

分布式虚拟环境中的定位服务^{*}

李 惠 贝 佳 翟 磊 孙元浩 龚震宇 潘金贵

(南京大学计算机软件新技术国家重点实验室 南京大学计算机科学与技术系 南京210093)

摘 要 针对分布式虚拟环境(DVE)中的静态数据分布式存储的问题,本文提出了一种适用于DVE的定位服务机制,以及一种在定位服务中使用的兴趣区域表达式的快速匹配算法。DVE中的定位服务类似于DNS域名服务,不同点在于定位服务根据空间和逻辑信息进行匹配。定位服务提供了对分布式存放的静态数据的有效、可靠的访问。本文提出的定位服务机制和匹配算法都具有很好的扩展性,可以方便地扩展到对其他更加复杂的兴趣描述进行精确定位。
关键词 分布式虚拟环境,定位服务,分布式系统,匹配算法

Location Service in Distributed Virtual Environment

LI Hui BEI Jia ZHAI Lei SUN Yuan-Hao GONG Zhen-Yu PAN Jin-Gui

(State Key Lab for Novel Software Technology, Department of Computer Science and Technology, Nanjing University, Nanjing 210093)

Abstract To the question of data distribution in Distributed Virtual Environment, Location Service, an applicable mechanism, is put forward here, together with a fast matching algorithm related to the interested area expression. There is an analogy between the Location Service and the DNS service, while the former does the matching according to the information of space and logic. The Location Service in DVE provides effective and reliable access to the distributed static data. The Location Service and matching algorithm presented in this paper is scalable. It is very convenient to extend to provide location service to other complex interest description.

Keywords Distributed virtual environment, Location service, Distributed system, Matching algorithm

分布式虚拟环境(Distributed Virtual Environment, DVE)是将虚拟现实技术和网络技术结合而产生的一个新的研究领域^[1]。DVE提供一个共享的虚拟空间,并能够通过计算机网络支持对象的仿真以及多个分布用户的交互和通讯。网络带宽是限制DVE性能的一个重要因素。针对这个问题,人们对DVE中的(静态)数据分布策略进行了很多研究。DVE中的静态数据主要包括3D图形数据、2D图像、mpeg、声音等组成虚拟场景的数据^[5]。按照数据的存放位置可分为集中式存储和分布式存储。按照客户端获得数据的方式,可分为本地存储和网络下载两种方式^[4]。

数据集中式存储的优点是便于管理,实现简单。但是缺点也显而易见,比如对服务器存储空间要求比较大、服务器易成为瓶颈等。数据的本地存储使得运行时刻在网络上只需要传输对象状态的改变信息,但这样降低了应用的灵活性,而且数据的一致性也难以保证。理想的模式应该是采用数据的分布式存储和网络下载来管理数据。所有的数据分布式存放在多个数据服务器上,客户机上只保留兴趣区域(Area Of Interest, AOI)内的数据。当客户端需要新的数据时,则向相应的数据服务器请求。

数据分布式存储模式下,客户端需要请求新的数据时,应该能够知道该向哪个数据服务器进行请求,因此在DVE中需要提供定位服务。客户端维护自己的兴趣区域,需要下载数据时,发送兴趣区域的描述给定位服务机制,请求对应的数据服务器的IP地址和端口号。定位服务机制根据兴趣区域进行

匹配,反馈给客户端相应的一个或者多个数据服务器地址和端口号,客户端再向数据服务器请求数据。数据库服务器需要预先向定位机制注册自身维护数据对应虚拟空间的描述信息。这种方式类似于DNS域名服务,不同点在于定位服务根据空间和逻辑信息进行匹配,而DNS根据域名字符串进行匹配^[2]。而且DNS中一般一个域名只对应一个IP地址,而定位服务中有可能一个兴趣区域落在多个数据服务器维护的空间区域内,因此一个兴趣区域有可能会对应多个数据服务器的地址。

本文将主要对分布式虚拟环境中的定位服务和快速匹配算法做一些研究。我们提出了一种可应用于分布式虚拟环境的定位服务机制以及一种快速的兴趣表达式匹配算法。定位服务机制已经应用在了我们的分布式虚拟环境AIMNET中。

1 定位服务机制

1.1 兴趣表达式和兴趣区域表达式

在我们的应用中,采用兴趣表达式来表示用户的兴趣,兴趣表达式采用属性关系表达式的集合来表示,每个属性都由一个三元组(类型,属性名,值或值域)构成。表1给出了一个例子。这个例子表明user agent为superli@mes.nju.edu.cn的主机参加名称为“e-classroom”的应用,并且感兴趣的范围是 $\{1 < x < 5, 0 < y < 1, -1 < z < 1\}$,感兴趣的对象属于“graduate”组,且是“avatar”类型,接收的消息类型是“AU-

^{*} 国家自然科学基金(No. 60473571)资助,李 惠 硕士研究生,主要研究领域为分布式虚拟环境、远程渲染;贝 佳 博士研究生,主要研究领域为分布式虚拟环境、网络组通讯技术;翟 磊 硕士研究生,主要研究领域为计算机图形学和分布式虚拟环境;孙元浩、龚震宇 硕士,主要研究领域为计算机图形学和分布式虚拟环境;潘金贵 教授,博士生导师,主要研究领域为多媒体信息处理、计算机图形学。

DIO”或者“TEXT”。

表1 兴趣表达式示例

类型	属性名	值或值域
String	user agent	superli@mes.nju.edu.cn
String	application	"e-classroom"
Float	spatial-x	$1 < x < 5$
Float	spatial-y	$0 < y < 1$
Float	spatial-z	$-1 < z < 1$
String	organization	"graduate"
String	function	"avatar"
String	medium-set	{"audio" "text"}

在进行静态数据的定位服务时,客户端需要精确描述自己的兴趣区域,数据服务器也需要将维护的空间区域(也就是数据服务器的兴趣区域)向定位服务器注册。兴趣区域用兴趣区域表达式来描述。如表1中的表达式 $\{1 < x < 5, 0 < y < 1, -1 < z < 1\}$ 表示点的坐标同时满足以上三个谓词的所有点组成的空间区域,实际上是一个各条边和坐标轴平行的长方体,也称为绑定盒(Axis-Aligned Bounding Box, AABB)。兴趣区域可以是一个或多个绑定盒描述的空间区域的并集。描述一个绑定盒对应空间区域的表达式称为简单兴趣区域表达式,多个简单兴趣区域表达式的析取式称为复合兴趣区域表达式,复合兴趣区域表达式对应的空间区域是多个简单兴趣区域表达式描述的空间区域的并集。简单兴趣区域表达式和复合兴趣区域表达式统称为兴趣区域表达式。由于兴趣区域的枚举特性,可以证明任何兴趣区域集合均可对应一个兴趣区域表达式,这个兴趣区域表达式可能是简单也可能是复合兴趣区域表达式。

此外为了满足快速匹配算法的需要,还需在兴趣区域表达式之上定义加、减两个二元操作。加法操作用于将两个兴趣区域表达式表示的区域合并,减法操作用于从第一个表达式表示的区域中减去彼此相交的部分。很容易证明,执行加法或者减法操作后的结果区域仍然能用一个兴趣区域表达式来表示。

简单兴趣区域表达式实际上使用了只包含“与”操作的谓词来表示。在具体实现时,表1中的简单兴趣区域表达式通常描述为以下6个用“与”操作关联起来的单操作符谓词: $X > 1$ 、 $X < 5$ 、 $Y > 0$ 、 $Y < 1$ 、 $Z > -1$ 、 $Z < 1$ 。

1.2 几何数据的分布式组织

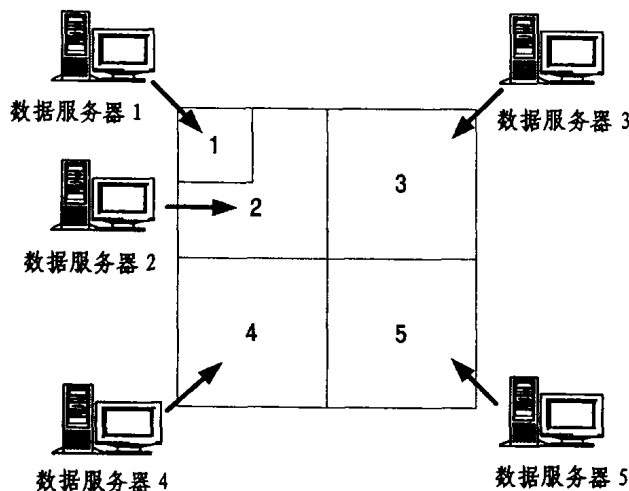


图1 空间划分在二维平面上的投影表示

分布式虚拟环境中的静态数据分布式的存储在多台数据服务器上,我们采用了一种简单的虚拟空间的划分方法^[4],把三维空间分成多个长方体。具体实现时,我们对划分做了一个简化,只在X、Y两维上进行划分,将整个虚拟空间划分为多个贯穿Z轴的子空间,每个数据服务器维护一个兴趣区域表达式表示的空间区域内的数据。如图1所示,将整个空间分为五部分,对应的静态数据分别存储在五台数据服务器上,数据服务器1、3、4、5维护的区域都可以用简单兴趣区域表达式来表示,而服务器2对应的区域则需要用复合兴趣区域表达式来表示。

1.3 定位服务器的组织

定位服务器接受客户端的定位服务请求,最终反馈给用户对应的数据服务器地址。在数据服务器数量较少时,可以用一台定位服务器来维护整个空间区域;数据服务器数量很多时,为了平衡每台定位服务器的负载,我们采用了一种层次式组织定位服务器的方法,将多台定位服务器组织成一棵树,树中的每个节点都知道对应的父节点和子节点的位置。每个节点都维护一个服务区域^[6]和一张谓词表。服务区域表示在该节点和它所有的子节点能够提供定位服务的虚拟空间区域的集合,也用兴趣区域表达式表示。谓词表由服务区域内各子区域的表达式对应的谓词组成,用于测试该区域是否和用户兴趣区域相交,如果相交,则表明两个区域匹配。根节点的服务区域对应整个虚拟空间,其他节点的服务区域对应虚拟空间中的一部分区域,同一层次的节点对应的服务区域彼此不相交。每台定位服务器对应的服务区域又分为可以精确匹配的区域和可以模糊匹配的区域。精确匹配表示匹配后可以直接得到数据服务器的地址,模糊匹配表示数据服务器的位置可以在子节点中找到,匹配后得到的是定位服务器的地址,客户端需要根据这个地址再次发送定位请求。如图2所示,每台服务器上服务区域内可以精确匹配的区域用灰色来表示,可以模糊匹配的区域用白色来表示。

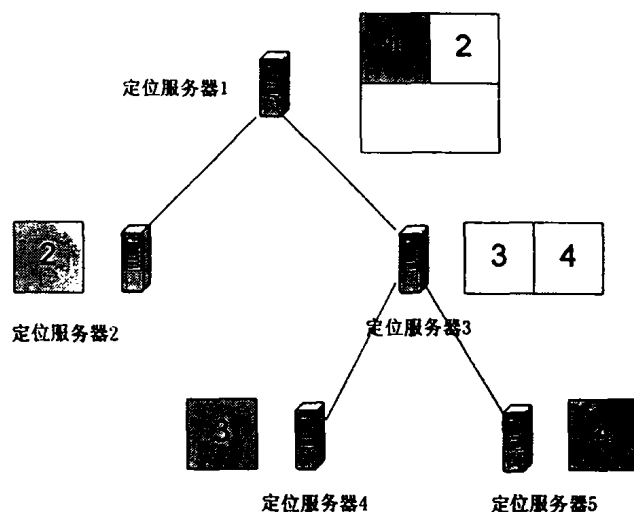


图2 定位服务器拓扑结构

1.4 匹配策略

初始状态下,每个客户端都有一个自己的登录服务器^[6],可以是任意一台定位服务器。每次请求定位服务时,都首先向自己的登陆服务器发送请求。定位服务机制接收用户的请求,并根据请求中的兴趣区域表达式返回给用户一个或者多个数据服务器的地址。每台定位服务器都首先将自己的服务区域和用户兴趣区域进行相交测试,用来判断这两个区域是否有

相交部分。根据测试结果,定位服务器有以下三种处理方法:

① 如果用户兴趣区域和定位服务器服务区域不相交,则向定位服务器父节点转交请求或者返回给客户端父节点的地址,客户端再向上一层定位服务器提交请求;

② 如果用户兴趣区域完全落在定位服务器的服务区域内,则根据谓词表进行匹配,返回给客户端对应的数据服务器地址或者对应的定位服务器子节点的地址;

③ 如果用户兴趣区域部分落在定位服务器的服务区域内,对于不在服务区域的部分,按照①的方法进行处理;对于在服务区域的部分,按照②的方法进行处理。

作为响应,定位服务器需要发送一个或者多个数据报给客户端,每个数据报中都包含一个服务器地址和兴趣区域表达式。如果服务器类型是定位服务器,兴趣区域表达式描述用户兴趣区域落在定位服务器服务区域内的部分,客户端还需根据用这个表达式再次向这台定位服务器请求定位服务;如果服务器类型是数据服务器,兴趣区域表达式描述用户兴趣区域落在数据服务器的兴趣区域内的部分,同时定位服务器还需发送数据服务器的兴趣区域给客户端,客户端将这些信息保存在本地缓存中,当再次需要请求新的数据的时候,首先将兴趣区域和本地缓存中保存的数据服务器的兴趣区域进行相交测试,只将不相交的区域向定位服务器请求定位服务。一般情况下,由于数据服务器的兴趣区域要远比用户的兴趣区域范围大,因此采用这种方法可以大大地减少用户请求定位服务的次数。

请求定位服务的过程类似于请求 DNS 服务的过程,比如图2中,如果有客户端将定位服务器3作为登录服务器,而请求定位服务的区域落在空间区域2表示的范围内,则定位服务处理流程如下:

① 客户端发送定位请求给定位服务器3,定位服务器3反馈给客户端定位服务器1的地址;

② 客户端再向定位服务器1发送定位请求,定位服务器1反馈给客户端定位服务器2的地址;

③ 客户端再向定位服务器2发送定位请求,定位服务器2反馈给用户对应的数据服务器地址,客户端就可以向这个数据服务器请求所需的数据。

1.5 快速匹配算法

如果客户端兴趣区域表达式和定位服务器的服务区域有相交部分,则需要将客户端兴趣区域表达式和谓词表中表示的兴趣区域表达式进行匹配。文[3]中描述了一种用于分布式虚拟环境中事件模型的快速匹配算法,稍做改动和简化,就可用于定位服务器的兴趣区域表达式匹配中。算法的基本思想就是将用户兴趣区域表达式中的简单兴趣区域表达式和服务器兴趣区域表达式中的简单兴趣区域表达式进行相交测试。需要用到的数据结构有一张谓词表和一个服务器地址列表,如图3所示。现对该算法简单描述如下(算法的详细描述参见文[3])。

谓词表由多行组成,每一行都对应一个谓词族,而且按序排列,一个谓词族包含了所有相同属性相同操作符的单操作符谓词。操作符为“<”的称为上阈值谓词族,操作符为“>”的称为下阈值谓词族。上阈值谓词族中的值按降序排列,下阈值谓词族中的值按升序排列。例如,有如下一个上阈值谓词族的排列序列:

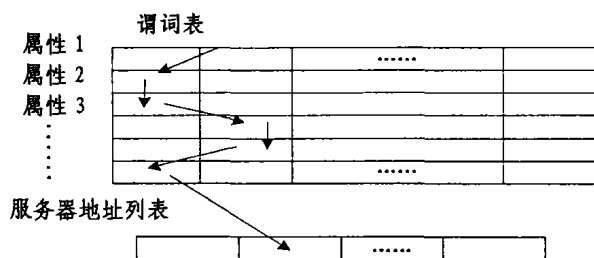


图3 快速匹配算法主要数据结构

"X<"	15	14	10	7	1
------	----	----	----	---	---

如果需要匹配的 X 值是9,则在7的位置之前的三个谓词都满足匹配。组成服务器的兴趣区域表达式的每一个简单兴趣区域表达式中所有的谓词都连接起来,指向地址列表中的对应的服务器地址。服务器地址可能是数据服务器地址,也可能是定位服务器的地址。

进行匹配时,在测试上阈值谓词族时,选取待测试的简单兴趣区域表达式中的下阈值谓词的属性值进行测试,采用二分法查找,把一行中的谓词分成左右两部分,左半部的谓词部分是匹配的;对于下阈值谓词族的测试,选取待测试的简单兴趣区域表达式中相应的上阈值谓词匹配,左半部的谓词部分是匹配的。逐行测试后,就可以得到所有与客户端兴趣区域表达式匹配的谓词集合,继而可以得到所有匹配对应的服务器的地址。匹配完成后,还要将用户兴趣区域表达式和匹配的服务器兴趣区域表达式执行减法运算,得到匹配区域的兴趣区域表达式。然后将每个匹配结果发送给客户端。

假定每台定位服务器上谓词表有 A 行,第 i 行的谓词数为 P,服务器地址列表的长度为 S,可以计算出整个匹配过程的时间复杂度为 $O(S)$ 。该算法可以很容易地扩展到对兴趣区域表达式中表示的其他属性进行匹配,而不仅限于对兴趣区域表达式进行匹配;对于其他一些非等值谓词,只需在谓词表中添加行即可;而对于等值谓词,在谓词表中添加行,匹配的时候只需要将等值谓词族中的值按序排列,待匹配的值和每一行中的值采用二分法比较即可。

1.6 场景服务器的注册

在定位服务器能够处理请求之前,数据服务器需要向其注册包含的场景描述信息,这些描述信息也用一个兴趣区域表达式来表示。由于数据服务器维护的静态数据状态一般比较稳定,因此可以通过手动配置,设定相应的定位服务器的服务区域。也可以借助于一些动态配置的算法,尽量平衡各个定位服务器服务区域的规模,动态地配置相应定位服务器的服务区域和谓词表。我们目前采用的是手动配置的方法。

2 实现方法

在编程实现时,简单兴趣区域表达式用类 LSInterestExp 来表示,客户端向定位服务器发送的请求数据报格式用类 LSPacketQuest 来表示。定位服务器返回给客户端的响应数据报用类 LSPacketReply 来表示。这三个类主要定义如下:

```
class LSInterestExp
{
    float pRanges[2][2];
}
class LSPacketQuest
{
    LSInterestExp interestExp;
}
class LSPacketReply
```

(下转第143页)

推导子图上 WMIS 问题的最优解。

命题4 该近似算法的近似度为 $1/\lceil(\Delta'+1)/3\rceil$, 其中 Δ' 是图 G 中点的最大度数。

证明: 将图 G 划分为 $\lceil(\Delta'+1)/3\rceil$ 个子图, Δ' 为图中点的最大度数。将每个点在图 G 中相关的所有边上的权值求和, 作为该点的权值, 求出每个子图中 WMIS (点带权最大独立集) 问题的最优解 $I[i]$ 。我们会看到图 G 上 EWMIS (边带权最大独立集) 问题的最优解 I^* 最多等于这 $\lceil(\Delta'+1)/3\rceil$ 个局部最优解的并集。因为经过划分后, I^* 中每个点都会属于 $\lceil(\Delta'+1)/3\rceil$ 个顶点子集中的其中一个, 它们所关联的边的权值总和不会超过它们所在的子集中的 WMIS 问题的最优解的权值。即使几个 I^* 中的点同时属于划分后的某个子集, 情况也是如此。

我们从这 $\lceil(\Delta'+1)/3\rceil$ 个最优解中找出权值最大的解 $I[max]$ 。用 $Weight(I[i])$ 表示子集 $I[i]$ 中所有点的权值之和, 则有 $Weight(I[max]) \geq Weight(I[i]), 1 \leq i \leq \lceil(\Delta'+1)/3\rceil$, 进而有 $\lceil(\Delta'+1)/3\rceil Weight(I[max]) \geq \sum_{1 \leq i \leq \lceil(\Delta'+1)/3\rceil} Weight(I[i])$; $Weight(I^*)$ 表示图 G 中 EWMIS 问题最优解 I^* 关联的所有边的权值之和。根据上面我们的分析, 有 $Weight(I^*) \leq \sum_{1 \leq i \leq \lceil(\Delta'+1)/3\rceil} Weight(I[i])$, 最后推出, $\rho(n) = Weight(I[max])/Weight(I^*) \geq 1/\lceil(\Delta'+1)/3\rceil$, 故该算法的近似度为 $1/\lceil(\Delta'+1)/3\rceil$ 。

命题5 算法的运行时间为 $O(|V|^2)$

(上接第134页)

```
{
    LSInterestExp interestExp;
    List<LSInterestExp> dataServerExp;
    sockaddr dataServerLoc;
    int nPort;
    int nAddrType;
}
```

类 LSInterestExp 中的4个值分代表一个绑定盒在 X 、 Y 两个坐标轴上的最大最小值; 类 LSPacketQuest 中的 interestExp 表示用户感兴趣的区域, 我们只用一个简单兴趣区域表达式来表示用户的兴趣区域; 类 LSPacketReply 中的 dataServerLoc 和 nPort 表示一个服务器的地址和端口号, 服务器可能是数据服务器, 也可能是定位服务器, nAddrType 用于标记服务器的类型; interestExp 表示客户兴趣区域和 dataServerLoc 表示的数据服务器的兴趣区域的相交部分, dataServerExp 则表示返回的数据服务器所对应的空间的兴趣区域表达式描述, 可能是一个, 也可能是多个简单兴趣区域表达式。

在我们的实现中, 定位服务机制和数据分布是相互独立的, 而且定位服务是一种在应用层之上的服务, 可以作为一个应用程序在计算机上运行, 也就是说同一台计算机既可以作为数据服务器, 也可以作为定位服务器。

定位服务和数据的网络传输会造成一定的延迟, 降低了客户端响应的实时性。一种有效的办法是在客户端使用行为预测方法, 根据用户的视点历史位置, 对视点将来的位置做预测, 客户端根据预测的位置向定位服务器发送请求, 然后向对应的数据服务器预取数据。在数据的传输和渲染时, 还需要累进式传输 (Progressive Transmission)、累进式渲染 (Progressive

证明: 步骤1可以在常数时间内完成, 步骤2运行时间为 $|V|$, 步骤3运行时间最多为 $|V| \cdot \lceil(\Delta'+1)/3\rceil$, 步骤4运行时间为 $|V|$, 步骤5, 6, 7可以在 $O(|V|^2)$ 的时间内完成。所以, 算法总的运行时间可以在 $O(|V|^2)$ 时间内完成。

结束语 本文从新的角度出发, 发展了原有的带权最大独立集问题的领域, 第一次系统地提出了边带权最大独立集问题, 而且通过对其属于 NP-Complete 性质的证明以及问题结构的研究分析, 进一步给出了一个近似度为 $1/\lceil(\Delta'+1)/3\rceil$ 近似算法。我们未来的研究重点将放在边带权最大独立集问题特有性质的进一步发掘上, 希望通过问题特有的结构找到具有更好近似度的算法。

参考文献

- 1 Hochbaum D S. Approximation Algorithm for Np-hard Problem. PWS Publishing Company, 1997
- 2 Corman T H, Leiserson C E, Rivest R L, Stein C. Introduction to Algorithms, Second Edition, The MIT Press, 2001
- 3 Arkin E M, Hassin R. On local search for weighted k-set packing. ESA'97
- 4 Baker B S. Approximation algorithm for NP-complete problems on planar graphs. J. ACM, 1994
- 5 Halldorsson M M, Lau H C. Low-degree graph partitioning via local search with applications to constraint satisfaction, max cut, and 3-coloring. J. Graph Algo. Applic. 1997

sive Rendering)等相关技术, 我们将另外著文说明。

结论和进一步工作 本文提出了一种用于分布式虚拟环境的定位服务机制, 可以在客户端只需知道自己的空间位置和感兴趣区域的情况下, 提供对虚拟环境中分布式存放的静态数据的有效、可靠的访问。这种方法可以独立于分布式虚拟环境动态通讯结构, 具有一定的通用性。我们已将这种机制应用到了我们的分布式虚拟环境 AIMNET 中。

目前的定位机制中, 各定位服务器的服务区域都是预先配置的, 而且这种机制只能处理静态数据的精确定位, 下一步我们将研究动态的服务区域配置算法以及分布式虚拟环境中的动态数据的精确定位机制。另外, 我们的虚拟环境在安全服务和用户权限管理方面有待加强, 我们会将安全服务和用户权限管理引入分布式虚拟环境, 并将其和定位服务机制相结合。

参考文献

- 1 潘志庚, 姜晓红, 张明敏, 石教英. 分布式虚拟环境综述. 软件学报, 2000, 11(4): 461~467
- 2 孙元浩. 可扩展主动兴趣管理. [南京大学硕士毕业论文]. 2003
- 3 龚震宇. 分布式虚拟环境中的事件模型. [南京大学硕士毕业论文]. 2003
- 4 姜晓红, 潘志庚, 杨孟洲, 石教英. 多服务器/客户 DVE 系统的分区数据管理模式. 计算机学报, 2000, 23(8): 872~876
- 5 杨孟洲, 姜晓红, 潘志庚, 石教英. 分布式虚拟环境中的信息服务和资源定位. 计算机研究与发展, 2000, 37(10): 1227~1232
- 6 Leonhardi A, Rothermel K. Architecture of a Large-scale Location Service. In: Proc. of the 22nd Intl. Conf. on Distributed Computing Systems (ICDCS'02)