- **时延**

《边缘计算中的算网融合趋势及研究进展_雷波》

《Computing Power Network: A Testbed and Applications with Edge Intelligence 》

- **网络平均拥堵指数**

它被定义为网络等待时间和资源利用率之间的比率。从网络的角度来看，我们的目标是最小化网络拥堵指数，以缓解网络的压力，并提供灵活的网络服务，这可以用以下方式来描述。

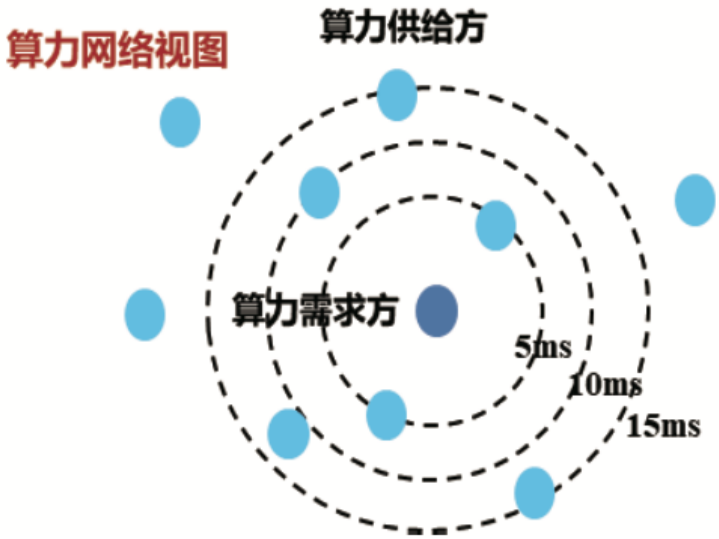
《Net-in-AI: A Computing-Power Networking Framework with Adaptability, Flexibility, and Profitability for Ubiquitous AI》

- **资源分配时间**

《Resource Reservation for Graph-structured Multimedia Services in Computing Power Network》

《边缘计算中的算网融合趋势及研究进展_雷波》

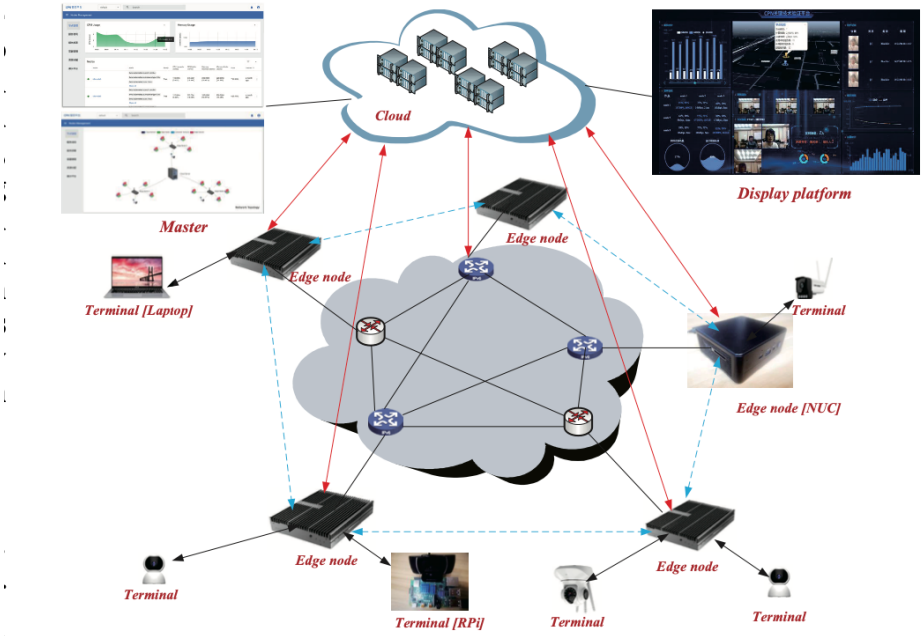
算力网络关键在于将算力资源的地址属性转换为网络时延属性，让使用者从模糊的、大概的距离判断， 转化成为精准的时延指标，以供各类智能业务按需调用资源，如图所示。



《算力网络：边缘智能的测试平台和应用》

摘要

算力网络（CPN）是一种新型的多接入边缘计算的发展，它有望以智能和灵活的方式应用无处不在的计算资源。在本文中，**我们实现了基于 Kubernetes 的微服务架构的 CPN 原型测试平台，实现了 CPN 的关键使能技术，包括计算建模、计算感知、计算公告和计算卸载。**我们**用具有智能推断功能的典型互联网服务评估了测试平台的性能，这些服务对延迟敏感且计算密集**。实验结果显示，与传统的边缘计算范式相比，我们的 CPN 测试平台可以实现更短的响应延迟和更好的负载平衡性能。



CPN 测试平台由三层组成：终端、边缘和云。

云：负责集群管理、数据持久化和网络性能指数监测
边缘：流式地处理和响应终端的各种边缘智能推理请求。

该测试平台的设计目标

- **延迟：**测试平台的优化目标是通过计算卸载策略使**请求的响应延迟最小化。**
- **分布：**所有的微服务都部署在集群的每个边缘节点上，其中要考虑数据同步、稳定通信和不同边缘智能应用之间的负载均衡策略；
- **智能：**平台应实现**智能边缘推理**、**智能服务调度**以及**智能资源管理**。

具体来说，CPN 测试平台的主要微服务有以下几个方面。

1. 计算意识

基于 Prometheus（Kubernetes 的官方监控套件），微服务定期收集各边缘节点的 ComNet 数据。

- 节点资源：计算和存储资源的总资源，以及实时资源利用率。
- 链路质量：链路带宽和建立 TCP 连接的延时。

2. 计算公告

通过 ZeroMQ 协议和分布式一致性算法，微服务聚集和同步其他边缘节点的计算能力信息，并合并到 ComNet 中。

3. 计算建模

把每个边缘节点提供的实时计算能力作为一个黑箱。根据预测试数据集中的 ComNet 和**响应延迟数据**，利用神经网络模型来评估边缘节点为几个边缘智能应用提供的计算能力，从而实现从 ComNet **到响应延迟**的服务导向映射。

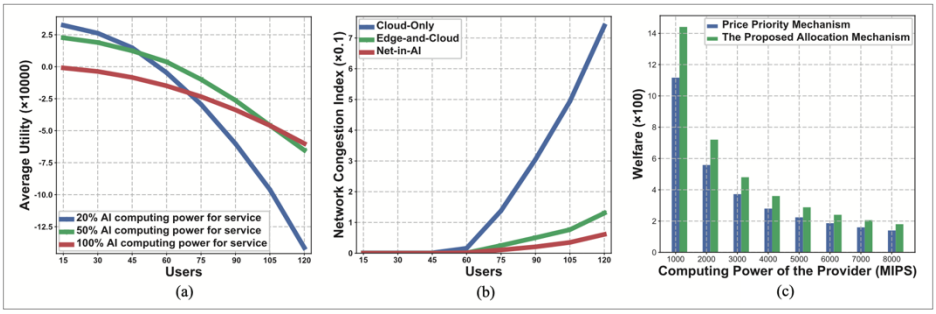
4. 计算卸载。

以**最小响应延迟**为优化目标，我们将计算卸载问题转化为加权双方图匹配问题，并提出一种改进的基于图论的算法（GT）来实现从请求到边缘节点的多对多匹配。我们还设计了一种自适应的算法选择机制，以实现不同网络状态下的最佳计算卸载性能。

《Net-in-AI: A Computing-Power Networking Framework with Adaptability, Flexibility, and Profitability for Ubiquitous AI》

贡献

- 提出了一个用于泛在人工智能的算力网络框架，通过在人工智能算力池中建立区块链和多级网络，表示为 Net-in-AI。这个框架实现了激励和智能计算网络化的融合
- 引入了一个人工智能计算能力分配机制，并将适应性（用户角度）、灵活性（网络角度）和盈利性（算力提供者角度）作为三个指标来评估综合人工智能计算网络系统的性能
- 从用户、网络和算力提供者的共同角度出发，提出了一个计算-网络分配问题。实验结果证明，与目前流行的框架相比，所提出的框架具有卓越的性能



《算力网络中图结构多媒体服务的资源预订》

摘要

在云、边缘和用户终端并存的计算网络环境中，计算密集型的多媒体业务得到了极大的发展。面对各种多媒体业务的高并发业务请求和对计算资源的不同需求，如何有效协调边缘节点和云节点的大规模计算能力，合理决定图结构多媒体业务的资源预留范围，以避免相互竞争的业务之间的资源冲突，是一个新的挑战。本文提出了一种针对并发多媒体服务的资源预留方法 CORA，该方法考虑了每个图结构多媒体服务中功能模块的交互以及多个并发服务之间的资源竞争。仿真结果表明，我们的方法可以获得合理的计算节点集进行资源预留，既能满足并发业务对计算资源的需求，又能避免多个业务之间的资源选择冲突。

背景

- **多媒体服务特点**

通常由各种不同的功能组件组成，共同完成多媒体服务的不同计算任，实现网络内边缘和云计算节点的任务协作。

- **传统的云计算和边缘计算不太适合多媒体服务**

他们通常只有一个集中的调度器来控制资源的分配。现有的研究通常集中在如何将服务分配给指定网络范围内的计算节点，但很少考虑当网络规模扩大时，集中式调度器无法及时获得所有资源节点的位置和计算能力。**因此，在调度多媒体服务和分配资源时，首选多个调度器。**

- **算力网络**

优点：可以自动收集本地的资源位置和计算力信息，然后在实施分布式控制协议后，为目标服务安排合适的节点。

缺点：

多种多媒体服务的并发请求让资源分配问题变得更加复杂。

不同的多媒体服务需要不同的计算能力。随着网络规模的扩大，寻找资源的效率变得越来越慢。

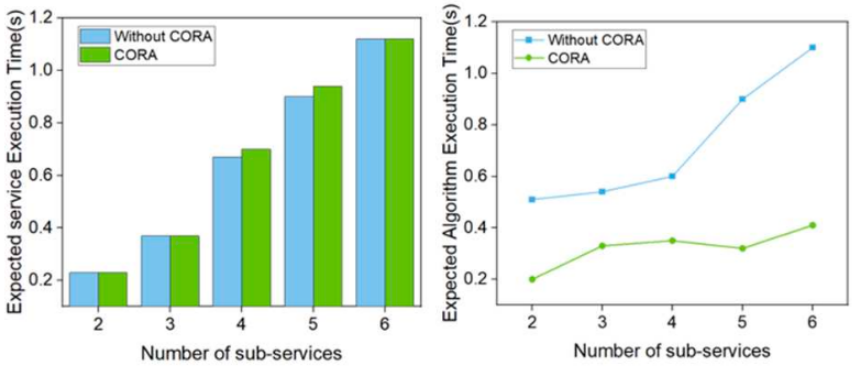
服务在多个调度器中寻找合适的节点，导致竞争冲突，影响服务的质量，并使服务不确定。

资源节点计算能力的变化导致网络中的调度器再次搜索其他合理的资源节点。因此，服务调度需要考虑鲁棒性，即在满足服务计算需求的前提下预留多个冗余的资源节点。

本文贡献

- 设计了一个基于利用率最大化的资源预留模型，即为每个服务保留一个合适的节点集，能够很好地满足每个多媒体服务的需求并防止冲突
- 基于这个优化模型，提出了 CORA 这种资源预留方法，它可以在有限的时间内获得可行的解决方案，并避免了传统方法中的高复杂性和慢收敛等问题。

- 实验对比指标


Without CORA
CORA
Expected service Execution Time(s)
1.2
1.0
0.8
0.6
0.4
0.2
2
3
4
5
6
Number of sub-services
Without CORA
CORA
Expected Algorithm Execution Time(s)
1.2
1.0
0.8
0.6
0.4
0.2
2
3
4
5
6
Number of sub-services