这是作者版本的文章，该文已被第九届网络机器学习国际会议（MLN 2026）接收。会议将于2026年12月1日至3日举行。

arXiv:2611.30699v1 [cs.NI] 23 Nov 2026

# Stress2Twin：面向 GuardedEdge 计算资源管理的基于配置文件的网络数字孪生

JulianJimenezAgudelo[1], PaolaSoto[1],AyatZaki-Hindi[2],Sébastien Faye[2],Nina Slamnik-Kriještorac[1], Johann Marquez-Barja[1], 和 Miguel Camelo Botero[1⋆]

[1] 安特卫普大学 – IMEC，IDLab，比利时
miguel.camelobotero@uantwerpen.be[2] 卢森堡科学技术研究院 (LIST)，卢森堡

**摘要。** 异构边缘系统需要资源管理器确定要部署多少应用副本、将其部署在何处，以及如何分配传入请求，同时满足服务级性能约束。通过详尽的多节点表征来获取这些决策所需的性能知识，需要大量测量；随着工作节点和配置数量增加，这一过程会变得愈发困难。Stress2Twin 是一个基于配置文件的网络数字孪生（NDT）框架，可对边缘节点进行独立性能分析，并将测得的行为组合成系统级资源管理模型。机器级预言机根据分配的请求速率和副本数量，对延迟、吞吐量、损失、排队和功耗进行表征。受保护优化器推导出运行区域，并在保留某一副本数量之前测试相邻的请求速率状态，从而在测得的非线性退化发生之前预留容量。对由三个工作节点组成的 Kubernetes测试平台进行独立性能分析，需要测量271个状态，约耗时4.5小时；相比之下，对所研究的集群配置空间进行详尽表征预计需要123小时。在单次递增评估中，所选策略在推导出的每秒请求数（RPS）1,650受支持范围内没有出现延迟或丢失阈值违规；与 RPS100 轮询（RR）部署相比，估计功耗降低了27.84%。在非单调工作负载下进行的另一项单次评估中，没有出现延迟违规，估计能耗也比稳定化 RPS50 RR 策略低19.67%。这些结果表明，独立获取的机器配置文件能够支持在测得的运行范围内进行容量感知资源整合。功耗节省采用理想化的活动工作节点

会计模型。

**关键词：** 网络数字孪生 · 多接入边缘计算 · 自动扩缩容 · 异构计算 · Kubernetes · 能源效率

⋆ 通讯作者。

## 1 引言

多接入边缘计算（MEC）将计算能力部署在靠近联网设备的位置。它支持对延迟敏感的物联网（IoT）分析以及由人工智能（AI）驱动的边缘推理。与同构数据中心不同，紧凑型边缘部署可能会结合处理能力、能耗成本和副本数量限制各不相同的机器。因此，此类环境中的资源管理需要共同决定应部署多少个应用副本、将其放置在哪里，以及如何在异构边缘计算节点之间分配传入的应用请求。作出这些决策需要了解候选配置在不同流量负载下的表现，但直接在集群级别获取这些信息很快就会变得成本高昂。近期关于零接触编排的研究同样强调，需要在异构边缘和云基础设施之间协调服务放置与扩缩容，同时考虑性能和能耗相关指标 [2]。

网络数字孪生（NDT）为表示此类性能知识并在将候选配置应用于物理基础设施之前对其进行评估提供了一种自然机制。国际电信联盟电信标准化部门（ITU-T）Y.3090 将 NDT 定义为一种虚拟表示，利用数据、模型和接口对物理基础设施进行分析、仿真和控制 [6]。以往研究已将数字孪生应用于能量感知边缘迁移 [3], 、仿真到物理的开放无线接入网（O-RAN）验证 [12], 、服务质量（QoS）感知的车载数据分发 [8], ，以及针对边缘资源的可信反事实假设分析 [1]。然而，这些研究并未解决这样的问题：如何在不进行详尽的系统级表征的情况下，根据独立测量的节点级行为，推导出联合扩展、服务部署位置和请求路由策略。

Stress2Twin 是一种面向异构边缘系统资源管理的配置文件驱动的 NDT 决策框架。它并不刻画完整的多节点配置，而是根据分配的应用程序请求速率和活跃应用副本，分别对每个边缘计算节点进行独立配置文件建模，组合所得的机器级模型，并推导联合扩展、部署位置和路由决策。其目标是确定资源高效的运行配置，在满足特定服务的延迟与丢失约束的同时，避免在接近实测饱和状态下运行。该模型涵盖多接入边缘计算基础设施的计算与服务路由域，而不包括完整的通信网络；无线接入条件、随时间变化的传输延迟以及网络拓扑动态不在当前范围内。

相关资源管理器以不同方式获取性能知识。Autopi-lot 将历史观测与启发式方法相结合，用于资源配置 [9],；Quasar 对工作负载进行分类，以实现面向服务质量的分配 [4], ；Ernest 则通过数量有限的实验推导性能模型 [11]。这些方法推动了数据辅助资源管理的发展，但并未基于对异构边缘工作节点进行独立测量所得到的配置文件，共同推导副本数量、部署位置和路由。因此，一个核心挑战是：如何在无需穷举性能分析的情况下，获取足够的性能知识，以支持这些联合决策

**表1**。与具有代表性的相关方法进行对比。

| 方法 | 知识 | 硬件异构性。联合[a] | 保护机制 | |
|---|---|---|---|---|
| Autopilot [9] | 历史记录/启发式方法 | – | – | – |
| Quasar [4] | 分类 | Yes | – | – |
| Ernest [11] | 稀疏实验 | – | – | – |
| 可信无损检测 [1] | 遥测数据/模型 | Yes | – | – |
| Stress2Twin | 每位工人档案 | Yes | Yes | Yes |

[a]联合副本数量、工作节点放置和请求路由。破折号表示所引工作未针对这一完整决策集。

完整的集群配置空间。表1将 Stress2Twin 与具有代表性的性能感知资源管理器和 NDT研究进行了区分。现有方法提供了有价值的历史模型、工作负载分类模型、学习得到的性能模型或假设分析模型，而 Stress2Twin 则专门将独立测量的异构工作节点配置文件，与副本、部署位置和路由的联合决策以及经验转换保护机制相结合。

随着工作节点数量和容量的增加，穷举集群性能分析会呈组合爆炸，这与数字孪生所面临的数据采集和验证挑战一致 [5]。对于最多有 $r_1$、.....、 $r_m$ 个副本的 $m$ 个工作节点，在考虑路由之前，每个流量级别就已有 $\prod_i (r_i + 1) - 1$ 种非空副本配置。例如，支持 8、4 和 4 个副本的三个工作节点，在每个提供负载级别下会产生 224 种配置。在以每秒请求数（RPS）衡量的 33 个提供负载级别上，若每种配置都测量 60 秒，则约需 123 小时。每当工作节点、工作负载或部署特征发生变化时，都重复进行这种穷举测量，这进一步限制了它们在构建和维护 NDT 时的实用性。

一种测量效率更高的替代方案是，针对每台机器所支持的分配负载与副本数量组合，分别进行性能分析，随后在集群级别组合这些测量结果。每个保留的配置文件状态代表一个经过测量的工作节点-负载-副本组合。对于每个副本数量，分配负载从 50 RPS 开始，每隔 60 秒以 50 RPS 为步长递增，直至观测到的请求丢失达到或超过 50%。仅保留损失低于 50% 的状态后，NUC0、NUC1 和 NUC2 共得到 271 个机器级配置文件状态。

对于递增评估中使用的每秒请求数为 50 的性能分析网格，每个状态测量约 60 秒，因此在保留的 271 个状态上总计需要 16,260 秒，即约 4.5 小时。这显著少于详尽集群级性能分析所需的约 123 小时，因为所提出的过程按照可接受的单机状态总数扩展，而不是按照集群范围配置的乘积扩展。这一缩减带来了第二个挑战：独立性能分析描述的是工作节点在隔离的稳态运行下的表现，不能直接被解释为安全的集群级运行点。

Stress2Twin 通过将可替换的机器模型组合成集群级 NDT 来弥合这一差距。受保护优化器根据性能分析结果推导运行区域，检查邻近负载，并在延迟与丢失约束下构建协调副本、活动工作节点和路由决策。这种容量感知的整合避免了仅基于负载的顺序扩展所带来的过载风险，也避免了轮询放置导致的工作节点持续激活。

本研究的主要贡献有三点。首先，Stress2Twin 提出了一种配置文件驱动的 NDT，将独立测量的机器级性能配置文件组合成集群级决策模型，从而减少对完整配置空间进行穷尽式特征分析的需求。其次，本文提出了一种受保护优化机制，根据延迟、吞吐量、损失和队列测量结果推导出每个工作节点的运行区域，并利用相邻负载状态制定协调的副本数量、工作节点激活和路由决策，同时避免在接近非线性退化的状态下运行。第三，本文在异构 Kubernetes 边缘部署上验证了由此得到的策略，量化了仿真到现实的准确性，以及与固定 RPS和延迟感知基线相比，在延迟违规、副本使用量、活跃工作节点和估计功耗方面的权衡；此外，还通过非单调工作负载评估补充了受控负载递增实验。

本文其余部分的结构如下。第 2 节阐述资源管理问题并介绍所提出的方法。第 3 节介绍实验平台和测量架构。第 4 节和第 5 节分别定义功能模型和基于配置文件的优化器。第 6 节评估由此得到的策略，第 7 节和第 8 节则讨论研究发现并总结全文。

## 2 问题形式化与所提方法

我们考虑一个由一组计算节点$\mathcal{N}$组成的异构边缘系统，这些节点在处理能力、能耗以及可承载的应用实例数量方面可能各不相同。我们将应用程序的每个可独立运行的实例称为副本。提供负载 $\Lambda$ 表示传入应用请求率。对于给定的 $\Lambda$，资源管理器必须确定三个相互关联的决策：部署多少个副本、这些副本应运行在哪些边缘计算节点上，以及应将多少比例的传入请求路由至每个活跃节点。目标是在满足特定服务的延迟与丢失约束的同时，避免不必要的副本部署和节点激活，从而限制资源消耗与功耗。

一个核心难点是获取评估候选配置所需的性能知识。直接进行集群级别的性能表征，需要测试副本数量、工作节点放置、流量分配和提供负载的各种组合。因此，候选配置的数量会随着工作节点数量和容量的增加而迅速增长；在硬件、软件或工作负载发生变化后重复进行这种表征，还会进一步增加测量成本。

Stress2Twin 通过将性能表征从集群级别转移到单个工作节点来解决这一问题。每个工作节点都会针对所支持的分配负载和副本数量组合进行独立分析。这些测量结果通过机器级预言机表示，用于估计候选运行点下工作节点的延迟、损失、吞吐量、队列行为和功耗。随后，集群级NDT会组合各个机器预言器的输出，以估计候选多工作节点配置的行为。

然而，仅有独立的稳态配置文件不足以保证安全运行。测得的状态可能满足最终的服务级约束，但同时已接近饱和转变；此时，分配流量稍有增加，就可能导致延迟或损失大幅上升。因此，Stress2Twin根据测得的配置文件推导出受保护运行区域，并评估当前候选状态以及相邻负载状态。由此得到的优化器会协同制定副本数量、活动工作节点和路由决策，同时在测得非线性退化发生前保留备用容量。

因此，该方法将特定于机器的行为（硬件或应用程序发生变化时可替换或更新）与基于组合机器模型生成集群级策略区分开来。下一节将在异构Kubernetes测试平台中具体应用这一建模方法。

## 3 系统与测量架构

### 3.1 边缘人工智能部署

该系统包含一个Kubernetes控制平面节点和三个推理工作节点，分别记为NUC0、NUC1和NUC2（图1）。控制平面节点负责协调实验并承载请求端点，但不运行推理Pod。NUC0是测得性能最强的工作节点，最多支持八个推理Pod。NUC1和NUC2各支持四个Pod。8-4-4的上限取决于可用硬件线程，每个Pod对应一个可运行的工作节点上下文。表2总结了计算平台。

**表 2**。实验计算平台。

| Node | 处理器 | 核心数/线程数 | 内存 |
|---|---|---|---|
| 控制节点 | Intel Core i3-1115G4 | 2/4 | 16 GB |
| NUC0 | Intel Corei7-8650U | 4/8 | 8 GB |
| NUC1 | Intel Corei5-7260U | 2/4 | 32 GB |
| NUC2 | Intel Corei5-7260U | 2/4 | 8 GB |

集群中运行的应用程序是用于 CIFAR-10 图像分类的容器化服务 [7]。每个副本分配一个线程和 1 GB 内存。

Envoy 使用权重 $W_1$、$W_2$ 和 $W_3$ 分别向 NUC0、NUC1 和 NUC2 分发请求。策略规定了每个工作节点上的副本数量和路由权重。优化器按照固定的按容量排序的整合策略激活工作节点，首先填充 NUC0；当提供负载超过活跃工作节点的受保护容量时，随后激活 NUC1 和 NUC2。

图1展示了物理部署和Stress2Twin数字孪生工作流。按照国际电信联盟电信标准化部门 ITU-T Y.3090 对网络数字孪生（NDT）的定义，NDT 是基于数据、模型和接口构建的虚拟表示；第4节中的功能模型提供了Stress2Twin的模型组件。物理系统包括Kubernetes集群、其边缘工作节点，以及提供流量服务和监控的组件。

针对不同提供负载和活跃副本数组合所采集的各工作节点测量数据，会汇总为可替换的机器预言器。功能模型将这些预言器组合起来，以估算候选配置的集群级性能。随后，受保护优化器利用这些估算结果选择副本数量、活跃工作节点和Envoy路由权重。通过这种方式，孪生体将物理系统测量数据与资源管理决策联系起来，并支持在集群上无需穷举测试每种配置即可评估配置。该实现采用由性能分析工作流填充的实测查找预言机。

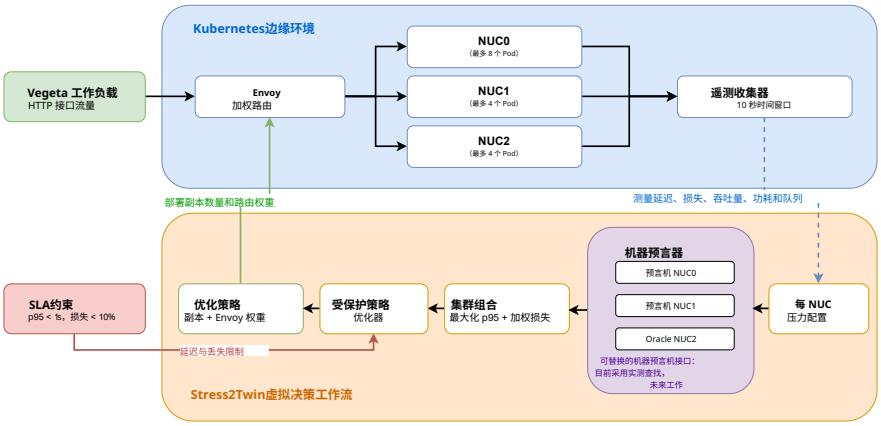


**图1**。Stress2Twin架构。各工作节点的测量数据用于构建可替换的机器预言器，这些预言器经过组合后，可在延迟与丢失约束下评估集群配置。受保护优化器会将副本数量和Envoy路由权重返回给Kubernetes部署。

### 3.2 遥测与阶段定义

Vegeta在同一局域网（LAN）上的一台独立客户端机器上运行。每个模拟客户端会从完整的CIFAR-10图像

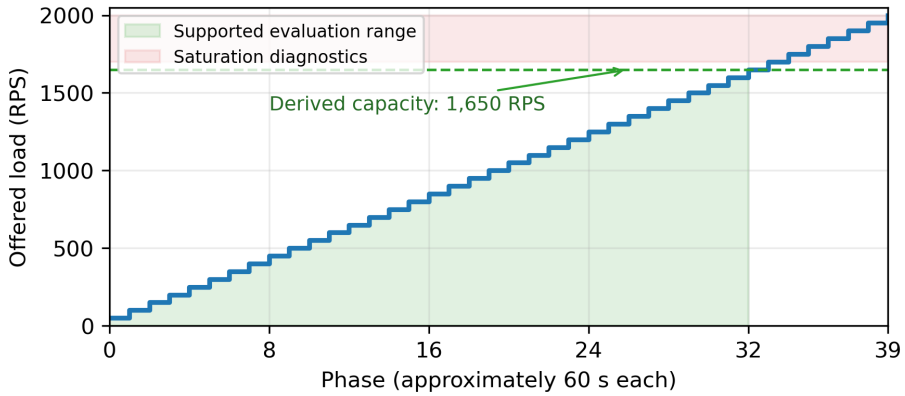


**图 2.** 40阶段评估负载递增及基于配置文件的支持范围。

数据集（共60,000张图像）中随机选择一张图像，将其序列化，并通过保持连接的超文本传输协议（HTTP）连接发送POST推理请求。保留的请求跟踪记录显示，请求大小中位数为2,492字节，响应大小中位数为1,622字节。Kubernetes（K8s）部署使用基于Python基础镜像构建的自定义$cifar\text{-}server$容器镜像，并配置了Envoy HTTPRoute权重和15秒的路由超时；未记录应用重试策略。kube-prometheus-stack会抓取该服务和采集器的数据。Kepler提供节点能耗指标，Grafana则用于检查。专用采集器聚合10秒时间窗口，将数据持久化到InfluxDB v2，并通过Kafka主题发布，供Prefect管理的NDT工作流使用 [1]。评估的部署使用K8s 1.35.7。原始请求和阶段摘要可得到成功请求的第95百分位延迟、损失、吞吐量、队列指标、副本数和功耗。损失等于1减去成功请求比例；观测到的失败包括HTTP 503响应和传输超时。

一个阶段是一个持续约 60 秒的恒定负载区间。评估由 40 个阶段组成，索引为 $k = 0, \ldots, 39$，对应的提供负载为 $\Lambda_k = 50(k + 1)$ 每秒请求数。因此，工作负载以每次 50 每秒请求数的步长从 50 增加到 2,000 每秒请求数。在每个阶段，都会应用与相应提供负载关联的副本数量和路由权重。图 2 展示了由此产生的工作负载递增。最高达到 1,650 每秒请求数的阶段构成支持的比较范围，而 1,700 至 2,000 每秒请求数的负载则保留用于饱和度诊断，并从性能分母中排除。

### 3.3 功耗核算

在任何跟踪记录中，物理工作节点都未被实际关闭。因此，我们通过在评估的策略保持某个工作节点处于活动状态时将其计入，来估算预期的部署功耗：

评估的策略会使其保持活动状态：

$$\begin{aligned} \widehat{P}_i &= 0, && \text{if } i \text{ is inactive,} \\ \widehat{P}_i &= 11\,\mathrm{W} + \max(P_{\mathrm{K},i} - P_{\mathrm{K0},i}, 0), && \text{otherwise.} \end{aligned} \tag{1}$$

11瓦项是独立测量的空闲墙上功率值，标准差为 0.3 W。此处，$P_{\mathrm{K},\,i}$ 是工作节点 $i$ 在评估状态下的 Kepler 估计值，而$P_{\mathrm{K0},\,i}$ 是该节点特定的无负载 Kepler 参考值。第二项保留了高于该参考值的负载相关 Kepler 分量。集群功率是所有活跃工作节点功率之和。由于各阶段时长相同，平均估计功耗的差异与评估轨迹中估计能耗的差异成正比。

## 4 Stress2Twin功能模型

### 4.1 机器级预言机

对每个工作节点，分别在测得的分配负载 $\lambda_i$ 和活跃副本 $n_i$ 组合下进行压力测试（图 3）。请求级测量值按工作节点、负载和副本数量进行汇总。对于每种状态，直接根据成功请求延迟计算第95百分位，而不是对 10 秒窗口百分位数取平均；采集器窗口则提供队列和能耗摘要。由此得到的预言机实现了

$$f_i(\lambda_i, n_i) \to \{p95_i, \ell_i, \mu_i, P_i, q_i^{95}, d_i^{95}\}, \tag{2}$$

其中，$p95_i$ 是成功请求延迟，$\ell_i$ 是请求丢失，$\mu_i$ 是成功吞吐量，$P_i$ 是式 (1) 中使用的、随负载变化的 Kepler 功率分量，$q_i^{95}$和 $d_i^{95}$分别是第95百分位排队时间和队列深度。完整的$\widehat{\ }$估计活跃工作节点功耗为 $P_i$。该接口与具体实现无关：查表预言机之后可以替换为学习模型，而无需更改优化器。在构建策略时，优化器根据测量得到的性能剖析网格状态推导其曲线。模拟过程中，对于不可用的负载，优化器会映射到相同副本数量下最近的测量负载；只有当请求的副本数量没有测量状态时，才会考虑其他副本数量。

### 4.2 集群组合

对于分配负载之和为集群负载 $\Lambda$ 的候选方案，集群预测采用保守方式组合：

$$p95_{\mathrm{cluster}} = \max_i p95_i, \qquad \ell_{\mathrm{cluster}} = \frac{\sum_i \lambda_i \ell_i}{\Lambda}. \tag{3}$$

最大延迟可防止负载较轻但已过载的工作节点被平均值掩盖。损失按请求加权，因为每个工作节点接收的

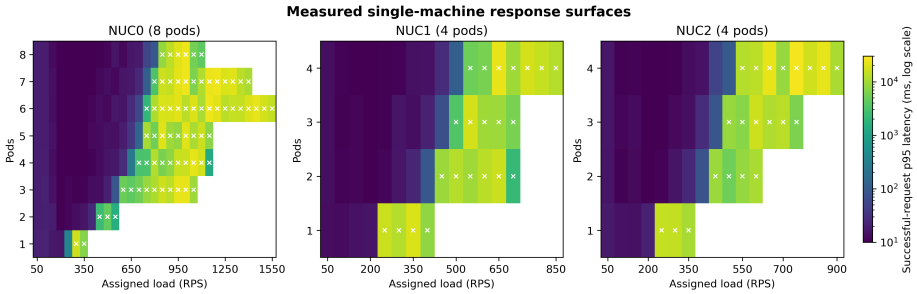


**图 3.** NUC0、NUC1 和 NUC2 的实测 p95 延迟配置文件。白色叉号标示违反一秒延迟SLA的状态。

offered流量份额不同。可行候选方案必须满足 $p95_{\text{cluster}} < 1$，000 毫秒和 $\ell_{\text{cluster}} < 10$ %。

延迟和丢失率阈值是服务特定输入。将 Stress2Twin 应用于其他服务时，需要在目标硬件上对其进行性能分析，并推导相应的运行区域边界和受保护容量。

## 5 基于配置文件的策略优化

### 5.1 稳态优化器

稳态优化器会枚举实测工作节点状态，使其分配负载之和等于请求的集群负载。在按递增负载水平构建策略时，它会剔除工作节点负载或副本数量减少的配置。候选配置按约束违反程度、活跃工作节点数量、副本总数和功耗的字典序进行排序。因此，优先选择同时满足两项服务约束的配置，其余目标则用于在这些可行配置中进行选择。此过程根据各个单独配置文件生成高效的运行点，并定义了未保护基线；随后，受保护策略会推导每个工作节点的副本曲线，并根据第 5.3 节的按容量排序的激活规则将其组合起来。由此得到的策略曲线在负载递增时保持单调。在可变流量情况下，当前负载决定采用与之对应的配置，而第 6.5 节所述的缩容保持时间则控制副本缩减。

### 5.2 实证运行区域

受保护优化器区分三个运行区域。稳定区域包含低延迟、吞吐量充足且排队受限的状态。肩部区域 是过渡区域，在此区域中，延迟和队列指标上升，但服务尚未崩溃。崩溃区域 包含最大分布偏移之后的状态，此时延迟或损失急剧增加，吞吐量

不再随提供负载变化。这些区域描述了每个工作节点的测量负载—副本性能空间。运行点由分配的请求速率和活跃副本数定义，而测量延迟、吞吐量、损失和队列指标则决定其所属区域。

边界是划分三个区域的阈值。这些阈值根据271个符合条件的配置状态得出，未使用集群策略结果。首先，log(1 + $p95$)中最大的相邻间隔将肩部区域与崩溃区域分开。其次，在崩溃前状态中，100毫秒稳定性锚点附近的相邻第95百分位间隔将稳定状态与肩部区域分开。该锚点用于选定观测到的间隔，其中点定义边界。由于两个第95百分位阈值对应的间隔是在对数刻度上识别的，因此采用几何中点；吞吐量阈值则采用其相邻线性刻度间隔的算术中点。计算保留了测得的端点：

$$\begin{aligned} b_{\text{stable}} &= \sqrt{(98.0+1)(111.0+1)} - 1 = 104.3 \text{ ms}, \\ b_{\text{collapse}} &= \sqrt{(372.38398+1)(1{,}346.20916+1)} - 1 \\ &= 708.243 \text{ ms}, \\ r_{\min} &= (0.9549 + 0.9807)/2 = 0.9678. \end{aligned} \tag{4}$$

队列保护阈值是稳定状态中观测到的上限，而队列外部限制则根据崩溃前状态得出。损失仍是实验约束，而非拟合出的边界。与 Kneedle [10]，在有序曲线上寻找单个拐点不同，此过程是在离散多指标状态集上划分区域。表3列出了相关证据及由此得出的阈值。

**表3**。由配置文件推导的运行区域边界。

| 边界 | 观察到的分离 | 阈值 |
|---|---|---|
| 稳定区/肩部区域 p95 | 98.0–111.0 毫秒 | 104.3 毫秒 |
| 肩部区域/崩溃区域 第95百分位 | 372.4–1,346.2 毫秒 | 708.2 毫秒 |
| 队列时间内侧/外侧护栏 | 稳定上限/崩溃前上限 47.2/153.6 毫秒 | |
| 队列深度内侧/外侧护栏 | 稳定上限/崩溃前上限 1/4 个请求 | |
| 吞吐量比 | 0.9549–0.9807 | 0.9678 |
| Loss | 实验约束 | $< 10\%$ |

对于每个工作节点，受保护容量是指能够通过当前状态护栏和邻近状态外包络测试的某个可接受副本数所对应的最大分配负载。应用表3可得，NUC0 的受保护容量为每秒请求数 750，NUC1 和 NUC2 则各为每秒请求数 450。三者之和构成每秒请求数 1,650 的受支持区域。固定激活顺序依据实测容量和整合目标确定。首先填满 NUC0，随后是 NUC1 和 NUC2。

### 5.3 两步邻域测试

对于给定的工作节点，副本数量 $n$ 决定了有多少个应用程序并行实例可用于处理其分配负载 $\lambda$。增加 $n$ 可提升处理能力，并在提供负载增长时减少排队和延迟，但需要消耗更多计算资源。相反，当 $\lambda$ 接近工作节点的容量时，若保留的副本过少，工作节点可能会进入肩部区域或崩溃区域。因此，邻域测试用于确定随着负载增加，当前副本数量能否安全保留，而不是等到观察到 SLA 违规后才扩增副本。

对于测量负载 $\lambda$ 和候选副本数量 $n$，所选优化器会检查当前状态以及接下来的两个 50 RPS 性能剖析点。只有当当前点位于稳定保护区域，且两个邻近点均处于崩溃前外包络范围内时，才会保留$n$ ：

$$\begin{aligned}\mathrm{keep}(n,\lambda) \Leftrightarrow\ & \mathrm{guard}(n,\lambda) \wedge \\ & \mathrm{outer}(n,\lambda+50) \wedge \mathrm{outer}(n,\lambda+100).\end{aligned} \tag{5}$$

如果当前副本数量未通过邻域测试，优化器就会增加副本数量，并选择能够为当前负载和邻近负载提供安全运行点的最小更大副本数量。这样既能避免不必要的过度配置，又能在工作节点达到测量饱和区域之前预留足够的处理能力。由此得到的每个工作节点的副本曲线均受到单调性约束。经过评估的受保护策略不会搜索任意部署位置：在集群级别，它会按照固定的按容量排序的整合策略激活工作节点，并根据分配负载计算整数路由权重。前瞻在负载空间中进行：50 和 100 RPS 的距离分别对应性能分析网格上的一个和两个点。

## 6 实验评估

### 6.1 基线与指标

真实部署评估包括两个基于规则的基线系列。固定 RPS 策略按每 50 或 100 每秒请求数分配一个副本。轮询（RR）在整个实验期间保持三个工作节点均处于活跃状态，按周期顺序放置后续副本，并根据活跃副本数量按比例分配流量。因此，这些基线中的工作节点数表示策略保持活跃的工作节点，而非某个阶段中仅承载副本的工作节点数。顺序激活会先填满 NUC0，再填满 NUC1，最后填满 NUC2，但没有基于实测容量的保护机制。两种固定 RPS 策略的上限均为平台的 16 副本上限。RPS50 在超过 800 每秒请求数时触及上限，RPS100 则在达到 1,650 每秒请求数时触及上限；若没有该上限，RPS50 的平均副本数将为 17.0，而不是 12.36；在最后阶段，RPS100 将请求 17 个而非 16 个副本。延迟感知型 Kubernetes 事件驱动

自动扩缩容（KEDA）控制器[3] 每 10 秒评估一次 30 秒窗口的 Prometheus 第95百分位值，并在其超过 500 毫秒时进行扩缩容，从而在 1 秒服务级别协议之前留出名义上的 50% 余量。它使用 1 至 16 个副本、15 秒的服务监控间隔和 300 秒的冷却时间，后者通过缩减稳定化实现。其跟踪数据不包含足够的逐工作节点路由状态，因此无法报告平均活跃工作节点数。原生的基于中央处理器（CPU）的 Kubernetes 水平 Pod 自动扩缩器（HPA）[4] 虽已配置，但未触发，且未保留时间对齐的 CPU 跟踪数据，因此不纳入评估。

Stress2Twin 是一种工作负载索引策略。对于固定 RPS 和 Stress2Twin 部署，所提供的阶段负载会选取预先计算好的配置，并在每个阶段边界前 15 秒应用该配置，且就绪超时时间为 120 秒。KEDA 通过滚动窗口检测性能退化。因此，固定每秒请求数策略提供了直接的配置层面对比，而 KEDA 则代表一种反应式的基于知识的控制器。

我们报告活跃副本平均数、活跃工作节点平均数、估计功耗、p95 延迟超过一秒的阶段数、请求丢失总量，以及丢失率超过 10% 的阶段数。总丢失率是指在比较区间内所有已提供请求中未能成功完成的比例。所有比较均使用相同的 33 个阶段，负载范围为每秒 50 至 1,650 个请求。评估围绕四个问题展开。第一，邻域保护能否改善部署策略相较于稳态选择的表现？第二，在受支持范围内外，预言机预测与部署结果的吻合程度如何？第三，所选策略在服务合规性和资源使用方面与基于规则的扩缩器相比如何？最后，在非单调工作负载下，同一种配置文件驱动方法是否仍然有效？为支持结果复现，用于生成所报告结果的数据集、分析代码和脚本将发布在 Zenodo 上。[5]

### 6.2 优化器消融

表4比较了三个级别的转换保护，同时保持相同的配置文件来源、组合规则和激活顺序。稳态优化器使用的副本数最少，但记录了21次p95违规。在相应的单次运行中，一步邻域出现一次违规，而第二步未出现违规；相较于一步前瞻，平均副本数仅增加0.39。平均功耗基本保持不变，因为备用副本通常会添加到已经提供服务的工作节点上。

[3] KEDA Prometheus 扩缩容器：https://keda.sh/docs/scalers/prometheus/.[4] Kubernetes HPA：https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale/. [5]https://zenodo.org/records/23804238.

**表4**。达到1,650每秒请求数的渐增部署优化器消融实验。

| 优化器 | 副本 功耗（W） | p95 > 1 s 损失 | > 10% | |
|---|---|---|---|---|
| 稳态配置 | 7.30 | 48.51 | 21 | 0 |
| 单步保护 | 8.03 | 52.41 | 1 | 0 |
| 两步保护 | 8.42 | 52.31 | 0 | 0 |

静态配置文件可以识别有用的运行区间，但最终服务级别协议对于选择边界而言过于宽松。对于观测到的负载爬升，在拐点之前预留容量后，随着邻域从零步增加到两步，违规次数逐渐减少。不同策略之间的对比表明，最终服务级别协议可行性与保护式可行性会导致实际部署行为出现实质差异，不过仍需进行重复运行才能量化其变异性。

### 6.3 预测策略结果与实际部署

图4将预言机预测与部署相同策略后获得的结果进行了比较。表5将受支持区域与七个饱和阶段区分开来。在受支持区域内，受保护策略的第95百分位均方根误差（RMSE）为75.8毫秒，预测偏差为 $-27.2$ 毫秒。尽管相对于一秒的SLA而言这一数值较小，但相对于稳定状态的几十毫秒延迟，75.8毫秒仍然不可忽视。log-p95均方根误差为0.305个数量级，相当于典型的2.02倍乘法误差因子。其损失平均绝对误差（MAE）为0.0005个百分点，功率均方根误差为2.12瓦。这些误差取决于各策略所选择的配置，并不比较相同运行状态下的预言机准确度。

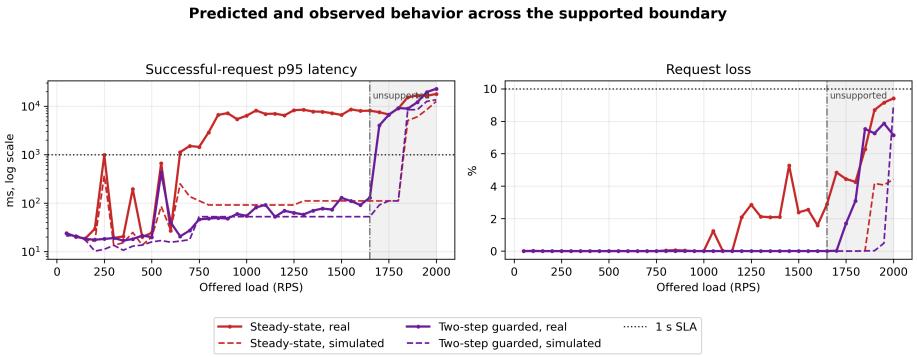


**图4**。 预测延迟、观测延迟与损失。阴影区域超出了推导得出的每秒 1,650 个请求的受支持范围。

**表5.** 按运行范围划分的预测误差。偏差是预测延迟减去观测延迟的平均值；错误安全表示预测值低于两个 SLA 限值，但在部署中违反了限值。

| 策略 | 范围 | p95 均方根误差 (ms) | 偏差 (ms) | 对数均方根误差 (几十年) | 错误安全阶段 |
|---|---|---|---|---|---|
| 稳态 | 50–1,650 | 5,241.3 | −3,943.9 | 1.411 | 21/33 |
| 两步 | 50–1,650 | 75.8 | −27.2 | 0.305 | 0/33 |
| 稳态 | 1,700–2,000 | 8,273.3 | −8,084.6 | 1.235 | 3/7 |
| 两步 | 1,700–2,000 | 6,377.9 | −5，653。5 | 1.172 | 3/7 |

假安全阶段是指预言机预测结果低于两个SLA限值，但部署结果违反了至少一个限值的阶段。因此，它衡量的是根据预言机输出所作出的错误可行决策。受保护策略在受支持区域内不存在假安全阶段。其首次违规发生在每秒1,700个请求（RPS）处，比推导容量高出一个50 RPS的网格步长；在每秒1,650个请求以上的全部七个阶段中，均违反了某项SLA。对于该策略在副本数量达到最大值时的情况，这一观察结果与工作节点容量总和相符，误差不超过一个网格步长。

### 6.4 基于规则的基线与资源效率

表6对比了仅采用整合的策略与同时使用实测单个工作节点容量的策略。

**表6**。 爬升至1,650 RPS的单次运行爬升结果。

| 策略 | 副本 | 有功功率 (W) | 第95百分位违规次数 | 损失 (%) | 损失违规次数 | |
|---|---|---|---|---|---|---|
| RPS50 轮询 | 12.36 | 3.00 | 71.40 | 2 | 0.0704 | 0 |
| RPS50 顺序策略 | 12.36 | 2.39 | 63.89 | 2 | 0.0377 | 0 |
| RPS100 RR | 8.73 | 3.00 | 72.49 | 2 | 0.0446 | 0 |
| RPS100 顺序策略 | 8.73 | 1.79 | 50.74 | 14 | 0.3210 | 0 |
| 延迟感知 500 毫秒[a] | 3.73 | – | ∼65.76 | 12 | 6.4716 | 8 |
| 稳态配置 | 7.30 | 1.73 | 48.51 | 21 | 1.3741 | 0 |
| 两步保护 | 8.42 | 1.82 | 52.31 | 0 | 0.0008 | 0 |

[a]由于无法使用逐工作节点路由，功率估算假设三个工作节点始终保持活动状态。

RPS100 顺序策略平均仅激活 1.79 个工作节点，在固定每秒请求数策略中功耗最低，为 50.74 W，但在填满 NUC0、未遵守其 750 每秒请求数的受保护容量时，记录了 14 次 p95 违规。RPS100 RR 记录了两次违规，但使所有工作节点保持活动状态，功耗为 72.49 W。在副本使用量相近的情况下，两步策略在任何阶段都未超过任一 SLA 阈值，功耗为 52.31 W，比 RPS100 RR 低 27.84%。

RPS50 顺序策略配置的副本数量约为 RPS100 顺序策略的 1.4 倍，但仍记录到两次违规。两步策略少使用 31.86% 的

副本，功耗也比该基线低 18.1%，且没有阈值违规。在 16 个副本下，按副本比例进行 8:4:4 路由会将 50% 的流量发送至 NUC0，当流量超过 1,500 RPS 时，超出其 750 RPS 的受保护容量。保留日志确认，RPS50 顺序策略、RPS50 轮询和 RPS100 轮询在 1,600 RPS 时首次违反第95百分位要求。因此，当路由导致某个工作节点过载时，仅增加副本并不足够。

结果与容量感知整合相符，即在服务级约束下协调副本、部署位置和路由。延迟感知轨迹记录了12次延迟违规和8次损失阈值违规，这与检测、调度和就绪完成之前，性能退化已进入尾部观测窗口的情况一致。更普遍地说，在异构集群中，仅凭副本数量不足以衡量服务容量：部署位置和路由还必须反映承载这些副本的工作节点的实测容量。

与保守的RPS50轮询相比，两步策略将副本数量减少了31.86%，将活跃工作节点减少了39.39%，并将估计功耗降低了26.74%。这一比较量化了过度配置的代价。与RPS100的比较则更具挑战性：整数阶段总数从288个副本降至278个，降幅为3.47%；同时，活跃工作节点整合使估计功耗降低了27.84%。因此，估计能源收益主要来自部署位置和路由，而非仅仅减少副本数量。

### 6.5 非单调工作负载下的策略评估

我们还分析了一条此前收集的、包含60个步骤的部署跟踪。每个步骤持续一分钟，提供负载以非单调方式在350至1,625 RPS之间变化。这一独立的 Stress2Twin 策略使用更密集的10-RPS性能分析数据集，以及配置为60 RPS的邻域；它并非递增过程中使用的100-RPS两步策略。上文报告的271状态、4.5小时性能分析成本仅适用于50-RPS递增数据集。Stress2Twin 和 RPS50 RR 基线都会在缩减规模前保留所需的副本数量五分钟。此配置用于评估该框架及其在负载变化下的缩减稳定化。它是对受控递增过程的补充，而不是重复上文使用的完全相同的50-RPS两步策略。图5显示了提供负载序列，而图6和表7报告了资源与能耗结果。

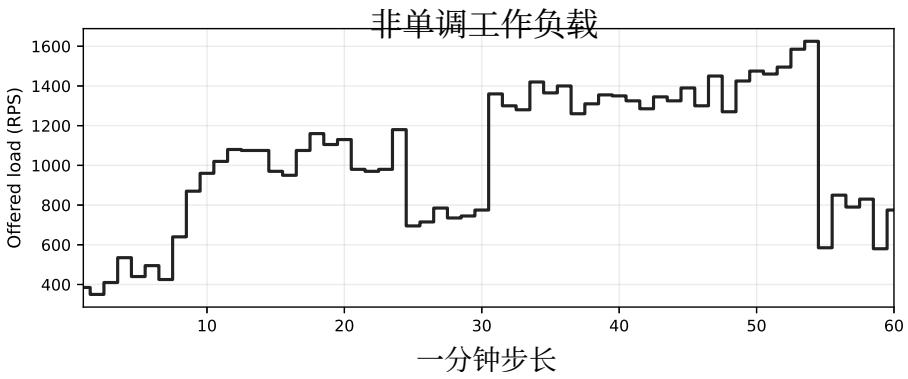


**图5**。 用于60步部署评估的非单调供给负载序列。每一步持续一分钟。

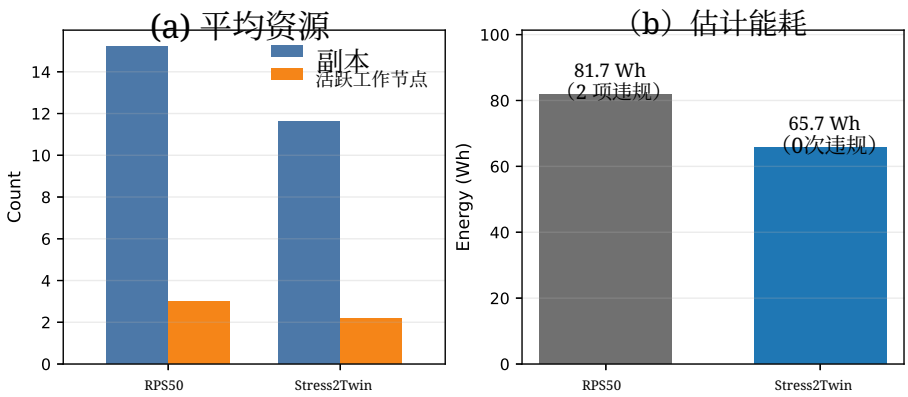


**图6**。 五分钟缩容稳定化期间的平均资源使用量和估计能耗。能耗标签还报告了第95百分位延迟超过一秒的一分钟步长数量。

**表7**。单次运行非单调工作负载结果。

| 策略 | 副本 | 有功功率 (W) | 能耗 (Wh) | 第95百分位 | 违规次数 | 损失 (%) |
|---|---|---|---|---|---|---|
| RPS50 轮询，保持5分钟 | 15.23 | 3.00 | 81.42 | 81.74 | 2 | 0.0113 |
| Stress2Twin ，保持5分钟 | 11.62 | 2.20 | 65.53 | 65.66 | 0 | 0.0005 |

在完整跟踪中，与稳定化 RPS50 RR 相比，Stress2Twin 将平均副本数减少了 23.74%，活跃工作节点减少了 26.67%，平均估计功耗降低了 19.52%，估计能耗降低了 19.67%。它没有出现 p95 违规，而基线出现了两次；两种策略都没有任何阶段的损失超过 10%。在不使用保持机制的 Stress2Twin 运行中，策略平均使用 10.40 个副本，并出现三次 p95 违规；稳定化运行平均使用 11.62 个副本，且没有出现违规。这一比较展示了负载下降和上升时资源与迁移安全之间的权衡。

## 7 讨论与实践意义

结果表明，独立获取的机器配置文件能够支持在所评估的异构部署中联合制定副本、部署位置和路由决策。这项工作的贡献在于性能知识的获取和使用方式：针对每个工作节点刻画其行为，在集群级别进行保守组合，并依据从配置文件状态推导出的运行边界进行筛选。不同于依赖历史集群观测 [9,4]或拟合作业级性能模型 [11], 的资源管理器，Stress2Twin 无需进行穷尽式集群级性能分析即可推导出策略，并且可以替换其机器预言器而无需修改优化器。

观察到的收益来源。 所评估的策略使整合、过渡保护和容量感知路由得以部分分离。在副本调度完全相同的情况下，与 RPS100 RR 相比，RPS100 顺序策略将估计功耗从 72.49 W 降至 50.74 W，但使第95百分位违规次数从 2 次增加到 14 次，这与整合将负载转移到超过其受保护容量的 NUC0 上相一致。在两个基于配置文件的策略之间，二者具有相同的组成和路由逻辑，而加入邻域测试后，违规次数从 21 次降至零次。

两步策略保持了与 RPS100 顺序策略相近的活动工作节点数量（1.82 对 1.79），但副本更少（8.42 对 8.73），且没有记录到违规，因此其收益并不能归因于增加了副本。RPS50 基线表明，当按副本比例进行的路由超过工作节点的受保护容量时，增加副本并不能恢复安全性。该评估未能完全分离容量比例路由权重与副本储备；采用受保护容量但不使用邻域测试的整合基线，可以将这两个组成部分区分开来。

能耗核算。 报告的功耗节省假设，未被策略保持活跃的工作节点不消耗功耗，而在整个跟踪期间，这些工作节点实际上始终处于通电状态。另一种空闲功率场景是为每个非活跃工作节点计入测得的 11 W 空闲功率。由于公式 (1) 关于活跃集合呈线性，这会使每种策略增加 11 $(3 - \bar{s})$ W，其中 $\bar{s}$ 是其活跃工作节点的平均数量。在这两种核算假设下，相对于 RPS100 RR 的估计功耗节省幅度为 9.93% 至 27.84%。

这一范围体现的是会计模型的敏感性，而非物理上可实现节省幅度的保证界限。实际的墙上功耗降低取决于可用的电源状态机制、工作节点空闲功耗，以及启动和恢复开销。

预测误差与保护机制的作用。 表5中的负偏差表明，稳态预言机倾向于低估部署延迟。稳态策略的21个误判安全阶段与选择孤立配置状态相一致：这些状态虽然满足最终服务级别协议（SLA），但接近肩部区域，在该区域中，分配负载或队列状态的微小偏差都会导致延迟发生大幅变化。受保护策略报告的较低误差是一种选择效应：保护机制排除了预测偏差可能改变可行性判定的状态，但不会改变固定状态下的预言机预测值。它不是置信区间，也无法量化预测不确定性。由于剩余的75.8毫秒第95百分位均方根误差相对于稳定延迟而言仍然不可忽略，因此，应使用孪生体对配置进行排序并界定安全区域，而不是预测绝对延迟。将逐状态测量变异性传播为明确的不确定性裕度，是一种自然的扩展方向。

参数选择。 表3中的运行区域边界由271个配置文件状态推导而来，未使用集群策略结果，而邻域深度则在评估斜坡上选定。在消融运行中，违规次数在第一步后由21次降至1次，在第二步后由1次降至0次，同时平均副本数增加了0.39。该结果为所选深度提供了依据，但并不构成样本外调优证据。根据配置文件本身推导深度，例如依据每个工作节点的保护区域与崩溃区域之间的负载距离，将消除这种依赖性，但仍有待评估。

工作负载动态与反应式控制。 Stress2Twin 是一种工作负载索引策略，因此它接收所提供的负载，而 KEDA控制器则根据尾随延迟窗口推断性能退化。因此，爬坡对比量化的是反应式检测的成本，而不是在信息相同的情况下对控制器进行排名；由负载驱动的 KEDA扩缩器可以提供匹配的输入。非单调轨迹将证据扩展到负载下降和不规则变化的情形。稳定化运行记录的违规次数少于未稳定化运行，但单次运行的测量结果无法将这一差异完全归因于五分钟保持期。由于稳态预言机忽略了积压、启动和就绪状态，该结果并不能证明系统能够应对任意突发负载或状态转换历史；这些情况需要显式的动态状态或保守稳定化。

可扩展性和可重复性。配置分析成本随着每个工作节点可接受状态数量之和增长，而不是随着集群配置数量的乘积增长，因此在增加工作节点时仍能保持采集过程可处理。策略推导的扩展方式有所不同：可行集群配置的无约束枚举会随着工作节点数量呈组合爆炸式增长

，而评估的受保护策略依赖于逐个工作节点的配置文件查找和固定的按容量排序的激活序列，从而避免搜索任意的工作节点组合。对于超出所评估配置的情况，其运行时间和内存扩展性，以及其在存在许多性能相近的工作节点时的表现，仍有待刻画。所有结果均来自每种策略的一次运行；因此，所报告的计数和误差仅具有描述性，其在重复运行中的变异性仍有待确定。

运行包络的有效性。 1,650 RPS边界是实测硬件、应用程序和服务级别协议的属性，并非通用的容量保证。受保护策略首次出现违规的位置比该边界高一个性能分析网格步长，这支持将其用作此次部署的运行包络。性能配置文件是在单个局域网上针对单个服务采集的，因此，共置服务、资源争用或传输波动都会使实测边界发生偏移。硬件、软件或工作负载发生重大变化后，必须重新分析受影响工作节点的性能，并重新推导其边界和受保护容量；由于性能分析是按工作节点进行的，因此只需重新测量发生变化的工作节点。

## 8 结论与未来工作

Stress2Twin 将各工作节点独立的压力配置组合起来，构建用于异构多接入边缘计算扩展的NDT决策模型。在所评估的部署中，仅考虑最终服务级别协议可行性在非线性转换附近并不安全；相比之下，两步式、基于配置文件的邻域测试在推导出的1,650 RPS受支持范围内，记录显示每个阶段均未超过延迟或丢失阈值。与RPS100 RR相比，该方法的估计功耗低27.84%；与RPS100 顺序策略整合相比，其p95违规次数为零，而非14次，功耗代价为1.6 W，约为3%。这一结果表明，每工作节点的容量信息有助于在未观测到服务级别协议违规的情况下进行整合。在相同的五分钟缩容保持时间下，非单调评估未出现p95违规，且估计能耗比RPS50 RR低19.67%。

尚待解决的挑战包括验证可重复性和实际功耗节省，在不通过评估斜坡选择邻域深度的情况下推导邻域深度，并表示由突发流量驱动的转换动态；从稀疏性能分析中学习可复用的逐工作节点模型；传播预测不确定性；检测模型漂移；以及将策略搜索扩展到穷举之外。应对这些挑战需要针对不同流量模式、服务和硬件开展更广泛的评估。运行时残差监测可以检测偏离已验证延迟和损失范围的情况，从而支持重新进行性能分析或采取保守的回退方案。

**致谢**。 网络数字孪生建模工作得到了6G-TWIN项目的支持。该项目由欧盟地平线欧洲研究与创新计划下的智能网络与服务联合事业资助，资助协议编号101136314。策略优化算法工作得到了imec.icon RAPIDNESS的支持，该项目由IMEC和佛兰德创新与创业机构共同出资（HBC.2024.0772）。

**利益披露**。 作者声明，不存在与本文内容相关的任何利益冲突。

## 参考文献

1. Agudelo, J.J., Soto, P., Zaki-Hindi, A., Sottet, J.S., Faye, S., Slamnik-Kriještorac, N., Marquez-Barja, J., Botero, M.C.: 迈向可信的6G网络数字孪生体：用于验证边缘计算资源反事实假设分析的框架。载于：EuCNC/6G峰会。第504–511页（2026）。https://doi.org/10.1109/ EuCNC/6GSummit68295.2026.11577404 2. Bello, R.C., Scivoletto, G., Landi, G., Mihai, R., Petrache, C.C., Camelo, M., Marquez-Barja, J., Slamnik-Kriještorac, N.: 6G计算连续体中的零接触跨域编排：真实场景下的框架评估。IEEE Open Journal of the Communications Society **7**, 1376–1395（2026）。https://doi. org/10 3662044 3 1109 2026 . /OJCOMS. . . Bozkaya-Aras, E.: 优化物联网边缘网络中的服务迁移：基于数字孪生的计算与节能方法。载于：IEEE WCNC。第1–6页（2025）。https://doi.org/10 10978782 4 1109 61545 2025 . /WCNC . . . Delimitrou, C., Kozyrakis, C.: Quasar：资源高效且具备QoS感知能力的集群管理。载于：Proc. ASPLOS。第127–144页（2014）。https://doi.org/10.1145/ 2541940.2541941 5. Fuller, A., Fan, Z., Day, C., Barlow, C.: 数字孪生：赋能技术、挑战与开放研究。IEEE Access **8**, 108952–108971（2020）。https: //doi.org/10 2998358 6 1109 2020 . /ACCESS. . . ITU-T：数字孪生网络：要求与架构。Y.3090建议书，国际电信联盟（2022年2月）7. Krizhevsky, A.: 从微型图像中学习多层特征。技术报告，多伦多大学（2009）8. Rosa, L., Calvio, A., Garbugli, A., Foschini, L.: 面向智慧城市中基于边缘的车联网数字孪生体的QoS感知数据分发平台。载于：IEEE WCNC。第1–6页（2025）。https://doi.org/ . /WCNC . . . Rzadca, K., Findeisen, P., Swiderski, J., Zych, P., Broniek, P., Kusmierek, J., Nowak, P., Strack, B., Witusowski, P., Hand, S., Wilkes, J.: Autopilot：谷歌的工作负载自动扩缩容。载于：Proc. EuroSys。第1–16页（2020）。https://doi.org/10. 1145/3342195.3387524 10. Satopaa, V., Albrecht, J., Irwin, D., Raghavan, B.: 大海捞针般寻找“Kneedle”：检测系统行为中的拐点。载于：IEEE ICDCS Workshops。第166–171页（2011）。https://doi.org/ . /ICDCSW. . . Venkataraman, S., Yang, Z., Franklin, M., Recht, B., Stoica, I.: Ernest：面向大规模高级分析的高效性能预测。载于：USENIX NSDI。第363–378页（2016）12. Yaqoob, M., Trestian, R., Nguyen, H.X.: Orbit-dt：通过数字孪生设计弥合仿真与现实之间的差距，面向超5G网络中的O-RAN。载于：IEEE WCNC。第1–6页（2025）。https://doi.org/10.1109/WCNC61545.2025.10978421