面向无状态计算服务的算力网络技术研究

张岩[1,2]，王立文[1,2]，曹畅[1,2]，唐雄燕[1,2]
（1. 中国联合网络通信有限公司研究院，北京 100048；2. 下一代互联网宽带业务应用国家工程研究中心，北京 100048）

摘　要：目前，信息技术（IT）领域的服务网格技术在跨域场景中缺乏与网络的协同调度能力，通信技术（CT）领域的算力网络技术聚焦于资源层调度和网络层解决方案，在面向无状态计算服务的算网协同与路由调度需求时，存在算力资源调度与用户应用层计算功能需求割裂、网络层连接方案与服务化应用程序编程接口（API）互联需求协议栈错位等问题。分析了典型算力业务对算力网络的差异化需求以及无状态计算服务的调度互联要求，提出了一种支持无状态计算服务路由调度的新型算力网络架构，给出了总体架构和关键技术分析。仿真结果表明，所提方案采用URL-PATH作为算力标识构建服务语义路由平面，可实现无状态计算服务的API粒度算网资源协同调度与差异化服务质量（QoS）保障能力提升，为算力网络从“资源供给”向“服务互联”演进提供了新的技术路线。

关键词：算力网络；无状态计算服务；模型即服务
中图分类号：TN92
文献标志码：A
DOI：10.11959/j.issn.1000–436x.2025160

Research on computing power network technology for stateless computing services

ZHANG Yan[1,2], WANG Liwen[1,2], CAO Chang[1,2], TANG Xiongyan[1,2]
1. China Unicom Research Institute, Beijing 100048, China
2. National Engineering Research Center of Next Generation Internet Broadband Service Application, Beijing 100048, China

Abstract: Currently, in the information technology (IT) field, service mesh technologies lack collaborative scheduling capabilities with networks in cross-domain scenarios. In the communication technology (CT) field, computing power networks focused on resource-layer scheduling and network-layer solutions. When addressing the requirements of computing-network collaboration and routing scheduling for stateless computing services, issues such as the disconnection between computing resource scheduling and user application-layer functional demands, as well as protocol stack mismatches between network-layer connectivity solutions and service-oriented application program interface (API) interconnection requirements, remain unresolved. An analysis was conducted on the differentiated requirements of typical computing services for computing power networks and the scheduling and interconnection demands of stateless computing services. A novel computing power network architecture was proposed to support routing and scheduling for stateless computing services, with the overall architecture and key technologies outlined. Simulation results demonstrate that the proposed scheme uses URL-PATH as computing service identifiers to construct a service semantic routing plane, enabling API-granular computing-network resource collaborative scheduling and enhanced differentiated quality of service (QoS). This approach provides new insights for evolving computing power networks from “resource provisioning” to “service interconnection”.

Keywords: computing power network, stateless computing service, model-as-a-service

收稿日期：2025–05–26；修回日期：2025–09–02
通信作者：王立文，wanglw97@chinaunicom.cn

0　引言

近年来，人工智能（AI, artificial intelligence）技术快速发展，尤其是生成式AI与大模型的成熟和应用，推动了模型调用服务成为数字经济的核心基础设施。如图1所示，模型调用服务是模型即服务（MaaS, model as a service）[1-2]的一种业务形态，用户不需要关注底层算力资源或模型部署细节，仅需通过应用程序编程接口（API, application program interface）调用即可获得AI推理服务。这种方式降低了AI技术的应用门槛，提高了技术效率，重构了AI产业分工模式，已覆盖人脸识别、智能问答、智能体等数百个场景[3-4]，尤其在高阶模型领域得到了广泛应用，如DeepSeek、ChatGPT等均支持通过API方式为用户提供计算服务。这一趋势标志着算力服务正从传统资源供应模式向轻量化的功能调用范式转变。

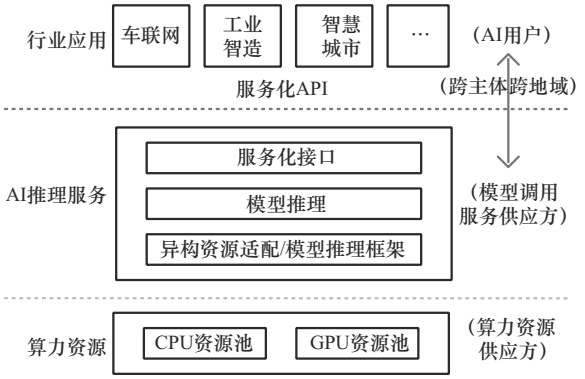


图1　模型调用服务的应用模式

无状态化设计是分布式计算领域的核心设计范式，其服务端不维护任何与客户端相关的会话状态信息。每个客户端的每次请求被视为独立的操作单元，服务端仅依据当前请求参数执行计算并返回结果。AI模型调用的计算过程是根据预先设计的算法和参数对用户输入数据进行处理得到结果，不会缓存中间数据，每次计算过程都是独立的，天然具有无状态性，符合无状态化设计理念。无状态化设计可以显著降低组件间的集成难度，提升系统的可扩展性，已在云计算、微服务架构等场景中得到了广泛应用[5]。传统无状态计算服务常用于同一集群内微服务间的松耦合设计，但AI类的无状态计算服务相较于传统无状态计算服务具有计算量大、供需跨主体等特点，服务和用户通常位于不同的数据中心，传统单体数据中心架构和静态资源分配模式已难以满足其高效、低时延和高并发的AI服务需求，在调度和互联时，跨域导致的算网状态协同、资源优化、按需连接保障等问题成为新的挑战。

算力网络[6-14]作为融合算力资源与网络技术的新型基础设施，正成为学术界与产业界的核心研究方向。其核心理念是将分布式的异构算力动态抽象为可全局调度的资源池，并基于网络拓扑与服务质量（QoS, quality of service）需求，实现算网资源的最优分配与协同计算。这一模式打破了传统云计算中心化架构的局限性，为跨域算力协同场景提供了低时延和高弹性的算力供给范式。然而，目前业界对算力网络的研究重点主要集中在面向智算训推的GPU资源供给等方面，而对于无状态计算服务的跨域调度与互联研究，仍主要采用网络层方案。这种方案通过网络地址标识算力服务，与实际基于HTTP的算力服务传输协议不匹配，存在路由设计受限、用户需求信息携带难等问题。

本文针对无状态计算服务的跨域调度互联需求，提出了一种原生支持无状态计算服务路由调度的新型算力网络方案。其核心创新在于将无状态计算服务的业务范式深度融入算力网络协议栈，通过定义适配服务化API的算力标识与服务路由机制，在网络层与应用层之间构建独立的逻辑平面，实现计算服务与网络地址的解耦。该方案支持客户端通过服务化API访问无状态计算服务，提供算力服务请求的解析、路由调度与QoS保障，实现基于服务语义的细粒度跨域调度互联。在不影响用户使用体验的前提下，实现跨层、跨域计算服务供需间的动态调度与协同优化，推动算力网络从资源供给向服务互联演进。

本文主要的研究工作如下。

1) 分析了不同算力服务对网络的差异化需求，明确了无状态计算服务对算力网络的调度和互联要求。

2) 提出了一种面向无状态计算服务的算力网络技术方案，阐述了总体架构和关键技术，并进行了仿真实验验证。

3) 论述了本文方案的应用场景及技术优势。

1　相关研究工作

服务网格与算力网络是实现无状态计算服务跨

域调度互联的研究基础。

1.1　服务网格

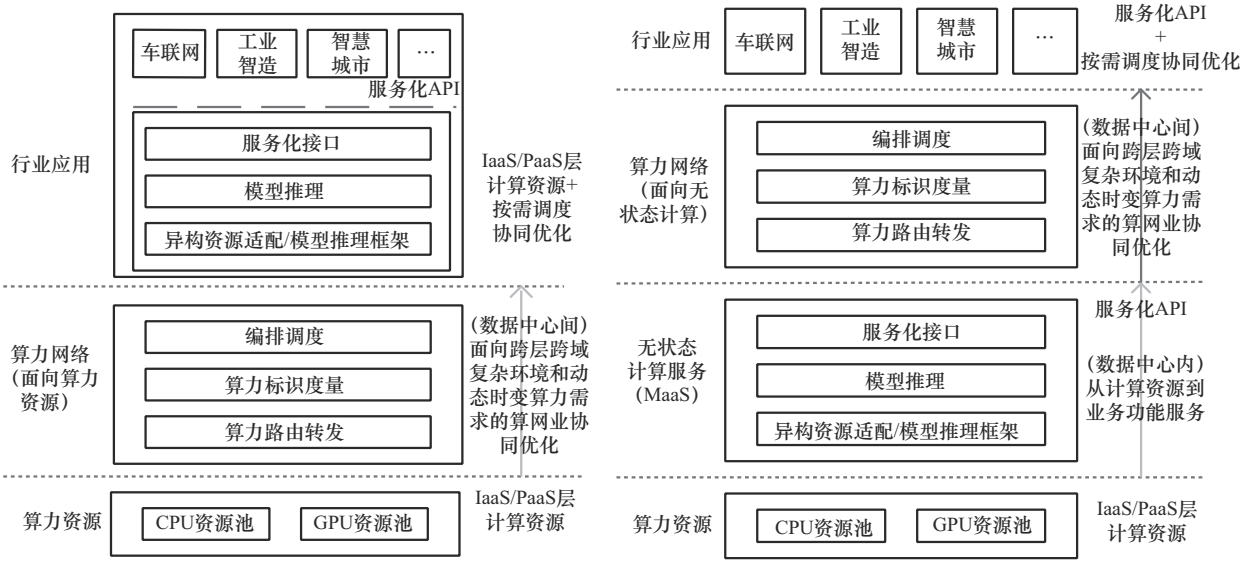
服务网格是一种专注于处理服务间通信的基础设施层，在TCP/IP上构建抽象层，实现服务路由、安全性和可观察性等功能[15-17]。其技术栈已相对成熟，典型方案（如Istio和Linkerd）通过控制平面统一管理流量策略，并依托Kubernetes生态实现自动化运维。服务网格的研究工作主要集中于数据中心内部环境。对于跨集群场景，虽然也提供了东西向网关等方案以实现跨集群的无缝访问，但主要是解决服务注册发现、身份识别认证、服务路由等问题，缺少跨域、跨主体算力服务开放所需的服务统一标识、域间网络的深度协同等机制。随着边缘计算场景的兴起，服务网格的研究正向云边协同架构延伸。边缘计算场景存在资源受限性、网络高波动性及分布式拓扑复杂等特点，传统服务网格方案面临显著挑战[18-19]，难以满足云边端复杂多变的算网环境中智能体、具身智能等AI业务的算网融合协同调度需求。

1.2　算力网络

算力网络[6-14]作为一种新型信息基础设施，能够深度融合泛在算力与网络，打破数据中心、超算中心、云计算和边缘计算之间的物理界限，构建智能、高效和按需的算网一体化服务体系。在产业界，算力网络的发展受到了高度重视。2025年5月，工业和信息化部印发《算力互联互通行动计划》，聚焦标准与标识体系建设，推动算力网络化、规模化应用，加快形成全国一体化算力体系。近年来，国内电信运营商也陆续发布算力网络相关白皮书推动产业生态发展，并与行业内研究机构、主要设备厂商在ITU-T、CCSA等标准组织中推动算力网络相关标准的制定。

在学术界，算力网络已成为当前新型网络领域的研究热点。国内外众多学者和科研机构致力于算力网络的基础理论研究、关键技术突破以及应用场景的拓展。从算力业务类型来看，现有研究多聚焦于算力资源层（如GPU集群、边缘节点）的互联调度优化[20]，通过算网协同调度实现虚拟机容器等异构算力资源或Serverless任务的动态分配。此类架构虽能提升资源利用率，但未充分考虑用户对计算功能的直接需求，导致从异构算力资源供给到用户AI功能需求间的供需链割裂，如图2(a)所示。例如，用户业务需要DeepSeek提供大语言模型计算服务，但算力网络提供的仅为支持DeepSeek部署的异构算力资源，需要用户在已分配的算力资源上部署DeepSeek模型并维护服务接口，为业务应用提供可直接调用的服务化API，增加了用户的使用难度。

针对MaaS这类无状态计算服务互联场景的算力网络方案尚处于初级阶段，多采用任播、域名系统（DNS, domain name system）等网络层方案实现[21-23]，利用IP地址、域名等对无状态计算服务



(a) 面向资源的算力网络的应用模式　　(b) 面向无状态计算服务的算力网络的应用模式

图2　算力网络的应用模式

进行标识/路由，但IP地址和域名的设计初衷是面向主机而非应用层的服务化API，协议栈层次错位会引入一些衍生问题。

1) 从调度粒度和时效看，任播方案以IP地址作为算力服务标识。当用户请求算力服务时，首先与目标IP地址建立TCP连接，再通过HTTP发送算力请求。调度发生在TCP会话建立的第一个握手报文阶段，因此只要TCP连接未断开，就不会再次触发调度。DNS以域名作为算力服务标识，当用户请求算力服务时，首先通过DNS协议请求域名对应的IP地址，此时通过控制返回的IP地址可实现算力服务实例的动态调度。然而，受限于DNS的多级缓存机制，域名对应的最优算力服务实例的IP地址更新需要数小时甚至数天才能完成全网同步，导致调度时效性较差。

2) 从协议匹配性看，如上所述，采用任播方案的算力网络使用IP地址标识算力服务。当客户端向算力网络发起服务请求时，以任播IP地址作为服务器地址建立TCP连接，再通过HTTP调用服务化API的统一资源定位系统（URL, uniform resource locator）。由于调度发生在TCP连接建立阶段，要求所有节点的服务化API的URL路径必须严格一致，才能确保任意算力服务实例和用户建立的TCP连接均可正常接收和处理HTTP层的算力请求。采用DNS方案的算力网络同样存在该问题。这种基于网络层协议进行服务调度，基于应用层协议进行传输实际算力请求的协议不匹配现象，会提高计算服务的客户端、服务端与算力网络的集成复杂度，也会影响算力服务的开放共享。

3) 从安全性看，使用IP地址或域名作为算力服务标识时，网络为用户与目标算力服务实例之间建立了IP地址可达的连接。此时，攻击者不仅可以发送正常的算力请求，还可能向目标服务实例发送恶意攻击报文，造成安全隐患。

2 需求分析

2.1 不同算力业务对算力网络的差异化需求

云计算技术发展至今，已衍生出裸金属、虚拟机、容器、对象存储、AI训推框架云服务、模型调用服务等多种算力业务形态，可以分为主机型、Serverless型和无状态计算服务型三类。

1) 主机型：如裸金属、虚拟机、容器等算力业务形态。用户可以像使用服务器设备一样，利用这些计算资源进行云门户网站、行业云应用、AI训推等各类业务部署。其典型特点是使用时间较长，从分配资源到完成计算任务并释放资源的周期可能需要数天甚至数年。业务连接协议因用户部署的具体业务而有所不同，通常采用L2层或L3层连接方案。

2) Serverless型：如对象存储、函数即服务（FaaS, function as a service）、AI训推框架云服务等[24-31]。用户直接将业务部署到服务商提供的能力平台中，不需要关注软件平台与资源的适配。由于使用过程中用户需要将自有数据、算法等部署到算力侧以实现业务，因此从算力使用的生命周期来看，与主机型类似，使用时间也比较长，但在业务连接协议方面，则通常是相对单一的特定业务协议。

3) 无状态计算服务型：如人脸识别、大模型等AI推理调用服务。无状态计算服务具有服务化、原子化等业务特征：① 服务化。无状态计算服务通过标准化API将算法功能暴露为可通过网络访问的计算服务，用户不需要关注其部署细节，仅需通过API调用即可使用其计算功能。目前业界广泛采用HTTP作为服务化API的主要交互方式（如RESTful、gRPC等）。② 原子化。当用户通过服务化API访问无状态计算服务时，每次API调用都是自包含的原子操作。即使采用了长连接模式，发起一次TCP连接可以实现多次API调用，但对于服务端而言，这些API调用在业务功能层面也是顺序无关、离散孤立的。因此，用户对算力资源的占用时间很短。以大模型为例，每次问答仅需要数秒甚至数毫秒，且连续问答也不需要算力侧缓存过程数据，连接协议则是基于HTTP的服务化API。

算力网络的本质是通过高效的资源分配和可靠的连接保障能力，实现分布式算力的全局协同与按需服务。不同算力业务因其特点各异，对算力网络的需求也有所不同。

主机型和Serverless型算力服务的任务生命周期较长，因此其算力资源分配的频次也较低，适合在算力网络的控制面中进行调度分配；无状态计算服务型的计算任务生命周期较短，适合将其调度过程与业务面报文转发过程融合；主机型算力服务随用户业务不同也有不同类型的业务互联协议，其连接需求类似于传统网络专线；Serverless型算力服

务和无状态计算服务通常使用特定的业务协议传输业务层内容，连接方案设计时需要考虑协议的适配性。

2.2　无状态计算服务对算力网络的调度要求

算力资源调度是算力网络实现算力服务供给的核心机制。在算力网络中，人脸识别、大语言模型等各类无状态计算服务分布于云、边、端多层节点，为用户提供服务。算力网络的目标之一是在多重约束条件下，通过调度分配算法将用户对算力服务的需求与算力服务实例进行精准匹配。

综上所述，无状态计算服务具有原子性，支持的最小调度粒度是“单次访问–单次响应”的服务化 API 调用。为此，可以将其调度过程与业务报文的处理过程融合，形成面向无状态计算服务调度的路由平面。其业务模型如图 3 所示，包括算力服务注册发现、算网状态感知交换、算力请求解析和路由调度等功能，主要工作流程如下。

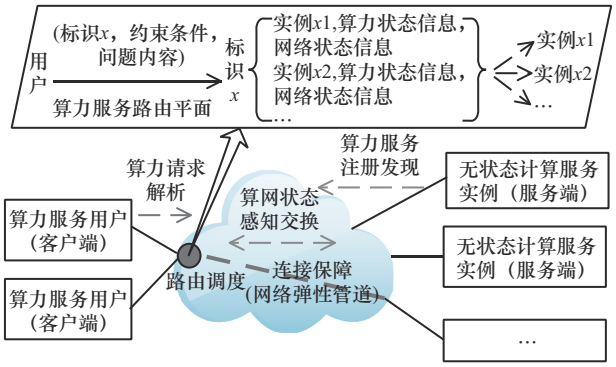


图3　面向无状态计算服务的算力网络业务模型

1) 算力服务注册发现。算力网络通过主动或被动方式获取算力实例，构建算力资源集合。

2) 算网状态感知交换。通过集中式、分布式或者混合式等方式实现算力服务、网络资源的状态信息，并完成网络节点间的数据交换，为算网融合调度提供数据基础。

3) 算力请求解析。在算力网络接入侧解析用户的算力请求，提取算力服务类型、算力服务的要求等用户需求信息。

4) 路由调度。算力网络根据算网状态信息进行算力资源调度计算，生成满足算力需求的转发策略，包括算力实例、传输路径及 QoS 等信息。

5) 连接保障。执行转发策略，构建用户与目标算力服务实例之间的网络弹性管道，完成算力请求和响应的传输。

分析上述流程中无状态计算服务的算网调度过程，对算力网络需有如下要求：一是应具备对每个算力服务请求进行逐一解析的能力，以支持根据用户的每次 AI 推理请求选择合适的算力服务实例，从而提高调度的精细度；二是应具备算网状态感知交换能力，以支持在算力服务调度决策过程中嵌入算网状态信息进行求解，从而实现时变跨域算网环境和用户需求下的算力服务优化配置和供给。

2.3　无状态计算服务对算力网络的互联要求

算力网络的另一目标是为跨域算力服务的供需双方提供连接能力，即在用户与目标算力服务实例之间构建显性或隐性的网络弹性管道作为业务流量的传输通路。通过对网络弹性管道配置不同的 QoS 参数实现差异化传输保障[32]。在针对无状态计算服务的互联方案设计中，考虑到无状态计算服务的服务化、原子性等业务特征，网络弹性管道需满足如下要求。

1) 灵活拆建能力。如光纤铺设、运营商网络专线等，不同网络技术在实现网络弹性管道时，其建立拆除难度、成本和效率有所不同。网络弹性管道的拆建效率直接影响算力网络为算力互联提供的“弹性”网络能力。无状态计算服务支持的最小调度粒度是“单次访问–单次响应”的服务化 API 调用，因此，从用户到算力服务实例之间的连接也呈现出细粒度、高动态变化的特征，这要求网络弹性管道需具备快速、灵活的拆建能力。

2) 协议匹配性。网络协议采用分层设计理念，可实现功能解耦和复杂通信的系统化管理。然而，这种分层架构也要求协议栈各层都需符合功能原语与接口规范才能互通。传统网络弹性管道技术通常提供 L2 层或 L3 层的连接能力，应用层互联需要在此基础上，先实现 IP 可达，再建立 TCP/UDP 连接。在跨网络域场景中，这种方式会增加网络配置的复杂性和运维成本，甚至成为制约用户业务方案可行性的关键因素。无状态计算服务基于 HTTP 通过服务化 API 实现算力请求与响应的传输。用户与算力服务实例之间实际需要传输的业务数据在 HTTP 部分，因此网络弹性管道应构建在 HTTP 层，实现算力请求与 TCP/QUIC 连接的解耦和路由终结，从而降低用户与算力服务实例侧的网络适配难度。

3) QoS 能力。QoS 能力通过量化网络弹性管道

的带宽、端到端时延、时延抖动及丢包率等参数，表征数据传输的性能保障强度。不同网络技术提供的QoS保障能力也各不相同[33-35]。网络需根据不同无状态计算服务的使用场景以及不同的文图影音内容，按需提供细粒度和差异化的QoS保障能力。

3　面向无状态计算服务的算力网络方案

3.1　设计原则

无状态计算服务的业务特征及其对算力网络弹性管道的特性需求，要求算力网络从面向算力资源连接向算力服务与业务应用升级。本文对面向无状态计算服务算力网络方案的设计遵循以下核心原则。

1) 原生支持无状态计算服务。在传统网络技术中，采用IP地址和域名作为算力服务标识的调度和QoS保障粒度与服务化API不匹配，导致网络资源浪费、暴露面广等问题。为此，需要设计基于服务化API粒度的算力调度和网络弹性管道方案。

2) 减少对无状态计算服务实例及算力用户应用的侵扰，便于集成。算力网络旨在为时变算网环境和需求下的算力用户提供高效、稳定和低成本的算力服务供给。在设计面向无状态计算服务的算力网络时，应充分考虑算力用户业务应用和无状态计算服务实例当前的软件技术生态，为算力供需双方提供低侵扰的算力互联能力，使计算服务和用户能够无感或低感地完成服务端、客户端与算力网络的三方集成。

3) 语义路由是一种根据用户意图或报文内容本身的语义来动态选择处理路径或转发策略的网络通信技术[36-41]，超越了传统基于IP地址或端口等固定标识的路由方式，致力于使网络更智能、更高效地满足多样化应用需求。在兼顾当前算力服务产业技术生态现状的同时，也借鉴未来网络语义路由的先进理念，从业务面报文中提取用户算力服务请求的语义构建路由平面。对于无状态计算服务，其表征用户所请求功能的是HTTP中的URL，因此可采用URL-PATH作为无状态算力服务标识方案来构建Overlay在网络层上的算力服务路由平面，从而实现无状态算力服务的灵活调度。

在此原则下，基于算力网络框架[42-43]开展了面向无状态计算服务的新型算力网络方案研究。

3.2　总体架构

图4为面向无状态计算服务的算力网络参考方案设计。本文方案是构建在传统Underlay网络上的Overlay网络方案，通过在Underlay网络的边界位置部署算力服务网关，构建算力服务路由平面和用户到目标算力服务实例的网络弹性管道，实现基于算力服务标识的算力服务路由转发处理和QoS保障。由于算力请求基于HTTP传输，并通过URL-PATH作为路由标识，该方案工作在网络协议栈的L7层。因此，网络弹性管道是面向算力服务应用层数据的传输管道，原生实现了Underlay网络的L3层路由终结，可支持跨网络域算力服务互通。算力服务应用层数据由算力服务网关调度分流，再利用Underlay网络的QoS流管道实现算力服务网关之间的传输保障。

功能逻辑如图5所示，本文方案主要包括控制层、业务层和设施层3层。

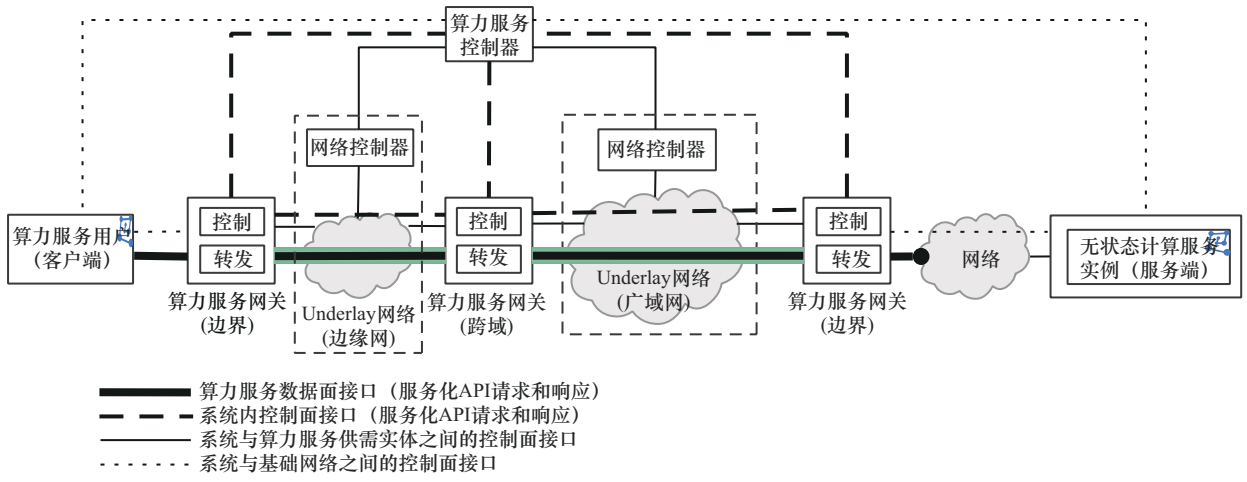


图4　组网架构

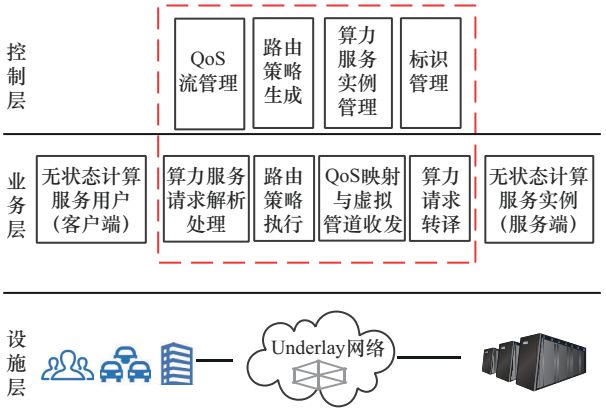


图5　功能逻辑

1) 设施层包括无状态计算服务实例所在算力节点、算力服务用户、Underlay 网络等算网资源，是面向无状态计算服务的算力网络方案的基础。

2) 控制层主要实现算力服务实例管理、网络状态感知、QoS 流管理、算力请求解析以及路由策略生成等功能。

① 标识管理。维护全局算力服务标识 URL-PATH 的描述信息，包括对应的算力服务功能、属性等，用于计算服务实例注册、用户侧查询等。算力服务标识是算力路由的基础，需要在全局范围内达成共识。

② 算力服务实例管理。实现算力服务实例的注册/发现、状态信息维护等功能。算力服务实例应归属于某一算力服务标识。当用户发起算力服务请求时，可通过算力服务标识获取所有算力服务实例的信息，以用于调度决策计算。

③ QoS 流管理。与 Underlay 网络交互，感知算力服务网关节点之间的网络状态，管理 Underlay 网络提供的 QoS 流和 QoS 能力，维护算力服务与 QoS 流之间的映射关系，为算力服务流量在算力服务网关之间的传输提供连接保障。

④ 路由策略生成。根据算力服务实例状态、网络状态和用户的算力服务需求，进行最优化计算，生成算力服务请求的路由策略表项。表项元数据包括算力服务标识、目标算力服务实例/下一跳网关、QoS 流等内容，用于控制网关对算力请求进行路由转发。

控制层可采用集中式、混合式和分布式部署等方案。其中，集中式部署方案采用全局集中控制器作为算网调度的大脑，负责算力服务和算网状态管理、路由策略生成等功能，网关主要作为执行器，根据控制器的路由策略进行转发处理，网关之间不需要交互；分布式部署方案则通过改进边界网关协议（BGP, border gateway protocol）等路由协议，实现网关之间的信息交互，并在网关本地完成路由决策生成与转发；混合式部署方案既有全局集中控制器，又存在网关之间的信息交互，2 种机制共同完成信息交换，如通过集中控制器完成算力服务发现、网络路径管理等功能，通过网关之间的直接交互完成算力服务状态的同步更新等功能。

3) 业务层实现算力用户与算力服务实例之间的算力服务请求与响应报文的处理、路由转发及传输等功能。

① 算力服务请求解析处理。模拟 HTTP 服务器监听来自用户侧的算力服务请求报文；解析算力服务请求报文，分析其 HTTP 头中的 URL 和首部字段，获取用户算力服务请求的算力标识及对算力服务的附加属性需求。

② 路由策略执行。通过从算力服务请求中提取的算力标识 URL-PATH 与控制面生成的路由策略表项进行匹配，查询该算力标识对应的最优算力服务实例，以及从当前网关到该算力服务实例所附着的算力侧网关之间的 QoS 流。

③ QoS 映射与虚拟管道收发。根据控制层与 Underlay 网络的交互，建立网关之间的 QoS 流，并根据路由策略建立算力服务请求与对应的 QoS 流的映射关系，将算力服务请求和响应报文按照对应的 QoS 流特性要求进行封装与解封装，完成算力服务报文在网关之间的收发。

④ 算力请求转译。在实际场景中，由于算力服务实例的运行环境、规划设计等原因，同类算力服务的各实例可能会使用不同的 URL，为降低算力网络方案与算力侧的适配和集成难度，使算力网络方案具有通用性，业务层支持算力服务标识 URL-PATH 与算力服务实例真实 URL 之间的转译。具体而言，该功能支持先将收到的算力服务请求中的 URL-PATH 信息转换为算力服务实例的真实 URL，再将其转发给算力侧处理。

3.3　关键技术

1) 基于 URL-PATH 的标识技术

在算力网络架构中，算力标识体系是资源调度与服务协同的基石。在无状态计算服务场景中，传统算力服务标识方案因算力的服务化、原子化等特

征，面临着适应性局限。

① 在服务化架构下，单一节点可以承载多种类型、多版本的算力服务（如不同功能的AI模型实例）。为了区分不同的算力服务，IP、域名等标识机制需要分配远超主机/节点数量的标识才能实现服务化API粒度的精准寻址与路由。

② 当算力服务请求经网络路由到达算力服务实例主机时，需要由算力服务用户确保应用层协议的匹配。

③ 在跨域协同场景中，算力服务需要突破物理边界，形成逻辑统一池。基于IP的传统标识既承担了网络层路由寻址功能，又承担了算力服务表征功能，在设计规划时存在局限性。

因此，构建兼具唯一性、自描述性和可路由性的算力标识体系，成为实现面向无状态计算服务的算力网络弹性调度与全局优化的先决条件。

服务化接口的演进为原生支持无状态计算服务的算力标识设计提供了范式指引。目前，算力服务通常采用基于HTTP的RESTful API、gRPC等服务化API来提供调用服务，其核心特征体现为协议中立性、语义显性化和层级可扩展性。

如图6所示，HTTP通过URL标识资源、通过首部字段携带交互过程中所需要的一些附加信息。

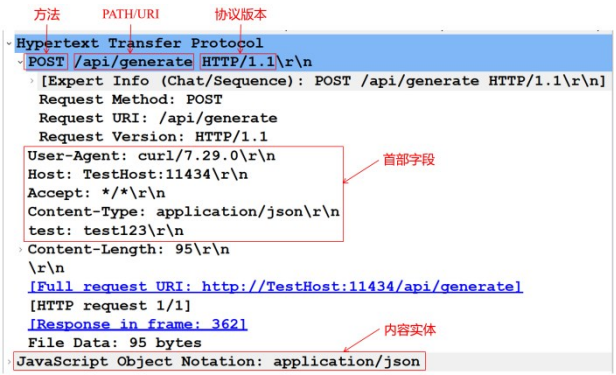


图6　HTTP请求报文的构成

URL：由协议、域名、端口号、路径和参数（可省略）等部分构成，其中路径字段（URL-PATH）天然支持多维度的服务描述，如通过/ai/deepseek/v1标识服务类型和资源版本等语义信息。

首部字段：HTTP的首部字段为客户端和服务器在处理请求和响应时提供所需要的扩展信息，采用“name:value”样式，并使用ASCII编码方式。首部字段可以包含丰富的信息，RFC2616、RFC4229等标准中定义了多种常用首部字段，涵盖了报文主体大小、认证信息等方面。此外，还可以灵活新增自定义的首部字段以满足新的需求。例如，在请求报文中添加表示算力价格信息的首部字段，用于给算力网络传递算力用户对算力服务价格的要求等。

在实际场景中，不同算力服务供应商会通过自行设计各自的URL来提供计算服务。而在算力网络中，需要对具有相同功能的算力服务实例进行聚类用于统一调度。因此，当用户向网络请求算力服务时，可以在算力请求中使用统一的算力标识URL-PATH。如图7所示，经过网络调度路由，算力服务请求到达目标算力服务实例的邻接网关后，通过算力请求转译功能将算力请求中的URL转译为算力服务实例的真实URL，从而实现对多实例的兼容纳管和灵活调度。

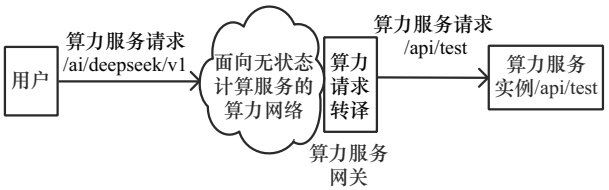


图7　算力请求中标识与实例真实URL的转译

基于URL-PATH的算力标识方案，通过分层结构与协议兼容性实现了服务化需求与网络能力的深度耦合，具备三重核心优势：首先，其层次化编码机制支持算力服务资源的逻辑聚合与细粒度切分；其次，语义显性化的URL-PATH与HTTP首部字段配合使用，包含可路由的算力请求与业务上下文，网络设备可直接解析算力请求报文以获取目标服务类型和服务水平协议（SLA, service level agreement）等属性，触发差异化的算力服务实例调度和流量工程策略，不需要依赖外部元数据系统；最后，URL-PATH基于HTTP承载，符合HTTP实现资源标识的标准，不需要对现有算力基础设施的业务面协议栈进行改造即可实现无缝兼容，显著降低集成成本，同时也与主流API网关天然适配，仅通过合理的控制面设计，即可实现URL-PATH标识驱动的算力服务智能路由，降低算力网络的开发和部署成本。

2) 业务面算力服务请求处理过程

业务面功能主要由部署在Underlay网络边界位置的算力服务网关共同完成，通过各节点对网络报文的处理，实现算力服务供需双方的互联。在算力

服务端到端互联中，各个节点的网络协议栈情况如图 8 所示。

① 算力用户。遵循传统服务化 API 客户端访问服务端的交互方式，从应用层发送 HTTP，经过主机协议栈的 TCP、IP 和 Ethernet 层层封装后，通过物理链路传输到邻接算力服务网关。

② 用户侧算力服务网关。监听并接受算力用户的服务化 API 调用报文，通过协议栈逐层解析获取 HTTP 报文，提取 HTTP 头中的 URL-PATH 和首部字段，完成算力服务路由调度，然后将 HTTP 报文封装到 Underlay 网络的虚拟管道协议中发送到对端网关。可以发现，经过网关的处理，算力用户与邻接算力服务网关之间的网络层路由被终结，仅发送了实际承载算力服务请求的 HTTP 报文。

③ 跨域算力服务网关。跨域算力服务网关通过在两侧网络域分别与对端网关建立虚拟管道，对收发的算力服务报文进行解封装、中转和封装操作，实现算力服务在跨网络域间的互联。

④ 服务侧算力服务网关。通过 Underlay 网络的虚拟管道接收算力服务报文，解封装后获取 HTTP 头，对 HTTP 头中的 URL 字段进行转译操作，替换为算力服务实例的真实 URL，再经过协议栈依次进行 TCP、IP 和 Ethernet 逐层封装后发送。

⑤ 计算服务实例。遵循传统服务化 API 服务端的网络协议栈处理方式，接收到网络报文后，通过主机协议栈进行解包处理，在应用层解析 HTTP 报文头和内容，并进行处理和响应。

以上为业务面算力服务请求发送阶段的协议栈处理过程，算力服务实例完成计算后返回的算力服务响应报文处理过程与之类似。经过反向转发后，将计算结果返回给算力用户。在实际应用时，还可以在网关之间互通协议的虚拟管道传输协议和 HTTP 层之间增加一层扩展协议，携带算力服务标识映射的数字 ID、加解密等信息，以提高路由匹配效率和传输安全性，增强方案的实用性。

4　仿真验证

本文通过仿真实验验证了所提新型算力网络方案的技术优势。实验以用户算力服务需求指标的达成情况作为衡量系统技术优势的指标，对比了本文方案与任播方案在算力网络中对用户的算力服务需求调度和保障能力方面的表现。

验证工作采用真实网元研制与算网环境模拟相结合的方式搭建仿真环境，主要包括算力服务网关、集中控制器以及算力服务用户/实例模拟软件。其中，算力服务网关基于开源代理软件 Envoy 和

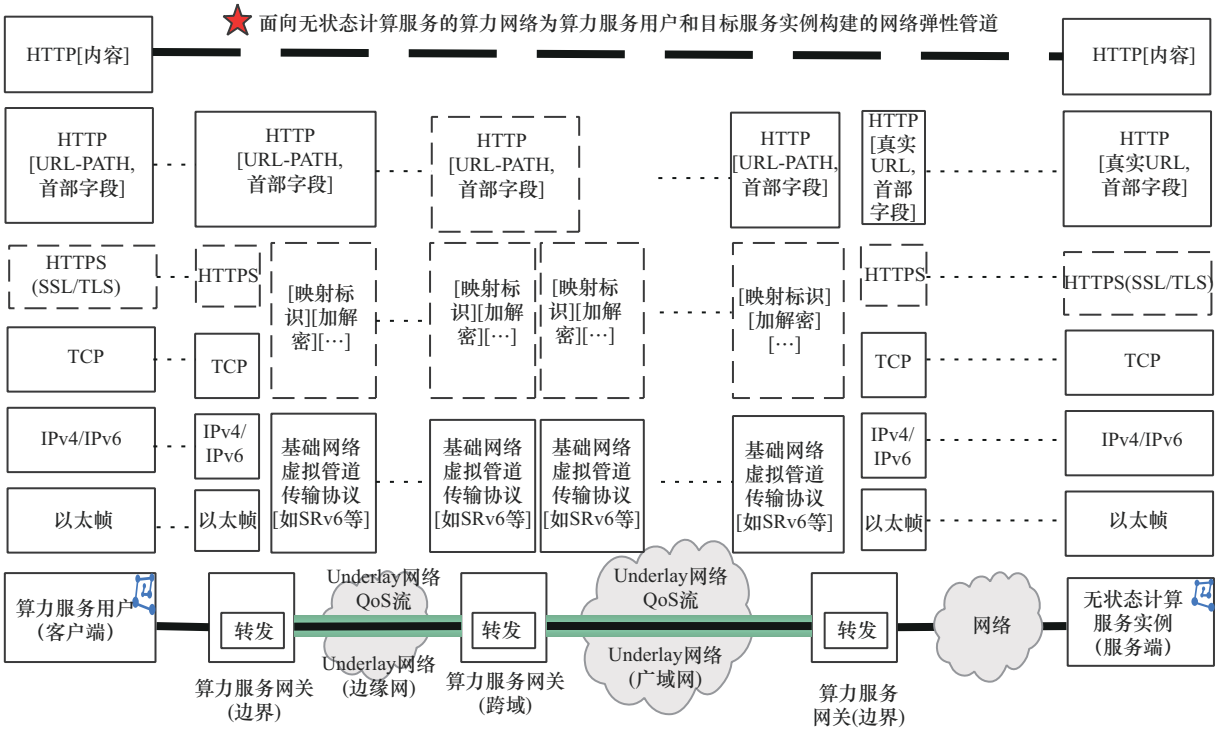


图 8　业务面转发协议栈

Linux系统路由功能，实现了本文方案与任播方案的路由调度策略执行及业务面转发能力；算力服务用户/实例模拟软件实现了算力服务供需之间的API访问和时延模拟；集中控制器支持算力服务实例注册、算网状态感知、路由调度策略计算与下发等基本功能。由于本次验证的目标是对比本文方案与现有网络层方案在调度灵活性方面的技术优势，而非针对不同场景和算网资源优化目标下的算法优势，因此采用了较为基础的贪心算法作为集中控制器中多约束条件下的路由调度计算工具。

4.1 数学建模

数学模型是实现路由调度计算的基础。在控制器研制过程中，对面向无状态计算服务的算力网络进行了数学建模。主要数学符号如表1所示。

表1 主要数学符号

数学符号	描述
R	无状态计算服务类型集合
r	R的元素，对应一个算力服务类型
S	某类无状态计算服务的实例集合
s	S的元素，对应一个算力服务实例
$\boldsymbol{A}$	算力服务实例的属性向量
α	$\boldsymbol{A}$的分量，对应算力服务实例的某一种属性
U	所有用户需求的集合
u	U的元素，对应某一用户的需求
$\boldsymbol{C}$	用户需求的约束条件向量
c	$\boldsymbol{C}$的分量，对应用户需求中某一要求
N	网络设备节点集合
n	N的元素，对应一个网络节点
E	网络节点之间的连接
$w(n_i,n_j)$	E的元素，描述节点间网络状态的权重函数
G	网络拓扑权重图
rID	算力服务标识ID
sID	算力服务实例ID
nID	算力服务实例附着到网络时分配的ID
$\boldsymbol{P}$	用户到某一算力服务实例的网络状态向量
p	$\boldsymbol{P}$的分量，对应网络状态的某一属性

参与算网融合调度的算力服务实例、用户和网络节点可以抽象为算力服务、用户需求以及节点间网络状态的集合。

1) 算力服务

设算力服务类型集合为

$$R=\left\{r_1,r_2,\cdots,r_m\right\} \tag{1}$$

每种算力服务类型r对应一种无状态计算服务功能，如人脸识别、物品分类、DeepSeek大模型等，在实际场景中，需要对算力服务的分类和标识进行统一设计和标准化。每种算力服务类型r中包含若干个实例，如不同地域部署的人脸识别服务节点就是该服务的实例，且支持动态注册和去注册。某一种服务的所有实例的集合记为S，表示为

$$S=\left\{s_1,s_2,\cdots,s_i\right\} \tag{2}$$

在调度时，需要将算力服务实例的属性状态作为依据。实例s_i的属性可以有多个维度，如并发容量、计算时延、售价等，表示为向量$\boldsymbol{A}$，如式(3)所示。

$$\boldsymbol{A}=\left[\alpha_1,\alpha_2,\cdots,\alpha_i\right] \tag{3}$$

其中，每个分量α表示算力服务实例的一种属性参数，在工程实施时，可以按需扩展设计。

2) 用户需求

设用户需求集合为U，表示为

$$U=\{u_1,u_2,\cdots,u_i\} \tag{4}$$

每个用户需求需指定所需资源类型$r\left(u_i\right)\in R$，也就是算力服务请求中的算力服务标识，同时，用户u_i也对算力属性提出约束，约束可以有多个维度，表示为向量$\boldsymbol{C}$，如式(5)所示。

$$\boldsymbol{C}=\left[c_1,c_2,\cdots,c_i\right] \tag{5}$$

例如，将分量c_i设定为单价上限，通过$c_i\leqslant x$元/次就可以表示用户对算力服务实例的调度要求之一是价格不高于x元/次。

3) 网络状态

设网络设备节点集合为

$$N=\{n_1,n_2,\cdots,n_i\} \tag{6}$$

其中，每个n_i表示一个网络设备。

网络设备间的物理或者逻辑连接关系表示为

$$E_N\subseteq N\times N \tag{7}$$

并由权重函数$w(n_i,n_j)$存储网络节点间的传输特性，如传输时延、带宽容量、经济成本等。

网络的拓扑则表示为

$$G=(N,E_N) \tag{8}$$

为了实现算网全局优化调度计算，算力网络还需要将算力资源R、算力用户需求U、网络状态G等进行统一建模。通过分析如图3所示的业务模型可以发现，算力网络需要获取算力服务、用户需求

及网络状态，再根据用户的算力请求进行调度。算力实例与用户通过邻接的网络设备动态绑定，由这些网络设备提供网络连接接入点、拓扑位置映射及全局唯一标识符等基础服务支撑。

算力网络可以通过算力与网络2个维度的标识映射，对算力服务、用户需求、网络状态等进行融合统一。算力维度有2个全局唯一的标识rID和sID，分别用于标识算力类型和算力实例。它们独立于网络层，当算力实例附着到邻接网络设备时，其算力维度的全局唯一标识sID将在网络维度上投影产生一个对应的网络维度的全局唯一标识nID，该网络维度全局唯一标识归属于其所邻接的网络设备，能够标定其在网络中的位置，支撑网络寻址。该过程可以表示为

$$\exists f_{\mathrm{rID}}: S \overset{\text{bijection}}{\rightarrow} \mathrm{rID} \tag{9}$$

$$\exists f_{\mathrm{sID}}: s \overset{\text{bijection}}{\rightarrow} \mathrm{sID} \tag{10}$$

$$\exists f_{\mathrm{nID}}: \mathrm{sID} \overset{\text{bijection}}{\rightarrow} \mathrm{nID} \tag{11}$$

$$\exists f_{n}: \mathrm{nID} \rightarrow N \tag{12}$$

在用户发起算力请求时，请求中会携带所需算力类型的标识rID。通过rID可以找到该类算力服务的实例集合S，S中包括所有同类型算力服务实例。对于任意算力实例$s \in S$可以通过上述映射函数获取到其邻接网络设备$n_s \in N$。

$$n_s = f_n(f_{\mathrm{nID}}(f_{\mathrm{sID}}(s))) \tag{13}$$

类似的，也可以获取到用户邻接的网络设备$n_u \in N$。再通过表征网络拓扑的权重图G可以获取n_u和n_s之间的权重函数$w(n_u, n_s)$，也就是从用户u到算力实例s的网络属性，可以记作$\boldsymbol{P}$。

$$\boldsymbol{P}(u,s) = \{p_1, p_2, \cdots, p_i\} \tag{14}$$

其中，p_i为时延、时延抖动、带宽等。

基于以上R、U和P的设定，根据系统目标（如最小化全局成本、单用户性能最优等）的不同，可构建相应的目标函数，进而通过合适的算法对问题进行求解，实现算力资源分配优化，形成算力需求路由调度决策。在调度计算过程中，将网络状态信息嵌入资源分配算法，实现了无状态计算服务跨域调度的算网协同优化。

4.2　验证分析

在仿真环境中，开展了如图9所示的无状态计算服务供需调度仿真实验场景。其中，网络时延为往返时延合计；服务实例负载高于50%、计算时延线性增加到100%时，时延翻倍。有s_1~s_5共5个运行物品分类算法的算力服务实例，且已通过服务化API将算法功能暴露到网络上，可以通过各自URL访问。算力服务具有容量、价格、时延等属性，其中容量均为50次/s；价格依次为2、1、2、1、2（每单位）；时延会根据负载情况变化。当负载低于50%时，5个算力服务实例的时延分别约为1 ms、3 ms、1 ms、3 ms和1 ms；当负载高于50%时，时延会线性增加。有u_1和u_2共2个用户，其中u_1对算力服务的指标需求是总时延（计算时延+网络时延）不高于10 ms，价格越低越好；u_2是时敏型业务，对算力服务的指标需求是总时延越低越好，价格不限。在用户侧和算力服务供给侧分别布置算力服务网关GW1~GW7，网关之间通过Underlay网络构建QoS流网络虚拟管道。网络状态属性主要是时延指标，其中GW1与GW3~GW7的双向传输时延分别是4 ms、6 ms、6 ms、6 ms和10 ms；GW2与GW3~GW7的双向传输时延分别是6 ms、4 ms、4 ms、4 ms和2 ms。

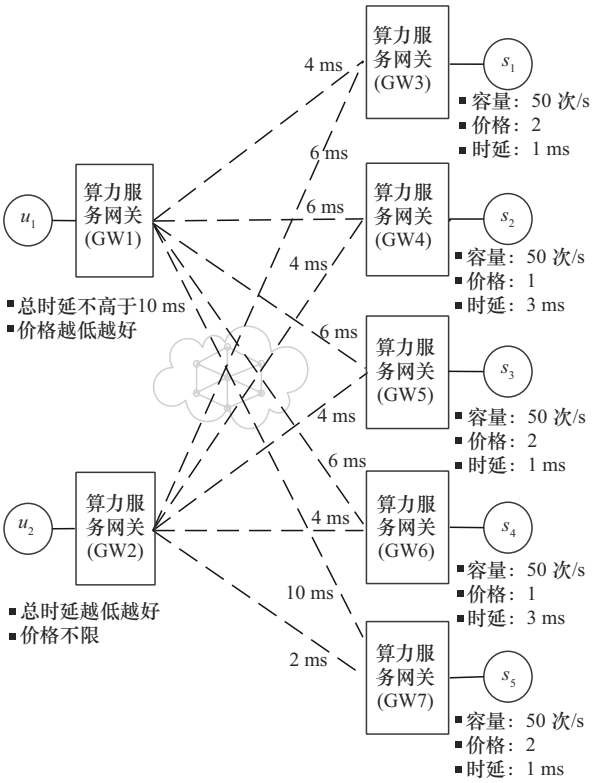


图9　仿真实验场景

基于上述数学模型，在模拟实验中，算力服务实例s_1~s_5属于同一种算力服务种类。算力服务实例的集合为

$$S=\{s_1,s_2,s_3,s_4,s_5\} \tag{15}$$

属性矩阵为

$$\boldsymbol{A}=\begin{bmatrix}50 & f_d(3,x) & 2\\ 50 & f_d(3,x) & 1\\ 50 & f_d(3,x) & 2\\ 50 & f_d(3,x) & 1\\ 50 & f_d(3,x) & 2\end{bmatrix}$$

其中，$f_d(a,b)$为

$$f_d(a,b)=\begin{cases}a, & b\leqslant 50\%\\ a+2(b-50\%)a, & b>50\%\end{cases} \tag{16}$$

用户算力需求集合为

$$U=\{u_1,u_2\} \tag{17}$$

约束条件矩阵为

$$\boldsymbol{C}=\begin{bmatrix}\text{时延不高于10 ms} & \text{价格越低越好}\\ \text{时延越低越好} & \text{价格不限}\end{bmatrix} \tag{18}$$

网络节点权重图的邻接情况如表2所示。

表2　网络节点权重图邻接情况

用户侧	时延/ms				
	GW3	GW4	GW5	GW6	GW7
GW1	4	6	6	6	10
GW2	6	4	4	4	2

以上为算力资源、用户需求和网络资源信息，在算网融合调度计算时，需要通过算力和网络维度的全局标识将这些信息进行关联。在不同的算力网络转发原理和工程实现方案中，可以采用不同的实现方法。

本文仿真实验中采用SRv6技术构建Underlay网络QoS流网络虚拟通道，表3是实验采用的算力标识方案对比。

1) 对于任播方案，算力服务类型rID通过任播地址进行标识，如通过（1.1.1.1）标识仿真实验所使用的物品分类算法，共有5个算力服务实例构成集合S，通过真实IP地址作为表征算力服务实例的算力维度全局唯一标识sID，与之对应的是在邻接算力服务网关中创建的SRv6 localsid作为网络维度全局唯一标识nID，通过SRv6 localsid即可获取到所属算力服务网关，进而获得用户到算力服务实例的网络状态信息$\boldsymbol{P}(u,s)$。当用户访问时，会发起向（1.1.1.1）的TCP连接请求，用户侧邻接算力服务网关收到目的地址为1.1.1.1的TCP握手报文后，通过目的地址获取目标计算服务类型rID，即任播地址（1.1.1.1）。随后，通过算力集合S、网络状态信息$\boldsymbol{P}(u,s)$等算网信息和调度算法计算最优算力服务实例，并完成路由转发。

2) 对于本文方案，算力服务类型rID通过URL- PATH进行标识，如通过（/yolo）标识仿真实验所使用的物品分类算法，同样有5个算力服务实例构成集合S，通过算力服务实例的真实URL作为算力服务实例的算力维度的全局唯一标识sID，进而与任播方案类似获取到nID、$\boldsymbol{P}(u,s)$等算网信息。用户访问时发起向邻接算力服务网关的TCP连接并发送http request报文，用户侧邻接算力服务网关收到http request报文后，提取到URL-PATH字段/yolo，即用户的目标算力服务类型rID，进而通过算力集合S、网络状态信息$\boldsymbol{P}(u,s)$等算网信息和调度算法计算最优算力服务实例并完成路由转发。

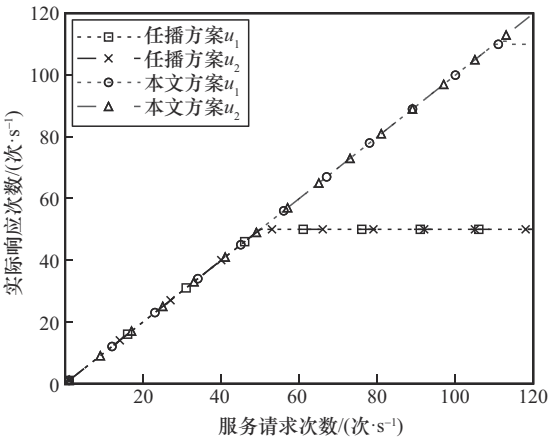
表3　算力标识映射关系

方案	任播方案	本文方案
用户算力请求的算力类型标识rID	表征算力服务的IP地址(1.1.1.1)	表征算力服务的URL-PATH (/yolo)
算力维度全局唯一标识sID	算力服务实例的真实IP地址	算力服务实例的真实URL
网络维度全局唯一标识nID	SRv6 localsid	SRv6 localsid
网络QoS流$\boldsymbol{P}(u,s)$	Srv6 sidlist	Srv6 sidlist

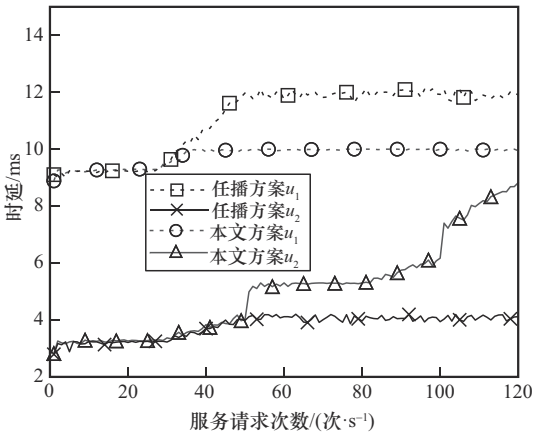
实验模拟了用户从1~120次/s算力服务请求时，任播方案和本文方案2种算力网络的服务响应情况。

图10(a)为任播方案与本文方案在不同强度的算力服务请求时网络实际响应用户请求的情况。可以看出，随着服务请求次数的增长，任播方案的响应次数极限能力为50次/s，本文方案的极限响应能力超过100次/s。其中用户u_1由于时延不高于10 ms的限制，在其约110次/s时达到响应极限。图10(b)为任播方案与本文方案在不同强度的算力服务请求时网络响应用户请求的最大时延情况。可以看出，

任播方案中用户 u_1 的最大时延随着负载压力的增加而增大，在服务请求次数达到 35 次/s 时便超过了 10 ms 的阈值限制；本文方案中用户 u_1 的总计算时延始终未超过 10 ms，满足用户 u_1 对算力服务时延不高于 10 ms 的要求。



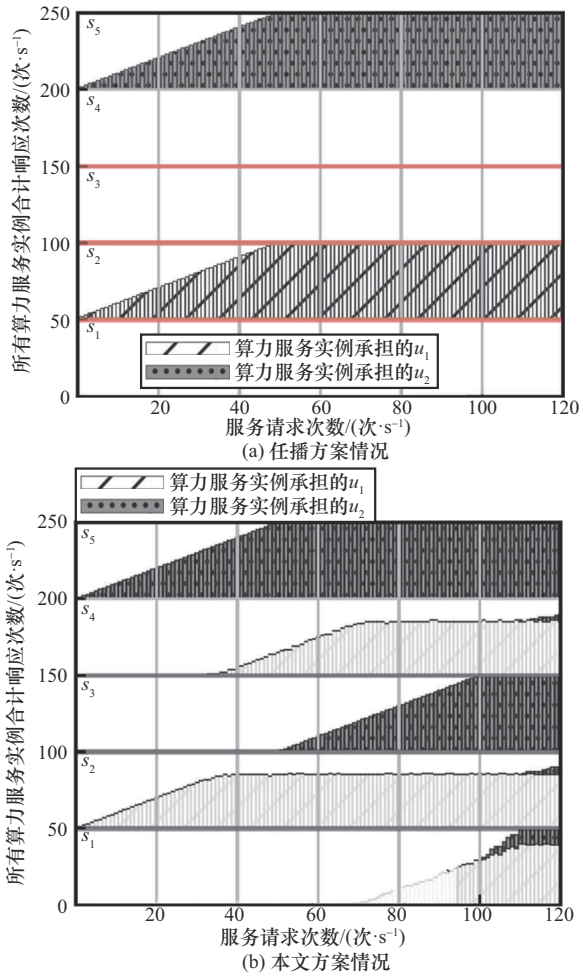
(a) 实际响应次数情况



(b) 时延情况

图 10　不同方案算力网络响应用户请求的情况

图 11 为采用任播方案和本文方案时算力服务实例的负载情况。纵坐标自下而上表示服务实例 s_1~s_5 的响应次数，合计 250 次/s。由图 11(a)可以看出，在任播方案下进行算力服务调度时，自始至终仅由 s_2 和 s_5 提供计算服务，s_1、s_3 和 s_4 处于空闲状态。由图 11(b)可以看出，在本文方案下进行算力服务调度时，当用户的服务请求次数低于单个算力服务实例处理极限时，用户 u_1 和 u_2 的请求分别由 s_2 和 s_5 提供计算服务，随着服务请求次数的增加，s_4、s_3 和 s_1 都逐渐被调度参与负载分担。例如，当用户 u_1 服务请求次数为 80 次/s 时，s_1、s_2 和 s_4 共同为其提供了总计 80 次/s 的服务响应。



(a) 任播方案情况

(b) 本文方案情况

图 11　不同方案算力服务实例负载情况

通过以上对比可以发现，即使采用了相同的算法，本文方案在对用户算力请求的响应能力方面也优于任播方案的算力网络。究其原因在于，任播方案以目的 IP 地址作为算力服务标识，构建的算力服务路由平面位于网络层，如图 12(a)所示，位于 TCP 层之下。在邻接算力服务网关解析报文时，为了避免将已建立连接的 TCP 会话报文流拆分到不同的算力服务实例，只能在收到用户侧发起的 TCP 握手报文时进行调度计算，并生成该 TCP 会话对应的 N 元组流表，以实现将后续报文转发到已建立连接的算力服务实例，即任播方案的最小调度粒度是 TCP 会话。相比之下，本文方案以 HTTP 的 URL-PATH 作为算力服务标识，构建的算力服务路由平面在应用层，如图 12(b)所示，位于 TCP 层之上。由于路由调度不受限于 TCP 连接状态，因此可以实现对每次算力服务 API 请求的独立调度。在实验中，因为用户侧模拟程序使用了长连接，所以出现了如图 10 所示

的算力服务实例的时延已不满足用户要求，但任播方案也无法为用户重新选择更合适算力实例，本文方案则能够有效解决这一问题。

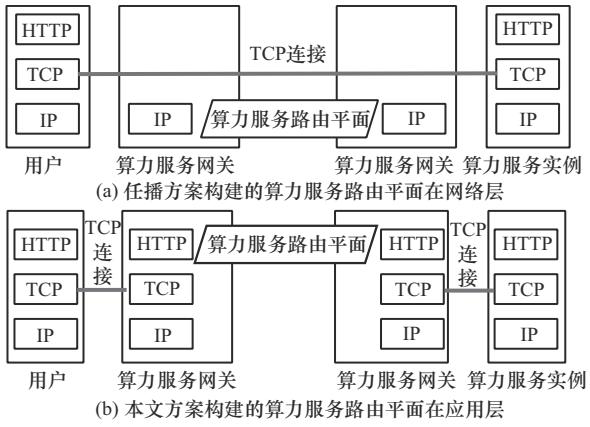


图12 不同方案的算力服务路由平面

如果用户侧采用短连接方式时，上述任播方案的问题将会有所改善，但这会限制用户的使用场景。此外，任播方案通过目的IP标识算力服务，对应用层未进行解析处理，所以要求服务实例s_1~s_5的真实URL必须一致才能保证用户构造的服务请求报文可以被任意算力服务实例正确识别。而本文方案以URL-PATH作为算力标识，并支持转发过程中将算力请求报文中算力服务标识URL-PATH转译为目标服务实例的真实URL，因此不限制算力服务实例服务化接口的真实URL是否一致，为用户提供更加友好的使用体验。

以上实验主要验证了本文方案基于URL-PATH标识构建算力服务路由平面，相对于任播地址作为标识的算力网络方案在支持API粒度调度和请求响应能力方面的技术优势。除了基于任播地址作为算力服务标识外，还有采用域名作为算力服务标识的算力网络方案，其利用DNS域名解析机制，在用户使用算力服务发送域名解析请求时，通过控制返回的IP地址来实现算力服务调度。该方案构建的算力服务路由平面同样位于网络层，除了存在前述TCP连接导致的调度问题外，还存在DNS缓存机制导致的调度响应滞后等问题。

本文方案为动态时变算网环境下的无状态计算服务灵活调度提供了架构基础。然而，在实际应用场景中，随着算网规模的扩大和业务目标的变化，还需要在本文方案和建模基础上进一步开展调度算法研究[44-45]，提高无状态计算服务跨域算网融合调度的能力。

5 应用场景

本文通过构建服务化API驱动的跨域算网协同架构，实现以服务化API为最小调度单元的算力资源动态编排与全局优化。其核心价值在于突破传统算力网络面向算力资源供给的调度模式，将计算服务功能、服务状态及网络质量纳入统一调度平面，支持跨异构网络域（如公有云、私有云、边缘节点）的细粒度算力服务匹配，为用户提供稳定高效、友好易用的服务化算力。这一能力原生支持当前业内采用服务化架构的信息与通信技术（ICT, information and communication technology）行业平台系统，易于集成，可广泛应用于车联网、智慧城市等领域，能够在用户系统无感或低感情况下显著提升算力服务的跨域供给能力。尤其在云边端协同、多智能体交互等高动态、强实时的复杂环境下，能够实现算力服务供给与业务需求的高效协同。以下从技术方案维度阐述两类典型应用模式作为参考示例。

5.1 算力网络增强的智能体方案

智能体是具备环境感知、自主决策与任务执行能力的计算实体[46-48]。近年来，大语言模型的突破显著提升了智能体的高阶认知能力，使其能够通过自然语言理解、推理与多模态交互完成复杂任务，成为业界研究的热点。以检索增强生成（RAG, retrieval-augmented generation）[49-50]为例，其业务流程包括问题向量化、知识检索、内容生成等阶段，如图13所示。在此过程中，智能体通常采用服务化API调用的松耦合协作模式动态调用外部知识库、模型推理服务等功能模块。

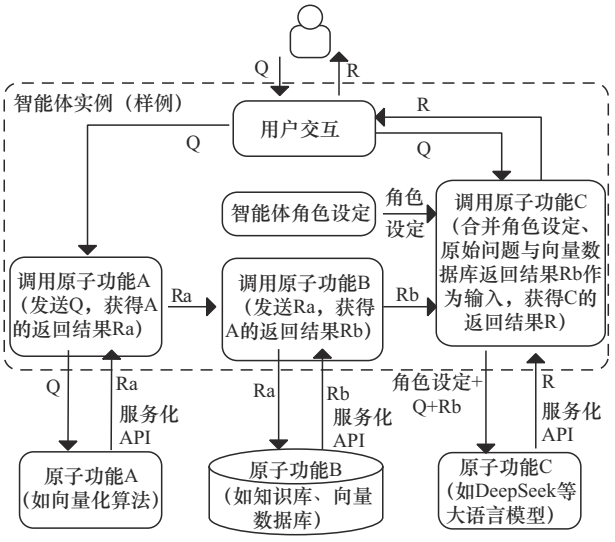


图13 智能体业务流程

传统智能体架构通过服务化 API 简化了组件交互，但其连接基础是组件间 IP 可达，算力分配未与网络协同，仍受限于静态部署模式和本地化调度策略，在跨域场景中缺乏与网络融合调度的解决方案。如图 14 所示，基于本文方案构建的面向无状态计算服务的算力网络可提供无状态计算服务的跨域调度互联能力：向量化算法、大语言模型等各类功能组件作为算力服务实例通过邻接算力服务网关接入到算力网络；当智能体实例根据业务需求调用这些功能组件时，不需要关注这些功能组件的部署位置，仅需向其邻接网关发送算力请求即可；算力网络根据算力服务请求中携带的算力服务标识及约束条件，在其构建的算力服务路由平面中，结合算网状态完成算力服务实例的路由调度，并利用网络能力提供从智能体到目标组件实例的连接保障，使智能体的主业务流程能够摆脱底层资源的约束，仅需向算力网络提交服务化 API 调用指令，即可实现全局算力服务的跨域供给与动态算网资源协同。

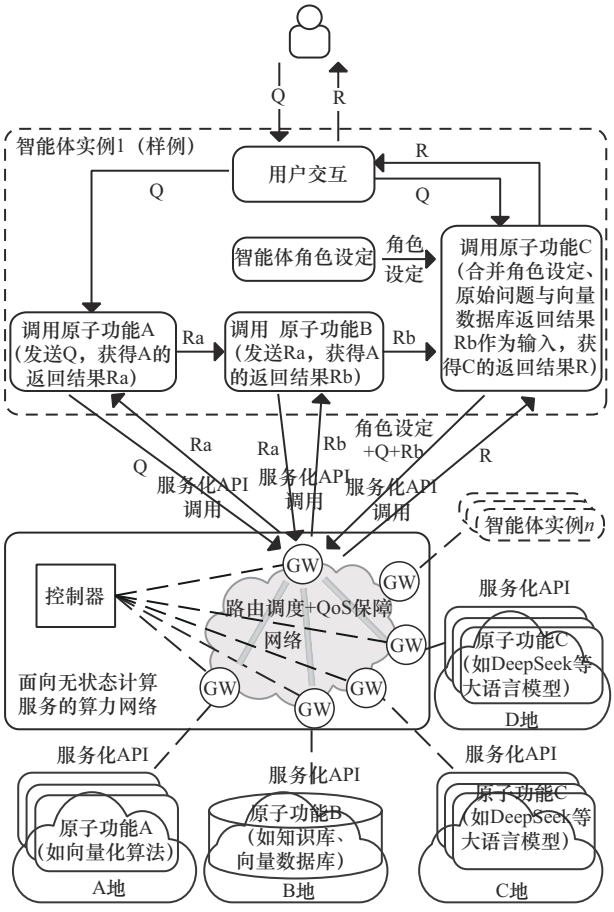


图 14　算力网络增强的智能体解决方案

这种算力网络驱动的智能体架构在工程实践中具有显著优势：业务流程与算力服务实现解耦，智能体开发者不需要关注功能组件的部署位置和运维细节，仅需向算力网络发起调用请求，即可实现跨域组件功能的集成，从而降低系统复杂性和维护成本，提高产品上线效率；算力网络通过实时采集网络带宽、算力服务负载、能源消耗、成本价格等多维度指标，构建多目标优化模型，实现算网业全局优化与协同计算。

算力网络与智能体服务化架构的深度融合，将为构建智能体、具身智能等泛在化、自适应的人工智能系统提供关键基础设施支撑。

5.2　多云算力统一入口与 QoS 保障方案

在传统多云业务部署中，企业通常将服务部署在本地或云端数据中心，并通过公网出口暴露服务访问点。如图 15 所示，服务提供商依赖 DNS[51-52] 实现域名与公网 IP 地址的动态绑定，以应对 IP 地址变更或故障切换。当跨域多实例部署时，可通过智能 DNS 的权重分配、地理位置或专线链路优选等策略，将用户请求引导至时延低或负载轻的数据中心[53]。

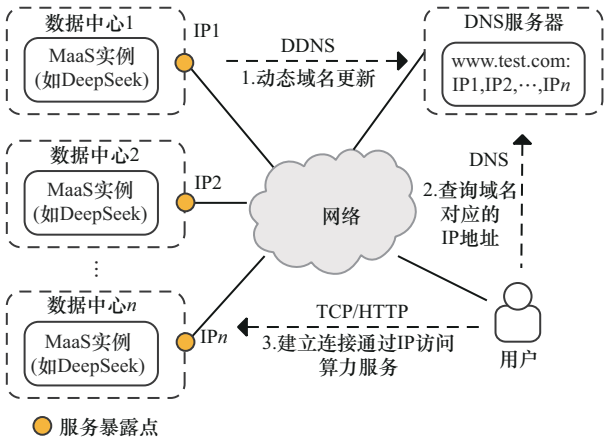


图 15　传统基于 DNS 的多云算力服务暴露方案

传统方案通过控制域名与 IP 地址的映射实现服务发布和多实例调度，但存在一些缺陷，如域名调度与网络状态缺少协同机制、DNS 缓存机制影响调度时效性以及调度后用户业务请求的传输保障过程依赖于网络的基础传输能力未与调度机制联通。本文方案通过构建算力服务路由平面实现算网协同调度，并通过算力服务网关解析算力请求实现业务分流，同时支持与 Underlay 网络协同提供 QoS

保障，为多云算力统一入口提供了新的解决方案。如图16所示，多云服务部署后，服务实例信息会向算力网络注册。算力网络将服务信息同步到边缘侧算力服务网关。边缘侧算力服务网关作为区域内的统一入口，负责服务暴露、接收用户请求、执行路由决策等。用户通过传统DNS解析获取边缘网关的IP地址，其请求首先抵达边缘网关；网关结合实时网络拓扑、云端算力负载及用户QoS需求（如时延SLA、带宽约束），选择最优数据中心。同时，算力网络基于Underlay网络构建边缘至云端的数据传输保障通道，确保服务传输链路的端到端性能。

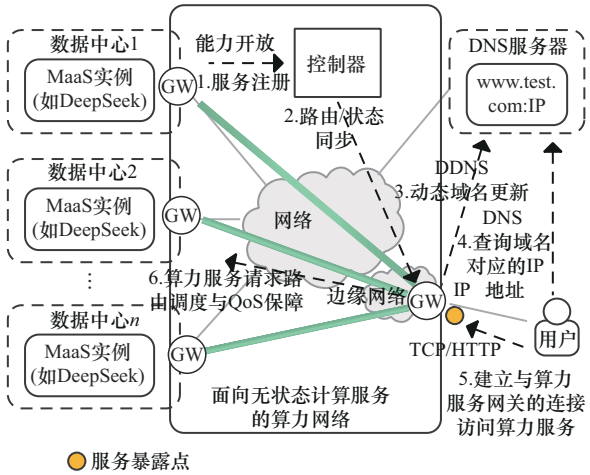


图16　基于本文方案的多云算力统一入口与QoS保障方案

本文方案重构了多云服务的暴露范式，将服务入口从数据中心公网出口下沉至边缘网络侧，通过算力网络的实时算网状态同步与动态路由决策机制，实现多云服务的全局优化调度。通过算力服务网关内置协议解析能力作为分流引擎，基于业务类型实施流量标记与优先级调度，实现细粒度的网络差异化服务能力。该方案可充分发挥通信运营商广域覆盖的多接入边缘计算（MEC, muti-access edge computing）优势与网络QoS保障能力，打破“网络管道化”困局，为用户提供更加灵活、动态和精细化的网络服务，支撑“网络即服务”商业模式的创新。

6　结束语

针对当前算力网络主要面向算力资源互联、缺乏对无状态计算服务等应用层算力服务的路由调度能力的问题，本文分析了算力网络的抽象模型及不同算力业务对算力网络弹性管道的特性需求，提出了一种原生支持无状态计算服务路由调度的新型算力网络方案。该方案以URL-PATH作为算力服务标识将服务化范式深度融入算力网络协议栈，在网络层与应用层之间构建独立的逻辑平面以实现服务路由和调度优化，并在用户和目标算力服务实例之间构建网络弹性管道提供传输保障。本文方案具有如下优点。

1) 支持API细粒度算力服务调度，提高算力服务响应能力。本文方案将算力服务的功能语义与网络层寻址、传输层连接解耦，通过应用层算力服务标识与语义路由机制，实现算力服务请求粒度的精细化调度。根据算力服务实例的实时负载、网络实时状态及用户的服务需求动态选择服务实例，实现算力服务实例的算网融合调度、算力服务流量的精细化分流与QoS保障，从而显著提升算力服务的供给能力。

2) 网络层路由终结特性有助于跨域互联与全局协同。本文方案采用应用层服务路由机制，通过算力服务网关实现网络层路由和传输层TCP会话的终结，天然支持跨域异构网络环境的服务互联。具体而言，服务标识在传输层上构建逻辑平面，使不同域内的服务注册、发现与路由策略可基于算力网络维护的统一算力服务元数据进行交互，而不需要依赖底层网络协议的强一致性。

3) 用户侧接口兼容性降低系统迁移与集成成本。本文方案在设计上保留了传统服务化接口（如RESTful API、gRPC）的兼容性，用户不需要修改现有调用逻辑即可接入新型算力网络。通过算力服务网关的协议转换功能，支持将服务标识映射为HTTP服务端点以监听用户的算力服务请求，降低了用户业务应用与算力网络的集成难度，可在用户无感或低感的情况下，根据算网状态完成算力服务的调度与优化，降低算力网络在优化算网资源供给时导致的用户系统迁移适配和集成成本。

本文方案既借鉴了未来网络的先进理念，又兼顾了当前算力服务产业技术生态的现状，实现了算力网络从资源供给向服务互联的演进，补充了业界针对不同算力业务类型的算力网络方案研究，有助于推进算网进一步融合和技术发展，为未来智能时代从异构算力资源到AI算法推理的泛在算力服务提供了可扩展的技术基座。

参考文献：

[1] GAN W S, WAN S C, YU P S. Model-as-a-service (MaaS): a survey[C]//

Proceedings of the 2023 IEEE International Conference on Big Data (BigData). Piscataway: IEEE Press, 2023: 4636-4645.

[2] WU P L, LIU Q, DONG Y J, et al. LMaaS: exploring pricing strategy of large model as a service for communication[J]. IEEE Transactions on Mobile Computing, 2024, 23(12): 12748-12760.

[3] ACHARYA D B, KUPPAN K, DIVYA B. Agentic AI: autonomous intelligence for complex goals: a comprehensive survey[J]. IEEE Access, 2025, 13: 18912-18936.

[4] XIE S Y, SANDOVAL E B, SHAIK K A, et al. Embodied generative AI art for enhanced human-robot interaction through a human-centric LLM-guided robotic arm drawing system[C]//Proceedings of the 2025 20th ACM/IEEE International Conference on Human-Robot Interaction (HRI). Piscataway: IEEE Press, 2025: 1727-1730.

[5] JAMSHIDI P, PAHL C, MENDONÇA N C, et al. Microservices: the journey so far and challenges ahead[J]. IEEE Software, 2018, 35(3): 24-35.

[6] 国家发展改革委, 中央网信办, 工业和信息化部, 等. 国家一体化大数据中心协同创新体系算力枢纽实施方案[R]. 2021.
National Development and Reform Commission, Cyberspace Administration of China, Ministry of Industry and Information Technology, et al. Implementation plan of computing power hub of national integrated big data center collaborative innovation system[R]. 2021.

[7] 国务院. “十四五”数字经济发展规划[R]. 2022.
The State Council. “14th Five Year Plan” for digital economy development[R]. 2022.

[8] 中国联通. 中国联通算力网络白皮书[R]. 2019.
China Unicom. China Unicom white paper on computing power network[R]. 2019.

[9] TANG X Y, CAO C, WANG Y X, et al. Computing power network: the architecture of convergence of computing and networking towards 6G requirement[J]. China Communications, 2021, 18(2): 175-185.

[10] SUN Y K, LEI B, LIU J L, et al. Computing power network: a survey[J]. China Communications, 2024, 21(9): 109-145.

[11] 何涛, 杨振东, 曹畅, 等. 算力网络发展中的若干关键技术问题分析[J]. 电信科学, 2022, 38(6): 62-70.
HE T, YANG Z D, CAO C, et al. Analysis of some key technical problems in the development of computing power network[J]. Telecommunications Science, 2022, 38(6): 62-70.

[12] GONG X M, BAI C C, REN S Y, et al. A survey of compute first networking[C]//Proceedings of the 2023 IEEE 23rd International Conference on Communication Technology (ICCT). Piscataway: IEEE Press, 2023: 688-695.

[13] LIU J L, SUN Y K, SU J Q, et al. Computing power network: a testbed and applications with edge intelligence[C]//Proceedings of the IEEE INFOCOM 2022-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). Piscataway: IEEE Press, 2022: 1-2.

[14] XU Q W, REN S Y, WANG J C, et al. A research of compute first networking based on information-centric networking cache mechanism[C]//Proceedings of the 2025 5th International Conference on Electronics, Circuits and Information Engineering (ECIE). Piscataway: IEEE Press, 2025: 783-791.

[15] KOSCHEL A, BERTRAM M, BISCHOF R, et al. A look at service meshes[C]//Proceedings of the 2021 12th International Conference on Information, Intelligence, Systems & Applications (IISA). Piscataway: IEEE Press, 2021: 1-8.

[16] PENG K, HU Y, DING H N, et al. Large-scale service mesh orchestration with probabilistic routing in cloud data centers[J]. IEEE Transactions on Services Computing, 2025, 18(2): 868-882.

[17] FURUSAWA T, ABE H, OKADA K, et al. Service mesh controller for cooperative load balancing among neighboring edge servers[C]//Proceedings of the 2022 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN). Piscataway: IEEE Press, 2022: 1-6.

[18] ELKHATIB Y, POYATO J P. An evaluation of service mesh frameworks for edge systems[C]//Proceedings of the 6th International Workshop on Edge Systems, Analytics and Networking. New York: ACM Press, 2023: 19-24.

[19] LIU J, XIONG H, YAN C. An open interconnection system for computing power based on service mesh[C]//Proceedings of the 2025 IEEE 5th International Conference on Power, Electronics and Computer Applications (ICPECA). Piscataway: IEEE Press, 2025: 266-269.

[20] 贾庆民, 丁瑞, 刘辉, 等. 算力网络研究进展综述[J]. 网络与信息安全学报, 2021, 7(5): 1-12.
JIA Q M, DING R, LIU H, et al. Survey on research progress for compute first networking[J]. Chinese Journal of Network and Information Security, 2021, 7(5): 1-12.

[21] LIU B, MAO J W, XU L, et al. CFN-dyncast: load balancing the edges via the network[C]//Proceedings of the 2021 IEEE Wireless Communications and Networking Conference Workshops (WCNCW). Piscataway: IEEE Press, 2021: 1-6.

[22] ITU. Architecture of dynamic-anycast in compute first networking (CFN-Dyncast)[S]. 2021.

[23] LI Y Z, HAN Z F, GU S H, et al. Dyncast: use dynamic anycast to facilitate service semantics embedded in IP address[C]//Proceedings of the 2021 IEEE 22nd International Conference on High Performance Switching and Routing (HPSR). Piscataway: IEEE Press, 2021: 1-8.

[24] FINGLER H, ZHU Z T, YOON E, et al. DGSF: disaggregated GPUs for serverless functions[C]//Proceedings of the 2022 IEEE International Parallel and Distributed Processing Symposium (IPDPS). Piscataway: IEEE Press, 2022: 739-750.

[25] LI Y K, LIN Y Y, WANG Y, et al. Serverless computing: state-of-the-art, challenges and opportunities[J]. IEEE Transactions on Services Computing, 2023, 16(2): 1522-1539.

[26] XU Z C, FU Y X, XIA Q F, et al. Enabling age-aware big data analytics in serverless edge clouds[C]//Proceedings of the IEEE INFOCOM 2023-IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2023: 1-10.

[27] RAZA A, AKHTAR N, ISAHAGIAN V, et al. Configuration and placement of serverless applications using statistical learning[J]. IEEE Transactions on Network and Service Management, 2023, 20(2): 1065-1077.

[28] IBRAHIM S, RANA O, BEAUMONT O, et al. Serverless computing[J]. IEEE Internet Computing, 2024, 28(6): 5-7.

[29] ZHAO M, JHA K, HONG S. GPU-enabled function-as-a-service for machine learning inference[C]//Proceedings of the 2023 IEEE International Parallel and Distributed Processing Symposium (IPDPS). Piscataway: IEEE Press, 2023: 918-928.

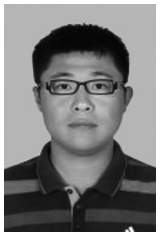
[30] MALEKABBASI M, PFANDZELTER T, SCHIRMER T, et al. GeoFaaS: an edge-to-cloud FaaS platform[C]//Proceedings of the 2024 IEEE International Conference on Cloud Engineering (IC2E). Piscataway: IEEE Press, 2024: 66-71.

[31] SABBIONI A, FOSCHINI L. SFIOC: a platform to support service dependency injection in serverless functions[C]//Proceedings of the 2025 34th International Conference on Computer Communications and Networks (ICCCN). Piscataway: IEEE Press, 2025: 1-6.

[32] GOMES J V, INÁCIO P R M, PEREIRA M, et al. Detection and classification of peer-to-peer traffic[J]. ACM Computing Surveys, 2013, 45(3): 1-40.
[33] ROUGHAN M, SEN S, SPATSCHECK O, et al. Class-of-service mapping for QoS: a statistical signature-based approach to IP traffic classification[C]//Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement. New York: ACM Press, 2004: 135-148.
[34] 黄韬, 汪硕, 黄玉栋, 等. 确定性网络研究综述[J]. 通信学报, 2019, 40(6): 160-176.
HUANG T, WANG S, HUANG Y D, et al. Survey of the deterministic network[J]. Journal on Communications, 2019, 40(6): 160-176.
[35] 贾庆民, 胡玉姣, 张华宇, 等. 确定性算力网络研究[J]. 通信学报, 2022, 43(10): 55-64.
JIA Q M, HU Y J, ZHANG H Y, et al. Research on deterministic computing power network[J]. Journal on Communications, 2022, 43(10): 55-64.
[36] REN P G, WANG X F, ZHAO B K, et al. OpenSRN: a software-defined semantic routing network architecture[C]//Proceedings of the 2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). Piscataway: IEEE Press, 2015: 101-102.
[37] IRTF. A survey of semantic Internet routing techniques[S]. 2021.
[38] IRTF. Challenges for the Internet routing infrastructure introduced by semantic routing[S]. 2022.
[39] IRTF. An introduction to semantic routing[S]. 2022.
[40] JABER G, PASTEI N, RAHAL F, et al. Naming and routing scheme for data content objects in information-centric network[C]//Proceedings of the 2020 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream). Piscataway: IEEE Press, 2020: 1-5.
[41] AMADEO M, CAMPOLO C, SERRANO S, et al. DT-assisted vehicular crowdsensing through semantic-aware NDN[C]//Proceedings of the 2025 IEEE International Mediterranean Conference on Communications and Networking (MeditCom). Piscataway: IEEE Press, 2025: 1-6.
[42] 中国联通研究院 算力网络架构与技术体系白皮书[R]. 2020.
China Unicom Research Institute. White paper on computing power network architecture and technology system[R]. 2020.
[43] 中国通信标准化协会 算力网络总体技术要求[S]. 2021.
China Communications Standardization Association. General technical requirements of computing and network convergence[S]. 2021.
[44] HU Z M, LI B C, LUO J. Time-and cost-efficient task scheduling across geo-distributed data centers[J]. IEEE Transactions on Parallel and Distributed Systems, 2018, 29(3): 705-718.
[45] KHALEEL M I. Region-aware dynamic job scheduling and resource efficiency for load balancing based on adaptive chaotic sparrow search optimization and coalitional game in cloud computing environments[J]. Journal of Network and Computer Applications, 2024, 221: 103788.
[46] LU Q H, ZHU L M, XU X W, et al. Towards responsible generative AI: a reference architecture for designing foundation model based agents[C]//Proceedings of the 2024 IEEE 21st International Conference on Software Architecture Companion (ICSA-C). Piscataway: IEEE Press, 2024: 119-126.
[47] JOVANOVIĆ M, CAMPBELL M. Self-directing AI: the road to fully autonomous AI agents[J]. Computer, 2025, 58(2): 71-77.
[48] DURGAPRASAD K V V B, ABOZIBID H K, HAWAS J N, et al. AI agents and conversation system[C]//Proceedings of the 2024 International Conference on Augmented Reality, Intelligent Systems, and Industrial Automation (ARIIA). Piscataway: IEEE Press, 2024: 1-7.
[49] TURAL B, ÖRPEK Z, DESTAN Z. Retrieval-augmented generation (RAG) and LLM integration[C]//Proceedings of the 2024 8th International Symposium on Innovative Approaches in Smart Technologies (ISAS). Piscataway: IEEE Press, 2024: 1-5.
[50] NEHA F, BHATI D, SHUKLA D K, et al. Exploring AI text generation, retrieval-augmented generation, and detection technologies: a comprehensive overview[C]//Proceedings of the 2025 IEEE 15th Annual Computing and Communication Workshop and Conference (CCWC). Piscataway: IEEE Press, 2025: 633-639.
[51] LI Q, QI X Q, LIU J M, et al. Design and implementation of traditional DNS protocol[C]//Proceedings of the 2017 International Conference on Computer Technology, Electronics and Communication (ICCTEC). Piscataway: IEEE Press, 2017: 1384-1390.
[52] WANG Z S, LIU Z. Research and design of decimal network DDNS[C]//Proceedings of the 2021 International Conference on Computer Network, Electronic and Automation (ICCNEA). Piscataway: IEEE Press, 2021: 153-158.
[53] LAI T L, TSAI M H. Design and implementation of a DNS server with geolocation capability[C]//Proceedings of the 2021 22nd Asia-Pacific Network Operations and Management Symposium (APNOMS). Piscataway: IEEE Press, 2021: 370-373.

[作者简介]

张岩（1983– ），男，河北保定人，博士，中国联合网络通信有限公司研究院正高级工程师，主要研究方向为算力网络、云网融合、未来网络体系架构等。



王立文（1984– ），男，河北衡水人，博士，中国联合网络通信有限公司研究院高级工程师，主要研究方向为云网技术等。

曹畅（1984– ），男，江苏无锡人，博士，中国联合网络通信有限公司研究院正高级工程师，主要研究方向为下一代互联网、算力网络等。

唐雄燕（1967– ），男，湖南永州人，博士，中国联合网络通信有限公司研究院正高级工程师，主要研究方向为宽带通信、下一代互联网、算力网络等。