



XXXX

## 基于坐标索引的算网标识解析机制

张月<sup>1</sup>, 谢人超<sup>1,2</sup>, 王媛<sup>3</sup>, 王芳<sup>3</sup>, 向梓豪<sup>3</sup>, 唐琴琴<sup>1</sup>, 黄韬<sup>1,2</sup>

(1. 北京邮电大学网络与交换技术全国重点实验室, 北京 100876;

2. 紫金山实验室, 江苏 南京 211111;

3. 中国电子科技集团公司第五十研究所, 上海 200063)

**摘要:** 算力网络深度融合了云端、边缘与终端原本分散且异构的计算资源, 满足了新兴业务对数据实时处理与高效分发的要求。标识解析服务是提供数据存储和检索以支持这些功能的基本前提。关键的问题是构建一种高效的数据索引机制, 来响应数据的检索和存储, 无论数据在算力网络中是如何缓存的。因此, 基于坐标索引方法 (Coordinate-Indexed Name Resolution, CNR) 提出了一种低延迟的算网标识解析机制。通过构造基于地理路由的三角剖分 (Delaunay Triangulation, DT) 图, 在 CNR 维护的虚拟空间上响应来自算力节点的任何查询请求, 极大减少跨服务器检索数据的路径长度, 从而实现高效的标识解析服务。实验表明, 与现有的解决方案相比, 采用 CNR 机制进行的算网标识解析缩短了 41.61% 的查找路径和 57.19% 的响应时间。

**关键词:** 算力网络; 数据共享; 低时延; 算网标识解析

**中图分类号:**

**文献标志码:**

**doi:** 10.11959/j.issn.1000-0801.

## Coordinate-based identification resolution mechanism for computing power networks

ZHANG Yue<sup>1</sup>, XIE Renchao<sup>1,2</sup>, WANG Yuan<sup>3</sup>, WANG Fang<sup>3</sup>, XIANG Zihao<sup>3</sup>, TANG Qinqin<sup>1</sup>,  
HUANG Tao<sup>1,2</sup>

1. State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

2. Purple Mountain Laboratories, Nanjing 211111, China

3. The 50th Research Institute of China Electronics Technology Group Corporation, Shanghai 200063, China

**Abstract:** The Computing Power Network (CPN) deeply integrates originally dispersed and heterogeneous computing resources from the cloud, edge, and terminals, meeting the stringent requirements of emerging services for real-time data processing and efficient distribution. The identity resolution service serves as a fundamental prerequisite for enabling data storage and retrieval to support these functionalities. The key challenge lies in constructing an efficient data indexing mechanism to respond to data retrieval and storage requests, regardless of how data is cached within the CPN. Therefore, based on the coordinate indexing method CNR, a low-latency identifier resolution mechanism was



proposed for computing power networks. By constructing a triangulation graph (DT) graph based on geographic routing, any query request from computing nodes was responded to within the virtual space maintained by CNR, significantly reducing the path length for cross-server data retrieval and thereby achieving efficient identifier resolution services. Experiments showed that compared to existing solutions, the identifier resolution mechanism using CNR reduced lookup path length by 41.61% and response time by 57.19%.

**Key words:** the computing power network, data sharing, low latency, computing power network identifier resolution

## 1 引言

随着大算力数据、计算密集、时延敏感型新兴业务的出现,目前部署的移动通信网络<sup>[1]</sup>无法满足这些智能应用无处不在、按需、实时的资源需求。算力网络<sup>[2]</sup>应运而生,它以计算与网络能力深度融合为特征<sup>[3]</sup>,旨在以计算服务的形式为未来智能社会提供计算支撑。标识解析服务是提供数据存储和检索实现算力网络这些功能的基本前提。而算力网络中算力节点之间的数据共享可以有效地减少数据的查找时延,从而降低网络的带宽消耗。在算网中实现数据共享依赖于层次化算力网络体系结构,主要包括位于顶层的云数据中心(Data Center, DC)以及广泛部署于网络边缘与终端的大量分布式算力节点。这些算力节点在组织上被进一步划分为多个逻辑或地理区域,每一个区域由区域DC统一管控。在这些地理位置分散的算力节点间进行数据共享可以减少数据检索时延,并且在远程云中心缓存的数据项也应该在算力节点之间共享。但是在算力网络中进行数据共享面临一些问题,首先在传统的云计算算力数据共享方式中,算力节点需先将数据上传至远程云,其他节点再从云端获取,这一过程将引入显著延迟<sup>[4]</sup>;其次,尽管算力节点之间的数据共享可以有效地减少标识请求的响应延迟<sup>[5][6]</sup>,但实际环境中算力节点可能跨越区域移动,导致数据分布动态变化,进一步增加了共享机制的复杂度。

要解决上述问题,关键的挑战是实现数据索引,它指示数据在算力网络中的位置,一个有效

的数据索引机制对算网标识解析服务是非常必要的。现有的基于域名服务(Domain Name Service, DNS)<sup>[7]</sup>的方法可能会解决上述问题。但是,在基于域名系统的索引方案中,数据请求可能经过多级递归查询,最终被路由至位于顶层的根域名服务器。该过程通常涉及跨多个网络层级的长路径转发,进而显著增加整体访问时延。同时,该方案采用集中式索引架构,依赖专用索引服务器统一维护全网数据索引信息。这种高度中心化的设计导致数据检索与存储过程过度集中于少数索引节点,一旦这些节点出现性能瓶颈或单点故障,将直接影响整个标识解析服务的可用性与效率。此外,该方案在系统容错能力与横向扩展性方面也存在局限,难以适应算力网络中动态、分布式的资源调度与高并发服务需求。另一种解决方法是基于分布式哈希表(Distributed Hash Table, DHT)<sup>[8]</sup>的方案,在DHT方法中,主要利用finger表来存储其他节点的信息,通过finger表节点可以快速地跳转到其他节点,而不是线性的遍历每个节点,但是在这种方法中,查找请求将经过最多 $\log(n)$ 个覆盖跳来定位数据,其中 $n$ 是算力网络中交换机的数量,路径仍然较长,远超过实际算力节点间的最短路径。因此,迫切需要一种高效的索引机制来支持算力网络中的标识解析服务。

本文提出了一种基于坐标的索引机制(Coordinate-Indexed Name Resolution, CNR),用于算力网络的标识解析服务。为了实现高效的索引机制,算力网络的控制平面维护一个虚拟2维空间,通过构造基于地理路由的三角剖分(De-

launay Triangulation, DT) 图, 在 DT 图上响应数据的请求查询。控制平面首先计算交换机和数据在 DT 图上的坐标, 然后通过 DT 图的构建方法连接这些点以构成虚拟空间的算力网络, 在 DT 图上通过贪婪转发将数据索引转发到距离该数据索引最近的交换机上, 最后交换机将数据索引转发到与其直接连接的所有算力节点对应的区域 DC 上。CNR 是 DHT 的扩展, 但在 DHT 索引机制下, 算力节点需要维护 finger 表, 用于在传统 DHT 下进行算力标识解析。同时, 对于每个所请求的数据索引, 查找请求需要经过算力节点之间的  $\log(n)$  个覆盖跳。在 CNR 机制下, 只需要从入口区域 DC 到目的区域 DC 一个覆盖跳来搜索数据索引, 而无需跨服务器搜索数据项。CNR 可以使用不同的网络规模, 响应来自所有算力节点的标识解析请求。CNR 机制实现了最短的路径长度和最少的算力节点中的响应时间来检索数据索引。此外, 在算力网络中会缓存多个数据副本以减少检索延迟, 而因为 CNR 机制将算力网络映射到 DT 图上, 当进行数据检索时, 数据解析请求总能路由到与待检索的数据副本位置最接近的交换机上, 由与该交换机直接相连的区域 DC 响应数据解析请求, CNR 的优势包括响应延迟小、实施开销低和查找路径短。本文的主要贡献如下:

- 设计了基于坐标的 CNR 索引机制来响应算力数据的查找, 解决数据共享中数据难以定位的问题, 从而提供低延迟的标识解析服务;

- 构建基于地理路由的 DT 图, 算力网络被映射到虚拟空间, 标识解析请求可以直接从入口算力节点传递到目的算力节点, 减少标识解析路径长度;

- 利用 Python 实现了 CNR 下的算力标识解析方法并与 DNS 和 DHT 方法进行了对比分析, 实验结果表明, 采用 CNR 机制进行的算力标识解析方法实现了更短的查找路径和更少的响应时间。

本文的其余部分按如下方式组织。第二节介绍了本文的系统架构。在第三节中, 详细介绍了算网标识解析方法。在第四节中, 展示了性能评估, 并和 DNS 和 DHT 方法进行比较。第五节对本文进行了总结。

## 2 系统架构

算力数据共享是减少算网标识解析延迟的关键, 其中面临的挑战是将数据索引放置在哪里以及如何搜索数据索引, 因此, 一个高效的数据索引机制对算网标识解析服务是非常必要的。在本文中, 提出了一种基于坐标索引方法的算网标识解析机制 CNR。

如图 1 所示, 在 CNR 机制中大量的算力节点被部署到不同的区域。每个区域中的所有算力节点通常由一个区域 DC 管理, 该区域 DC 更多的计算和存储容量。区域 DC 负责管理所有涉及的算力节点, 并调度该区域中的用户请求。在一个区域中可以有多个区域 DC, 它们可以在分层体系结构中扮演相同的角色来保证数据查找的准确性。CNR 机制的主要目标包括快速的算网标识解析服务、可扩展性和低的实现开销。

为了实现 CNR 机制, 控制平面维护一个虚拟的 2D 空间, 并将算力网络映射到虚拟空间, 构建基于地理路由的 DT 图, 在 DT 图上响应数据的查找。控制平面首先从交换平面收集网络拓扑, 然后在第 3 节中计算交换机的坐标。控制平面计算虚拟空间中交换机的坐标, 然后构建 DT 图以连接虚拟空间的这些点, 构成虚拟空间中的算力网络, 数据项的坐标通过散列其标识符得到。在 DT 图上, 数据索引被交换机贪婪转发到与其位置最接近交换机上, 与该交换机相连的算力节点将存储这些数据项。

在 CNR 机制下, 存储了一些共享算力数据的算力节点将这些算力数据索引发布到与其直接相连的区域 DC, 而不是同一区域中的所有其他算

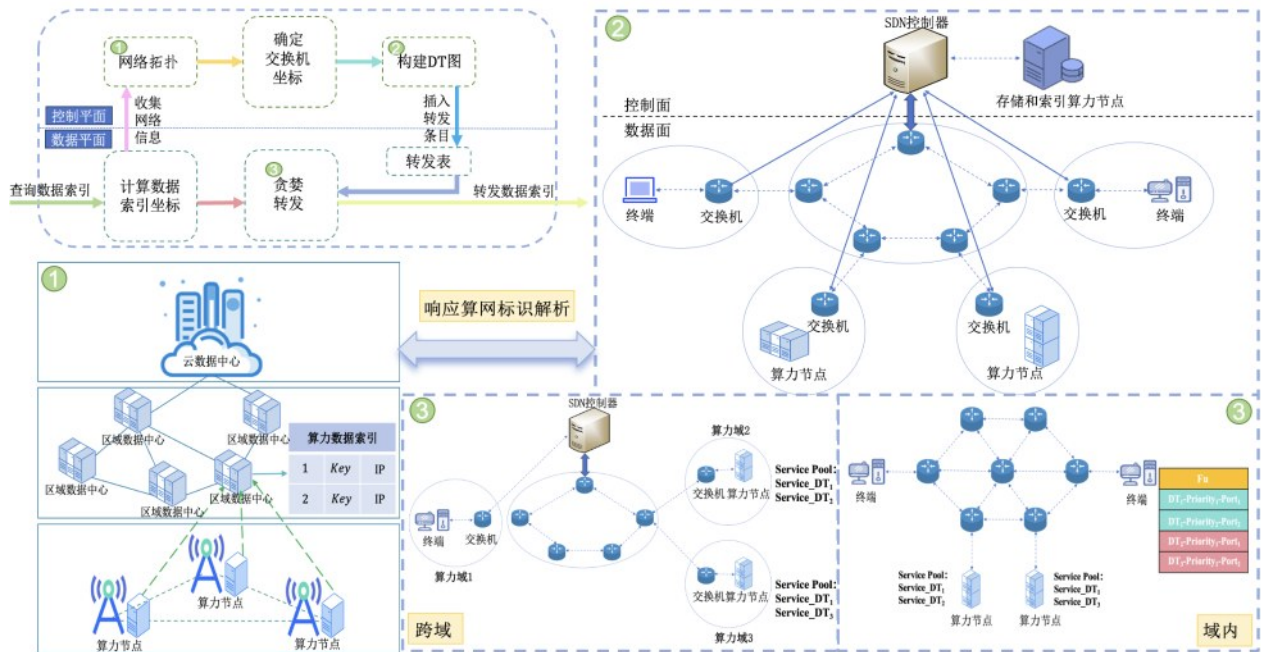


图1 CNR:算网标识解析机制

力节点，这将有效减少带宽消耗。算力数据索引以字典 $\langle key, value \rangle$ 的形式存储，其中键是算力数据的标识符，值是算力节点的IP地址。

具体地，控制平面收集数据平面中物理网络的拓扑信息，生成基于地理路由的DT图，当算力节点发起标识解析请求时，数据平面的交换机将计算数据索引的坐标，并将标识解析请求提交给控制平面，解析请求在DT图上被贪婪转发，按照最短路径原则与该数据项距离最近的交换机直接相连的区域DC对数据检索做出响应，控制平面将虚拟空间中的每一跳决策映射为物理网络中的具体转发端口，交换平面按照控制平面做出的标识解析响应决策进行数据转发，最终到达目的区域数据中心。对于算力域内的标识解析请求，数据检索请求响应直接由与该请求算力节点直接相连的区域DC上，由该区域DC存储的数据项提供标识请求服务；对于跨算力域的标识解析请求，数据检索请求在DT图上被贪婪转发到与该数据项位置最接近的交换机上，由交换机将请求转发到该算力域所在的区域DC上，实现从入

口区域DC到目的区域DC的数据快速定位。

控制平面基于物理实际链路和交换机的拓扑信息构建算力网络的虚拟空间DT图，来响应数据的检索，交换平面基于控制平面做出的索引决策被记录在转发表上的信息来进行数据项的转发。CNR机制基于数据位置和交换机在虚拟空间中的位置贪婪地转发数据请求，交换机位置的确定是实现CNR优势的关键。这是因为不好的虚拟位置会导致算力节点之间的路由路径较长，负载均衡较差。因此，在下一节中，本文首先详细介绍DT图的构建过程。

### 3 基于DT图的算网标识解析

在算力网络中，大量算力数据存储在那些地理上分布的算力节点和远程云中。为了有效地查找这些算力数据，CNR机制构造基于地理路由的DT图，算力数据索引和交换机的坐标在DT图上得以体现，其中交换机的相对位置和物理位置保持一致，在响应标识解析请求时，交换机通过贪婪转发的方式，将查询请求路由到距离查询数

据最近的算力节点上,并且由与该交换机直接相连的区域DC对查询请求做出响应。相比于分布式的数据查询DHT方法,此方法实现了O(1)的DHT,标识解析请求只需从入口算力节点到目的算力节点一个覆盖跳,相比于查询覆盖跳数为 $n$ 的DHT方法响应时间更短、查询路径跳数更小;相比于基于DNS的标识解析方案,数据请求无需采用逐层检索的方式递归查询,通过CNR可以直接获得指示数据位置的数据索引时,进一步减少标识解析延迟。下面将详细介绍DT图是如何构建的,包括如何确定DT图上交换机和数据索引的坐标,以及数据索引的放置和标识解析过程。

### 3.1 确定交换机的坐标

在云DC或区域DC中使用一个或多个控制器来收集区域DC之间的网络状态和链路信息<sup>[9][10]</sup>,该信息用于计算这些交换机之间的标量积矩阵 $\mathbf{S}$ <sup>[11]</sup>,如式(1)给出。假设算力网络中交换机的个数为 $n$ ,则由 $n$ 个交换机即 $s_1, s_2, \dots, s_n$ 之间的测试路径长度构成的标量积矩阵 $\mathbf{S}$ 是 $n \times n$ 维的。

$$\mathbf{S} = \mathbf{R}\mathbf{R}^T \quad (1)$$

其中, $\mathbf{R}$ 表示 $m$ 维空间中 $n$ 个点的 $n \times m$ 坐标矩阵,每个 $n \times n$ 对称矩阵 $\mathbf{S}$ 均可分解为若干矩阵的乘积,如式(1)所示,符号 $T$ 表示矩阵的转置运算。

然后把矩阵 $\mathbf{S}$ 通过构造特征向量的方式进行对角化,如式(2)所示。

$$\mathbf{S} = \mathbf{Q}\mathbf{A}\mathbf{Q}^T \quad (2)$$

其中, $\mathbf{Q}$ 是 $m \times m$ 维标准正交矩阵(即满足 $\mathbf{Q}\mathbf{Q}^T = \mathbf{Q}^T\mathbf{Q} = \mathbf{I}$ ), $\mathbf{A}$ 是对角矩阵。

接下来需要讨论标量积矩阵 $\mathbf{S}$ 的特征值极性。由于 $\mathbf{S}$ 是一个对称矩阵,而实对称矩阵的所有特征值都是实数,因此任意对称矩阵 $\mathbf{S}$ 均可按照式(3)进行分解。

$$\mathbf{S} = \mathbf{P}^T\mathbf{P} \quad (3)$$

并且对于任意一个非零向量,都有

$$\mathbf{x}^T\mathbf{S}\mathbf{x} = \mathbf{x}^T(\mathbf{P}^T\mathbf{P})\mathbf{x} = (\mathbf{P}\mathbf{x})^T(\mathbf{P}\mathbf{x}) = \|\mathbf{P}\mathbf{x}\|^2 \quad (4)$$

此处 $\|\mathbf{P}\mathbf{x}\|^2$ 表示向量 $\mathbf{P}\mathbf{x}$ 的欧几里得范数平方。由于任何实数的平方均为非负值,因此该结果恒为非负数。这意味着对于所有非零向量 $\mathbf{x}$ ,都满足 $\mathbf{x}^T\mathbf{S}\mathbf{x} \geq 0$ 。由此可以判定 $\mathbf{S}$ 是一个半正定矩阵(根据定义,半正定矩阵的所有特征值均为非负数)。因此可得出结论:对称矩阵 $\mathbf{S}$ 的特征值必然非负。基于上述推导,采用 $\mathbf{R} = \mathbf{Q}\mathbf{A}^{\frac{1}{2}}$ 来表示交换机的物理坐标。

基于上述分析,本文设计了一种基于物理网络计算虚拟空间坐标位置的算法(如算法1所示)。算法1借鉴了经典多维尺度分析(Multidimensional Scaling, MDS)的核心思想<sup>[12]</sup>,保持DT图上坐标点的距离与算力网络中交换机距离的一致性。

首先,算法1以最短路径距离矩阵 $\mathbf{D} = [d_{ij}]$ 作为输入,其中每个元素 $d_{ij}$ 表示第 $i$ 个与第 $j$ 个交换机之间的最短路径长度,其中 $i, j = 1, 2, \dots, n$ ,这些信息均由网络控制平面提供<sup>[13][14]</sup>。在计算过程中基于多维尺度分析的理论基础,通过双中心化处理(步骤2)将距离矩阵转换为标量积矩阵 $\mathbf{S}$ 。随后在步骤4中,通过对矩阵 $\mathbf{S}$ 进行特征值分解,并将特征向量与其对应的特征值相乘,最终得到交换机在虚拟空间中的坐标矩阵 $\mathbf{V}$ 。

#### 算法1: 计算DT图上交换机的坐标

因此对于交换机 $s_i$ 和 $s_j$ 来说,通过算法1输出的虚拟坐标 $\{v_i, v_j\}$ 与交换机的物理坐标 $\{r_i, r_j\}$ 满足一个重要特性:虚拟空间中的欧氏距离 $\|v_i - v_j\|^2$ 能够有效逼近实际网络距离 $\|r_i - r_j\|^2$ ,从而保持网络拓扑的空间关系。通过这种方式可以在不同的算力网络规模中得到DT图上坐标点的位置,具有较好的扩展性。其中,需要明确的是,基于MDS的虚拟空间坐标位置确定算法是建立在网络状况相对稳定的基础上的。在算力网络调度决策的时间粒度上,网络动态性可视为有界扰动,基



**输入:** 最短路径矩阵  $D$

**输出:** 交换机的虚拟坐标  $R$

1: 计算  $D^{(2)}$ , 即  $D$  的平方距离矩阵

2: 通过将平方距离矩阵  $D$  与  $M = I - \frac{1}{n}A$  相乘构建标量积矩阵  $S$ , 即  $S = -\frac{1}{2}MD^{(2)}M$ 。其中,  $A$  是所有元素均为 1 的方阵,  $n$  表示交换机的数量

3: 正交化之后的标量积矩阵  $S$ , 包含由特征向量  $\alpha_1, \alpha_2, \dots, \alpha_m$  构成的矩阵  $Q_m$  以及由这些特征向量对应的最大特征值  $\lambda_1, \lambda_2, \dots, \lambda_m$  构成的对角矩阵  $A_m$

4: 通过  $V = Q_m A_m^{-\frac{1}{2}}$  计算交换机在虚拟空间中的坐标, 其中  $Q_m$  由矩阵  $S$  的前  $m$  个特征向量组成,  $A_m$  是由  $S$  的前  $m$  个特征值构成的对角矩阵。

于此前提, 该拓扑模型在算力网络宏观资源优化问题中具有适用性, 保留了网络基础设施的固有时空特征。

### 3.2 完善交换机的坐标

基于算力网络中交换机的物理拓扑得到的 DT 图上坐标点的位置只能初步确定不同坐标点之间的相对位置, 下一步还应该考虑这些坐标点之间的负载平衡问题, 使得标识解析请求被均匀的在坐标点之间转发。

为此, 本文引入了 K-means 聚类算法原理, 通过迭代优化进一步精确调整交换机的空间坐标配置, 使得标识解析请求不会集中转发到某一个坐标点上。在 DT 图上, 每个数据项都被映射为一个具有坐标的位置点。K-means 算法的核心思想是将这些数据点划分到 K 个簇 (对应 K 个交换机), 每个簇内的数据点与所属簇中心的距离之和最小。在初始随机分配的交换机位置情况下, 如图 2-1 所示, 由于各个交换机负责的区域范围大小不一, 当数据项在虚拟空间中呈现均匀分布时, 必然导致不同交换机承载的数据量存在显著差异, 从而造成负载不均衡现象。如图 2-2 所示, 我们可以观察到初始随机选择的 K 个质心 (交换机位置) 与其实际负责区域的几何中心重合时, 可以实现不同的簇直接的负载均衡。通过 K-means 算法的迭代优化过程, 我们可以获得一个接近最优的交换机布局方案——当算法收敛时, 每个交换机的坐标位置恰好等于其所负责数据区域的质心, 此时达到局部最优解状态。从几何意

义上理解, K-means 算法可以被视为在离散数据点集合上实现的质心 Voronoi 剖分 (Centroidal Voronoi Tessellation, CVT) [15]。该算法通过交替执行以下两个步骤实现优化: 首先, 将每个数据点分配给最近的交换机 (质心), 形成 K 个数据簇; 其次, 重新计算每个簇的质心作为新的交换机位置。这一过程不断迭代直至收敛, 最终使得每个交换机的实际位置与其负责区域的质心完全重合。

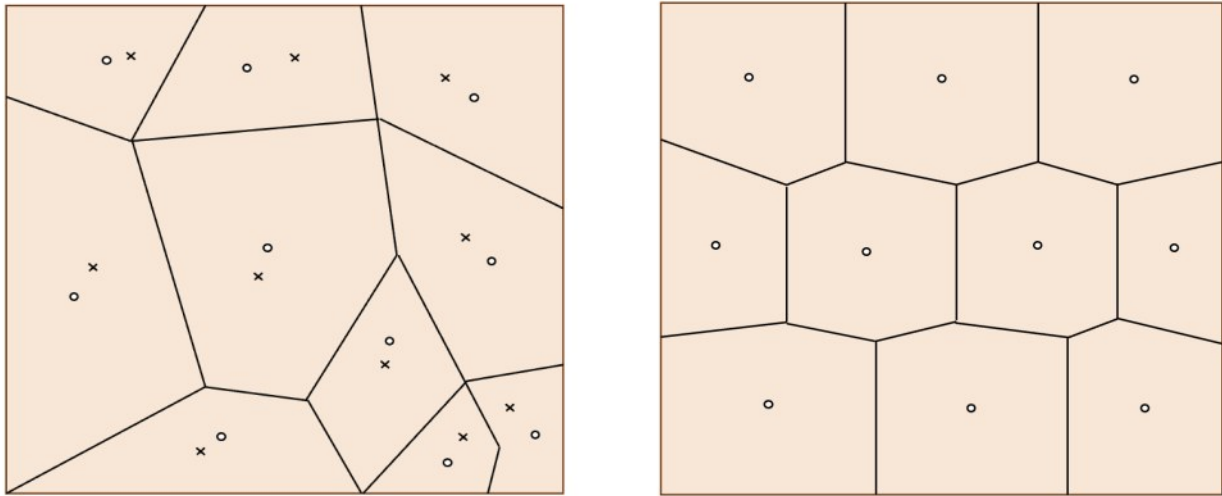
K-means 算法的核心约束条件可表述为: 每个聚类中心必须等于其对应簇内所有数据点的质心。给定一个数据区域  $\Omega$  和定义在该区域上的密度函数  $\rho(r)$ , 对于任意子区域  $R \subseteq \Omega$ , 其质量质心  $r^*$  可定义为:

$$r^* = \frac{\int_R r \rho(r) dr}{\int_R \rho(r) dr} \quad (5)$$

考虑在度量空间  $\Omega$  中布置  $n$  个交换机, 令  $(v_k)_{1 \leq k \leq n}$  表示这些交换机在虚拟空间中的坐标位置, 这些坐标由算法 1 得到, 定义距离函数  $\phi(r, q)$  表示点  $r$  与位置  $q$  之间的度量距离, 则点  $r$  到点集  $P$  的距离可表示为  $\phi(r, P) = \inf \{ \phi(r, q) | q \in P \}$ 。基于此, 我们可以定义与交换机  $v_k$  相关联的数据区域  $R_k$ :

$$R_k = \left\{ r \in \Omega | \phi(r, v_k) \leq \phi(r, v_j), j = 1, 2, \dots, n, j \neq k \right\} \quad (6)$$

区域  $R_k$  包含了  $\Omega$  中所有到交换机  $v_k$  距离不大



(1) 一般分布

(2) 均匀分布

图2 K-means算法原理

于到其他任何交换机距离的数据点集合。在 K-means 算法的框架下，这些区域  $R_k$  实际上构成了数据空间的一种划分，即各个簇的覆盖范围。进一步地，对于定义在  $\Omega$  上的密度函数  $\rho(\bullet)$ ，我们可以建立 K-means 聚类问题的目标函数，即聚类误差平方和：

$$F((v_i, R_i), i = 1, 2, \dots, n) = \sum_{i=1}^n \int_{r \in R_i} \rho(r) |r - v_i|^2 dr \quad \#(7)$$

式(7)度量了所有数据点到其所属聚类中心距离的加权平方和，其最小化过程正是 K-means 算法的核心优化目标。与质心 Voronoi 剖分理论类似，当该目标函数达到最小值时，系统达到一种均衡状态：每个交换机的坐标位置恰好等于其负责数据区域的质心。

基于 K-means 算法的优化原理，本文设计了相应的坐标调节方法，如算法 2 所示，用于进一步优化由初始算法获得的交换机位置配置。该调节算法采用一种有效的采样技术，通过对虚拟空间进行随机采样来近似估计聚类误差函数。定理 1 明确给出了 K-means 目标函数  $F$  最小化的必要条件：对于所有  $1 \leq i \leq n$ ，区域  $R_i$  必须是相对于交换机坐标  $v_i$  的最近邻区域（即 Voronoi 区域），同时交换机坐标  $v_i$  必须是对应区域  $R_i$  的质心。

算法 2: K 调节算法：在实现负载均衡的同时细化虚拟空间中的交换机的坐标要求

定理 1. 聚类误差平方  $F$  达到最小值的必要条件是，对于所有  $1 \leq i \leq n$ ，每个聚类中心  $v_i$  等于其对应簇  $R_i$  的质心，且每个簇  $R_i$  必须包含所有距离该聚类中心最近的数据点。

**输入：** 交换机的坐标  $V$

**输出：** 交换机  $V$  的更新坐标

1: 初始化聚类中心：设置  $V$  为  $V^*$ ；对所有的  $i=1, 2, \dots, n$ ，设置  $j_i=1$ ，其中  $v_i \in V$ ，表示第  $i$  个聚类中心及其权重计数器

2: 数据采样与分配：从由控制平面以等概率构造的拟空间  $\Omega$  中随机抽取 1 000 个点的样本  $W$

3: 最近邻分配：对每个点  $w \in W$ ，找出与最近的聚类中心  $v_i$ ；用  $i^*$  表示该聚类中心的索引

4: 更新聚类中心和对应计数：设置  $v_{i^*} \leftarrow \frac{j_{i^*} v_{i^*} + w}{j_{i^*} + 1}$  和  $j_{i^*} \leftarrow j_{i^*} + 1$ ，其中， $j_{i^*} - 1$  表示聚类中心  $v_{i^*}$  已被更新的次数

5: 如果这组新的聚类中心满足收敛准则（质心移动距离小于阈值，或达到最大迭代次数）则终止；否则，返回步骤 2。



证明：给定一个区域  $\Omega$ ，一个正整数  $n$ ，一个密度函数  $\rho(\bullet)$  定义在  $\Omega$  上，令  $\{v_i\}_{i=1}^n$  表示属于  $\Omega$  的  $n$  个交换机坐标的集合，令  $\{R_i\}_{i=1}^n$  表示  $\Omega$  的任意划分。聚类目标函数定义为：

$$F((v_i, R_i), i=1, 2, \dots, n) = \sum_{i=1}^n \int_{r \in R_i}^0 \rho(r) |r - v_i|^2 dr \quad (8)$$

首先，考察  $F$  相对于某一特定坐标的第一次变化，考虑扰动  $v_j + \epsilon c$ ，其中  $c$  为任意满足  $q_j + \epsilon c \in \Omega$  的向量，则函数变化可表示为：

$$F(v_j + \epsilon c) - F(v_j) = \int_{r \in R_j}^0 \rho(r) \left\{ |r - v_j - \epsilon c|^2 - |r - v_j|^2 \right\} dr \quad (9)$$

当聚类目标函数  $F$  达到最小值时，其关于坐标的一阶变分必须为零。将上式除以  $\epsilon$  并取极限  $\epsilon \rightarrow 0$ ，可以得到最小值的必要条件：

$$v_j = \frac{\int_{R_j}^0 r \rho(r) dr}{\int_{R_j}^0 \rho(r) dr} \quad (10)$$

式 (10) 表明当目标函数最小时，每个交换机的坐标  $v_j$  必须是区域  $R_j$  的质心。接下来，固定交换机的坐标  $\{v_i\}_{i=1}^n$ ，考察区域划分的选择对目标函数的影响。如果将任意划分  $\{R_i\}_{i=1}^n$  替换为 Voronoi 划分  $\{\widehat{R}_j\}_{j=1}^n$ ，两者的目标函数值，

$$F((v_j, \widehat{R}_j), j=1, 2, \dots, n) = \sum_{j=1}^n \int_{r \in \widehat{R}_j}^0 \rho(r) |r - v_j|^2 dr \quad (11)$$

式 (11) 在特定的  $r$  值下，根据 Voronoi 区域的定义，若  $r$  属于对应于  $v_j$  的 Voronoi 域  $\widehat{R}_j$ ，则必须满足：

$$\rho(r) |r - v_j|^2 \leq \rho(r) |r - v_i|^2 \quad (12)$$

当  $r$  到其最近交换机  $v_j$  的距离不大于到任何其他交换机  $v_i$  的距离时，式 (12) 才会成立，因此由于  $\{R_i\}_{i=1}^n$  不是  $\Omega$  的 Voronoi 划分时，在  $\Omega$  的某个可测集合必须满足严格不等式。

$$F((v_j, \widehat{R}_j), j=1, 2, \dots, n) < F((v_i, R_i), i=1, 2, \dots, n) \quad (13)$$

目标函数  $F$  取得最小值时，区域划分  $\{R_i\}_{i=1}^n$  被选择为与交换机坐标  $\{v_i\}_{i=1}^n$  对应的 Voronoi 划分，对应交换机的更新坐标。采用 K 调节算法对交换机的坐标进行优化改变了物理拓扑和虚拟空间的精确映射关系，K 调节算法通过优化节点的空间分布，使得任意节点与所属区域中心的距离尽可能短，且各区域规模均衡，通过局部收缩调整并未改变不同区域节点之间的全局相对位置。K 调节算法微调了区域间的拓扑几何关系，通过优化区域内部的坐标使得区域质心更能代表整个区域的路由坐标，构建了更利于贪婪转发规则的虚拟拓扑几何。

### 3.3 确定算力数据坐标

在算力网络架构中，控制平面负责构建并维护一个与物理网络对应的二维坐标空间  $\Omega$ ，用于实现对全网算力资源与交换设备的统一空间化映射与寻址。标识解析请求检索的数据项以及算力网络中的交换设备被映射到二维坐标空间上，它们被分配在虚拟空间中的唯一坐标来确定在 DT 图上的位置。

针对每一个数据项  $d$ ，对其唯一身份标识进行散列运算获得在 DT 图中的位置。具体实现上，使用 SHA-256<sup>[16]</sup> 散列函数得到长度为 256 位的二进制哈希值。同时为了保持数据项均匀分布以及提高数据项映射效率，截取哈希值的末 64 位（8 字节），并进一步将其解析为两个 32 位二进制数值，记为  $dx$  与  $dy$ 。最后通过归一化处理，得到数据在二维虚拟空间中的标准化坐标：

$$\left( \frac{d_x}{2^{32}-1}, \frac{d_y}{2^{32}-1} \right)$$

数据项的坐标采用十进制形式存储, 每个坐标分量占用 32 位存储空间, 从而唯一地确定算力数据在虚拟空间中的拓扑位置。算力数据索引以键值对  $\square$  标识符, 地址  $\square$  的结构形式分布于各索引算力节点之中。其中, “键” 为数据的唯一标识符, “值” 为该数据实际所在的存储算力节点的网络地址。为了提高标识解析服务效率, 各索引算力节点内部采用高效哈希表结构存储索引数据。通过插入函数  $\text{put}(\text{标识符}, \text{地址})$ , 将索引关系映射到哈希表上; 通过查询函数  $\text{get}(\text{标识符})$ , 则可基于数据标识获取对应存储节点的位置信息。若指定标识符在本地哈希表中不存在映射关系, 函数将返回空值, 表明该索引未存储于当前节点。

### 3.4 发布算力数据索引

在算力网络的虚拟化拓扑构建中, 采用三角剖分图的构造方法保障路由传输的确定性。三角剖分图具有明确的几何特征: 在任意三角形的外接圆范围内不存在其他节点, 如图 3 所示。左图不符合该特征, 因为点  $D$  位于三角形  $\langle A, B, C \rangle$  的外接圆内部; 右图则满足此条件, 构成符合构造原理的三角剖分结构。基于虚拟空间中交换机的坐标分布, 控制平面构建对应的三角剖分图。当图中相邻节点  $v_i$  和  $v_j$  之间缺乏实际物理链路时, 数据无法在二者间直接传输, 此时需借助中间节点进行多跳转发, 此时  $v_j$  称为  $v_i$  的多跳三角邻接节点<sup>[17]</sup>。为实现确定性交付, 系统利用三角剖分图的拓扑属性, 设定转发规则: 每个交换机将数据索引转发至其相邻节点中与目标数据索引坐标最接近的那个交换机, 从而保证路由路径始终朝着目标方向收敛。此时已有基于点集  $\{v_1, v_2, \dots, v_{i-1}\}$  构建的三角剖分图  $DT(v_1, v_2, \dots, v_{i-1})$ , 考虑节点  $v_i$  的插入过程。为将  $v_i$  纳入结构中, 需对原有三角剖分进行局部调整, 从而生成新的三角剖分

$DT(v_1, v_2, \dots, v_i)$ 。调整步骤如下: 首先确定  $v_i$  位于原有三角网格中的哪一个三角形 (或某条边) 内; 随后将  $v_i$  与该三角形的三个顶点相连, 形成三个新三角形 (若  $v_i$  落在边上, 则将其与包含该边的两个三角形的四个顶点相连, 形成四个新三角形)。由于新生成的边以及受影响的原有边可能不满足三角剖分的几何条件, 因此需通过“边翻转”操作使其全部满足要求, 最终得到更新后的  $DT(v_1, v_2, \dots, v_i)$ 。以图 3 中三角剖分  $DT(A, B, C, D)$  为例: 当两个相邻三角形不满足外接圆准则时, 将它们的公共边  $BC$  替换为另一条对角线  $AD$ , 从而生成两个符合三角剖分条件的新三角形。这一替换过程称为“边翻转”。整个三角剖分构建过程的时间复杂度为  $O(n)O(n)$ , 其中  $n$  为网络中交换机的总数。需要注意的是, “边翻转”操作具有严格的局部性, 新节点加入仅触发其直接邻居节点与相关边的拓扑调整, 无需更改网络中其它部分的虚拟坐标与连接关系, 无需进行全局性的拓扑重构。同时, 当节点离开时, 仅其直接邻居节点感知到连接失效, 并触发局部重连接, 在  $DT$  图上与删除节点相关的边会被删除, 然后在邻居节点之间重新建立新连接以维持网络平面的连通性。此过程同样被严格限制在失效节点的原邻域内。这种局部更新特性, 确保了系统扩展的低开销与实现效率。

为确保数据传输的可靠性, CNR 机制基于多跳三角剖分 (Multi-hop Delaunay Triangulation, MDT) 的特性进行路由决策<sup>[18]</sup>: 各交换机将数据索引转发至其邻接节点中, 该节点与数据索引在虚拟坐标空间的位置距离最小。为了实现这一决策, 控制平面依据三角剖分图的连接结构, 在交换机转发表中记录与标识解析路径对应的转发条目。如算法 3 所示, 在三角剖分图中, 交换机  $v_i$  将算力数据索引  $d$  向相邻算力节点  $v_j$  发送的条件是: 该节点  $v_j$  在虚拟空间中的坐标位置比  $v_i$  更接近于数据索引  $d$  的坐标, 其判断准则由式 (14)



图3 DT图的构造

给出。

$$\{|d-r_j| < |d-r_i|, i=1, 2, \dots, n, i \neq j\} \#(14)$$

否则，将数据索引发送到交换机  $v_i$ ，并将算力数据索引直接转发到连接此交换机的区域 DC 中。

为确保 CNR 机制下的数据传输可达性，控制平面首先在虚拟空间内基于所有交换机的坐标位置构建三角剖分图结构。此后，各交换机采用贪心转发策略，将数据索引传输至其相邻节点中坐标与该数据索引坐标最接近的那个邻居节点。如图4所示，左图为十个节点的实际物理连接状况，节点 A 与 E 间缺乏直连链路，无法直接进行消息投递；右图则展示了这十个节点所形成的多跳三角化网络，节点 A 至 E 通过左图中的多跳路径  $\langle A, D, E \rangle$  实现数据包转发，因此节点 E 被视为 A 的多跳三角邻接节点。通过多跳数据传送，保证

标识解析请求在 DT 图上的可靠传送。

在此基础上，基于三角剖分图的贪心路由机制可实现确定性数据传送。具体而言，给定目标位置  $z$ ，数据包传输过程始终在所有节点中距离  $z$  最近的节点处终止，从而保证数据最终抵达目标位置。

然而，一个关键挑战是如何确保每个交换机均能有效向其三角剖分邻居传输数据分组。这是因为在三角剖分拓扑中定义的邻居关系，这种邻接关系在算力网络物理拓扑结构中不一定存在。未与索引算力节点直接相连的交换机不参与三角剖分的构建过程，它们仅作为中继节点，协助将数据索引传递至相应的三角剖分邻居。如图5所示，假设在虚拟空间中交换机  $s_1$  的三角剖分邻居为  $s_3$ ，但在实际数据平面中， $s_1$  与  $s_3$  之间并无直接物理链路。因此，为实现确定性传递，每个交

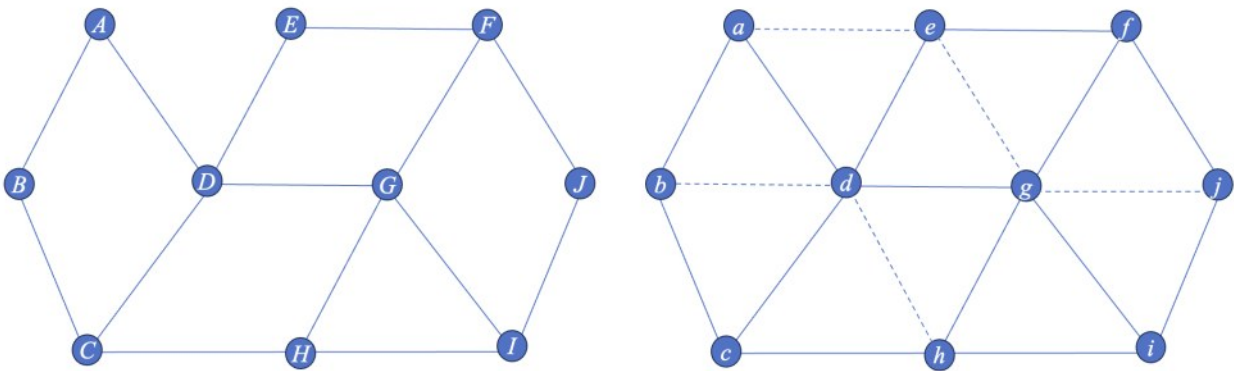


图4 MDT图的构造

交换机必须配置双转发机制：其一将数据包转发至物理邻接节点，其二则负责将请求转发至三角剖分邻居。未直连至索引算力节点的交换机不参与三角剖分结构的构建，仅作为中继节点将数据索引进一步在DT图上传送。

交换机在控制平面维护的虚拟空间中被分配相应的坐标。每个交换机具备自身坐标、其物理邻接节点的坐标以及其三角剖分邻接节点的坐标。任意两个算力节点之间的欧氏距离可通过它们的坐标计算得出，该距离已在前述算力网络嵌入过程中完成映射。在交换机（例如  $s_x, x=1, 2, \dots, n$ ）上实施贪心转发：对于标识为  $d$  的数据，其存储位置由哈希值  $H(d)$  决定，该哈希值将被转换为虚拟空间中的坐标。 $s_x$  会将数据包转发至其三角剖分邻接节点中距离  $H(d)$  坐标最近的那台交换机。若该邻居为物理邻接节点，则直接转发

数据包；否则，将通过虚拟链路将该数据包转发至该最邻近的三角剖分邻居。如果不存在比  $s_x$  自身更接近  $H(d)$  坐标的邻居节点，则证明  $s_x$  即为距离  $H(d)$  最近的交换机，此时数据包转发至目标区域。当数据到达最接近  $H(d)$  坐标的交换机后，该交换机会确定用于实际存放数据的唯一对应交换机。具体算法流程在算法3中详细阐述。

算法3: 发布算力数据索引

3.4 查询算力数据索引

当查询算力数据索引时，数据解析请求首先会被转发到DT图上与该数据项位置最近的交换机上，然后与该交换机直接相连的区域DC会响应数据解析请求。如图6所示，当与交换机  $s_1$  直接相连的区域DC想查询数据  $d$  的数据索引时，在DT图中，查询请求被  $v_1$ （交换机  $s_1$  对应的虚拟空间坐标）贪婪转发，按照  $\langle v_1, v_3, v_7, v_{11} \rangle$  的转

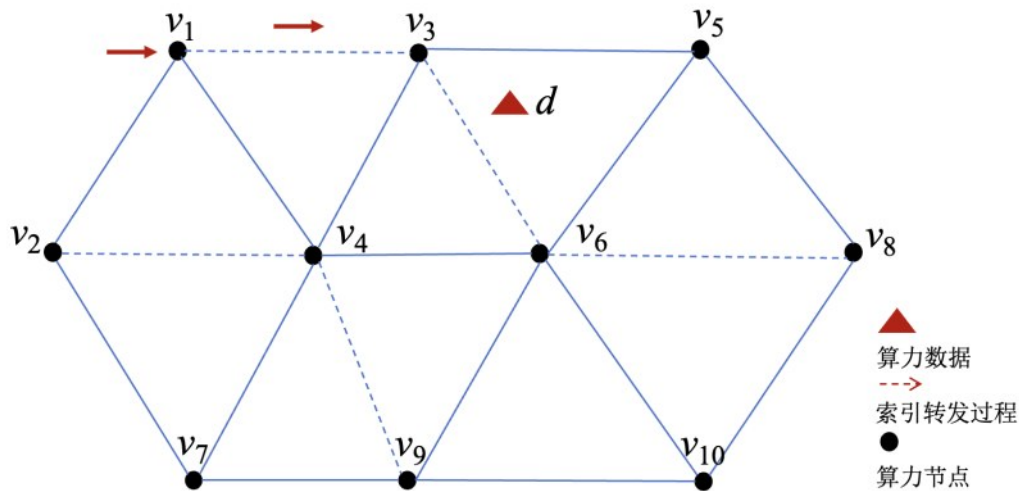


图5 转发数据索引流程

- 
- 1: 在DT图中，对于任意交换机  $v_i$ ，它与算力数据  $d$  的欧氏距离用  $\text{Dis}(v_i, d)$  表示，并且  $v_i$  在DT图上的临接节点集合用  $\{v_x\}$  表示
  - 2:  $\text{Dis}(v_i, d) = \min \text{Dis}(\{v_x\}, d)$
  - 3: **If**  $\text{Dis}(v_i, d) < \text{Dis}(v_j, d)$  **then**
  - 4: 将算力数据  $d$  从  $v_i$  转发到  $v_j$
  - 5: **AND**
  - 6:  $v_i$  将算力数据索引转发到与其直连的区域DC
  - 7: **Else**
  - 8:  $v_i$  将算力数据索引转发到与其直连的区域DC
  - 9: **End if**
-



发路径定位到交换机  $s_{11}$ ，即  $s_{11}$  在虚拟空间中的坐标  $v_{11}$  与算力数据  $d$  的距离最小，然后与  $v_{11}$  直接相连的区域 DC 对此次算力数据索引的查询做出响应，返回指示算力网络中的数据位置的对应数据索引，最后，数据请求者可以使用某种路由方案来检索数据，完成标识解析服务。

在算力网络中，待检索数据项可能会存在多个数据副本，存储在地理位置分散的算力节点中，以此来响应不同位置的标识解析请求。在 DT 图上，针对存在多个数据副本的情况需要对 CNR 机制进一步优化。

针对算力数据项，通过散列其标识符得到其在虚拟空间中的坐标，当存在多个共享数据项时，因为散列值相同，数据会被存储在相同的索引算力节点中。出于负载均衡和容错性考虑，CNR 机制通过设计多副本数据检索情况将共享数据项存储在算力网络的不同算力节点中，来提高 CNR 机制的标识解析效率。具体地，针对数据项  $d$  的第  $i$  个数据副本，通过将数据项的标识符  $H(d)$  与  $i - 1$  进行拼接，再进行散列得到  $H(H(d) + i -$

1) 以此得到第  $i$  个数据副本在 DT 图上的位置。

当算力网络中存在多个数据副本时，标识解析请求会从入口算力节点距离最近的位置检索数据。在 DT 图上，算力网络的物理拓扑结构被映射到虚拟空间中，坐标点的相对距离与物理网络保持一致。当网络中存在多个数据副本时，在 DT 图上这些数据项被转发到距离这些数据项坐标位置最近的交换机上，并存储在直接连接该交换机的数据中心上。

在查询数据索引时，无需遍历所有索引副本，在 DT 图上标识解析请求会被转发到距离该节点最近的数据副本处。如算法 4 所示，与入口区域 DC 直接相连的交换机将标识解析请求转发到最近的数据副本处，存储该数据副本的目的区域 DC 对数据检索请求作出响应。因此，在 CNR 机制框架下能够高效选取路径最短的最优索引副本，从而实现快速的标识解析服务。

算法 4: 选择最优索引副本

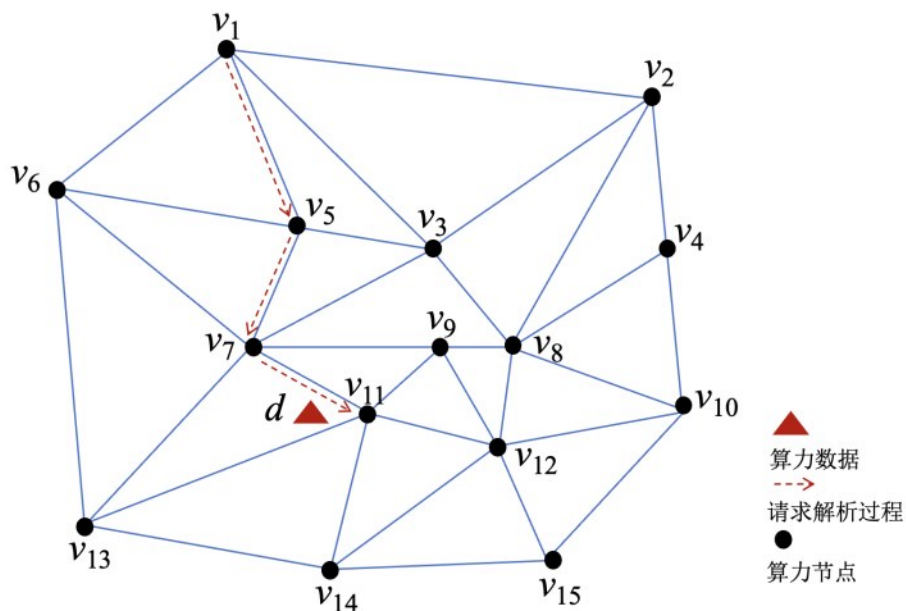


图6 查询数据索引过程

**输入:** 索引副本数  $l$ , 数据索引  $d$ , 与入口区域 DC 直接连接的交换机  $s_j$

**输出:** 最佳索引副本的标识符  $H(d) + i - 1$

1: 对于每个索引副本  $j$ ,  $R_j = \text{Dis}(s_j, H(d) + (i - 1))$  表示虚拟空间中从  $s_j$  到  $H(d) + (i - 1)$  的欧几里德距离

2:  $R_i = \min\{R_j\}, 1 \leq j \leq l$

3: 入口区域 DC 将标识符  $H(d) + i - 1$  传送到交换机  $s_j$ 。

## 4 性能评估

在本节中, 评估基于坐标索引的算网标识解析机制的性能, 性能指标包括查找请求的路径长度和 CNR 机制的响应时间。在进行算网标识解析时, 依据 CNR 机制维护的由算力网络在虚拟空间中的映射位置以及被查询的数据项构成的 DT 图进行数据项的发布与检索。如图 7 所示, 本文搭建的网络拓扑由交换机和算力节点组成。本文基于网络拓扑实现了 DNS、DHT 和本文的 CNR 机制, 并进一步比较了三种不同标识解析机制的性能。本文实现了 CNR 机制, 包括交换平面和控制平面功能, 控制平面中的函数用 Python 编写。

在大规模模拟中, 本文采用基于 Waxman 模型的 BRITE 工具生成层次化拓扑, 从真实网络拓扑采样的子图作为物理网络拓扑的基础。拓扑规模下交换机节点构成网络的骨干, 每个骨干节点下挂载的算力节点形成一个算力域。Waxman 模型的参数设置如下: 链路概率系数  $\alpha = 0.15$ , 短边偏好系数  $\beta = 0.2$ , 以生成与广域网特性相近的、具有一定局部聚集性的拓扑。链路带宽采用分层配置: 核心交换机互联链路为 100 Gbps, 算力节点接入链路为 25 Gbps。传播时延由节点在  $1\ 000\ \text{km} \times 1\ 000\ \text{km}$  地理区域内的坐标决定, 按光在光纤中的速度 ( $2.05 \times 10^5\ \text{km/s}$ ) 计算。在链路上施加随机泊松背景流量, 链路利用率  $\rho$  在 5% - 70% 范围内调节; 实验将  $\rho \leq 35\%$  定义为稳定状态, 此时排队时延方差低于传播时延的 15%。排队时延基于 M/G/1 模型计算, 交换机处理时延固定为  $5\ \mu\text{s}$ 。

在查找算力数据索引的过程中, 采用了一种

基于距离优化的方法来提高数据检索的效率。具体来说, 当需要检索算力数据索引时, 该索引会被首先发送到虚拟空间中与目标算力数据索引地理距离最近的交换机。这一步骤是基于 DT 图上的坐标来实现的, 确保了标识解析的准确性和高效性。接下来, 这个交换机会将算力数据索引进一步转发到与之直接相连的区域 DC。这个区域 DC 负责处理和响应对特定算力数据索引的查询请求。这样的设计允许利用数据共享的优势, 即通过减少数据传输的距离来降低延迟, 并提高响应速度。

### 4.1 混合数据共享框架评估

本节对混合数据共享框架进行性能评估, 重点分析查找路径长度与请求响应时间两项指标。其中用平均查找路径长度衡量每次标识解析请求成功时在逻辑覆盖网中经过的跳数均值, 为网络内所有算力节点的查找路径的平均值; 用平均响应时间衡量标识解析请求的时延性能, 即从请求发出到响应返回的端到端物理时延均值, 所有结果均在 95% 置信区间下重复实验 10 次取平均。混合数据共享包括域内和跨域部分, 域内采用最短路径路由, 跨域采用跨数据中心混合路由。对比方案包括: (1) 基于坐标索引 (CNR) 的贪婪转发机制; (2) 基于分布式哈希表 (DHT) 的 finger 表查询方案; (3) 基于域名系统 (DNS) 的递归查询方式。

首先评估不同区域规模下各方案的查找路径长度。实验设置区域数量从 10 到 100, 每个区域包含 10 个算力节点。图 8-1 表明, CNR 方案的路径长度显著低于 DHT 与 DNS 方案。在 DHT 方案中, 跨域查找跳数随区域数增加而显著上

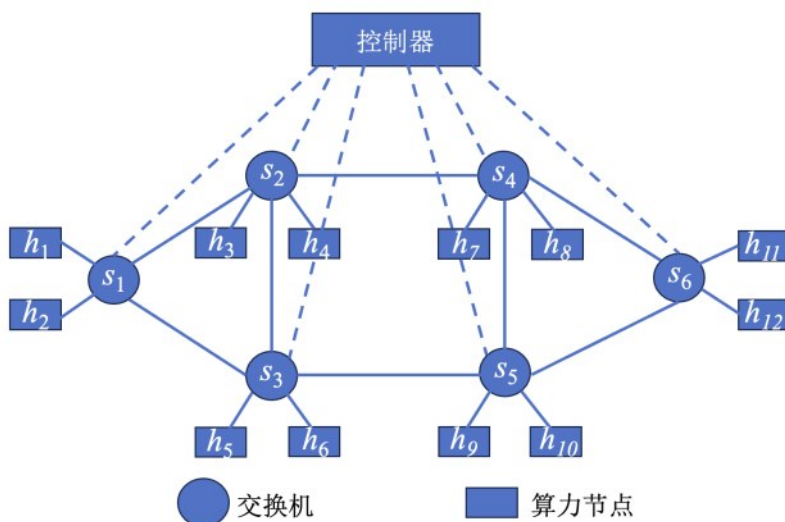


图7 网络拓扑示意图

升；而在 DNS 方案中部分标识解析请求需绕行远程云节点，导致路径延长。

其次考察单个区域内算力节点数量对路径长度的影响（区域数固定为 100，节点数由 10 增至 100）。其次为验证混合框架中跨域路由机制的选型合理性，本文比较了分别集成 CNR、DHT、DNS 作为跨域路由组件时，系统跨域查询路径长度随区域内算力节点数量增长的变化（区域数固定为 100，节点数由 10 增至 100）。如图 8-2 所示，DNS 与 DHT 方案的路径长度随节点数增加明显上升，而 CNR 方案解析请求路径长度受节点数量的影响较小。这是因为 CNR 基于本地拓扑进行单跳转发，避免了 DHT 的多跳查询与 DNS 的集中式绕行。

最后分析区域数量对请求响应时间的影响。随着区域数增加，CNR 的响应时间保持稳定且明显低于 DHT 与 DNS 方案。DHT 需经  $\log(n)$  跳覆盖网络转发，DNS 则依赖集中式云节点，均导致延迟显著上升。此外，DHT 与 DNS 方案还存在交换机负载不均的问题，进一步影响响应效率。

因此，通过以上分析得出结论，不论是在算力节点数量增加还是区域数目的增加的影响下，和 DNS 和 DHT 标识解析方法相比，基于 CNR 的

方案实现了更短的路径长度和更少的响应时间。

#### 4.2 索引机制评估

下面本文比较了基于不同标识解析方案下索引机制的性能指标。首先进行不同算力节点数量下平均查找路径的长度比较。为评估本文提出的完整 CNR 标识解析机制的综合性能，将其与独立的 DNS、DHT 方案进行对比，测量全局平均查找路径长度随总网络规模（区域内算力节点数量）增长的变化。如图 9-1 所示，本文的 CNR 机制实现的平均路径长度接近 DNS 实现的平均路径长度，并且明显短于 DHT 实现的平均路径长度。这是因为采用 DHT 进行标识解析的方法依赖于网络中的节点分布和节点间的连接关系，算力节点数量的增加意味着节点间的距离可能增加，从而导致路径长度的增加。因此当算力节点数量增加时，在基于 DHT 的方案下，路径长度的明显增加，而在 CNR 机制下基于算力标识的算网标识解析方法，算力节点的数量对平均查找路径长度有适度的影响，这是因为在网络规模变大时，DT 图中节点数也会增加，进而带来查找路径的增加。

其次，本文比较了基于不同标识解析方案下索引机制用于检索数据索引的响应时间。图 9-2

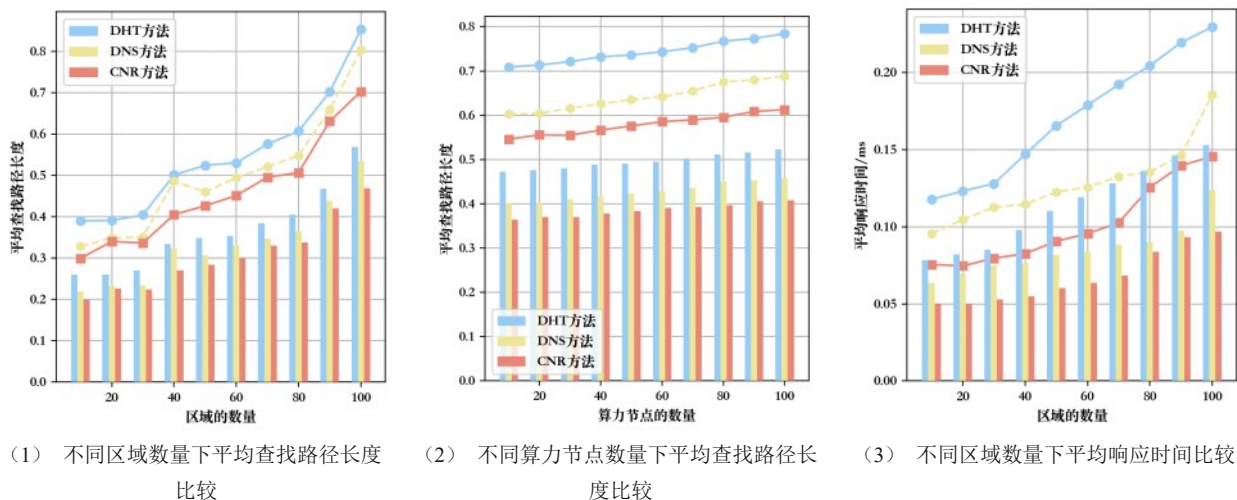


图8 在不同网络规模下CNR、DNS和DHT性能比较

示出了在不同索引机制下每个交换机的平均转发时间。如图9-2所示，与其他两种标识解析机制相比，本文的CNR机制在交换机中实现的响应时间更少。在本文所提出的CNR机制中，响应标识解析请求的时延仅取决于当前交换机的邻接节点数量，具有稳定的局部计算特性。相比之下，在基于DNS和DHT的架构下，响应时间随着算力网络规模的扩大而显著增加，其性能受全局负载与网络状态的影响更为明显。

同时，本文评估了数据副本对CNR机制性能的影响。本文在网络中存储了1 000个数据项，数据大小从10 KB到10 MB不等。然后，本文随机生成一些数据请求，并测试在不同索引机制下检索数据索引和数据副本的延迟。图9-3显示由于CNR机制可以基于虚拟空间中的交换机和索引副本的坐标快速地选择最优的索引副本，因此在三种标识解析机制中检索索引副本的平均延迟最小。本文可以发现，在图9-3中，DNS机制和CNR机制之间的差距很小。这主要是因为本文的网络规模很小。此外，本文还测试了多个数据副本对检索数据的延迟的影响。如图9-3所示，在三种机制中，CNR机制在不对所有数据副本进行采样的情况下检索数据的平均等待时间最短。在

图9-3中，DNS和DHT的数据的平均等待时间非常接近，因为它们都在获得数据索引之后使用最短路径路由来检索数据。

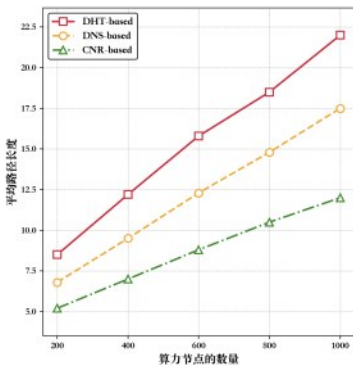
### 4.3 大规模模拟

本文采用交换机级合成拓扑进行评估，每台交换机连10个算力节点。控制平面负责在虚拟空间中构建并维护DT图，虽然DT图的计算过程会带来一些时间开销，但是DT图构建之后，各交换机可基于本地信息独立执行贪婪转发，无需与控制平面交互，从而显著降低控制平面的负载。

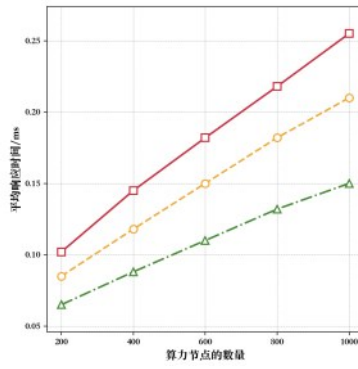
为验证CNR机制的可扩展性，实验在交换机数量为10至100的大规模网络上进行，对应算力节点规模从100到1 000。结果表明，CNR机制具备与软件定义网络相当的可扩展能力，且网络规模越大，其性能优势越明显。

首先评估不同标识解析机制下数据索引对单数据副本检索的平均路径长度。图10-1显示，CNR与DNS机制的平均路径长度相近，均显著低于DHT机制。随着交换机数量增加，DHT的路径长度明显上升，而CNR与DNS机制增长较慢。

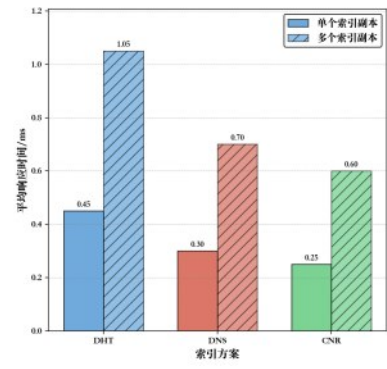
进一步评估每个共享数据包含三个索引副本时的性能表现。实验结果如图10-2所示，其趋势



(1) 不同算力节点数量下平均查找路径长度比较



(2) 不同算力节点数量下平均响应时间比较



(3) 不同副本数量下平均响应时间比较

图9 不同索引机制的性能比较

与单副本情况基本一致：CNR机制的平均路径长度接近DNS机制，并明显短于DHT机制。并且，DNS作为一种集中式索引机制，在容错性与可扩展性方面存在局限。

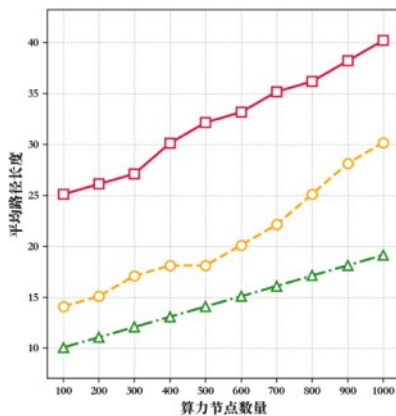
图10-3表明，增加索引副本可缩短各机制下的检索路径，其中DHT机制的改善幅度最为明显。尽管如此，DHT的整体路径长度仍高于CNR与DNS。具体数据表明，在单副本与三副本情况下，CNR机制的路径长度分别比DHT机制短57%，比DNS机制短33%。

综上所述，CNR机制在保持分布式优势的同时，能够实现接近集中式DNS机制的检索效率，

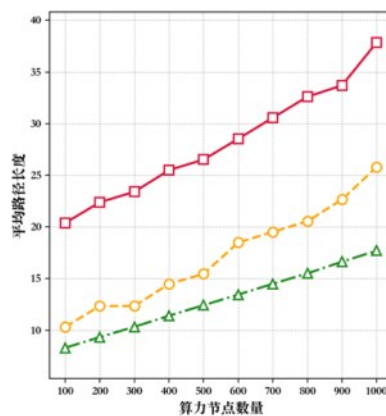
且显著优于分布式哈希表方案，适用于大规模算力网络环境。

### 5 总结

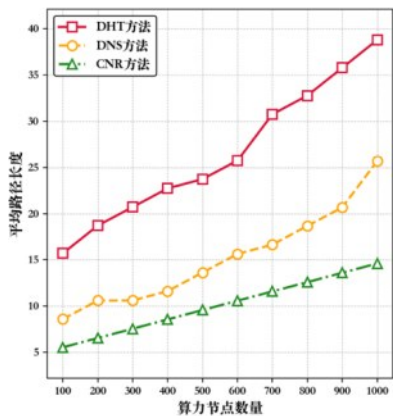
本文基于算力网络整合异构算力资源的背景，针对新兴应用的低延迟需求，提出了一种低延迟的算网标识解析机制CNR。本文设计一种基于坐标索引和地理路由DT图的高效算网标识解析服务。通过在DT图上映射数据索引与交换机坐标，在CNR机制维护的虚拟空间上响应任意数据查询请求，从而显著优化了标识解析过程。实验表明，与现有的解决方案相比，采用CNR机制



(1) 单数据副本索引检索下性能比较



(2) 三个数据副本索引检索下性能比较



(3) 多个数据副本索引检索下性能比较

图10 大规模网络中不同索引机制性能比较

进行的算网标识解析缩短了 41.61% 的查找路径和 57.19% 的响应时间。CNR 机制为下一代 DNS 向软件定义方向发展提供了切实可行的技术支撑,对推动算力服务向低延迟、高性能方向优化具有重要理论与应用价值。

参考文献:(参考文献格式参照 2015 年新标准 GBT 7714-2015)

[1] Abkenar F S, Ramezani P, Iranmanesh S, et al. A survey on mobility of edge computing networks in IoT: State-of-the-art, architectures, and challenges[J]. *IEEE Communications Surveys & Tutorials*, 2022, 24(4): 2329-2365.

[2] 姚惠娟,陆璐,段晓东. 算力感知网络架构与关键技术[J]. *中兴通讯技术*, 2021, 27(03): 7-11.

YAO Y H, LU L, DUAN X D. Computing-aware network architecture and key technologies[J]. *ZTE Technology Journal*, 2021, 27(3): 7-11.

[3] Z. Kang, S. Jieli, W. U. Meixi. Research on heterogeneous computing power network scheduling [J]. *Information and Communications Technology and Policy*, 2023, 49(5): 42.

[4] ZENG Y, HUANG Y, LIU Z, et al. Joint online edge caching and load balancing for mobile data offloading in 5G networks[C]//*Proceedings of 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. Piscataway: IEEE Press, 2019: 923-933.

[5] XIE J, QIAN C, GUO D, et al. Efficient data placement and retrieval services in edge computing[C]//*Proceedings of 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. Piscataway: IEEE Press, 2019: 1029-1039.

[6] YI S, HAO Z, QIN Z, et al. Fog computing: Platform and applications[C]//*Proceedings of 2015 IEEE HotWeb*. Piscataway: IEEE Press, 2015:

73-78.

[7] Tehrani P F, Osterweil E, Schmidt T C, et al. A security model for web-based communication [J]. *Communications of the ACM*, 2024, 67(10): 83-90.

[8] Tahir H, Abid S A, Ali A. Scalable and resilient routing for in-band SDNs with a distributed hash table-based approach[J]. *International Journal of Communication Systems*, 2025, 38(16): e70248.

[9] Ansah F, Olaya S S P, Krummacker D, et al. Controller of controllers architecture for management of heterogeneous industrial networks[C]//*Proceedings of IEEE 16th International Conference on Factory Communication Systems (WFCS)*. Piscataway: IEEE Press, 2020: 1-8.

[10] Ahmad S, Mir A H. Scalability, consistency, reliability and security in SDN controllers: A survey of diverse SDN controllers[J]. *Journal of Network and Systems Management*, 2021, 29(1): 1-59.

[11] Duursma I, Wang H P. Multilinear algebra for minimum storage regenerating codes: A generalization of the product-matrix construction[J]. *Applicable Algebra in Engineering, Communication and Computing*, 2023: 1-27.

[12] F. Wickelmaier. An introduction to MDS [J]. *Aalborg University Department of Communication*, 2003(1).

[13] Babarczy P. Resilient control plane design for virtual software defined networks[J]. *IEEE Transactions on Network and Service Management*, 2021, 18(3): 2557-2569.

[14] Saeed N, Nam H, Haq M I U, et al. A survey on multidimensional scaling[J]. *ACM Computing Surveys*, 2018, 51(3): 1-25.

[15] Du Q, Faber V, Gunzburger M. Centroidal Voronoi tessellations: Applications and algorithms

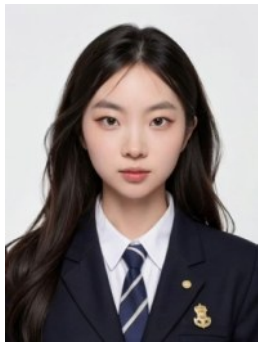


[J]. SIAM Review, 1999, 41(4): 637-676.

[16] KAMMOUN M, ELLEUCHI M, ABID M, et al. FPGA-based implementation of the SHA-256 hash algorithm[C]//Proceedings of 2020 IEEE International Conference on Design & Test of Integrated Micro & Nano Systems (DTS). Piscataway: IEEE Press, 2020: 1-6.

[17] BOISSONNAT J D, DYER R, GHOSH A. Delaunay triangulation of manifolds[J]. Foundations of Computational Mathematics, 2018, 18: 399-431.

[18] Lam S S, Qian C. Geographic routing in d-dimensional spaces with guaranteed delivery and low stretch[J]. IEEE/ACM Transactions on Networking, 2013, 21(2): 663-677.



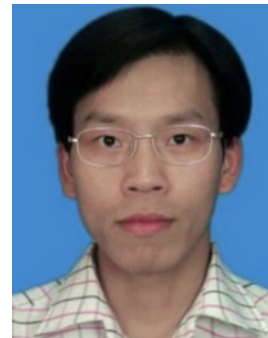
张月 (2000-), 女, 北京邮电大学硕士, 主要研究方向为算力网络标识解析和下一代DNS。

谢人超 (1984-), 男, 博士, 北京邮电大学教授、博士生导师, 主要研究方向为信息中心网络、工业互联网、算力网络、边缘计算、数字孪生。

唐琴琴 (1994-), 女, 博士, 北京邮电大学副研究员、硕士生导师, 主要研究方向为算力网



络、网络人工智能、网络孪生等。



黄韬 (1980-), 男, 博士, 北京邮电大学教授、博士生导师, 主要研究方向为路由与交换、软件定义网络、内容分发网络、确定性网络、算力网络等。