摘要进行压缩编码, 以提高查询匹配的

效率和降低查询在节点间传播的开销。

1.1 RDF 编码

RDF 是由 W3C 制定的专门用于描述 Web 资源的框架, 它定义了一种用于描述资源及其相互关系的简单模型, 同时也提供了一种机制来显式地表示服务、处理和商业模型。RDF 语言是语义 Web 实现的关键技术之一, 也是语义信息描述的有效手段。

从逻辑上看, RDF 数据都可以被视为一个由若干 RDF 三元组(RDF triple)所组成的集合。每一个三元组由主语(subject)、谓语(predicate)和宾语(object)组成, 形如(subject, predicate, object)。其中, 主语是被描述的对象, 代表对象哪一个方面被描述的对象称为谓词, 代表某个属性的值的对象称为宾语。为了让计算机能阅读 RDF, 使用互联网中的 URI(uniform resource identifiers)标志不同的主语、谓语和宾语的系统, 使计算机不至于混淆名称相似的不同对象; 使用 XML 作为表达这种三元关系的标准语言, 便于在不同计算机上传输。将 RDF 通过 XML 来表达的语法称为 RDF/XML, 这并不是 RDF 三元组的唯一表示方法, 然而, 无论采用什么格式, RDF/XML 所表示的语义都是唯一的。

1.2 Bloom filter

布隆过滤器(Bloom filter)是由 Burton Bloom 在 1970 年提出来的, 它是一个简单的、空间效率极高的位向量结构。通常

收稿日期: 2012-09-15; 修回日期: 2012-10-26

作者简介: 徐韵洁(1988-), 女, 安徽庐江人, 硕士研究生, 主要研究方向为计算机网络、语义网、资源发现(xuyunjie621@126.com); 冉晓旻, 男, 副教授, 硕士, 主要研究方向为计算机网络、网络体系结构。

应用在一些需要快速判断某个元素是否属于集合,但是并不严格要求 100% 正确的场合。Bloom filter 有以下几个特点:a)不存在漏判(false negative),即某个元素在某个集合中,肯定能判断出来;b)可能存在误判(false positive),即某个元素不在某个集合中,可能也被误报出来;c)确定某个元素是否在某个集合中的代价和总的元素数目无关。

Marzolla 等人^[4]提出在叠加网络(overlay network)中采用位图索引(bitmap index)。本文采用 Bloom filter 对描述资源的 RDF/XML 进行摘要编码,以将其转换为一种代价较小的数据结构,减少索引信息的存储和传输开销。另外,由于布隆过滤器具备良好的信息压缩编码特性,并且编码后形成的向量结构极适合于在网络上传输。对 RDF 数据进行编码的过程也就是对其中的 RDF 三元组进行编码的过程。使用 Bloom filter 对三元组进行编码的示意图如图 1 所示。

RDF 查询语句也需要用布隆过滤算法进行编码,编码后的查询语句是一串与 RDF 三元组过滤器结构相同的位向量,这样就能将查询匹配过程转换为位向量间的位运算过程。尽管索引仅仅只是资源的近似,并且 Bloom filter 存在一定的误判,但最终用户可以检查准确的 RDF 来确认匹配。

2 资源发现机制

2.1 网格资源的组织

根据资源在地理上的分布情况,将网格资源组织为一个分层模型,如图 2 所示。最底层是资源层,其中有各种资源、服务及用户;上面一层是汇聚层,低层的资源、服务及用户在物理上或在管理策略上汇聚在一起;最上面一层是由 VO 构成的分布式覆盖网络层。VO 之间是平坦的关系,VO 内部在逻辑上以树型结构来组织节点。

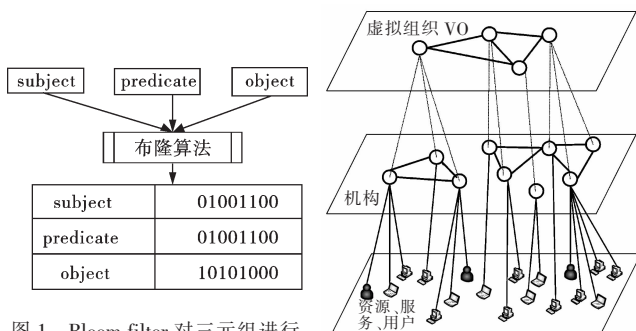


图 1 Bloom filter 对三元组进行编码的示意图

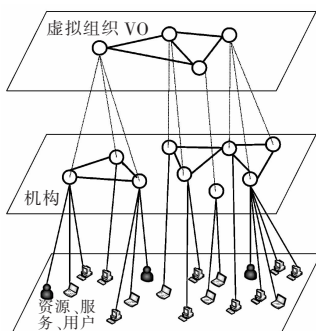


图 2 网格资源组织模型

集中性、层次化的资源发现机制随着网格规模的扩大会成为性能的瓶颈。因此本文对资源请求处理分为两个部分:本地部分和远程部分。本地部分即是查询请求在本地 VO 资源索引目录中进行查找;远程部分即是若本地没有匹配资源,则采用蚁群算法对整个网格资源进行查找。这种双重资源发现机制的主要步骤如下:

- 用户提出资源查询请求,后台将查询转换成 RDF 查询语句并进行编码得到一串位向量,提交给所在 VO;
- VO 在本资源域内进行集中式查找,若找到满足要求的资源,则转 d);
- 将查询请求位向量继续转发给其他 VO 进行匹配,转 b);
- 返回匹配的资源或将地址返回给用户;

e) 结束。

如果蚂蚁每到一个 VO 就要对该 VO 中存储的本地资源进行集中式查找,这无疑浪费了不少的时间。因此以 VO 为单位,对其中的所有资源索引在 VO 节点整合,如图 3 所示。

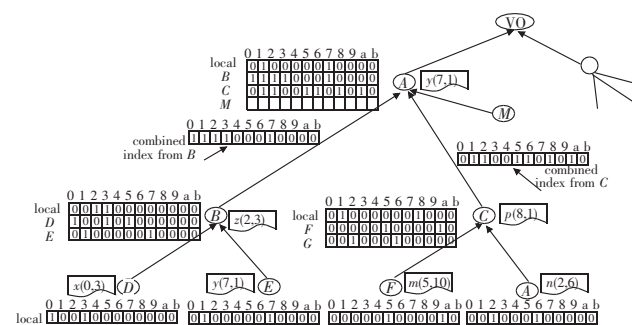


图 3 VO 内资源索引整合示意图

图 3 中用 12 位的布隆位图和 2 个散列函数来映射资源,现实中位图的规模和散列函数的数量要大得多。图 3 中每个节点维护一个包含一个或几个资源的位图,一个位图是该节点的资源,其他的是下层节点的。每个节点又将聚合后的位图发给上层节点。例如,节点 B 的资源表包含一个本地位图和两个下层节点 D 和 E 的位图。B 节点中资源 z 被映射到两个位置:位图 2 和 3 ($H_1(z) = 2, H_2(z) = 3$)。所以在 B 的本地位图中, $B_2 = 1, B_3 = 1$ 。B 采用位或运算将这三个位图合并,合并后的位图代表 B 和它后代所有的资源,然后将合并后的位图传给它的上层节点 A。A 又将本地资源位图和其下层所有节点的资源合并,将合并后的位图传给 VO。VO 将收到的下层位图合并最终形成一个位图,即是本地所有资源位图的概括视图。蚂蚁每到一个 VO 时首先匹配这个位图,若匹配则该 VO 内可能存在所需资源,再进行集中式查找;若不匹配则该 VO 内肯定没有所需资源,蚂蚁直接转移到下一个 VO。

2.2 蚁群资源搜索算法

由于蚁群算法起初是用于解决路径的优化问题,所以本文对算法进行了修改以适合于网格资源的搜索应用。覆盖层 VO 节点模型可定义为一个加权无向图 $G = (V, E)$, 其中 $V = \{v_i | i = 1, 2, 3, \dots, N\}$ 表示 VO 节点的集合, $E = \{e_{ij} | i, j = 1, 2, 3, \dots, N, \text{且 } i \neq j\}$ 表示图中边的集合,对每条边 e 赋予权值 w_e 表示网格传输的延时、延时抖动、带宽、包丢失率等。在基于蚁群算法的资源搜索机制中,查询信息包是蚂蚁,存在符合搜索要求的 VO 节点是食物。当某节点 V_{query} 发出搜索请求时,就相当于派出蚂蚁在网格中按照一定规则沿着路径 e 寻找在 V_{provide} 节点的食物,其中的规则约束就是利用权值 w_e 来计算到达下一节点的概率,指导蚂蚁前进的方向。

2.2.1 关键参数

1) 邻接点表 AT 在每个 VO 节点中用 AT 来存放与其他 VO 节点邻接关系的 $n \times n$ 矩阵,节点 i 和节点 j 连通则 AT_{ij} 置 1,不连通则置 0。

2) 信息素 τ 信息素就是蚂蚁运动时在路径上释放的一种特殊的分泌物,蚂蚁通过它来寻找路径。每个 VO 都存储着本域内和邻域资源的信息素,即维护一张信息素表。当节点发出资源搜索请求时,就相当于派出一批蚂蚁根据信息素表在 VO 覆盖网络层中并行地寻找食物。初始时刻,各个 VO 中的资源信息素都相等,设 $\tau = C$ (C 为常数)。

每只蚂蚁都维护一个路径表,用来记录访问节点的序号和

先后顺序。当所有蚂蚁构建完一条路径后,算法将根据信息素更新规则进行信息素更新。当第 i 只蚂蚁停止前进时,对路径表中所有 VO 节点的信息素按照式(1)进行局部更新。局部更新规则使相应的信息素减少,可以有效地避免蚂蚁收敛到同一条路径上。

$$\tau_{ij}(t+1) = \varphi \cdot \tau_{ij}(t) + (1 - \varphi) \cdot \tau_{ij}(t) \quad (1)$$

其中: $\tau_{ij}(t) = (nh)^{-1}$; h 为蚂蚁前进的跳数; n 为 VO 的数目; φ 为信息素挥发系数, $0 < \varphi \leq 1$ 。

每批蚂蚁完成搜索后,各路径上的信息素浓度需要根据式(2)作全局更新,其目的是为了充分利用历史搜索经验。较近的资源节点之间信息素增量较大,因此下次被选择的概率较大。

$$\tau_{ij}(t+1) = \rho \cdot \Delta\tau_{ij} + (1 - \rho) \cdot \tau_{ij}(t) \quad (2)$$

其中: $\Delta\tau_{ij} = (h_{\text{found}})^{-1}$, h_{found} 为找到资源节点的蚂蚁前进的跳数; ρ 为信息素挥发系数。

3) 状态转移概率 p_{ij}^k 蚂蚁 k ($k = 1, 2, \dots, m$) 在运动过程中的 t 时刻,从节点 i 转移到节点 j 的状态转移概率 $p_{ij}^k(t)$ 可按式计算:

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{s \in \text{tabu}_k} [\tau_{is}(t)]^\alpha [\eta_{is}(t)]^\beta} & \text{若 } j \neq \text{tabu}_k \\ 0 & \text{否则} \end{cases} \quad (3)$$

4) 生存时间 TTL 设置 TTL (time to live) 控制搜索深度,蚂蚁前进一步 TTL 就减 1,若所在 VO 存在符合请求的资源则停止前进,沿原路返回一个 QueryHit,否则直到 TTL 为 0 时停止前进。为了保证 QoS,算法设置迭代次数 K ,找出每次迭代中路径最短(即跳数最少)的资源节点返回给查询用户。

2.2.2 基于 ACA 的资源搜索流程

a) 初始化各参数。

b) 将 m 只蚂蚁放在发出请求查询的节点上,对本地 VO 进行搜索,若有匹配,记下资源地址或者备份资源,跳转到步骤 e); 否则继续 c)。

c) 每只蚂蚁 k 按式(3)转移位置,蚂蚁每走一步都查询 VO 中的资源位图。若匹配成功,则该 VO 进行本地搜索,将资源地址返回给本地 VO,并按式(2)全局更新信息素;否则蚂蚁继续按概率转移进行搜索,若达到 TTL,则按式(1)局部更新各 VO 的信息素。

d) 判断是否达到迭代次数 K 。若是,则跳转到 b); 否则继续 e)。

e) 根据 QoS 输出搜索资源结果。结束。

2.3 仿真与评估

实验采用 MATLAB 对资源发现的蚁群算法进行仿真。实验 VO 节点数为 1 000,蚁群算法的参数设置为: $\varphi = \rho = 0.2$, $\alpha = 1, \beta = 5, \eta$ 为 1000×1000 的全 1 矩阵。仿真实验中省略资源匹配环节,采取随机标定符合查询请求的资源如表 1 所示,模拟资源节点的动态加入或退出。

多次实验结果表明,第一代蚂蚁数量 $m \geq 80$ 时查全率(recall) 稳定在 100%,如图 4 虚线所示;搜索的错误率一直为 0,如图 5 所示,说明没有出现误判。随着查询次数 times 的增加,在 recall 值保持一定的条件下 m 逐渐减少,图 4 实线所示的是第 3 次循环迭代时 m 和 recall 的关系,说明了在利用历史搜索经验后,在查询信息量较少的情况下可以达到较高的查全率,减少了网络通信量。图 6 为蚁群算法($m = 20$)与随机转发方法搜索资源的比较。可以看出,随着搜索范围的扩大,蚁群算法所消耗的

时间上升得比较平缓,对比随机转发有较好的效果。

表 1 基于 ACA 搜索的资源信息

请求资源节点 V_{query}	标定提供资源节点编号 V_{provide}	QoS 标准
1	5, 50, 116, 221, 257, 301, 399, 400, 460, 520, 521, 605, 666, 777, 888, 900, 945, 960, 985, 999	路径长度

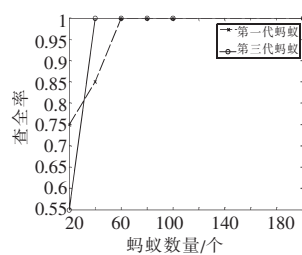


图 4 蚂蚁数目与查全率

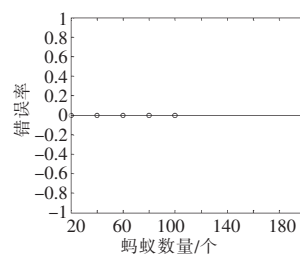


图 5 蚂蚁数目与错误率

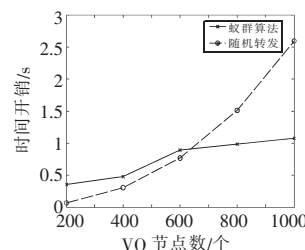


图 6 蚁群算法与随机转发方法时间开销比较

3 结束语

本文将蚁群算法引入到网格资源发现中,采用了最短路径作为度量标准。另外此算法还可以采用多种度量来选择路由,如延时、可靠性、带宽、成本等,通过加权运算构成一个复合度量,再填入信息素表中作为寻径的标准。该发现机制具有良好的扩展性和灵活性,在资源查询请求较多的情况下,具有较大的效率优势。

参考文献:

- [1] 李红,刘鲁. 分布式网格资源发现研究综述[J]. 计算机应用与软件, 2010, 27(2): 16-18.
- [2] DORIGO M, STÜTZLE T. Ant colony optimization[M]. 张军, 胡晓敏, 译. 北京: 清华大学出版社, 2007: 1-3.
- [3] 邓小清, 彭小利, 梁弼. 基于蚁群算法的网格资源发现模型研究[J]. 网络安全技术与应用, 2010(9): 30-31.
- [4] MARZOLLA M, MORDACCHINI M, ORLANDO S. Resource discovery in dynamic grid environment[C]//Proc of the 16th International Workshop on Database and Expert Systems Applications. [S. l.]: IEEE Press, 2005: 356-360.
- [5] 陈巧, 熊秋娥. 基于蚁群算法的 P2P 网络资源发现服务研究[J]. 电脑知识与技术, 2011, 7(28): 6934-6936.
- [6] 王峰, 周佳骏, 周小发. 基于蚁群算法的对等网络自适应寻径协议[J]. 计算机工程与应用, 2010, 46(17): 101-104.
- [7] JEANVOINE E, MORIN C. RW-OGS: an optimized random walk protocol for resource discovery in large scale dynamic grids[C]//Proc of the 9th Grid Computing Conference. [S. l.]: IEEE Press, 2008: 168-175.
- [8] 刘经宇, 方彦军. 蚁群算法在城市交通路径选择中的应用[J]. 西南交通大学学报, 2009, 44(6): 912-917.
- [9] 刘宴兵, 尚明生, 肖云鹏. 网格高性能调度与资源管理技术[M]. 北京: 科学出版社, 2010: 239-241.
- [10] 程相, 孙继银, 李琳琳. 基于 RDF 的军事网格资源描述方法[J]. 计算机技术与发展, 2011, 21(10): 129-132.