# 基于资源属性空间的网格资源查找算法

张树东[1] 廖乐健[2] 刘晶晶[3]
(中国科学院软件研究所多媒体通信和网络工程研究中心 北京 100080)[1]
(北京理工大学计算机学院 北京 100081)[2] (中央民族大学经济学院 北京 100081)[3]

**摘 要** 提出了一种基于资源属性空间的网格资源查找算法,资源之间通过属性的相似度来确立彼此之间的邻接关系,每个邻居代表属性的一种变化趋势,这样在资源查找时可以沿着与目标资源属性越来越近的方向查找,因此具有较高的查找效率。在理论上证明了算法的完备性;通过试验验证了算法的查找效率。

**关键词** 计算网格,资源查找,空间模型

## A Computing Grid Resource Locating Algorithm Based on n-dimensional Space Model

ZHANG Shu-Dong[1] LIAO Le-Jian[2] LIU Jing-Jing[3]
(Engineer of Multimedia Communication and Network Engineering Center, ISCAS, Beijing 100080)[1]
(School of Computer, Beijing Institute of Technology, Beijing 100081)[2]
(School of Economic, Central University for Nationalities, Beijing 100081)[3]

**Abstract** A novel Resource Locating Algorithm is presented for computing grid environment. The relation among resources is established according the similization of their attributes. Each neighbor represents a type of varying direction of resource attributes. So the resource discovery can be carried through toward the direction of target resource attributes. The perfectibility of the algorithm is proven in theory. The experiment results show that each resource only need store little neighbors' information, the algorithm has higher efficiency.

**Keywords** Computing grid, Resource locating, Space model

## 1 引言

在传统的单计算机系统和集群系统中,计算资源的分布比较集中,计算作业在使用资源之前可以快速、可靠地进行资源定位,资源的查找操作对计算性能的影响很小。作为一种计算基础设施,网格是广域的、大规模的分布式环境,它具有如下特点:

1)网格资源地理分布极广,资源之间、资源和使用者之间往往通过广域网连接;

2)资源以及网格的状态是动态变化的;

3)作为一种公共基础设施,网格资源的数量巨大;

4)网格用户数量巨大。

由于网格是分布在广域网上的,受网络带宽、网络延迟和网络不可靠性的影响,网格资源的定位将在很大程度上影响网格计算的性能。网格是一个动态的环境,它利用遍布全球的闲置资源,网格资源可以随时加入或退出网格系统,同时,受本地作业的影响,网格资源的可用度也是不断变化的,所以需要一种有效的机制来及时发现网格系统中满足作业要求的可用网格资源。

在网格环境下,由于资源数量巨大且不断变化,建立全局资源视图非常困难的。同时,由于网格用户数量巨大,服务请求数量众多,集中的资源索引方式很可能成为系统的性能瓶颈和网络的拥塞点,因此需要研究出一种分布的、基于局部视图的、快速的资源定位和查找方法。

资源发现是把资源和资源请求者之间联系起来的重要环节,有了资源发现机制,请求者才能使用自己所需要的网格资源,否则,大量的资源放在网格上,请求者不知道自己能够使用哪些资源,因此,资源发现是资源和资源请求者之间连接的纽带。

资源发现是把资源和资源请求者之间联系起来的重要环节,有了资源发现机制,请求者才能使用自己所需要的网格资源,否则,大量的资源放在网格上,请求者不知道自己能够使用哪些资源,因此,资源发现是资源和资源请求者之间连接的纽带。 资源发现是根据作业请求的资源描述,从网格上为请求者找到满足要求的资源的过程。

## 2 现有资源查找方法

现有网格系统中,资源查找主要有三种方式:穷举查找方法、集中查找方法和基于路由转发机制的资源查找算法。下面给出这三种算法的简要描述。

集中查找算法

对于集中查找方法来说,网格系统拥有一个全局的网格信息中心,该中心存储着所有资源节点的位置信息和资源与资源节点的位置信息。在资源查找时,直接访问网格信息中心,查找所需的资源。

穷举查找算法

对于穷举查找方法来说,系统中每个节点存储着所有资源节点的位置信息,不包括资源与资源节点之间的对应关系。在资源查找时,逐个访问各个资源节点,直至找到所需的资源。

基于路由转发机制的资源查找算法

对于基于路由转发机制的资源查找方法来说,系统中的每个节点都存放着所有资源的路由信息。在资源查找时,系统按资源路由表中的信息,访问相邻的网格节点,直至找到所需的网格资源。

张树东 博士,高级工程师;廖乐健 博士,教授,博士生导师。

集中查找方发简单、便捷，系统容易构建。但单一的网格信息中心容易造成单点故障，另外，随着系统规模的扩大，中心节点很可能造成系统的性能瓶颈和网络拥赛。穷举查找算法和基于路由转发机制的资源查找算法虽然能够解决单点问题，但系统需要维护全局网格视图，当网格系统规模非常大时，全局网格视图的维护是非常困难的。另外穷举查找算法的效率比较低。

## 3 网格资源的空间模型

在计算网格系统中，资源描述语言 RDF 用 CPU 速度、内存大小、操作系统、磁盘空间容量等参数对资源进行描述，这些参数彼此之间是正交的，参数的个数是给定的、有限的。假设用来描述资源的参数有 $d$ 个，这样，以这 $d$ 个参数为坐标轴，可以构造一个 $d$ 维线性空间，用 $R^d$ 表示。每个资源根据自己的坐标值对应 $d$ 维坐标空间中的一个点，用 $r(c_1,c_2,\cdots,c_d)$ 表示，其中 $c_i$ 是一个资源属性。如果我们在这 $d$ 维线性空间上定义内积

$\vartheta(r_i,r_j)=\sqrt{(c_{i1}-c_{j1})^2+(c_{i2}-c_{j2})^2+\cdots+(c_{id}-c_{jd})^2}$，则线性空间 $R^d$ 成为一个 $d$ 维欧氏空间，我们称这个 $d$ 维欧氏空间为网格资源空间。网格资源在网格资源空间对应的点称为这个网格资源在网格资源空间中的网格资源点。内积 $\vartheta(r_i,r_j)$ 是资源 $r_i$ 和 $r_j$ 之间在 $R^d$ 中的直线距离，资源之间的距离越小，表明资源之间的相似程度越高，所以内积 $\vartheta(r_i,r_j)$ 也称为资源 $r_i$ 和资源 $r_j$ 之间的资源相似度。

在网格系统中，系统由一个个资源节点和资源节点之间的网络连接组成，每个资源节点拥有 1 到多个资源。资源的查找过程就是从相邻的一个资源节点出发，查找到满足要求的资源的过程。在网格空间模型中，空间中的点是网格资源，不是网格资源节点。由于每个资源均属于一个资源节点，因此，可以从这个相邻的资源节点中选择一个与要查找资源特性最相似的资源出发来查找目的资源。这样，资源的查找过程可以抽象为从网格资源空间中的一个资源点出发，查找到目的资源的过程。

## 4 基于资源空间模型资源查找算法

### 4.1 资源组织结构

与其它资源发现算法不同的是，在基于资源空间模型的资源发现算法中，每个资源不保存资源的全局视图信息，它只存储一些在资源空间中与其临近的资源信息。

假设用来描述资源的参数有 $d$ 个，形成一个 $d$ 维资源空间，设 $r_0$ 是资源空间中的一个资源点，那么以 $r_0$ 为中心的 $d$ 个坐标轴将资源空间分割成 $2^d$ 个子区域，如图 1 所示。

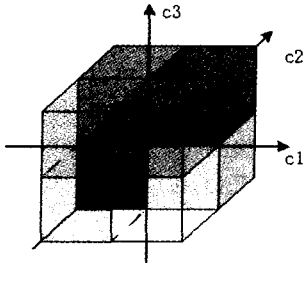


图 1 坐标中对资源空间的分割图

在每个子区域中选择与 $r_0$ 在每个坐标方向最接近的 $d$ 个点。如图 4-2 所示，$r_{01}$ 在 $c1$ 方向上与 $r_0$ 最相似，$r_{02}$ 在 $c2$ 方向上与 $r_0$ 最接近，$r_{03}$ 在 $c3$ 方向上与 $r_0$ 最接近。$r_{01}$、$r_{02}$、$r_{03}$ 可以是同一个点，也可以为空，如果 $r_{01}$、$r_{02}$、$r_{03}$ 为空，说明 $r_0$ 的这个子区域中没有网格资源。

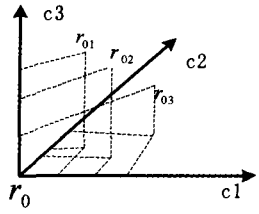


图 2 资源邻接点示意图

所以，$r_0$ 中最多需要存储邻接资源点的个数是：

邻接资源点的个数 $=2^d \cdot d$

这样，网格中的资源在资源空间中，通过彼此之间的邻接关系形成一个联络网，彼此之间互相连通（资源之间的连通性，在下一节中给出证明）。资源发现的过程，就是从联络网中的一个资源点出发，通过联络网找到需要的资源点的过程。

### 4.2 资源的连通性证明

设 $x$ 是 $d$ 维资源空间的一个坐标轴，那么资源空间中的资源可以按 $x$ 方向优先的顺序排列成一个线性链表。由线性链表性质可知，线性链表中的任何一个节点，沿着链表均可以找到链表中的任何另外一个节点，所以按这样的组织方式，资源空间中的资源是相互连通的。

下面我们证明上节描述的资源组织方式涵盖了以 $x$ 为索引的线性链表。

设 $r_0$ 是资源空间中的任意一点，$r^+$ 是 $r_0$ 在以 $x$ 为索引的线性链表中的上一点（$x$ 增长的方向），$r^-$ 是 $r_0$ 在以 $x$ 为索引的线性链表中的下一点（$x$ 减少的方向）。现在我们只要证明 $r^+$ 和 $r^-$ 在 $r_0$ 存储的 $2^d$。$d$ 邻接资源信息之中，就可以说明上节描述的资源组织方式涵盖了以 $x$ 为索引的线性链表。

证明：

1）以 $r_0$ 为原点，以 $d$ 个参数为坐标轴资源空间，被 $d$ 个坐标轴分割成 $2^d$ 个子区域，如图 1 所示。

2）其中与 $x$ 轴正向邻接的子区域有 $2^{d-1}$ 个，如图 3 所示。我们用区域 $a_1,a_2,\cdots,a_{2^{d-1}}$ 表示这 $2^{d-1}$ 个子区域；所以 $r^+ \in a_1 \cap a_2 \cap \cdots \cap a_{2^{d-1}}$。

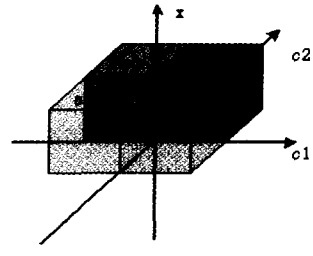


图 3 区域邻接图

3）设 $r_i^x$ 是子区域 $a_i$ 中在 $x$ 方向中与 $r_0$ 最接近的资源，这样我们得到了一组资源列表 $(r_1^x,r_2^x,\cdots,r_{2^{d-1}}^x)$，它们是区域 $a_1,a_2,\cdots,a_{2^{d-1}}$ 中在 $x$ 方向中与 $r_0$ 最接近的资源。

4）设 $r_i$ 是 $(r_1^x,r_2^x,\cdots,r_{2^{d-1}}^x)$ 中与 $r_0$ 在 $x$ 方向上最接近的资源，也就是说 $r_i$ 是 $a_1 \cap a_2 \cap \cdots \cap a_{2^{d-1}}$ 中与 $r_0$ 在 $x$ 方向上最接近的资源。

5）又由于 $r^+ \in a_1 \cap a_2 \cap \cdots \cap a_{2^{d-1}}$，且 $r^+$ 是在 $x$ 正向上与 $r_0$ 最接近的资源；

6)所以 $r_i=r^+$；

7)所以 $r^+$ 在 $r_0$ 的邻接资源信息中；

8)同理可以证明 $r^-$ 也在 $r_0$ 的邻接资源信息中；

9)所以，上节描述的资源组织方式涵盖了以 $x$ 为索引的线性链表；

10)又由于以 $x$ 为索引的线性链表示连通的，因此在上节描述的资源组织方式中资源是连通的。

证毕。

### 4.3 基于资源空间模型的资源查找算法

#### 4.3.1 资源查找算法描述

资源查找过程首先是从起始 $r_0$ 出发，按资源之间的邻接关系，沿着与目标资源 $r_k$ 不断接近的方向，查找目标资源 $r_k$ 的过程。

A. 通过比较 $r_0$ 和 $r_k$ 的属性值，判断 $r_k$ 落在以 $r_0$ 为中心的哪个子区域中；

B. 取这个子区域中的 $d$ 个邻接资源点信息，判断 $r_k$ 是否在其中，如果在，返回结果，查找结束；否则选择与 $r_k$ 距离最近的邻接资源点作为下一跳，并记录 $r_k$ 在相对于 $r_0$ 的方向，转到下一跳；

C. 判断 $r_k$ 是否是本跳的一个邻接点，如果是，则资源查找结束；

D. 判断上一跳 $r^{i-1}$ 和 $r_k$ 是否落在本跳资源点 $r^i$ 的同一其区域内，

a. 如果是，则检查本区域内是否有比上一跳资源点及本跳资源点更接近 $r_k$ 的资源点，如果有，则将这个资源点作为下一跳，继续 $C$ 步操作。

b. 否则，计算 $d$ 个邻接资源点、上一跳资源点、本跳资源点计算与 $r_k$ 沿各坐标方向的最短距离：

$$d^{\min}=\min(d_{c1}^{\min},d_{c2}^{\min},\cdots,d_{cd}^{\min}) \tag{4}$$

$$d_{cj}^{\min}=\min(|c_{r^{i-1}}^{j}-c_{r_k}^{j}|,|c_{r^i}^{j}-c_{r_k}^{j}|,|c_{r_1^i}^{j}-c_{r_k}^{j}|,|c_{r_2^i}^{j}-c_{r_k}^{j}|,\cdots,|c_{r_d^i}^{j}-c_{r_k}^{j}|) \tag{5}$$

其中，$d_{cj}^{\min}$ 是与 $r_k$ 在 $c^j$ 方向上最短距离；$c_r^j$ 是资源 $r$ 在 $c^j$ 上分量；$r_n^i$ 是本跳资源 $r^i$ 的第 $n$ 个邻接点。

c. 从距离最短的资源点，沿着最短方向，逐个查找，直至找到满足条件的资源或返回资源不存在信息。

E. 如果不是，选择下一跳资源节点，如果没有下一跳资源节点，返回空；否则转交下一跳节点，重复 $C$ 到 $E$ 步骤直至找到资源 $r_k$ 或返回资源不存在信息；

F. 存在与 $r_k$ 的相似度大于于 $r_0$ 与 $r_k$ 的相似度的关联点，则将查找任务交给 $r_k$ 与的相似度最最大的资源重复上述查找过程，直至找到资源 $r_k$ 或返回资源不存在信息。

#### 4.3.2 资源加入算法描述

资源加入算法如下：

A. 首先，从网格中任意选择一个资源点，作为起始资源点；

B. 从该起始资源点出发，查找要加入资源点自己，找出与其最接近的资源点；

C. 构建该资源点在 $2^d$ 个子区域中的 $d.2^d$ 个邻接资源点；

D. 由这 $d.2^d$ 个邻接资源点出发，更正由于该资源点的加入，部分资源节点邻接关系的改变。

#### 4.3.3 资源退出算法描述

资源退出算法如下：

从退出资源点的 $d.2^d$ 个邻接资源点出发，更正由于该资源点的退出，部分资源节点邻接关系的改变。

### 4.4 模拟试验

为了验证算法的正确性和有效性，我们在实验室中搭建了试验环境，来模拟网格环境中的众多资源，使用空间模型结构来组织资源间的邻接关系，使用基于资源空间模型的资源发现算法来进行资源查找。

#### 4.4.1 试验环境

试验软件用 Java 语言开发，主要软件主要包括资源点对象 Node、邻接对象 Neighbor 和主控对象 Server 组成。Node 对象描述资源实例信息，包括资源属性信息、资源邻接信息；Neighbor 对象描述资源在一个子区域中的邻接资源点信息，包括邻接方向和邻接资源点信息；主控对象 Server 包含一个资源列表和模拟的主要控制流程。

试验硬件环境为一台 HP 至强服务器(4CPU，2GB 内存)，Redhat Linux 9，SUN JDK1.4.2。

#### 4.4.2 试验数据

对于描述计算资源的属性参数，可以通过某种变换，使其取值范围在 0 和 1 之间。为了便于分析和测试系统的调试，用三维属性数据来描述描述资源。测试数据由一系列资源点组成，每个资源点由三个属性数据来描述，属性数据的值由随机函数随机产生，取值在 0 和 1 之间。共产生了 18 组测试数据。

表 1 试验数据

| 组别 | 节点数 |
|---|---|
| 1 | 100 |
| 2 | 1000 |
| 3 | 2000 |
| 4 | 3000 |
| 5 | 4000 |
| 6 | 5000 |
| 7 | 6000 |
| 8 | 7000 |
| 9 | 8000 |
| 10 | 9000 |
| 11 | 10000 |
| 12 | 12000 |
| 13 | 15000 |
| 14 | 20000 |
| 15 | 25000 |
| 16 | 30000 |
| 17 | 40000 |
| 18 | 50000 |

#### 4.4.3 试验过程

a)从输入数据文件中依次读取各资源信息，按资源之间的相似度构造邻接关系；

b)随机选择一个资源点作为资源查找起始点；

c)随机选择一个资源点作为资源查目标点；

d)从起始资源点开始，按照基于资源空间模型的资源发现算法寻找目标资源点；

e)记录查找步数(所经过的资源点个数)；

f)重复 2)至 5)步骤一万次；

g)分别使用 18 组试验数据重复步骤 1)至 6)。

#### 4.4.4 试验结果和分析

表 2 所示是试验系统的试验结果。

表 2 试验结果

| 组别 | 节点数 | 平均查找步数 |
|---|---|---|
| 1 | 100 | 2.8809 |
| 2 | 1000 | 7.8433 |
| 3 | 2000 | 10.6713 |
| 4 | 3000 | 12.9757 |
| 5 | 4000 | 14.9045 |
| 6 | 5000 | 16.7572 |
| 7 | 6000 | 18.5037 |
| 8 | 7000 | 20.1106 |
| 9 | 8000 | 22.0903 |
| 10 | 9000 | 23.0067 |
| 11 | 10000 | 24.6342 |
| 12 | 12000 | 27.4529 |
| 13 | 15000 | 31.6142 |
| 14 | 20000 | 36.7902 |
| 15 | 25000 | 43.4994 |
| 16 | 30000 | 47.6794 |
| 17 | 40000 | 58.0572 |
| 18 | 50000 | 66.2148 |

我们将试验结果转换成平均查找步数与资源点数之间的关系图，如图 4 所示。

其中横坐标是资源点数，纵坐标是平均查找步数；黑色曲线(带点)是试验结果曲线，红色曲线是拟合函数曲线，拟合函数是 $y=x^{0.6}/10$。

**结论** 本文提出了一种描述网格计算资源的空间模型，在该模型下，不需建立全局的资源视图，系统不会随着资源数量的快速增加而变得不可用。另外由于资源的搜索总是朝着有利的方向发展，因此能够提高资源的搜索速度比较快。

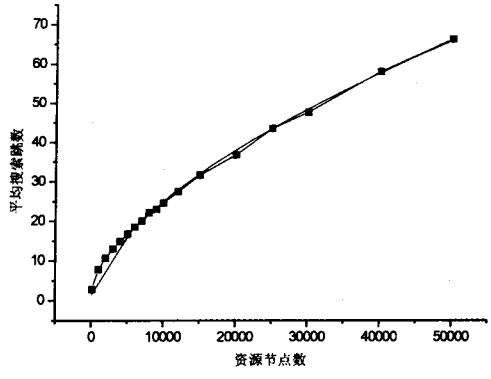


图 4

## 参 考 文 献


1 Foster I, Kesselman C, et al. The Grid : Blueprint for a New Computing Infrastructure. Morgan2Kaufmann, San Francisco, CA, 1998

2 徐传福，陈海涛，等. 基于 DHT 的层次式 P2P 资源定位模型. 计算机工程与应用，2004，18：156～158

3 李伟，徐志伟，等. 网格环境下一种有效的资源查找方法. 计算机学报，2003，26(11)：1546～1549

4 董方鹏，龚奕利，等. 网格环境中资源发现机制的研究. 计算机研究与发展，2003，40(12)：1749～1755

5 李伟，徐志伟. 一种网格资源空间模型及其应用. 计算机研究与发展，2003，40(12)：1756～1762

6 Rajkumar B. High Performance Cluster Computing. Beijing, Posts & Telecom Press, 2002

7 Foster I, Kesselman C. Globus: a meta- computing infrastructure toolkit[J]. International Journal of Supercomputer Applications, 1997, 11(2): 115～128


(上接第 102 页)

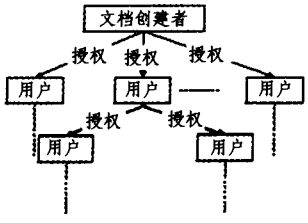


图 5 授权关系树

图 5 表示在一个有 $n$ 个实体元权限的实体上，$n$ 棵授权关系树中的一棵。其它的 $n-1$ 棵授权关系树都与此类似。假设这是实体权限为 $x$ 的授权树，那么有且只有该树上的用户才具有对文档创建者所创建的文档享有 $x$ 操作权限。由于电子文档的元权限只有有限的几种，因此采用这种方法使得授权关系在逻辑上更简明。当对某一用户授予对文档的一种操作权限时，就在该权限树上添加该用户作为授权用户的子结点。当对某一用户削除权限时，必须且只能由其上级结点进行并采用一种级联式削权，即在删除该结点的某项权限之前必须扫描子结点 找出具有该权限的子结点并删除该权限。通过这种递归式扫描一直到叶结点，这样以该结点为起点的子树将被删除，如图 6 所示。

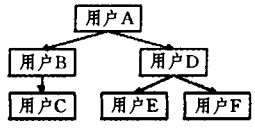


图 6 削权前的授权树

图 6 是从一个授权关系网中分离出来的关于权限 $x$ 的一棵授权关系子树，用户 B,D 的 $x$ 权限只能由用户 A 取消，并且在这棵树上用户 A 也仅能取消其子树的 $x$ 权限(假设用户 B、C 还拥有其它权限)。当用户 A 取消 B 的 $x$ 权限时，用户 C 的 $x$ 权限也会被取消。同理，当用户 A 取消用户 D 的 $x$ 权限时，用户 E 、用户 F 的 $x$ 权限也会被取消 。结果如图 7 所示。

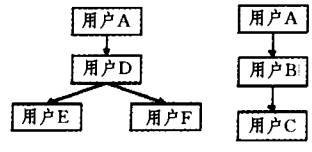


图 7 削权后的授权树

这样通过一个基于授权关系树的 DAC 模型就能灵活地根据政府运作的需要对相关文档进行授权管理跟踪，弥补了 RBAC 的不足。

通过以上对应政府角色的层层授权，结合 PKI 体系进行角色管理，政府信息集成平台中的电子文档可以达到综合安全而高效的管理。

## 参 考 文 献


1 盛小钢，王克，陈庆荣. 电子公文同样可信——基于 PKI 的电子印章系统的设计与实现. 计算机安全，2004(12)

2 朱靓，叶志坚. 基于 PKI 的交叉认证技术在电子政务中的研究与应用. 福建电脑，2005 (12)