面向算力互联网的高可用智算互联互通平台体系架构

鄢智勇[1]，陈浩[1]，丁立戈[1]，魏本洁[1]，陈晓帆[1]，胡建锋[1,2]
（1.天翼云科技有限公司，北京 100082；2.翼速云科技有限公司，福建 厦门 361001）

摘　要：针对全国范围算力资源信息互联、计算任务互通和基础设施高可用的需求，侧重任务式和资源式智能算力应用场景，设计了一种互联互通平台体系架构，通过平台连接算力需求方和供给方，支持算力互联互通并保证可用性。基于分层解耦、分布式互联、池间任务直连互通的设计原则，提出算力插件实现资源互联，借助算网调度实现任务跨资源池互通。采用主备多活故障保护实现平台高可用，为降低运维开销，对算力互联网平台体系架构的可用性进行建模，提出可用性感知的模块划分方法。模拟结果表明，所提方法可在满足不同可用性要求前提下，最小化微服务数量，从而降低运维开销。研究结果可为未来多算力运营主体间的互联互通打好基础。

关键词：算力互联网；算力插件；高可用；智算互联互通
中图分类号：TN393
文献标志码：A
DOI：10.11959/j.issn.1000–436x.2025153

High available intelligent computing interconnected and migration platform architecture for Internet of computing power

YAN Zhiyong[1], CHEN Hao[1], DING Lige[1], WEI Benjie[1], CHEN Xiaofan[1], HU Jianfeng[1,2]
1. China Telecom Cloud Technology Co., Ltd., Beijing 100082, China
2. China Telecom Yisu Cloud Technology Co., Ltd., Xiamen 361001, China

Abstract: In response to the national demand for interconnectivity of computing resources, computing tasks, and high availability of infrastructure, a platform architecture for interconnectivity was designed with a focus on task-based and resource-based intelligent computing application scenarios. The platform connected computing power demanders and suppliers to support interconnectivity and ensure availability. Based on the design principles of hierarchical decoupling, distributed interconnection, and direct interconnection of tasks between pools, a computing power plugin was proposed to achieve resource interconnection, and task cross resource pool interconnection was achieved through network scheduling. The active/standby multi active fault protection was used to realize the high availability of the platform. In order to reduce the operation and maintenance costs, the availability of the Internet of computing power platform architecture was modeled, and the module division method of availability awareness was proposed. The simulation results show that the proposed method can minimize the number of microservices while meeting different availability requirements, thereby reducing operational costs. The research results can lay a solid foundation for the interconnection and intercommunication among multiple computing power operators in the future.

Keywords: Internet of computing power, computing plugin, high availability, intelligent computing interconnection

0　引言

自1969年互联网[1]诞生之日起，伴随着应用的不断丰富，逐渐产生了多种互联网应用模式，如移动互联网、消费互联网和工业互联网[2]。近年来，

收稿日期：2025–04–30；修回日期：2025–08–20
通信作者：陈浩，chenhao3@chinatelecom.cn
基金项目：国家科技重大专项基金资助项目（No.2025ZD1302500）
Foundation Item: The National Science and Technology Major Project of China (No.2025ZD1302500)

算力和应用正在不断发展[3]，应用模式正在发生转变。一方面，在技术上，当本地计算资源类型不满足要求时，需将任务迁移至其他边缘计算节点执行[4]；另一方面，我国在政策层面上正在加快引导算力互联[5]。随着算力应用逐渐成为焦点，算力互联网将是未来极其重要的新型互联网应用模式。算力互联网是在互联网体系架构[6]上增加算力标识、算力调度等新功能，实现全网异构算力的感知使用和计算任务的最佳算力资源部署决策，进而形成标准化的算力互联互通体系，也是支撑我国算力应用发展的新型基础设施。

随着人工智能[7]等任务式智能算力应用[8]和算力卡[9]等资源式智能算力应用场景的兴起，市场对于智能计算的需求也越来越多。但是，我国的智算与网络发展现状却难以满足日益增长的算力需求。

1) 资源可互联：用户需要全面获取各种资源或产品信息并从中选择，但由于资源从属多主体，资源尚未完全互联，用户难以获取多主体算力资源信息。更难的是智算资源异构，性能不一，纵使知晓全部资源信息，也难以为智算应用选择最佳资源。

2) 应用易互通：用户需要在各资源池自由部署迁移智算应用[10-11]，但智算任务在不同主体资源池部署需使用不同的接口，导致任务在跨主体资源池间互相流通变得困难。

3) 体系高可用：用户需要高可用的算力资源服务，因为只需小时级的服务故障就能造成上百万的经济损失[12-13]，并且随着功能的不断丰富，算力互联互通体系将不可避免地变得复杂，从而更容易出现故障，导致总体可用性下降。

为满足上述需求，可采取已有的技术方案应对。对于需求1)和2)，一种直接的方法是采用多云管理平台[14]来建设算力互联互通体系，这类平台可支持多主体资源池的管理，获取所接入云资源池的产品信息，在不同云池上部署人工智能（AI）智算任务，可实现基本的资源互联和应用互通能力，但这类平台不能满足用户对资源的最佳选择需求。为此，一些工作考虑添加算力调度能力，例如，云代理[15]在多云管理平台的基础上增加了算力感知模块和算力调度模块，实现了最优算力决策的功能，但它无法为用户提供统筹算网的最优调度决策。尤其是在智算场景下，AI应用对网络的需求相比通算更高，例如，AI训练数据集可达TB级[16]，网络带宽需求大[17]；实时AI推理任务对首Token响应要求较高[18]，服务运营者有降低网络访问时延的需求。为此，需要具备多云场景下决策最佳网络传输路径的功能，云代理无法满足这一需求。同时，为从性能不一的多主体资源中选择最佳资源，还需要新增算力度量功能以实现异构资源的性能统一度量，现有平台[14-15]则缺少相关功能。此外，对于需求3)，现有方案例如云代理[15]虽可通过增加云资源提供商数量来提高用户服务的可用性，但它们对平台自身的可用性并未进行深入研究，直接采用现有技术方案建设算力互联网会导致智算互联互通平台体系自身可用性不足。

综上所述，现有的解决方案在算网调度能力和可用性上还不能很好地满足用户要求，需要设计高可用的智算互联互通平台体系架构来支撑不断增长的智算应用需求。在当前市场发展阶段，智算互联互通平台体系架构的设计尤其需要平衡好可用性与有限资源间的矛盾，其中资源包括设备和人力资源。一方面，通过增加冗余备份可显著提高可用性，但也会产生更多的资源消耗，因而通常需要控制备份资源的使用量；另一方面，当前市场环境以降本增效为目标，考虑到更多的部署服务实例需要配备更大规模的运维团队，因而尽可能减少部署的服务实例数量同样重要。如何将各种功能模块编排组合划分形成合理的体系架构，在实现智算互联互通功能的前提下，满足可用性要求，同时尽可能减少相关资源开销，是本文要解决的一个重要科学问题。

在这一科学问题下，设计智算互联互通平台体系架构面临着2个挑战。1)要实现全国范围内智算资源池的互联互通，体系架构必须具备极高的扩展性。一方面，我国至少有百级别的算力提供商[19]，不同提供商的接入方式都有差异，如何实现众多异构算力服务提供商的互联互通是极具挑战性的；另一方面，资源提供商通常具有多种网络接入方式，如公网、专网和专线等，如何实现多类型网络的接入和调度同样具有挑战性。2)作为新型基础设施，智算互联互通平台体系架构必须具备很高的可用性。现有云计算服务的可用性通常可达到99.9%[20]，但当增加了算网调度和算力度量等多种功能后，体系架构的复杂度变高，此时如何保持智

算互联互通平台体系架构自身的高可用将变得有挑战性。

针对上述问题与挑战，本文设计了面向算力互联网的高可用智算互联互通平台体系架构。走增量式路线增加算网调度等功能以满足智算互联互通需求，不同于现有方法[15]采用的模块化单体架构，本文采用分层微服务架构，以便扩展大量新功能。该架构以现有的云管和网管平台作为基础设施层；在其上结合多云管控模块，并加入多网管控模块构建互联管控层；在此之上，新增算网调度层；并在最顶层加入平台互联、算网交易和运营功能形成算网交易运营层。各层的模块均通过微服务部署，形成四层微服务体系架构。考虑到多主体资源性能不一致的现状，本文在多云管控模块中引入算力度量功能，实现对异构云资源的性能统一度量，可将度量结果作为算网调度决策的依据。为实现计算任务跨主体跨域迁移，本文还引入算数协同调度与数据快递功能。在实现智算互联互通功能后，本文通过增加微服务备份实例的方式，提高总体可用性。

针对挑战 1)，现有方案[15]采用模块化单体架构，通过提供器和执行器直接对接各云厂商接口的方式实现多云接入，该方案不适合支撑全国范围大规模多云接入。为克服现有方案的不足，本文提出算力插件与网络插件，可通过适配器模式屏蔽异构云网资源接口，实现异构算力和网络资源的感知与灵活按需接入，通过微服务化独立部署，可实现横向扩展与热插拔能力，支撑大规模多云多网接入和调度。

针对挑战 2)，常见的方法是增加备份模块，在备份资源有限的情况下，仍需寻找其他提高体系架构可用性的方法。为进一步优化体系架构以满足可用度要求，本文观察到将模块中不同子模块分别以独立微服务部署，可增强体系架构的可用性（见第 3 节），然而划分出过多微服务实例数会造成运维资源开销。为此，本文对所提出的体系架构建立可用性模型并进行分析，针对可用性与有限资源间的矛盾，本文提出可用性感知的最少微服务数量模块划分问题，并提出了最优微服务划分方法，在满足可用性要求的同时尽可能减少微服务数量。相比于当前成熟微服务划分方法[21]，本文提出的划分方法更容易满足可用性要求，且在满足要求的情况下，保持相同的微服务数量开销。

本文的主要贡献如下。

1)提出智算互联互通平台体系架构，可支持异构智算资源发现、接入与使用。通过算网调度，同时满足用户的算力与网络两方面要求。

2)提出算力插件和网络插件技术，可接入指定云网资源，支持热插拔和按需动态加载，提高整体架构的扩展性和灵活性。

3)基于主备多活故障保护和可用性计算方法，考虑有限的运维资源，建立最优可用性的模块划分问题及划分算法，保障整体架构的可用性。

1　相关工作

近年来，中国算力市场蓬勃发展，算力规模已跃居全球第二。然而，国内算力资源呈现碎片化分布，利用率普遍偏低[3]。为破解算力分散化、异构化、供需失衡等难题，亟须构建能够实现全网算力感知、实时发现和随需使用的算力互联网。不同于传统互联网偏向于底层设计的体系结构研究，算力互联网的实现以网络设施层为基础，聚焦于算力互联网平台互联体系构建。当前互联网体系架构研究已取得系列成果，具体如下。文献[6]提出多维可扩展性概念，显著提升互联网管理能力。文献[22]介绍了与互联网体系结构发展密切相关的五种特性的基本评估模型，提出了一种基于演进式的互联网体系结构发展思路。文献[23]明确了互联网体系架构的演进方向，构建了实验验证平台。文献[24]从知识表征、意图驱动、分布式AI和AI可解释性4个维度，探讨了6G网络智能化提升路径。文献[25]提出应用按需定制网络服务质量的概念，通过可声明、细粒度和端到端能力实现网络即服务。文献[26]构建了涵盖网络、平台、安全三大体系的架构，为工业互联网实施提供指导。这些工作主要研究网络基础设施的体系架构，而本工作研究平台体系架构，两者不在同一层级。

算力互联网的目标是实现资源互联和应用互通，相关工作包括多云和算力调度两方面研究。在多云方面，文献[27]提出了应用复制、应用系统分层、应用逻辑分片和应用数据分片4种多云计算架构，有效增强数据和系统安全性。文献[28]通过成本效益分析和动态定价策略，论证了多云策略在资源优化方面的经济优势。文献[29]对比了Terraform和Cloudify，证实Terraform在部署速度、资源消耗

和易用性方面更优。文献[30]设计了分层故障容错模型，通过故障的自我检测和自动恢复，可提高物联网（IoT）应用的可靠性并解决异构IoT环境中的基础设施级故障。文献[31]提出的KubeTelecom引擎实现了多云多容器集群的统一管理，满足5G网络切片业务的云化部署需求。文献[15]与本文工作最为相关，它通过云平台代理创建一个细粒度的双边市场，将独立且互不兼容的云平台进行整合，并通过自动化任务调度和资源优化，支持用户在不同云平台间迁移计算任务。但该工作主要聚焦多云管理，且其架构采用提供器和执行器直接对接各云厂商接口的方式，不适合支撑全国范围大规模多云接入，此外该工作也未考虑多网环境现状。总之，上述工作侧重研究多云纳管内的功能实现，而本文则侧重研究多云多网统一管理的顶层架构。

在算力调度方面，文献[32]提出基于交叉熵的集中式不可分割任务调度算法，显著降低系统平均代价。文献[33]将任务调度问题建模为二维多重背包问题，采用动态规划算法充分利用空闲计算资源。文献[34]构建了基于XGBoost算法的任务资源需求预测模型，并提出自适应资源伸缩调度算法，有效提升资源利用率。文献[35]利用粒子群算法优化算力资源节点选择，提高了大数据资源调度效率。这些工作主要关注于调度算法研究。文献[36]创新性地将网络质量、带宽等参数纳入调度机制，实现了算力与网络的协同调度。通过K8s实现容器调度，利用可编程交换机实现网络参数采集，集中式控制调度能力。但该方法依赖可编程网络，仅适用于单云内算力调度，不适合本文的跨域多云和多网场景。文献[37]围绕AI应用场景，提出了支持云、网、边深度融合算力网络方案，采用算力调度和网络调度串行执行结构，但调度决策和调度部署功能的强耦合可扩展性弱，难以应对大规模多云多网场景。文献[38]将区块链技术应用于算力网络协同资源调度，在动态网络中展现出高鲁棒性。文献[39]提出的泛在化视频传输调度方案通过主/增强描述层分解和算力网络结合，实现了核心网负载卸载。文献[40]设计了一个算力网络资源协同调度平台，通过网络连接，整合多级算力和存储，统筹分配和调度计算任务，提供最佳资源分配方案，实现整网资源的按需最优分配使用。但该体系架构中算网模块直接对接算力资源池，难以适应大范围多云多网资源接入。综上，上述相关工作未关注多云多网场景，或相关设计不满足大范围场景可扩展性，不能直接解决本文所提问题。

为解决可用性感知的最少微服务数量模块划分问题，本文提出了最优微服务划分方法。为此本文调研了关于软件模块划分的相关研究。文献[41]提出一种多层贪婪模块化聚类方法以提升准确性、模块化质量及可扩展性。文献[42]使用扩展蚁群优化对软件系统进行高效自动再建模的方法，具备更高的模块化质量值及更低的时间复杂度。文献[21]提出了快速聚类算法，其根据各子模块间的依赖关系构造关联矩阵作为输入，然后依据子模块关联程度的强弱进行聚类划分。这些研究聚焦于软件模块的自动划分，主要满足内聚性目标，不适用于解决本文提出的问题。本文提出的划分方法考虑了模块可用度和划分后的微服务数量，可给出最优划分方案。

2 体系架构设计

本节从需求角度出发，设计互联互通平台体系架构。首先，分析市场上的各角色及角色间的关系。其次，基于各角色需求，形成设计原则。最后，基于设计原则，设计智算互联互通平台体系架构。

2.1 需求推演

当前的算力市场上有诸多角色，包括算网消费方、资源提供方、算网连接方和监管方。算网消费方是购买并使用算网资源的企业和个人。资源提供方是提供算力的厂商，如云服务和互联网数据中心（IDC）运营商等。随着消费方对互联互通需求的增多，市场上出现了算网连接方这一新角色。连接方的需求是通过连通提供方和消费方，提供算网一体化产品和解决方案。算网连接方要保证中立可信，一般由电信运营商和政府认可的企业承担该角色。监管方是各级政府机关单位，其需求是掌握全国算力资源底数，监管算力市场和制定政策等。

图1展示了各角色间的关系，算网消费方调用算网连接方的服务获取资源；算网连接方整合所有资源提供方的能力，为消费方提供算网资源调配服务；资源提供方向算网连接方提供云计算服务；算网连接方和资源提供方同时接受监管方的监管。为满足各方需求，需仔细设计算力互联网平台体系架构，确保可落地，支持增量式演进逐步升级算力互联网。本文采用演进式思路，基于现有基础设施，

分析哪部分成熟不易改变，哪部分可进行修改优化，基于分析结论来确定算力互联网平台体系架构的设计重点。

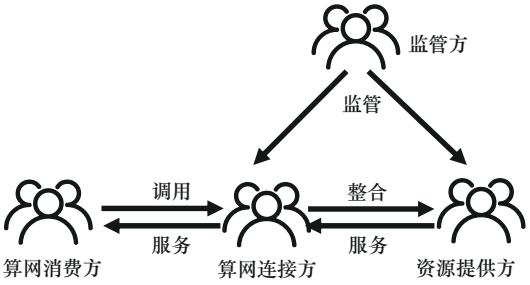

图 1　角色关系

当前阶段，算网消费方中的个人用户通常使用终端设备访问算力互联网，而企业通常采用成熟的私有化平台访问算力互联网，这些终端设备或平台成熟不易修改，并不在本文的设计范围内。资源提供方中的云服务商已有成熟的云管平台，其架构不易改变，需设计的部分在于其与其他平台的对接方式。算网连接方负责连接消费方与提供方，并提供多样的算网功能，因而需要构建技术平台。从政治和市场两方面需求考虑，可分为区域和行业平台。区域平台由地方政府投资的企业运营，纳管调度地方算力；行业平台则由各行业中的企业运营，纳管调度企业内外部算力。如何让区域/行业平台连通各资源池是体系架构的设计要点。电信运营商是特殊的算网连接方，其具有成熟的网管平台，可提供网络调度能力，如何实现网管平台和其他平台的连接也是设计要点。监管方由于需要收集全网信息，涉及复杂的数据处理功能，因此需要建设监管平台。上述分析到的各类平台将构成算力互联网的主要组成部分，本文给出算力互联网的平台关系并标出本文关注的架构设计范围，如图2所示。

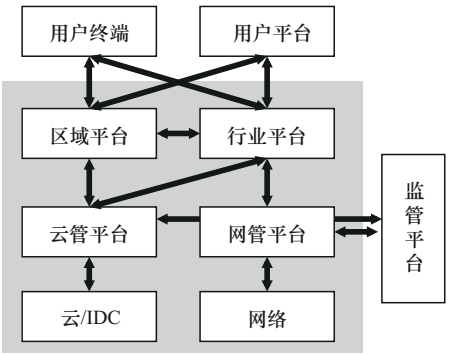

图 2　平台关系

最上层是用户终端和用户平台，通过连接区域平台或行业平台进行算力交易和管理。区域平台和行业平台向下连接云管平台和网管平台，实现多种云化资源和网络资源的统一管理。监管平台连接云管平台和网管平台。云管平台管理云或IDC资源，网管平台则管理网络。

2.2　设计原则

根据各角色需求，本节给出智算互联互通平台体系架构的设计原则。

1)分层解耦：从平台关系看，算力互联互通平台体系适合采用分层架构，按照平台关系自顶向下分成多个层级，层级间具有严格的上下调用顺序，避免跨层调用。分层架构天然适配角色分工，且功能解耦内聚，便于不同平台独立迭代，可增强互联互通体系的落地概率。虽然分层架构会造成性能和可用性的下降，但通过减少层级数量，仍可最大限度维持互联互通平台体系架构的性能和可用性。

2)分布式互联：从市场角度看，各区域/行业平台间适合采用分布式互联架构，即部分平台间建立连接关系。分布式互联天然适配市场环境，因为不同运营主体间关系复杂，只有建立合作关系的主体间才会进行资源产品服务等信息的发布。一种特殊情况是由企业机构团体组成的星形互联，即通过一个全局平台与其他平台连接，将各类信息统一汇聚并广播发布。这种架构需要在企业机构团体内达成一致的前提下才可能实现。从实施难度上看，广义的分布式资源互联更容易推广落地。

3)池间任务直连互通：从成本角度看，在计算任务跨资源池迁移时，宜采用池间直连互通的方式，即任务由一个资源池直接迁移到另一个，不通过第三方中转。直连互通的好处在于可减少任务迁移的网络成本开销，并最大程度保障性能；缺点在于需对每对资源主体间的迁移功能进行适配，有一定的复杂度。另一种方式是通过第三方平台中转，其好处在于只需适配中转平台和其他资源池间的迁移功能，复杂度较低。其缺点在于网络成本较高，需要额外第三方中转网络计费，难以落地。因此，当前更适于采用池间任务直连互通的方式实现跨多主体资源池的任务迁移。

2.3　体系架构

本节根据设计原则给出智算互联互通平台体系架构，并阐述各层的具体功能及相互联系，如图3

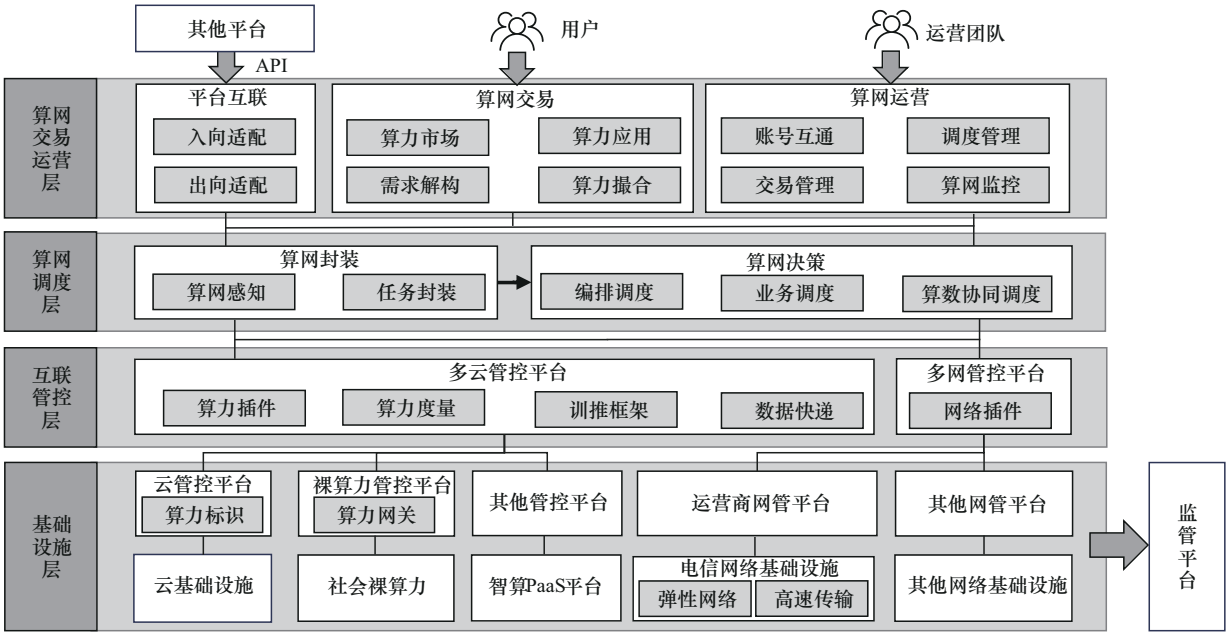

图3　智算互联互通平台体系架构

所示。按照分层架构原则，自顶向下分别是算网交易运营层、算网调度层、互联管控层和基础设施层。首先，算网交易运营层主要满足用户获取各种资源或产品信息的需求，其中用户关注算网交易、运营团队关注算网运营。其次，算网调度层根据用户需求，进行算网封装和算网决策，进行算网一体化调度，满足在多主体异构资源中选择最佳资源或产品的需求。然后，互联管控层基于多云管控和多网管控，负责对全国所有异构算力和网络资源进行统一管理控制，屏蔽底层接口异构性，感知算网资源并上报给算网调度层，同时接收算网决策并将命令下发给基础设施层，满足用户自由部署迁移智算应用的需求。最终，基础设施层基于上层管控指令，为用户提供算力资源，并通过基础网络连通各资源池，为平台间、用户到平台、用户到资源池间的通信提供网络服务。

接下来分析各层间及模块间的接口标准。对应到工业界，本文设计的体系结构内的上三层通常由同一家算网连接方建设和运营，因而算网交易运营层、算网调度层、互联管控层间的连接为内部接口，通常采用远程过程调用（RPC, remote procedure call）协议。而互联管控层与基础设施层间的连接通常只能采用云管平台暴露的北向OpenAPI，且这一接口不易定义统一标准，因为云厂商间存在天然的接口差异。上述提到的各层都包含多个功能模块，并基于微服务架构实现，将每个模块打包为独立镜像，以提高架构可扩展性。下面分别详细介绍各层的功能。

2.3.1　算网交易运营层

算网交易运营层包括平台互联、算网交易和算网运营模块。3个模块相互连接，共同实现算网交易运营能力。平台互联模块分别和算网交易、算网运营模块连接，将相关功能对外做统一暴露。算网交易需将账户信息及交易数据发送给算网运营，反之，算网运营需为算网交易提供账户和交易管理能力。下面分别介绍各模块的功能。

平台互联模块为不同用户、区域和行业平台提供适配接口实现对接功能，包括资源查询、产品上下架、账户认证和产品交易结算等。按交易方向分为入向和出向适配子模块。入向适配指本平台向其他平台查询信息或买入产品资源，其他平台作为提供方时对接本平台入向适配子模块，反之，则对接出向适配子模块。

算网交易提供产品资源应用视图、任务互通操作界面，可依据用户需求提供最佳算力产品的功能，分为算力市场、算力应用、需求解构和算力撮合等子模块。算力市场提供算力产品，用户可在此直接下单使用全网各种算力资源产品服务。算力应用提供算力软件即服务（SaaS）服务，如DeepSeek等。需求解构提供自动化用户需

求分析封装，为调度提供标准化输入。算力撮合提供最佳算力匹配功能，为用户提供最佳算力候选集合。

算网运营模块为运营团队提供对平台的统一管控功能，可分为账号互通、调度管理、交易管理和算网监控等子模块。账号互通可对用户账号进行生命周期管理，并与其他平台保持同步。调度管理可管理算网调度算法实例，支持算法的A/B测试，可评估算法效果。交易管理可对交易信息进行管理，维护所有算力服务商的交易状态。算网监控可分析处理算网资源数据，并为监控大屏提供统计数据。

2.3.2　算网调度层

算网调度层负责算网一体化调度，包括算网封装和算网决策模块。2个模块串行连接，算网封装将封装后的算网状态数据和用户任务信息发送到算网决策模块进行决策，共同实现算网调度能力。

算网封装包括算网感知和任务封装子模块。任务封装基于云原生设计，支持将用户提交的计算任务封装打包为标准化的容器镜像，以便在不同主体的算力资源池间进行无缝迁移。同时，还支持将计算任务与相关数据关联，实现计算存储的统一元数据封装，便于在计算任务迁移时定位相关数据源存储位置。算网感知可整合下一层模块上报的算网信息，通过数据分析模型建模等手段，为算网决策提供算网状态实时数据，支撑各类算网决策。

算网决策包括编排调度、业务调度、算数协同调度子模块。编排调度作为算网决策的核心模块之一，承担着计算任务与资源供给的智能匹配功能。该模块通过构建多维资源调度模型，采用强化学习等智能算法，在决策阶段综合评估算力节点地域分布、算力负载、网络状态及传输成本等多维度约束条件，实现算网资源的合理调度和分配。通过编排调度能力，结合算网封装子模块的输入，实现全域资源的创建和资源的最佳分配，解决算力资源的供需不匹配问题。此外，不同业务场景都有不同的固定基本指标需求。为此，本模块支持预置调度能力，如成本最优、距离最近、性能最优、综合最优等调度策略，实现全域资源根据特定业务匹配最佳资源的能力。该模块的技术突破在于实现了从任务解析、算网调度决策、算网资源交付的全流程自动化，将传统人工选择所需的数小时缩短至分钟级。

业务调度为用户提供访问算力服务的业务流调度能力，例如，针对不同区域部署的DeepSeek服务节点，为全国用户优化访问策略。适用于AI分布式推理和大数据等各种场景，这类场景通常需要大规模的计算资源，且对最终用户的访问体验有严格要求。在这种业务场景下，可定义不同智算任务的访问需求，结合智算资源负载利用率、访问时延、业务流量变化等指标，实现不同业务流按需合理调度，从而实现资源的高效利用。

算数协同调度是算存网融合的创新能力，通过存算网协同，可实现算力、网络和数据的联合调度决策，支撑计算任务的跨资源池迁移。通过全局资源视图实时感知各算力节点的负载状态、存储及缓存资源分布、传输路线成本，再基于传输任务特征，可分析生成最优资源配置方案。用户不需要关注算力与训练数据所在位置，平台自动帮助用户找到最优算力资源池和传输路径，实现训练数据跟随算力以高效或低成本方式自动迁移，达到存随算走的效果。

2.3.3　互联管控层

互联管控层负责对全国所有异构算力网络资源的统一管理控制，包括多云管控平台和多网管控平台。本层中的多云管控和多网管控采用分列方式，将云和网的管控能力解耦，相互间不进行连接，适配产业现状，以满足增量式演进需要。

多云管控平台的主要功能是纳管云计算资源、裸计算资源（即无云管数据中心）和智算平台即服务（PaaS）平台等，为上层提供统一的资源视图和管理功能，是智算互联互通体系的重要组成部分。其主要组成包括算力插件、算力度量、训推框架和数据快递模块。

算力插件为异构算力平台接入提供特异性适配和统一接入能力。为了将各种算力接入平台，一种直观简单的架构方案是按照各云管平台的特异性接口开发相应对接功能，并集成于一体，这种架构的缺点在于其功能过于耦合，造成软件僵化，不利于扩展。本文提出采用适配器模式，屏蔽异构资源接口，将各云管平台的对接功能解耦到每个插件中，提高体系架构扩展性和灵活性，如图4所示，每个云管平台的对接功能由一个独立的算力插件实现。

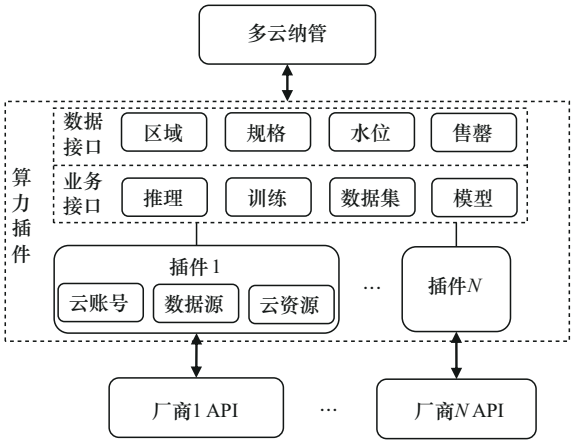

图4　算力插件子模块架构

具体来说，算力插件包含云账号、数据源和云资源3个部分功能。其中，云账号功能包含账号的创建与鉴权、实名认证等；数据源功能涉及区域、规格、水位等数据查询；云资源功能包含云资源的生命周期管理和查询等。针对异构资源的接入，算力插件首先收集现有资源产品列表，并针对接口进行一一映射，实现北向对管控平台内部多云纳管子模块提供统一抽象的接口，最终调用各厂商云管平台的南向应用程序接口（API）。北向接口分为业务和数据两种类型，业务接口主要功能是支持多云管控平台向各云管平台下发任务，数据接口主要用于多云管控平台向各云管平台查询资源池的实时状态。

为进一步增强架构可扩展性，本文还提出采用云原生方式实现算力插件部署，实现插件的增加删除查询能力，以便支持按需加载和热插拔。算力插件以独立服务方式开发，打包成镜像后，通过K8s容器部署。增加插件时，按需加载后，首先注册到多云管控平台，平台感知到新插件，通过统一接口调用插件实现连接。删除插件时，采用了云原生容器，解耦独立部署微服务，具备无状态特性，可直接删除，不会影响到上层模块的正常运转。插件的功能进行变更或扩展后，不需要重启多云管理平台，将插件重新部署即可使相关功能生效，从而实现插件热插拔。平台支持对于算力插件的心跳信号检查，可实时查询其在线状态。以上方案支持算力插件的快速横向扩展，在生产环境中可支撑至少百级别的算力资源提供商统一接入与异构算力资源感知，并通过K8s能力实现扩缩容，达到动态管理的能力。

算力度量的功能是对异构算力资源及业务进行性能和服务能力等指标的统一综合量化评估，可支撑算网决策和算网交易，协助算网调度进行更为准确的决策，其评估准确性将极大影响用算决策和服务体验。算力度量指标分为静态指标、动态指标及综合模型评估指标。静态指标包含了逻辑运算能力、存储容量等硬件固有属性，动态指标包括CPU利用率、内存剩余量、网络吞吐率等实时状态参数。综合模型评估指标通过计算、存储和网络带宽等指标，以数值化形式评估服务整体性能。

训推框架是指为智算训练推理提供统一的框架，如PyTorch、TensorFlow等，可适配各种国内外智算芯片，如英伟达、华为昇腾等。训推框架以镜像的方式提供，根据用户的算力需求，多云管控平台将通过云原生管理方式，在指定的资源池上识别芯片类型，并部署启动相应的训推框架镜像，以满足用户的智算训推需求。

数据快递是指为计算任务跨主体跨资源池间互通而提供的数据传输能力，该子模块接收算数协同调度的指令，并执行指令的部署动作。计算任务的跨池互通涉及计算程序和状态数据两部分。其中状态数据通常持久化在资源提供商的存储系统中，伴随计算任务的迁移，相应的状态数据也需要从源存储传输到目的存储中。基于池间任务直连互通原则，数据快递将直接打通跨主体资源池间的传输链路，其采用的技术方案是通过在源或目的资源池开通数据中转客户端，实现源和目的存储的读写功能，通过读取功能将数据载入内存，并通过写功能将内存中的缓存数据直接写入目的存储。通过这一方式，可避免为每对主体开发迁移功能，降低复杂度。

多网管控平台的功能是以统一视角管理多个网络基础设施，支持以全局视角聚合电信和其他异构网络管控平台，构建统一的网络资源视图，实时监测多类网络状态，智能分析流量需求与业务场景，动态调度跨网络资源。其技术创新性体现在通过插件化方式实现多种异构网络控制器标准化接入，提高系统架构扩展性。

网络插件子模块是多网管控平台与下层网络管控平台的桥梁，它将电信运营商网络管控平台、其他网络管控平台的差异化接口进行标准化处理，确保多网管控平台能无缝对接不同类型的网络管理系

统，消除异构网络间的交互障碍，确保不同网络管控平台在统一框架下协同工作。

2.3.4　基础设施层

基础设施层负责提供算力资源，并通过基础网络连通各资源池，为平台间、用户到平台、用户到资源池间的通信提供网络服务。基础设施层的算力资源包括云管控平台及云基础设施、裸算力管控平台及社会裸算力、其他云管控平台及智算 PaaS 平台；网络资源包括电信运营商网管平台和网络基础设施、其他网管平台和网络基础设施。

1) 算力资源

云管控平台是各云服务提供商的平台，其主要功能是提供算力资源管理 API。此外，为满足国家对算力互联互通的监管需求，还需在云管平台上增加算力标识功能子模块。算力提供商可借助算力标识子模块自动向政府监管平台注册，上报算力监测信息。目前，行业内普遍推行的是以中国信息通信研究院制定的算力标识行业标准。该标准定义了算力标识体系，通过多维标签对算力资源进行精准标识。标识由系列编码 ID 构成，涵盖城市、行业、企业、资源类型、数据中心、服务类型、计算能力、存储容量、网络配置、芯片类型及芯片唯一编号等关键维度。算力供应商需依据算力标识规范提交算力标识注册表，并通过算力互联互通平台获取唯一的算力标识注册代码，以证明其具备在特定可用区向互联互通平台上报算力资源的资质。

裸算力管控平台是指互联网数据中心等提供商的管控平台，它们通常没有完备的云管能力。为将其纳管接入算力互联网中，一种直观的方式是采用现有裸算力管控平台的基本能力，但这种方式会导致部分能力薄弱的资源池无法支撑智算任务的跨池互通需求。为此，本文提出采用算力网关实现对裸算力管控平台的云管能力增强。算力网关是一种可便捷部署的算存网资源一体管控平台，可对裸算力资源池进行快速轻量云化管理，使其具备算力管理、互联网络、存储管理和可信计算。算力管理支持虚拟私有云（VPC）划分、云硬盘、虚拟化等功能。互联网络支持零信任安全接入、专线或公网接入，实现跨资源池的安全数据传输。数据能力支持键值和文件缓存，支持智算场景下数据的快速访问。安全能力可支持对数据的文件加解密，并提供可信计算设备，实现可信计算能力。

2) 网络资源

电信运营商网络管控平台专注于电信网络基础设施的精细化管理，平台具备弹性网络与高速传输两大特性。弹性网络功能通过实时监测电信网络资源的使用状态，基于业务需求动态调整网络资源，提升资源利用率；高速传输功能通过技术手段保障数据传输满足稳定低时延、高带宽等要求，为智算等电信业务提供可靠支撑。

其他网络管控平台主要负责管理电信网络之外的多元网络基础设施。通过其他网络管控平台，多网管控平台可以间接对接多样化网络设施，实时获取网络状态数据，协调资源分配，保障其他网络基础设施的稳定运行，与电信运营商网络管控平台形成互补，共同构建多网融合的管理生态，满足复杂场景下的多元化网络需求。

2.4　互联互通流程

基于本文提出的体系架构，如何实现资源产品信息的互联和任务跨资源池的互通是一个重要的问题。本节分别给出平台内/平台间的资源信息互联，以及平台内的任务互通流程。对于跨平台任务互通，因涉及多运营商算力网络，在算力互联网初期推行难度大，因此可作为未来工作深入研究。

2.4.1　资源/产品信息互联流程

资源/产品信息互联流程如图 5 所示。对于平台内资源信息互联，首先由多云管控平台借助算力插件向各管控平台发起产销品列表查询请求，算力插件分析查询列表发送给云厂商，并在端口间进行一一映射，并由各管控平台返回产销品列表。最后由多云管控平台向算网封装模块更新产销品列表。当用户在算网交易模块中浏览产销品时可获得最新信息。

对于平台间资源信息互联，首先，由远端平台的互联模块以平台身份发起登录认证，由本地平台的互联模块进行身份认证，并返回登录结果。然后，由远端平台发起产销品列表查询请求，由本地平台的互联模块转发到算网封装模块进行检索。接着，由算网封装模块查询可向远端平台销售的产销品列表。最终，由平台互联模块返回所有结果。基于上述过程，即可实现资源产品信息的互联。

2.4.2　计算任务互通流程

计算任务互通流程如图 6 所示。任务互通涉及计算程序和状态数据的跨资源池流通。对于计算程

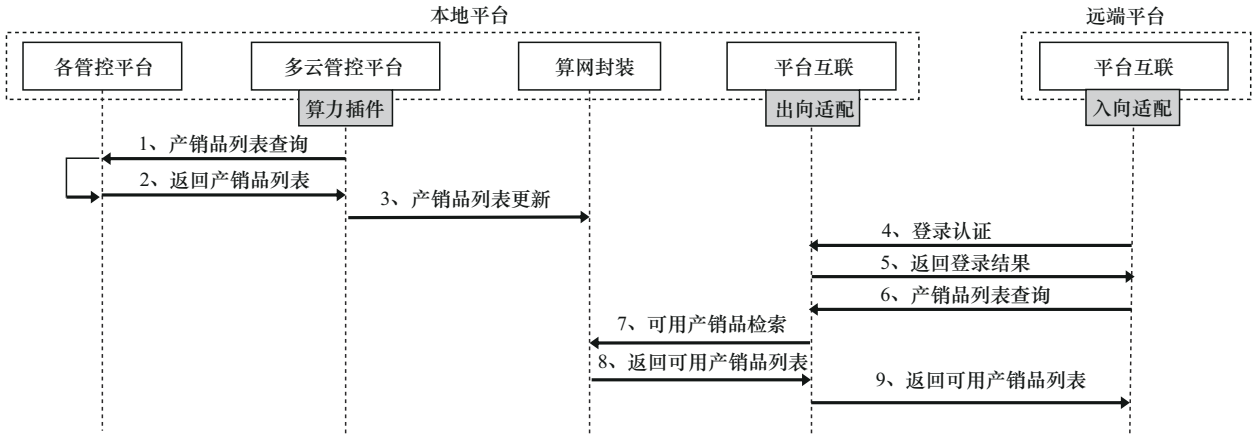

图5　资源/产品信息互联流程

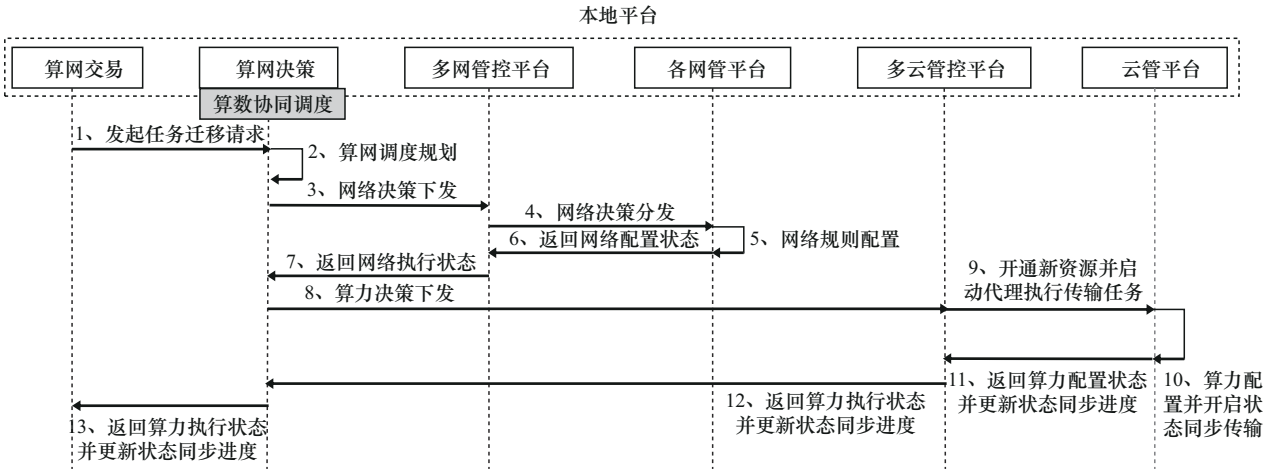

图6　计算任务互通流程

序的迁移，本质是在目的资源池上再次部署程序，这一部分的流程即算力部署流程。对于状态数据的迁移，主要通过算数协同调度、数据快递和弹性网络实现跨域数据流通。

首先，用户在算网交易模块发起任务迁移请求，请求包含源资源池信息、计算任务信息、目的资源池需求信息。请求将由算网决策模块进行算网调度规划，决策跨网传输路径后，将网络决策下发到多网管控平台。然后，由多网管控平台将网络决策分发到各网管平台，并执行网络规则配置，打通弹性网络传输路径，返回网络配置状态。在算网决策模块收到网络执行成功状态后，需要先在目的资源池开通资源用于部署计算程序，再传输状态数据，实现任务互通。为此，算网决策模块将算力决策下发到多云管控平台，并由该平台向目的云管平台开通新资源部署计算程序，同时由数据快递启动客户端代理执行状态数据的传输任务，实现计算任务的状态迁移。具体地，需要向代理提供状态数据的源地址和目的地址。最后，云管平台将同步资源开通、程序部署和状态同步的进度到多云管控平台，并逐级反馈到算网交易模块，为用户实时展示任务互通的进度。基于上述过程，即可实现智算任务的互通。

3　体系架构可用性分析及问题建模

为进一步优化算力互联互通平台体系架构的可用度，本节首先介绍相关可用度理论。然后基于理论对互联互通平台体系架构建模，并阐述可用度与微服务数量间的矛盾。最后定义问题并进行形式化。

3.1　串并联结构可用度

本文采用可用度理论中的串并联结构可用度计算方法对所提出的体系架构进行建模。一个串并联结构包含M个串联的模块集合，每个模块集

合 i 包含了 n_i 个并联的模块实例，其结构如图 7 所示。

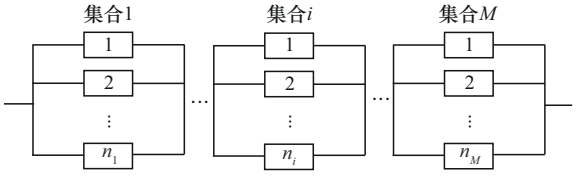

图 7　串并联结构

图 7 中，模块集合 1 由 n_1 个模块实例并联组成，其他集合以此类推。假设 A_i 为模块集合 i 的可用度，模块集合中各模块实例可用度均为 a_i，模块集合 i 中有 n_i 个模块实例，则可计算模块集合 i 的可用度为

$$A_i = 1 - (1 - a_i)^{n_i} \tag{1}$$

相对应地，串并联结构的总体可用度为

$$A = \prod_{i=1}^{M} A_i = \prod_{i=1}^{M} [1 - (1 - a_i)^{n_i}] \tag{2}$$

本文提出的智算互联互通平台体系架构中算网交易、算网运营、算网封装、算网决策和多云管控平台等部分可看作串并联结构中串联的若干模块。各模块包含若干子模块，以算网交易模块为例，包含算力市场、算力应用、需求解构、算力撮合 4 个子模块。各子模块以串联形式组合。同一模块内的多个功能相近子模块可划分到同一微服务中以降低运维成本，并以微服务为单位进行并联形式的资源备份。架构总体可用度模型为一个大串并联结构。

3.2　问题定义

本文采用微服务架构，在同一个模块内，可将每个子模块打包成一个微服务镜像，也可将多个子模块打包成一个微服务镜像。微服务数量越多，需要配备的运维团队规模越大，当前市场状况需尽可能降低运维成本，因此要减少微服务数量。需注意的是，将不同子模块打包到不同微服务中，会影响体系架构的总体可用度。下面以算网交易模块为例解释如何在一个模块内进行子模块的划分，并将它们打包进不同的微服务，如图 8 所示。该模块包含 4 个子模块，假设每个子模块的可用度为 0.9，所有子模块均正常运行时，模块才能正常运行，则算网交易模块的总可用度为 0.656 1。在无备份情况下，无论怎样将各子模块划分到微服务中（图 8 中虚线框代表一个微服务），该模块的可用度均为 0.656 1。但在实际部署中，为提高总体可用度，需部署备份微服务实例。此时模块的可用度将受到子模块划分方式的影响，且一个子模块故障会导致整个微服务故障。例如，将每个子模块打包为一个微服务，并给每个微服务增加单备份，则每个子模块的可用度将达到 0.99，模块总可用度达到 0.96，微服务数量为 4（不计算备份微服务数量）。如果将子模块两两划分到 2 个微服务中，则模块可用度达到 0.93，总计 2 个微服务。如果将 4 个子模块打包到一个微服务中，则总体可用度降低到 0.88。

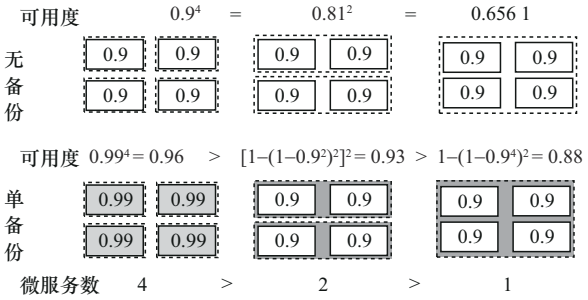

图 8　不同模块微服务划分对可用度的影响

从图 8 可以看出，微服务数量越少，总体架构的可用度越低。因而要合理地将不同子模块划分到各微服务中，使得在给定微服务备份实例数要求，且满足总体可用度要求的前提下，尽可能减少微服务数量，即可用性感知的最少微服务数量模块划分问题。下面对这一问题进行数学形式化。假设体系架构中总共有 M 个模块，其中第 m 个模块可被划分到 I_m 个微服务中，微服务索引为 i。记模块 m 包含 J_m 个子模块，子模块索引为 j，子模块 j 的可用度为 a_j^m，若该子模块被划分到微服务 i 中，则记二元决策变量 $x_{ij}^m = 1$，否则记为 0。则第 i 个微服务的可用度为

$$a_i^m = \prod_{j=1}^{J_m} [(1 - x_{ij}^m) + a_j^m \cdot x_{ij}^m] \tag{3}$$

为了得到 a^m，现引入二元变量 z_i^m，其值为 1 时表明微服务 i 中至少包含一个子模块，其计算方式为

$$z_i^m = 1 - \prod_{j=1}^{J_m} (1 - x_{ij}^m) \tag{4}$$

为方便管理，通常每个微服务备份实例数都相同，记为 k，可得第 m 个模块的可用度计算方式为

$$a^m = \prod_{i=1}^{I_m} z_i^m [1-(1-a_i^m)^k] + (1-z_i^m) \tag{5}$$

总体架构的可用度计算方式为

$$a = \prod_{m=1}^{M} a^m \tag{6}$$

在微服务备份数量为K、体系架构总体可用度要求为A时，通过为各模块找到最优的子模块到微服务划分策略X^m，在保证整体系统可用度满足要求的情况下，最小化微服务数量z。问题形式化描述为

$$\begin{aligned} &\text{obj}.X^{m*} = \arg\min_{X^m} \sum_m \sum_i z_i^m \\ &\text{s.t.}\mathrm{C}_1: k = K \\ &\mathrm{C}_2: a \geqslant A \\ &\mathrm{C}_3: \sum_{i=1}^{I_m} x_{ij}^m = 1 \end{aligned} \tag{7}$$

其中，约束C_1对约束备份资源数量为K，约束C_2保证系统可用度至少为A，约束C_3表明每个子模块至多分配到一个微服务内，优化目标中的决策变量为x_{ij}^m。

4 可用性感知的模块划分方法

本文提出的智算互联互通平台体系架构包含了平台互联、算网交易等模块。每个模块可被打包成多个微服务，也可为模块分配更多微服务数量预算以提升系统可用性。对应地，一个微服务可容纳同一模块内的多个子模块。本文提出基于两层决策的可用性感知模块划分方法，外层从1开始遍历额外微服务数量预算，并将微服务数量预算分配给各模块；内层则按照预算分配方案，在每个模块内生成最优子模块划分方案。在找到可行解后外层循环终止。

4.1 外层——微服务预算分配方法

为最小化微服务数量z，本文在原定每个模块分配一个微服务的基础上，额外增加h'个微服务数量预算，其中$h' = \sum_m I_m$，范围为$1 \leqslant h' \leqslant \sum_m J_m - M$，其中，$\sum_m J_m$是子模块总数，$M$是模块总数，$h'$也可称为微服务数量预算。每个模块$m$最多可拥有的微服务数量等于其子模块数。可从数量预算$h' = 1$的情况开始遍历，生成该微服务数量预算下的所有分配方案，并找到其中可用度最高的分配方案，增加h'重复上述过程直到总体可用度满足要求。

预算分配方案可通过动态规划算法生成，定义状态dp[p]表示前q个模块获得p个微服务数量预算的方案数（dp[p]关联了具体方案），初始时dp[0] = 1。对于每个模块，遍历所有可能的微服务数量，并更新状态数组。其具体流程如下。

初始化算法参数：数组dp[0⋯h']设置为0，dp[0] = 1

1)循环迭代次数m = 1,2,3,⋯,M；

2)创建临时数组next_dp，初始化为0；

3)循环微服务数量p = 0,1,2,⋯,h'；

4)若dp[p] == 0，则跳过；

5)计算模块最多可分配到的微服务，预算数计算方式为max_c = min(J_m,$h' - p$)

6)循环q = 0,1,2,⋯,max_c：

7) next_dp[$p + q$] += dp[p]

8) next_dp[$p + q$]基于dp[p]更新分配方案

9)将dp替换为next_dp

10)返回dp[h]中具有最大可用度的微服务预算分配方案。

4.2 内层——子模块划分方法

一个模块m包含J_m个子模块。若该模块的微服务数量预算是I_m个，则内层问题即如何划分J_m个小模块到I_m个微服务中以最大化该模块的可用度。其划分方案为第二类斯特林数$S(J_m,I_m)$对应的求解方法。其递推表达式为

$$S(J_m,I_m) = S(J_m - 1,I_m - 1) + I_m \cdot S(J_m - 1,I_m) \tag{8}$$

递推中止的边界条件为

$$\begin{aligned} &S(0,0) = 1, S(J_m,0) = 0 (J_m > 0), \\ &S(J_m,I_m) = 0 (I_m > J_m) \end{aligned} \tag{9}$$

所有划分方案可通过回溯法生成，最终在所有方案中选择具有最大可用度的划分方案即可。

4.3 时空复杂度分析

1) 微服务预算分配方法

微服务预算分配方法求解时使用动态规划表存储，dp及next_dp共同构成的表格大小为$(h' + 1) \times (M + 1)$，需要遍历所有分配情况，J_m是模块m包含的子模块数，则M个模块包含的平均子模块数为$\overline{M} = \frac{\sum_m J_m}{M}$，则时间复杂度为$O(M \cdot h' \cdot \overline{M}^M)$。

空间复杂度方面，需要存储搜索到的各类对应方案，复杂度与时间复杂度同阶。该方法适合在子模块数目较少时寻找最优解。

2) 子模块划分方法

子模块划分方法求解时维护一个二维表格 dp[J_m+1][I_m+1]，需要填充 J_m 行、每行最多 I_m 列（因为 $I_m \leqslant J_m$）。对于每个从 1 到 J_m 的循环，内层循环最多执行 $\min(i, I_m)$ 次。因为 $I_m \leqslant J_m$，总次数为 $I_m+I_m+\cdots+I_m$（共 J_m-k+1 次）加上 $1+2+\cdots+k$，整体为 $O(I_m J_m)$。递推公式需进行迭代循环，最坏情况整体时间复杂度为 $O(I_m{}^{J_m} \cdot J_m)$，但实际情况下剪枝后要远小于该值。空间复杂度方面，假设最终求解的第二类斯特林数为 $S(J_m, I_m)$，每个方案存储 J_m 个子模块的分配情况，则空间复杂度为 $O(S(J_m, I_m) \cdot J_m)$。

5　体系架构评估

本节在不同参数下，对所提出的智算互联互通平台体系架构的可用性和部署所需微服务数量开销进行评估。

5.1　实验设置

本文采用数值模拟开展实验，为合理评估采用子模块划分优化后的体系架构可用度和微服务数量开销，选择不划分作为基本对比方案，此外，还选择全划分，以及微服务主流划分方法——快速聚类算法（FCA, fast clustering algorithm）[21]作为对比方案，具体如下。

1)不划分：不对每层的模块进行划分，每个模块作为一个微服务，例如，平台互联和算网交易作为 2 个独立的微服务。不划分方法是最直接的方法，其微服务数量是最少的，但其可用性较差。

2)全划分：将每层的模块划分成最小单元，即以各子模块作为一个微服务，例如，平台互联划分为入向适配和出向适配 2 个独立的微服务。全划分可最大化体系架构可用性，但其微服务数量是最多的。

3)可用性感知划分：即本文提出的基于动态规划和回溯法的两层划分方法，可根据总体体系架构可用度，决策最佳划分方法。

4)FCA 划分：该算法根据子模块间的依赖关系构造关联矩阵作为输入，然后依据子模块关联程度的强弱进行快速聚类划分。其执行速度快，但会将不同模块的子模块划分到一个微服务中，导致架构可用性下降。本文根据图 3 构建各模块的连接关系作为算法输入。

除了对比方法外，还需对实验参数进行设置，包括每个子模块的可用度，总体体系架构可用度要求和备份实例数量。主流云厂商服务水平协议（SLA，service level agreement）中单应用可用度，以及软件可用度相关研究[43]的应用可用性参数范围为 0.99~0.999。为测试各类方法在低可用性情况下的表现，本文将子模块的可用度取值范围设置为“0.9,0.95,0.99,0.995,0.999”。总体体系架构可用度要求取值范围设置为“0.99, 0.999, 0.999 5, 0.999 9, 0.999 99”。备份实例数量取值范围设置为“1,2,3,4,5”。考虑到云计算行业会采用两地三中心等容灾方式，因此默认备份实例数量设置为 2。基于上述实验参数设置，本文对体系架构总体可用性和微服务数量 2 个指标进行评估。

本文采用演进式思路，基于现有基础设施，通过添加新功能（即各子模块）的方式建设算力互联网。现有成熟的基础设施具有较高的可用度，本节在计算可用度时，主要计算包含新功能模块的可用度（共计 10 个模块和 24 个子模块），以分析本文所提出的划分方法。

5.2　架构可用性评估

本节评估所提出的智算互联互通平台体系架构能否通过子模块划分优化的方法满足不同总体架构可用性要求，并比较分析不同划分方法间的结果差异，如表 1 所示，其中，可用性感知划分按照不同可用性要求分别展示相应结果。

从表 1 可以看出，采用可用性感知划分后的体系架构可用度，介于不划分和全划分之间。以第三列为例，相比于不划分方法，采用可用性感知划分方法对体系架构进行优化，可提高可用性；采用全划分方法，可将总体架构可用性最大化；采用 FCA 划分后的体系架构可用性较低，因为该算法以快速聚类为目标，对可用性的优化不足，其划分出的不同微服务包含的子模块数量不均衡，导致总体可用度不高。此外，当全划分方法能满足总体架构可用度要求时，可用性感知划分方法同样能满足要求，但不划分方法则不能保证。

5.3　微服务数量评估

本节对采用可用性感知划分方法后的智算互联互通平台体系架构微服务数量开销进行评估。作为

表1　架构总体可用度

方法	可用度为0.9	可用度为0.95	可用度为0.99	可用度为0.995	可用度为0.999
不划分	84.508 37	97.504 55	99.976 36	99.996 99	99.999 98
99划分	97.627 40	99.027 47	99.976 36	99.996 99	99.999 98
99.9划分	97.627 40	99.700 43	99.976 36	99.996 99	99.999 98
99.95划分	97.627 40	99.700 43	99.976 36	99.996 99	99.999 98
99.99划分	97.627 40	99.700 43	99.990 58	99.996 99	99.999 98
99.999划分	97.627 40	99.700 43	99.997 60	99.999 03	99.999 98
全划分	97.627 40	99.700 43	99.997 60	99.999 70	99.999 99
FCA	79.288 17	96.151 07	99.958 29	99.994 59	99.999 96

参考，全划分的微服务数量是24个，不划分和FCA划分的微服务数量均为10个。图9展示了在2个备份实例时，不同子模块可用度和总体可用度要求对微服务数量的影响。

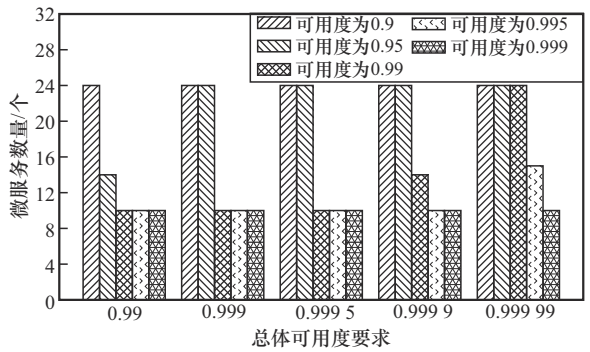

图9　子模块可用度对微服务数量的影响

从图9中可以看出，随着总体可用度要求的提高，微服务数量也随之增长，尤其是当子模块的可用度较低时（如0.95），更加需要通过划分模块的方式来提供总体可用度，因此导致微服务数量的快速增多，达到24个。当子模块的可用度极高时（如0.999），不需要对模块进行划分即可满足总体可用度要求。

图10评估了备份实例数和子模块可用度对微服务数量的影响。随着备份实例数量的增长，微服务数量逐渐下降，因为足够多的备份实例可以极大提高系统总体可用度，不需要划分出更多微服务。此外，当备份实例数量固定（如为2）时，随着子模块可用度的增加，微服务总数量快速下降。一个特殊的情况是备份实例数量为1且子模块可用度不高于0.99时，即使按照全划分方法将体系架构划分为24个微服务，也不能满足可用度要求。这一结果说明，为满足总体可用度要求，在必要时仍然需要提高备份实例数。

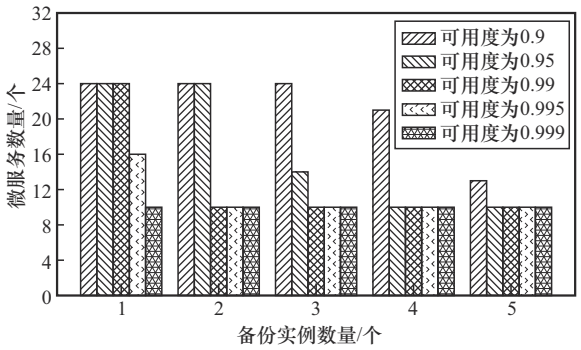

图10　备份数量对微服务数量的影响

6　结束语

本文为算力互联网的智算互联互通需求提出了一种高可用体系架构。通过算力插件和算网一体化调度技术，可支持算力资源信息互联、计算任务跨池互通。通过可用性感知模块划分方法，规划体系架构微服务划分方案，实现体系架构的高可用，并最小化微服务数量，降低运维开销。模拟结果表明，本文提出的划分方法可满足智算互联互通平台体系架构高可用要求。展望未来，智算互联互通体系结构仍需在多方面开展进一步研究，包括平台互联接口协议、多主体异构算力接入方法、AI驱动的智能化调度算法等方面，这些方向可作为未来研究进一步探索。

参考文献：

[1] CLARK D. The design philosophy of the DARPA Internet protocols[J]. ACM SIGCOMM Computer Communication Review, 1988, 18(4): 106-114.

[2] 余晓晖, 张恒升, 彭炎, 等. 工业互联网网络连接架构和发展趋势[J]. 中国工程科学, 2018, 20(4): 79-84.
YU X H, ZHANG H S, PENG Y, et al. Networking architecture and development trend of industrial Internet[J]. Strategic Study of CAE, 2018, 20(4): 79-84.

[3] 温小振, 常金凤, 吴美希. 综合算力发展现状与趋势分析[J]. 信息通信技术与政策, 2024, 50(2): 7-11.
WEN X Z, CHANG J F, WU M X, et al. Analysis of current situation and trend of comprehensive computing power development[J]. Information and Communications Technology and Policy, 2024, 50(2): 7-11.

[4] JAIN S, NGUYEN V, GRUTESER M, et al. Panoptes: servicing multiple applications simultaneously using steerable cameras[C]//Proceedings of the 2017 16th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN). Piscataway: IEEE Press, 2017: 119-130.

[5] 张惠瑀. 算力互联互通背景下的算力合作规制: 内涵、困境与实现路径[J]. 南京邮电大学学报(社会科学版), 2025, 27(4): 85-93.
ZHANG H Y. Cooperation regulation of computing power in the context of computing power interconnection: connotation, dilemmas, and implementation pathways[J]. Journal of Nanjing University of Posts and Telecommunications(Social Science Edition), 2025, 27(4): 85-93.

[6] 吴建平, 刘莹, 吴茜. 新一代互联网体系结构理论研究进展[J]. 中国科学(E辑), 2008(10): 1540-1564.
WU J P, LIU Y, WU X. Research progress of new generation Internet architecture theory[J]. Science in China (Series E), 2008(10): 1540-1564.

[7] MAYER R, JACOBSEN H A. Scalable deep learning on distributed infrastructures[J]. ACM Computing Surveys, 2021, 53(1): 1-37.

[8] BHARDWAJ R, XIA Z X, ANANTHANARAYANAN G, et al. Ekya: continuous learning of video analytics models on edge compute servers[C]//19th USENIX Symposium on Networked Systems Design and Implementation. Berkeley: USENIX Association, 2022: 119-135.

[9] ZHENG L, LI Z, ZHANG H, et al. Alpa: automating inter- and intra-operator parallelism for distributed deep learning[C]//16th USENIX Symposium on Operating Systems Design and Implementation. Berkeley: USENIX Association, 2022: 559-578.

[10] STOICA I, SHENKER S. From cloud computing to sky computing[C]//Proceedings of the Workshop on Hot Topics in Operating Systems. New York: ACM Press, 2021: 26-32.

[11] JAMSHIDI P, AHMAD A, PAHL C. Cloud migration research: a systematic review[J]. IEEE Transactions on Cloud Computing, 2013, 1(2): 142-157.

[12] GUNAWI S H, HAO M, SUMINTO R O, et al. Why does the cloud stop computing [C]//Proceedings of the 7th ACM Symposium on Cloud Computing. New York: ACM Press, 2016: 1-16.

[13] NALDI M. Evaluation of customer's losses and value-at-risk under cloud outages[C]//Proceedings of the 2017 40th International Conference on Telecommunications and Signal Processing (TSP). Piscataway: IEEE Press, 2017: 12-15.

[14] FROIS J, PADRÃO L, OLIVEIRA J, et al. Terraform and AWS CDK: a comparative analysis of infrastructure management tools[C]//Proceedings of the 38th Brazilian Symposium on Software Engineering (SBES 2024). Piscataway: IEEE Press, 2024: 623-629.

[15] YANG Z, WU Z, LUO M, et al. SkyPilot: an intercloud broker for sky computing[C]//Proceedings of 20th USENIX Symposium on Networked Systems Design and Implementation. Berkeley: USENIX Association, 2023: 17-19.

[16] SCHUHMANN C, BEAUMONT R, VENCU R, et al. LAION-5B: an open large-scale dataset for training next generation image-text models[C]//Advances in Neural Information Processing Systems. Massachusetts: MIT Press, 2022: 25278-25294.

[17] BROWN T B, MANN B, RYDER N, et al. Language models are few-shot learners[C]//Advances in Neural Information Processing Systems. Massachusetts: MIT Press, 2020: 1877-1901.

[18] HUANG M L, ZHU X Y, GAO J F. Challenges in building intelligent open-domain dialog systems[J]. ACM Transactions on Information Systems, 2020, 38(3): 1-32.

[19] 第十四届中国 IDC 产业年度大典[J]. 中国会展(中国会议), 2019, (22): 99.
The 14th annual ceremony of IDC industry in China[J]. China Conference & Exhibition, 2019(22): 99.

[20] HAUER T, HOFFMANN P, LUNNEY J, et al. Meaningful availability[C]//Proceedings of the 17th USENIX Symposium on Networked Systems Design and Implementation. Berkeley: USENIX Association, 2020: 545-557.

[21] TEYMOURIAN N, IZADKHAH H, ISAZADEH A. A fast clustering algorithm for modularization of large-scale software systems[J]. IEEE Transactions on Software Engineering, 2022, 48(4): 1451-1462.

[22] 徐恪, 朱敏, 林闯. 互联网体系结构评估模型、机制及方法研究综述[J]. 计算机学报, 2012, 35(10): 1985-2006.
XU K, ZHU M, LIN C. Internet architecture evaluation models, mechanisms and methods[J]. Chinese Journal of Computers, 2012, 35 (10): 1985-2006.

[23] 吴建平, 林嵩, 徐恪, 等. 可演进的新一代互联网体系结构研究进展[J]. 计算机学报, 2012, 35(6): 1094-1108.
WU J P, LIN S, XU K, et al. Advances in evolvable new generation Internet architecture[J]. Chinese Journal of Computers, 2012, 35(6): 1094-1108.

[24] 李文璟, 喻鹏, 张平. 6G智能内生网络架构及关键技术分析[J]. 中兴通讯技术, 2023, 29(5): 2-8.
LI W J, YU P, ZHANG P. Architecture and key technologies of 6G intelligent endogenous network[J]. ZTE Technology Journal, 2023, 29 (5): 2-8.

[25] 黄韬, 张晨, 肖玉明, 等. 服务定制网络体系架构的设计与思考[J]. 通信学报, 2024, 45(2): 1-17.
HUANG T, ZHANG C, XIAO Y M, et al. Design and research of service customized networking architecture[J]. Journal on Communications, 2024, 45(2): 1-17.

[26] 余晓晖, 刘默, 蒋昕昊, 等. 工业互联网体系架构 2.0[J]. 计算机集成制造系统, 2019, 25(12): 2983-2996.
YU X H, LIU M, JIANG X H, et al. Industrial Internet architecture 2.0 [J]. Computer Integrated Manufacturing Systems, 2019, 25(12): 2983-2996.

[27] BOHLI J M, GRUSCHKA N, JENSEN M, et al. Security and privacy-enhancing multicloud architectures[J]. IEEE Transactions on Dependable and Secure Computing, 2013, 10(4): 212-224.

[28] JOE W. The economics of the hybrid multicloud fog[J]. IEEE Cloud Computing, 2017, 4(1): 16-21.

[29] CARVALHO L R D, ARAUJO A P F D. Performance comparison of terraform and cloudify as multicloud orchestrators[C]//Proceedings of the 2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID). Piscataway: IEEE Press, 2020: 380-389.

[30] ALMURSHED O, RANA O, LI Y H, et al. A fault-tolerant workflow composition and deployment automation IoT framework in a multicloud edge environment[J]. IEEE Internet Computing, 2022, 26(4): 45-52.

[31] 武宇亭, 王旭亮, 全硕. KubeTelecom: 一种面向5G网络切片的多云多容器集群管理与运维引擎[J]. 电信科学, 2021, 37(12): 72-83.
WU Y T, WANG X L, QUAN S. KubeTelecom: a multi-cloud multi-container cluster management and operation engine for 5G network slicing[J]. Telecommunications Science, 2021, 37(12): 72-83.
[32] 巩宸宇, 舒洪峰, 张昕. 多层次算力网络集中式不可分割任务调度算法[J]. 中兴通讯技术, 2021, 27(3): 35-41.
GONG C Y, SHU H F, ZHANG X. Centralized unsplittable task scheduling algorithm for multi-tier computing power network[J]. ZTE Technology Journal, 2021, 27(3): 35-41.
[33] 李成华, 石胜涛, 李孝天, 等. 基于边缘算力协同系统的视频智能分析任务动态调度方法[J]. 电子与信息学报, 2023, 45(12): 4458-4468
LI C H, SHI S T, LI X T, et al. Dynamic scheduling method for video intelligent analysis tasks based on edge computing power collaborative system[J]. Journal of Electronics & Information Technology, 2023, 45 (12): 4458-4468.
[34] 杨明烜, 洪学海, 唐宏伟. 基于任务资源需求预测的人工智能算力调度[J]. 高技术通讯, 2024, 34(5): 475-485.
YANG M X, HONG X H, TANG H W. Artificial intelligence computing power cluster scheduling based on task resource demand prediction[J]. Chinese High Technology Letters, 2024, 34(5): 475-485.
[35] 金天骄, 栗蔚. 基于算力网络的大数据计算资源智能调度分配方法[J]. 数据与计算发展前沿, 2022, 4(6): 29-37.
JIN T J, LI W. An intelligent scheduling and allocation method of big data computing resources based on computing power network[J]. Frontiers of Data & Computing, 2022, 4(6): 29-37.
[36] 李铭轩, 曹畅, 杨建军. 基于可编程网络的算力调度机制研究[J]. 中兴通讯技术, 2021, 27(3): 18-22, 61.
LI M X, CAO C, YANG J J. Computing power scheduling mechanism based on programmable network[J]. ZTE Technology Journal, 2021, 27 (3): 18-22, 61.
[37] 雷波, 刘增义, 王旭亮, 等. 基于云、网、边融合的边缘计算新方案: 算力网络[J]. 电信科学, 2019, 35(9): 44-51.
LEI B, LIU Z Y, WANG X L, et al. Computing network: a new multi-access edge computing[J]. Telecommunications Science, 2019, 35(9): 44-51.
[38] 衷璐洁, 王目. 区块链赋能的算力网络协同资源调度方法[J]. 计算机研究与发展, 2023, 60(4): 750-762.
ZHONG L J, WANG M. Blockchain-enpowered cooperative resource allocation scheme for computing first network[J]. Journal of Computer Research and Development, 2023, 60(4): 750-762.
[39] 张旭光, 陈鸣锴, 魏昕. 算力网络支撑下的泛在化视频传输调度[J]. 计算机研究与发展, 2023, 60(4): 786-796
ZHANG X G, CHEN M K, WEI X. Ubiquitous video transmission scheduling supported by computing power network[J]. Journal of Computer Research and Development, 2023, 60(4): 786-796.
[40] 彭开来, 王旭, 唐琴琴. 算力网络资源协同调度探索与应用[J]. 中兴通讯技术, 2023, 29(4): 26-31.
PENG K L, WANG X, TANG Q Q. Collaborative scheduling of computing power network resources: exploration and application[J]. ZTE Technology Journal, 2023, 29(4): 26-31.
[41] SÖZER H. Evaluating the effectiveness of multi-level greedy modularity clustering for software architecture recovery[C]//Software Architecture. Berlin: Springer, 2019: 71-87.
[42] VARGHESE R B G, RAIMOND K, LOVESUM J. A novel approach for automatic remodularization of software systems using extended ant colony optimization algorithm[J]. Information and Software Technology, 2019, 114: 107-120.
[43] LI J Z, LU Q H, ZHU L M, et al. Improving availability of cloud-based applications through deployment choices[C]//Proceedings of the 2013 IEEE Sixth International Conference on Cloud Computing. Piscataway: IEEE Press, 2013: 43-50.

[作者简介]

鄢智勇（1981– ），男，湖北天门人，天翼云科技有限公司高级工程师，主要研究方向为算力互联网络、AI系统、AI应用、HPC网络等。

陈浩（1995– ），男，北京人，博士，天翼云科技有限公司初级工程师，主要研究方向为算力互联网络、SDN/NFV、互联网体系架构、算力度量等。

丁立戈（1997– ），男，河南南阳人，博士，天翼云科技有限公司初级工程师，主要研究方向为物联网、云计算、群智感知、强化学习等。

魏本洁（1984– ），女，广东梅州人，天翼云科技有限公司初级工程师，主要研究方向为云计算、算力互联网络等。

陈晓帆（1987– ），男，广东揭阳人，博士，天翼云科技有限公司高级工程师，主要研究方向为算网调度、算力度量、SDN/NFV、网络安全等。

胡建锋（1985– ），男，福建莆田人，天翼云科技有限公司初级工程师，主要研究方向为算力互联网络、算网调度、AI应用等。