

基于主动推理的拟态云服务冗余度裁决在线自适应优化算法

张乾坤^{1,2}, 刘冰¹, 盛章骏¹, 邓贤君¹, 袁斌¹, 张帅³, 梁浩³, 程国振³, 丁国栋⁴

(1. 华中科技大学网络空间安全学院, 湖北 武汉 430074; 2. 网络空间安全教育部重点实验室, 河南 郑州 450002;
3. 网络空间部队信息工程大学, 河南 郑州 450002; 4. 嵩山实验室, 河南 郑州 450046)

摘要: 针对拟态云服务中冗余度裁决参数配置难以适应动态环境, 且安全收益、服务时延与恢复效能难以协同优化的问题, 提出一种基于主动推理的在线自适应优化 (AIOAO) 算法。首先, 面向拟态云服务, 建立裁决参数与安全收益、服务时延及恢复效能之间的量化映射模型, 将裁决参数配置问题形式化为动态不确定环境下的多目标在线优化问题; 其次, 提出 AIOAO 算法, 引入主动推理机制实现裁决参数在线自适应更新。实验表明, AIOAO 算法能够根据系统状态变化动态调整裁决参数, 在服务时延可控的前提下提升安全收益与恢复效能。

关键词: 拟态云服务; 冗余度裁决; 主动推理

中图分类号: TP309

文献标志码: A

doi: 10.11959/j.issn.1000-436x.TXXB260195

Online adaptive optimization algorithm for redundancy adjudication in mimic cloud services based on active inference

Zhang Qiankun^{1,2}, Liu Bing¹, Sheng Zhangjun¹, Deng Xianjun¹, Yuan Bin¹, Zhang Shuai³, Liang Hao³,
Cheng Guozhen³, Ding Guodong⁴

1. School of Cyber Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China
2. Key Laboratory of Cyberspace Security, Ministry of Education of China, Zhengzhou 450002, China
3. Information Engineering University, Zhengzhou 450002, China
4. Songshan Laboratory, Zhengzhou 450046, China

Abstract: To address the problem that redundancy adjudication parameter configuration in mimic cloud services is difficult to adapt to dynamic environments, and that security gain, service latency, and recovery effectiveness are difficult to optimize jointly, an active inference-based online adaptive optimization (AIOAO) algorithm was proposed. Firstly, for mimic cloud services, a quantitative mapping model between adjudication parameters and security gain, service latency, and recovery effectiveness was established, and the adjudication parameter configuration was formulated as a multi-objective online optimization problem in a dynamic and uncertain environment. Secondly, the AIOAO algorithm was proposed, in which an active inference mechanism was introduced to adaptively update adjudication parameters online. Experimental results show that the AIOAO algorithm can dynamically adjust adjudication parameters according to system state changes, and improve security gain and recovery effectiveness while keeping service latency under control.

Key words: mimic cloud service, redundancy adjudication, active inference

收稿日期: 2026-04-10; 修回日期: 2026-05-15

通信作者: 邓贤君, dengxj615@hust.edu.cn

基金项目: 国家自然科学基金资助项目 (No.62302183); 网络空间安全教育部重点实验室开放基金资助项目 (No.KLCS20240401)

Foundation Items: The National Natural Science Foundation of China (No.62302183), The Open Foundation of Key Laboratory of Cyberspace Security, Ministry of Education of China (No.KLCS20240401)

0 引言

随着数字化转型不断深入,云服务已成为政务、金融、工业互联网和互联网平台等关键业务场景的重要基础设施^[1-3]。在云原生^[4]环境下,云服务系统能够实现服务实例的按需扩缩容、迁移和重构,从而显著提升系统敏捷性与资源利用率。然而,系统攻击面也会随服务规模扩张和运行状态变化而持续扩大。在该环境下,微服务数量快速增长,服务调用链日趋复杂,容器化部署支持细粒度伸缩和迁移,使系统攻击面呈现出显著的动态特征,传统依赖静态边界和固定规则的防护手段难以有效应对未知漏洞利用与持续渗透等复杂威胁^[5]。特别是在关键业务场景中,云安全防护不仅需要抑制异常输出,还需要在高并发请求和服务连续性约束下保持稳定响应能力。针对这类问题,拟态防御^[6-8]在不改变系统既有服务功能的前提下,削弱攻击者对固定攻击面的持续利用能力,从而形成了拟态云服务^[9-10]这一新的安全防护思路。

动态异构冗余架构^[11]是拟态防御的核心实现形式,其通过对多个功能等价而实现差异化的执行体进行动态选择和并行调度,以提升系统对未知攻击与单点失效的抵御能力。在该架构中,多余度裁决^[12]是连接异构执行、异常感知与系统重构的关键机制。对于同一服务请求,系统首先将请求复制并分发至多个动态选取的异构执行体进行并行处理,然后对各执行体的输出结果进行一致性比较,从中选择相对正确的结果作为最终服务响应。这一过程被抽象为 $\frac{k}{n}$ 裁决机制^[13],即对于 n 个执行体的处理结果,若至少有 k 个执行体输出一致,则将该一致结果作为裁决输出。

裁决参数 n 和 k 并非单纯的工程配置项,而是影响拟态云服务安全收益、服务时延和恢复效能的关键控制变量。较小的冗余规模 n 有利于降低服务开销,但可能削弱异常输出抑制能力;较大的冗余规模 n 和更严格的一致性阈值 k 有助于提升防护能力,却可能引入更高的请求分发、实例等待和裁决处理时延。因此,如何在动态不确定的云服务环境中对多余度裁决参数进行在线自适应优化,是拟态云服务从架构设计走向稳定运行部署必须解决的关键问题。

然而,在拟态云服务场景下,多余度裁决仍面

临若干亟待解决的问题。一方面,现有研究多将裁决参数视为静态配置,而云服务中的请求负载、资源占用、实例健康状态以及攻击强度均具有显著时变性,固定的裁决参数难以长期适应复杂运行环境^[14]。另一方面,多余度裁决参数不仅影响系统对异常输出的抑制能力,还直接关系到服务时延以及异常感知后的恢复效率^[12],使得参数设计同时受到安全收益、服务时延和恢复效能等多重目标的共同制约。此外,云环境系统状态往往具有时变性和不完全可观测性,单纯依赖离线枚举方法难以满足在线优化需求,而强化学习方法又可能因学习代价高和收敛过程长而难以直接适用于服务连续性要求较高的拟态云服务场景^[15]。

针对上述问题,本文面向拟态云服务场景,提出了基于主动推理的在线自适应优化(AIOAO)算法,给出动态不确定环境下兼顾安全收益、服务时延和恢复效能的裁决参数优化解决方案。首先,针对裁决参数静态配置难以适应云服务动态运行环境的问题,分析拟态云服务中异构执行、裁决输出与反馈重构的闭环运行关系,建立裁决参数 n 和 k 与安全收益、服务时延及恢复效能之间的映射模型。其次,针对裁决参数设计受多重目标共同制约的问题,将参数配置问题形式化为动态不确定环境下的多目标在线优化问题。最后,针对系统状态时变且不完全可观测的问题,提出AIOAO算法,引入主动推理机制对系统隐含状态进行推断,并根据运行观测实现裁决参数的在线自适应更新,从而获得兼顾安全防护与服务性能的参数配置方案。进一步地,本文通过构建拟态云服务实验场景,对比固定裁决参数方法及其他基线方法,从安全收益、服务时延和恢复效能等方面对AIOAO算法进行评估。本文的主要贡献如下。

(1) 建立了面向拟态云服务的多余度裁决参数优化模型,刻画裁决参数 n 和 k 对系统安全收益、服务时延及恢复效能的影响机理,为拟态云服务中的裁决参数优化提供了形式化建模基础。

(2) 构建了多余度裁决参数的多目标在线优化问题,并提出AIOAO算法。综合考虑安全收益提升、服务时延控制和恢复效能增强等目标,将裁决参数配置问题形式化为动态不确定环境下的多目标在线优化问题,并设计基于主动推理的参数在线自适应更新机制,实现多余度裁决参数 n 和 k 的动态

优化。

(3) 验证了AIOAO算法在拟态云服务动态环境下的有效性。实验结果表明,所提算法能够根据系统状态变化实现裁决参数自适应调整,在兼顾服务性能的同时提升系统安全性与恢复效能。

1 相关研究工作

本节围绕本文研究问题,从拟态云服务、多余度裁决、动态决策优化3个方面对相关研究进行综述,并在此基础上分析现有工作的不足。

1.1 拟态云服务

拟态云服务是面向云服务场景构建内生安全能力的重要技术路径,其核心思想是在不改变云服务对外接口和业务语义的前提下,将云平台提供的应用服务节点构造为具有动态性、异构性和冗余性的服务形态。拟态云服务的运行机制如图1所示,系统通过异构实例并发执行、结果裁决和反馈重构形成闭环防护过程。普黎明等^[9]针对单执行体云应用服务缺乏异构性和动态性、难以应对未知漏洞和后门威胁的问题,提出面向云应用的拟态云服务架构,并讨论了策略调度和裁决机制等关键运行机制。Pu等^[10]进一步提出拟态云服务建模与仿真环境,支持拟态云服务中动态异构冗余结构、调度机制和裁决机制的建模与仿真。Li等^[16]面向5G核心网服务化架构提出拟态安全云架构,通过多容器执行、异构组件和交叉校验提升云化核心网服务的安全性。上述研究表明,拟态云服务已从理论架构逐

步扩展到云应用、仿真平台和5G核心网等具体场景。

现有拟态云服务研究更多回答“如何构造拟态云服务”和“如何验证拟态云服务安全能力”等问题,对拟态云服务连续运行过程中的裁决参数优化关注不足。

1.2 多余度裁决

在拟态云服务场景中,多余度裁决负责对多个服务响应进行比较与判定,从而形成统一的服务输出。多余度裁决主要包括服务响应收集、一致性比较和结果反馈3个环节。首先,系统对 n 个在线服务实例返回的响应结果进行收集与对齐;其次,裁决模块依据可编程语义解析判定^[17]、深度语义特征判定^[18]、自适应表决判定^[19]等规则,对多个响应进行一致性比较,判断是否存在满足阈值条件的响应集合;最后,若存在至少 k 个一致结果,则输出该结果作为最终服务响应,否则将当前请求标记为异常,并触发实例调整。这一过程被抽象为 $\frac{k}{n}$ 裁决机制,其中, n 表示当前参与并发处理的实例数量, k 表示形成最终输出所需满足的一致性阈值($\frac{n}{2} < k \leq n$),参数 n 和 k 共同决定了多余度裁决的冗余规模和严格程度。李卫超等^[12]对基于拟态防御架构的多余度裁决进行了建模与风险分析,从防御能力和运行效率等角度揭示了多余度裁决对系统安全性和服务的影响。Zheng等^[13]研究了具有功能

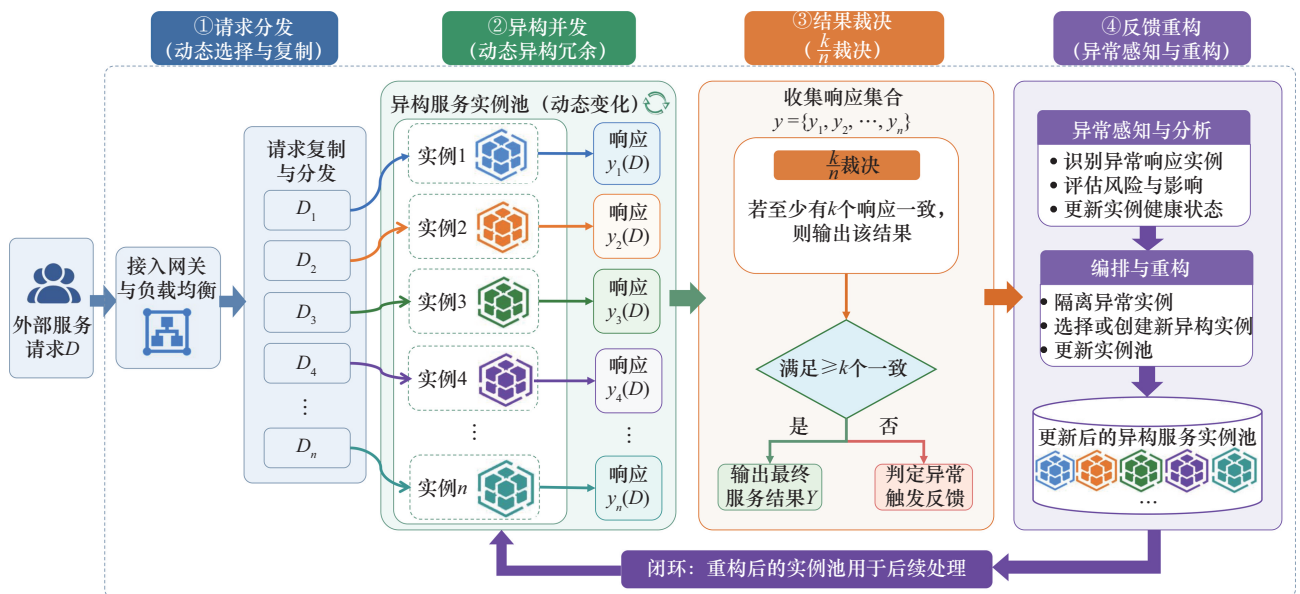


图1 拟态云服务的运行机制

依赖关系的可重构 k -out-of- n 系统可靠性评估问题, Xiahou 等^[20]进一步利用贝叶斯网络和动态贝叶斯网络对具有功能依赖关系的模块化 k -out-of- n 系统进行可靠性建模, 为刻画冗余阈值和组件状态依赖之间的关系提供了参考。

在拟态云服务连续运行场景中, 裁决参数 n 和 k 本身也是影响系统性能的重要决策变量。较大的 n 能提高冗余度和异常抑制能力, 但会额外增加系统开销; 较大的 k 能提高裁决严格性, 但可能增加等待时延并改变异常恢复触发条件。现有工作较少将裁决参数 n 和 k 与安全收益、服务时延和恢复效能进行统一建模, 也缺少在实例状态变化和攻击风险变化下对裁决参数进行在线自适应优化的方法。

1.3 动态决策优化

云服务运行环境具有负载波动、资源状态变化和服务实例动态调整等特点, 因此动态决策优化方法被广泛用于云服务资源分配与服务调度。Xia 等^[14]研究云原生微服务中的自适应资源分配问题, 说明动态调度方法能够提升云服务系统在变化负载下的性能表现。强化学习方法能够通过与环境交互学习长期收益最大化策略, 已在动态资源调度、安全策略选择和复杂系统优化中得到应用。Arulku-
maran 等^[21]对深度强化学习方法进行了综述, Gu 等^[22]系统总结了安全强化学习的方法、理论和应用, Zhu 等^[23]将强化学习用于提升复杂优化问题求解效率。主动推理是一种面向部分可观测动态环境的闭环决策方法。Friston 等^[24]提出主动推理过程理论, Sajid 等^[25]对主动推理进行了系统阐释和比较, Malekzadeh 等^[15]进一步分析了主动推理与强化学习在部分可观测条件下的联系。

现有动态决策优化方法多面向资源调度与负载均衡, 难以刻画拟态云服务中安全收益、服务时延与恢复效能的耦合关系。强化学习依赖交互探索, 学习代价和收敛时延较高; 同时, 系统隐状态不完全可观测, 仍需适合在线推断的裁决参数优化方法。

2 多余度裁决优化模型

本节通过构建拟态云服务系统模型, 刻画多余度裁决参数与系统的安全收益、服务时延和恢复效能之间的关系, 形式化描述多余度裁决参数优化问题。

2.1 拟态云服务系统模型

为刻画拟态云服务中服务请求、裁决参数与系

统运行指标之间的关系, 本文建立拟态云服务系统模型。在时刻 t , 拟态云服务系统可表示为:

$$\mathcal{M}_t = (\mathcal{E}, \mathcal{A}_t, \eta_t, \mathbf{z}_t, \mathbf{h}_t, \mathbf{y}_t, \mathbf{q}_t) \quad (1)$$

其中, 各参数含义如下。

(1) $\mathcal{E} = \{E_1, E_2, \dots, E_m\}$ 表示异构服务实例池。异构服务实例池由 m 个功能等价而实现差异化的候选服务实例构成。

(2) $\mathcal{A}_t = \{A_1, A_2, \dots, A_{n_t}\} \subseteq \mathcal{E}$ 表示时刻 t 的在线服务实例集合。系统根据当前运行状态, 从异构服务实例池中动态选择 n_t 个实例构成当前在线服务集 ($0 < n_t \leq m$), 用于并发处理同一服务请求。

(3) η_t 表示时刻 t 的服务请求到达状态, 用于刻画当前业务负载强度。

(4) $\mathbf{z}_t = (n_t, k_t)$ 表示时刻 t 的多余度裁决参数, 其中, n_t 表示参与并发处理的在线服务实例数量, k_t 表示形成最终服务输出所需满足的一致性阈值 ($\frac{n_t}{2} < k_t \leq n_t$)。

(5) $\mathbf{h}_t = (h_1(t), h_2(t), \dots, h_{n_t}(t))$ 表示时刻 t 的在线服务实例健康状态向量, 其中, $h_i(t)$ 表示在线服务实例 A_i 在时刻 t 保持正常服务输出的概率。

(6) $\mathbf{y}_t = \{y_1, y_2, \dots, y_{n_t}\}$ 表示 n_t 个在线服务实例针对同一服务请求返回的响应集合。

(7) $\mathbf{q}_t = (G_t, \tau_t, R_t)$ 表示性能指标, 其中, G_t 表示安全收益, τ_t 表示服务时延, R_t 表示恢复效能。

基于上述模型, 拟态云服务的运行过程可描述为: 当外部服务请求在时刻 t 到达时, 系统根据请求到达状态 η_t 和系统状态向量 $\mathbf{h}_t$, 从异构服务实例池 $\mathcal{E}$ 中动态选取在线服务实例集合 $\mathcal{A}_t$, 并依据裁决参数 $\mathbf{z}_t = (n_t, k_t)$ 将请求副本分发至各在线服务实例并发处理。随后, 系统收集响应集合 $\mathbf{y}_t$, 通过多余度裁决进行一致性判定并生成最终服务输出; 若响应结果存在异常, 则触发实例调整与反馈重构。最终, 系统在当前参数配置下表现出的安全收益、服务时延和恢复效能, 分别由 G_t 、 τ_t 和 R_t 表征。

2.2 安全收益度量

安全收益 G_t 用于刻画在给定裁决参数配置下, 拟态云服务相对于单实例服务在异常服务输出风险上的抑制效果。其基本思想是比较“单实例直接输出”与“多实例经裁决输出”两种情况下, 系统保持正常服务输出能力的差异。

设 $A_i \in \mathcal{A}_t$ 为时刻 t 的一个在线服务实例, 记随机变量 $T_i \sim \text{Exp}(\lambda_i)$ 为实例 A_i 在攻击作用下产生异常服务响应的的时间。于是, 实例 A_i 在时刻 t 保持正常服务输出的概率可表示为:

$$h_i(t) = \Pr(T_i > t) = e^{-\lambda_i t} \quad (2)$$

其中, λ_i 表示在线服务实例 A_i 在当前攻击环境下产生异常服务输出的风险强度, λ_i 越大, 说明该实例越容易在给定时间内输出异常响应。该建模能够反映攻击者对系统认知逐步加深后, 实例异常输出概率随时间持续增长并逐渐趋于稳定的过程。

(1) 单实例服务基线

在不采用多余度裁决的情况下, 系统服务结果由单个服务实例直接给出。由于时刻 t 的在线服务实例集合为 $\mathcal{A}_t$, 本文将在线服务实例平均保持正常服务输出的概率作为单实例服务基线, 记为:

$$P_t^{\text{base}} = \frac{1}{n_t} \sum_{A_i \in \mathcal{A}_t} h_i(t) \quad (3)$$

(2) 多余度裁决服务

在采用 $\frac{k_t}{n_t}$ 裁决机制后, 只要至少有 k_t 个在线服务实例保持正常服务输出, 系统即可形成裁决结果。假设不同在线服务实例在给定时刻的服务状态相互独立。因此, 裁决后的正常服务输出概率为:

$$P_t^{\text{judge}} = \sum_{\substack{\mathcal{C} \subseteq \mathcal{A}_t \\ |\mathcal{C}| \geq k_t}} \left(\prod_{A_i \in \mathcal{C}} h_i(t) \prod_{A_j \in \mathcal{A}_t \setminus \mathcal{C}} (1 - h_j(t)) \right) \quad (4)$$

其中, $\mathcal{C}$ 表示在时刻 t 保持正常服务输出的在线服务实例子集。

式(4)中的独立性假设适用于在线服务实例在依赖组件、镜像构建链和部署位置等方面充分异构的场景。当执行体异构性不足或存在共性漏洞时, 多个在线服务实例可能因相同漏洞产生相关异常输出, 甚至形成一致错误响应并通过裁决。为刻画该影响, 设 $\chi_t \in [0, 1]$ 表示时刻 t 的共性脆弱性因子, 即由共因失效导致多个执行体同时异常的概率。令 P_t^{com} 表示共性漏洞触发时系统仍能输出正常结果的概率, 则考虑共性脆弱性的裁决正常输出概率。

$$P_t^{\text{judge, c}} = (1 - \chi_t) P_t^{\text{judge}} + \chi_t P_t^{\text{com}} \quad (5)$$

在共性漏洞导致一致错误输出的极端情况下, P_t^{com} 可接近于 0, 此时 $P_t^{\text{judge, c}} \approx (1 - \chi_t) P_t^{\text{judge}}$ 。由此

可见, 随着 χ_t 增大, 增加 n_t 带来的安全收益将受到明显抑制, 甚至不再随冗余规模增加而持续提升。

基于上述定义, 本文考虑异构性较充分条件下的裁决增益, 将时刻 t 的安全收益表示为裁决后正常服务输出概率相对于单实例基线的提升比例, 即

$$G_t = \frac{P_t^{\text{judge}}}{P_t^{\text{base}}} \quad (6)$$

若 $G_t > 1$, 则说明引入多余度裁决后系统安全性得到提升, 且 G_t 越大, 表示当前裁决参数配置带来的安全收益越显著; 当 $G_t = 1$ 时, 说明当前裁决配置相对于单实例服务未带来额外安全增益。

2.3 服务时延度量

服务时延 τ_t 用于刻画拟态云服务在给定裁决参数配置下, 完成一次外部服务请求处理并返回最终裁决结果所需的总时间。与传统单实例服务不同, 拟态云服务中的一次请求处理过程包括请求复制与分发、异构在线服务实例并发处理以及多余度裁决输出等环节。

(1) 请求分发时延

对于时刻 t 的外部服务请求, 系统首先将请求复制并分发至在线服务实例集合 $\mathcal{A}_t$ 中的 n_t 个异构实例。设单位请求副本的平均分发时延为 d , 则请求分发时延可表示为:

$$\tau_t^{\text{dis}} = d n_t \quad (7)$$

(2) 实例处理时延

各在线服务实例对请求副本进行并发处理。设 η_t 表示时刻 t 的服务请求到达强度, 即单位时间内到达的外部服务请求数量; μ_i 表示在线服务实例的服务率, 即在当前资源配置下实例 $A_i \in \mathcal{A}_t$ 单位时间内能够完成的请求处理数量。将在线服务实例 A_i 的处理过程近似建模为 $M/M/1$ 排队系统, 则其平均处理时延可表示为:

$$\tau_i^{\text{pro}}(t) = \frac{1}{\mu_i - \eta_t}, \quad \eta_t < \mu_i \quad (8)$$

将平均处理时延按从小到大排序, 记为:

$$\tau_{(1)}^{\text{pro}}(t) \leq \tau_{(2)}^{\text{pro}}(t) \leq \dots \leq \tau_{(n_t)}^{\text{pro}}(t) \quad (9)$$

在 $\frac{k_t}{n_t}$ 裁决机制下, 系统在获得至少 k_t 个有效

响应后即可形成裁决结果, 因此并发处理阶段的实例时延可近似表示为:

$$\tau_t^{\text{wait}} = \tau_{(k_t)}^{\text{pro}}(t) \quad (10)$$

(3) 裁决处理时延

当达到裁决阈值后, 系统还需对返回结果进行收集、对齐和一致性比较。设单位响应的收集与对齐时延为 θ_1 , 单位响应比较时延为 θ_2 , 则裁决处理时延可表示为:

$$\tau_i^{\text{adj}} = \theta_1 k_i + \theta_2 \frac{k_i(k_i - 1)}{2} \quad (11)$$

综合上述3个环节, 时刻 t 的总服务时延为:

$$\tau_t = \tau_t^{\text{dis}} + \tau_t^{\text{wait}} + \tau_t^{\text{adj}} \quad (12)$$

其中, n_t 越大, 请求分发时延越高; k_t 越大, 系统等待有效响应和执行裁决比较所需时间越长。进一步地, 在满足服务质量要求的场景下, 系统服务时延需满足预设的服务等级约束, 即 $\tau_t \leq \tau_{\text{SLO}}$, 其中, τ_{SLO} 为最大允许服务时延。

2.4 恢复效能度量

恢复效能用于刻画拟态云服务在异常被裁决感知后, 通过实例隔离、替换、迁移或重建等机制恢复正常服务能力的综合表现。根据恢复过程对外部干预依赖程度的不同, 拟态云服务的恢复方式分为在线恢复与离线恢复。在此基础上, 本文综合考虑恢复结果与恢复速度, 以恢复成功概率和平均恢复时间刻画系统恢复效能。

(1) 恢复成功概率

设系统在时刻 t 由多余度裁决检测到异常分歧后, 在线服务实例 $A_i \in \mathcal{A}_t$ 的恢复方式选择概率分别为 p_i^{on} 和 p_i^{off} 。若实例 A_i 采用在线恢复, 则其在当前恢复周期内成功回到可用服务状态的概率为 u_i^{on} ; 若采用离线恢复, 则其成功回到可用服务状态的概率为 u_i^{off} 。因此, 实例 A_i 在当前恢复周期内重新变为可用状态的概率可表示为:

$$r_i(t) = p_i^{\text{on}} u_i^{\text{on}} + p_i^{\text{off}} u_i^{\text{off}} \quad (13)$$

在 $\frac{k_t}{n_t}$ 裁决机制下, 系统恢复后的目标并不是

使全部实例都重新上线, 而是保证至少有 k_t 个在线服务实例重新达到可用服务状态, 从而恢复系统形成有效裁决输出的能力。为便于分析, 进一步假设不同在线服务实例的恢复结果相互独立, 则时刻 t 的恢复成功概率可表示为:

$$P_t^{\text{rec}} = \sum_{\substack{C \subseteq \mathcal{A}_t \\ |C| \geq k_t}} \left(\prod_{A_i \in C} r_i(t) \prod_{A_j \in \mathcal{A}_t \setminus C} (1 - r_j(t)) \right) \quad (14)$$

(2) 平均恢复时间

除恢复结果外, 恢复过程所需时间也是衡量恢

复效能的重要因素。设在线服务实例 A_i 的异常隔离时间、在线恢复时间、离线恢复时间以及重新加入在线服务集合的时间分别为 t_i^{iso} 、 t_i^{on} 、 t_i^{off} 和 t_i^{join} , 则实例 A_i 的期望恢复时间可表示为:

$$t_i^{\text{rec}} = t_i^{\text{iso}} + p_i^{\text{on}} t_i^{\text{on}} + p_i^{\text{off}} t_i^{\text{off}} + t_i^{\text{join}} \quad (15)$$

进一步地, 时刻 t 的平均恢复时间定义为:

$$T_t^{\text{rec}} = \frac{1}{n_t} \sum_{A_i \in \mathcal{A}_t} t_i^{\text{rec}} \quad (16)$$

基于恢复成功概率与平均恢复时间, 本文将恢复效能理解为系统在单位期望恢复时间内重新达到有效裁决状态的能力。为避免极短恢复时间导致比值异常放大, 进一步引入恢复时间下界 $T_{\min} > 0$, 则时刻 t 的恢复效能定义为:

$$R_t = \frac{P_t^{\text{rec}}}{\max(T_t^{\text{rec}}, T_{\min})} \quad (17)$$

其中, $T_{\min}$ 表示系统恢复流程中由实例隔离和编排调度等不可避免操作所决定的最小有效恢复时间。当 R_t 取值较大时, 表示系统能够在较短时间以较高概率恢复到满足裁决条件的可用服务状态, 恢复效能较好。式(17)主要适用于恢复时间波动处于可控范围的场景。若恢复时间呈现明显长尾分布, 则可将 T_t^{rec} 替换为分位数恢复时间 $Q_\alpha(T_t^{\text{rec}})^{[26]}$ 或条件风险值 $\text{CVaR}_\alpha(T_t^{\text{rec}})^{[27]}$, 从而得到更保守的度量。本文恢复时间由异常隔离、在线恢复、离线恢复和重新加入在线集合等有界环节构成, 因此采用平均恢复时间 T_t^{rec} 能够较稳定地反映系统恢复过程。

2.5 多目标在线优化问题

基于前述建立的安全收益、服务时延和恢复效能度量, 本文进一步将拟态云服务中的多余度裁决参数配置问题形式化为多目标在线优化问题。其目标是在动态不确定环境下, 为时刻 t 的拟态云服务系统选择合适的裁决参数 $\mathbf{z}_t = (n_t, k_t)$, 在提升系统安全收益与恢复效能的同时, 控制服务时延满足服务质量要求。该多目标在线优化问题为:

$$\begin{aligned} & \max_{\mathbf{z}_t = (n_t, k_t)} \{ G_t, -\tau_t, R_t \} \\ & \text{s.t. } \frac{n_t}{2} < k_t \leq n_t \leq m, \quad \tau_t \leq \tau_{\text{SLO}}, \quad n_t, k_t \in \mathbb{Z}^+ \end{aligned} \quad (18)$$

由于服务请求到达状态、在线服务实例健康状态向量以及异常后的恢复状态均随时间动态变化, 系统在不同时刻对应的最优裁决参数配置并不固

定,需根据实时观测结果进行持续更新。因此,该问题又具有显著的在线性与动态性。与此同时,系统真实运行状态具有时变性和不完全可观测性,使得传统静态配置或离线枚举方法难以满足实时决策需求。综上,拟态云服务中的多余度裁决参数配置问题可归结为一个面向动态不确定环境的离散多目标在线优化问题。

3 AIOAO 算法

本节基于前文给出的多目标在线优化问题,提出 AIOAO 算法。

3.1 主动推理框架

主动推理^[19-21]是一种面向部分可观测动态环境的闭环决策方法,其核心思想是通过运行观测持续更新对系统隐状态的信念,并在目标偏好约束下通过最小化期望自由选择当前动作。对于状态隐含、观测受限且服务连续性要求较高的拟态云服务场景,强化学习^[22-24]方法通常需要通过持续试错积累大量交互样本,易带来较高的学习代价和较长的收敛过程,并可能因频繁探索对服务连续性和性能稳定性产生不利影响。相比之下,主动推理能够在部分可观测条件下依据当前观测对系统隐状态进行推断,并结合目标偏好开展动作选择,降低了对大规模试错探索的依赖,因此更适合用于拟态云服务中面向动态不确定环境的多目标在线优化问题。

主动推理由隐状态、观测、动作3个基本要素构成。设 $s_t \in \mathcal{S} = \{s^{(1)}, s^{(2)}, s^{(3)}, s^{(4)}\}$ 表示时刻 t 的系统隐状态,用于刻画拟态云服务在当前时刻的真实运行态势,其中 $s^{(1)}$ 、 $s^{(2)}$ 、 $s^{(3)}$ 和 $s^{(4)}$ 分别表示正常稳定状态、轻度异常状态、高风险状态和恢复调整状态。设 $\mathbf{o}_t = (\delta_t, \hat{\tau}_t, \rho_t)$ 表示时刻 t 的系统观测向量,用于描述系统在运行过程中能够直接获取的外部反馈信息,其中, δ_t 表示响应一致性观测量, $\hat{\tau}_t$ 表示当前时刻实际观测到的服务时延, ρ_t 表示恢复反馈观测量。设 $\mathbf{a}_t \triangleq \mathbf{z}_t = (n_t, k_t) \in \Omega_t$ 表示时刻 t 的决策动作,本文对应于裁决参数配置。此外,定义目标偏好函数,用于刻画系统对未来性能结果的评价准则,从而体现对高安全收益、低服务时延和高恢复效能的综合优化需求。

在主动推理执行过程中,系统首先获取当前时刻 t 的运行观测信息 $\mathbf{o}_t$,并结合生成模型更新对当前隐状态 s_t 的信念。随后,基于当前状态信念以及

候选裁决动作 $\mathbf{a}_t$,预测系统在下一时刻 $t+1$ 可能到达的隐状态 s_{t+1} 及其对应观测 $\mathbf{o}_{t+1}$ 。在此基础上,系统结合目标偏好,计算各候选动作的期望自由能(期望自由能用于综合衡量候选动作对未来目标偏好的匹配程度以及对系统状态不确定性的降低能力)。最后,选择期望自由能最小的动作作为当前时刻的裁决参数配置,并将其作用于系统运行过程。因此,主动推理可将拟态云服务中的裁决参数在线优化过程统一表述为隐状态推断与裁决动作选择相结合的闭环决策过程。

3.2 隐状态推断

隐状态推断旨在根据当前时刻可直接获取的系统观测 $\mathbf{o}_t$,推断系统隐状态 s_t 的后验信念分布。由于隐状态不能被直接观测,本文通过观测生成模型 $P(\mathbf{o}_t | s_t)$ 与状态转移模型 $P(s_t | s_{t-1}, \mathbf{a}_{t-1})$ 对其进行概率推断。其中,观测生成模型用于刻画在给定隐状态 s_t 条件下产生当前观测 $\mathbf{o}_t$ 的可能性,状态转移模型用于描述系统在上一时刻隐状态 s_{t-1} 和上一时刻裁决动作 $\mathbf{a}_{t-1}$ 共同作用下向当前隐状态 s_t 的演化规律,二者对应的条件概率分布均可由历史运行数据预先估计获得。

为表征系统在当前时刻对各类隐状态的认知程度,定义时刻 t 的隐状态后验信念分布为:

$$b_t(s_t) = P(s_t | \mathbf{o}_{1:t}, \mathbf{a}_{1:t-1}) \quad (19)$$

其中, $\mathbf{o}_{1:t}$ 表示截至时刻 t 的观测序列, $\mathbf{a}_{1:t-1}$ 表示截至时刻 $t-1$ 的动作序列。该分布反映了系统在融合历史信息与当前观测后,对各候选隐状态的置信度。

后验信念分布的更新采用预测与校正两步机制。首先,根据上一时刻的信念分布 $b_{t-1}(s_{t-1})$ 和裁决动作 $\mathbf{a}_{t-1}$,对当前时刻的隐状态进行先验预测。

$$\bar{b}_t(s_t) = \sum_{s_{t-1} \in \mathcal{S}} P(s_t | s_{t-1}, \mathbf{a}_{t-1}) b_{t-1}(s_{t-1}) \quad (20)$$

其中, $\bar{b}_t(s_t)$ 表示在尚未引入当前观测之前,系统对当前隐状态的先验估计。随后,结合当前时刻的观测信息 $\mathbf{o}_t$,利用观测生成模型对上述先验估计进行校正,得到当前时刻的后验信念分布:

$$b_t(s_t) = \frac{P(\mathbf{o}_t | s_t) \bar{b}_t(s_t)}{\sum_{s \in \mathcal{S}} P(\mathbf{o}_t | s) \bar{b}_t(s)} \quad (21)$$

后验信念分布比单一状态估计更完整地保留

了系统对各候选状态的不确定性描述, 后续动作选择将基于这一分布而非仅基于单一状态标签进行。

3.3 裁决参数更新机制

在获得时刻 t 的隐状态后验信念分布 $b_t(s_t)$ 后, AIOAO 算法需要进一步确定当前时刻的裁决参数配置, 以在满足裁决约束与服务等级约束的前提下, 实现安全收益、服务时延与恢复效能的综合优化。考虑到拟态云服务中的裁决参数更新并非仅依据当前观测进行经验式调整, 而是需要在隐状态推断基础上完成参数决策, AIOAO 算法将裁决参数更新建模为基于后验信念的在线动作选择过程。

结合前文给出的多目标在线优化问题, 定义时刻 t 的可行裁决动作集合为:

$$\Omega_t = \left\{ \mathbf{a} = (n, k) \mid \frac{n}{2} < k \leq n \leq m, \tau(\mathbf{a}) \leq \tau_{\text{SLO}} \right\} \quad (22)$$

其中, $k \in \mathbb{Z}^+$, $n \in \mathbb{Z}^+$ 。

对于任一候选动作 $\mathbf{a} \in \Omega_t$, 系统首先根据当前后验信念分布 $b_t(s_t)$ 与状态转移模型, 对执行该动作后下一时刻可能到达的隐状态进行前瞻预测。记对应的预测状态分布为:

$$\bar{b}_{t+1}(s_{t+1}|\mathbf{a}) = \sum_{s_t \in \mathcal{S}} P(s_{t+1}|s_t, \mathbf{a}) b_t(s_t) \quad (23)$$

为与主动推理框架保持一致, AIOAO 算法进一步将系统性能指标 $\mathbf{q}_t = (G_t, \tau_t, R_t)$ 理解为由系统状态与裁决动作共同决定的结果变量, 即:

$$\mathbf{q}_t = \mathbf{q}(s_t, \mathbf{a}_t) = (G(s_t, \mathbf{a}_t), \tau(s_t, \mathbf{a}_t), R(s_t, \mathbf{a}_t)) \quad (24)$$

据此, 可得到候选动作 $\mathbf{a}$ 对应的预测性能期望向量。

$$\bar{\mathbf{q}}_{t+1}(\mathbf{a}) = \sum_{s_{t+1} \in \mathcal{S}} \mathbf{q}(s_{t+1}, \mathbf{a}) \bar{b}_{t+1}(s_{t+1}|\mathbf{a}) \quad (25)$$

为体现系统对高安全收益、低服务时延和高恢复效能的长期优化需求, AIOAO 算法采用预先设定的偏好效用函数 U 进行刻画。具体地, 定义为:

$$U(\bar{\mathbf{q}}_{t+1}(\mathbf{a})) = \omega_G \tilde{G}_{t+1}(\mathbf{a}) - \omega_\tau \tilde{\tau}_{t+1}(\mathbf{a}) + \omega_R \tilde{R}_{t+1}(\mathbf{a}) \quad (26)$$

其中, $\omega_G > 0$ 、 $\omega_\tau > 0$ 和 $\omega_R > 0$ 分别表示安全收益、服务时延和恢复效能对应的偏好权重, 且在算法初始化阶段预先设定; $\tilde{G}_{t+1}(\mathbf{a})$ 、 $\tilde{\tau}_{t+1}(\mathbf{a})$ 和 $\tilde{R}_{t+1}(\mathbf{a})$ 分别表示对 $\bar{\mathbf{q}}_{t+1}(\mathbf{a})$ 中各指标分量在当前

候选动作集合上进行最小-最大归一化处理后的结果。

在此基础上, 定义候选动作 $\mathbf{a}$ 在时刻 t 的期望自由能为:

$$\mathcal{F}_t(\mathbf{a}) = -U(\bar{\mathbf{q}}_{t+1}(\mathbf{a})) + \lambda H(\bar{b}_{t+1}(s_{t+1}|\mathbf{a})) \quad (27)$$

其中, $H(\cdot)$ 表示信息熵, λ 为权衡系数。第一项用于衡量候选动作预测结果与系统目标偏好的匹配程度, 当动作能够带来更高的安全收益与恢复效能、同时保持更低的服务时延时, 其效用值更大, 对应的自由能更小。第二项用于刻画动作执行后未来状态分布的不确定性, 该项越小, 说明动作越有利于降低系统未来运行态势的不确定性, 提升裁决参数更新的稳健性。

由此, 时刻 t 的最优裁决动作可表示为:

$$\mathbf{a}_t^* = \arg \min_{\mathbf{a} \in \Omega_t} \mathcal{F}_t(\mathbf{a}) \quad (28)$$

若存在多个候选动作取得相同的最小期望自由能, 则优先选择并发实例规模 n 较小的动作; 当并发实例规模相同时, 进一步选择一致性阈值 k 较小的动作, 以减少不必要的调用开销并保持服务连续性。

考虑到拟态云服务场景中观测噪声和短时波动可能导致裁决参数频繁切换, 本文进一步引入带门限的平滑更新机制。记上一时刻实际执行的裁决动作作为 $\mathbf{a}_{t-1}$, 则当前时刻的参数更新规则定义为:

$$\mathbf{a}_t = \begin{cases} \mathbf{a}_t^*, & \mathcal{F}_t(\mathbf{a}_{t-1}) - \mathcal{F}_t(\mathbf{a}_t^*) > \varepsilon \\ \mathbf{a}_{t-1}, & \mathcal{F}_t(\mathbf{a}_{t-1}) - \mathcal{F}_t(\mathbf{a}_t^*) \leq \varepsilon \end{cases} \quad (29)$$

其中, ε 为平滑更新门限。仅当新动作相对于当前动作带来的期望自由能下降超过预设阈值时, 系统才执行裁决参数切换; 否则保持原有配置不变。

AIOAO 算法的完整流程如算法 1 所示。该算法以当前运行观测为输入, 先完成隐状态后验信念更新, 再对可行裁决参数进行前瞻预测与期望自由能评价, 最终经平滑更新得到实际执行的裁决参数。

算法 1 AIOAO 算法

输入 观测生成模型 $P(\mathbf{o}_t|s_t)$, 状态转移模型 $P(s_t|s_{t-1}, \mathbf{a}_{t-1})$, 偏好权重 ω_G 、 ω_τ 、 ω_R , 不确定性权衡系数 λ , 更新门限 ε , 服务等级阈值 τ_{SLO}

输出 裁决参数序列 $\{\mathbf{a}_t\}_{t=1}^T$

1) 初始化后验信念分布 $b_0(s_0)$, 裁决参数 $\mathbf{a}_0 =$

- (n_0, k_0)
- 2) for $t \in \{1, 2, \dots, T\}$ do
- 3) 获取当前时刻运行观测 $\mathbf{o}_t = (\delta_t, \hat{\tau}_t, \rho_t)$
- 4) 根据式(20)计算先验信念分布 $\bar{b}_t(s_t)$
- 5) 根据式(21)计算后验信念分布 $b_t(s_t)$
- 6) 根据式(22)构造可行裁决参数集合 Ω_t
- 7) for $\mathbf{a} \in \Omega_t$ do
- 8) 根据式(23)计算候选裁决参数 $\mathbf{a}$ 对应的预测状态分布 $\bar{b}_{t+1}(s_{t+1}|\mathbf{a})$
- 9) 根据式(25)计算候选裁决参数 $\mathbf{a}$ 对应的预测性能期望向量 $\bar{q}_{t+1}(\mathbf{a})$
- 10) 对 $\bar{G}_{t+1}(\mathbf{a})$ 、 $\bar{\tau}_{t+1}(\mathbf{a})$ 和 $\bar{R}_{t+1}(\mathbf{a})$ 进行最小-最大归一化处理, 分别得到 $\tilde{G}_{t+1}(\mathbf{a})$ 、 $\tilde{\tau}_{t+1}(\mathbf{a})$ 、 $\tilde{R}_{t+1}(\mathbf{a})$
- 11) 根据式(26)计算候选裁决参数 $\mathbf{a}$ 的偏好效用值 $U(\bar{q}_{t+1}(\mathbf{a}))$
- 12) 根据式(27)计算候选裁决参数 $\mathbf{a}$ 的期望自由能 $\mathcal{F}_t(\mathbf{a})$
- 13) end for
- 14) 根据式(28)确定当前时刻最优候选裁决参数 $\mathbf{a}_t^*$
- 15) if 存在多个候选裁决参数取得相同的最小期望自由能 then
- 16) 优先选择并发实例规模 n 较小的候选裁决参数
- 17) if 并发实例规模 n 相同 then
- 18) 进一步选择一致性阈值 k 较小的候选裁决参数
- 19) end if
- 20) end if
- 21) if $\mathcal{F}_t(\mathbf{a}_{t-1}) - \mathcal{F}_t(\mathbf{a}_t^*) > \varepsilon$ then
- 22) $\mathbf{a}_t \leftarrow \mathbf{a}_t^*$
- 23) else
- 24) $\mathbf{a}_t \leftarrow \mathbf{a}_{t-1}$
- 25) end if
- 26) end for

3.4 AIOAO 算法复杂度分析

为分析 AIOAO 算法在在线环境下的实时性, 设系统隐状态数量为 $S = |\mathcal{S}|$, 时刻 t 的可行裁决动作数量为 $L = |\Omega_t|$, 异构服务实例池规模为 m 。在每个决策周期内, AIOAO 算法首先根据式(20)和

式(21)完成隐状态信念更新, 其中状态预测需要对状态转移概率进行加权求和, 复杂度为 $O(S^2)$, 观测校正复杂度 $O(S)$ 。随后, 算法对每个候选裁决动作进行前瞻预测和期望自由能计算。对单个候选动作, 预测状态分布的复杂度为 $O(S^2)$, 偏好效用与信息熵计算的复杂度为 $O(S)$ 。

对于安全收益 P_t^{judge} 和恢复成功概率 P_t^{rec} , 若直接枚举所有实例子集, 复杂度会随 n_t 指数增长。本文在实现时采用 Poisson-binomial 动态规划方式^[28]计算“至少 k_t 个实例正常/恢复成功”的概率, 其复杂度为 $O(n_t k_t)$, 上界为 $O(m^2)$ 。因此, AIOAO 算法在单个时隙内的总体计算复杂度可表示为:

$$O(S^2 + L(S^2 + m^2)) \quad (30)$$

由于 $L \leq \sum_{n=1}^m \left\lceil \frac{n}{2} \right\rceil = O(m^2)$, 算法仅需在有限状态和有限动作空间上进行前向推断与评价。

4 仿真实验与分析

本节采用离散时间仿真方式构建拟态云服务运行场景, 并通过对比实验验证 AIOAO 算法在动态不确定环境下载决参数在线自适应更新的有效性。

4.1 仿真设置

仿真实验平台为一台通用机架式服务器, 服务器型号为 Dell PowerEdge R760, 采用双路 Intel Xeon Scalable 处理器, 内存配置为 128 GB DDR5 ECC RDIMM, 存储配置为 2 TB NVMe SSD, 操作系统为 Ubuntu 22.04 LTS。实验程序采用 Python 3.10 实现。为增强仿真场景的真实性, 本文采用 Alibaba Cluster Trace 数据集^[29]对系统运行负载进行驱动。考虑到本文研究对象为单个拟态云服务内部的在线裁决参数更新问题, 而非大规模资源编排问题, 本文将异构服务实例池规模设为中等规模, 以兼顾拟态冗余机制的表达能力和仿真复杂度。整个仿真过程划分为 T 个离散时隙, 在每个时隙内, 系统依据状态转移模型生成当前真实隐状态, 根据当前隐状态生成系统观测向量, 并将观测向量作为 AIOAO 算法的输入, 用于完成裁决参数求解。为降低随机因素对实验结果的影响, 每组实验在相同参数设置下独立重复 30 次, 并以平均结果作为最终统计值。仿真参数配置如表 1 所示。

表1 仿真参数配置

仿真参数	取值
异构服务实例池规模 m	10
仿真时隙数 T	500
低负载请求到达强度 η_t^L	0.3
平稳负载请求到达强度 η_t^S	0.6
高负载请求到达强度 η_t^H	0.9
实例服务率 μ_i	[1.2, 2.0]随机值
单位请求分发时延 d	0.01
单位响应收集与对齐时延 θ_1	0.005
单位响应比较时延 θ_2	0.002
正常稳定状态下风险强度 $\lambda_i^{(1)}$	[0.01, 0.03]随机值
轻度异常状态下风险强度 $\lambda_i^{(2)}$	[0.04, 0.08]随机值
高风险状态下风险强度 $\lambda_i^{(3)}$	[0.10, 0.18]随机值
恢复调整状态下风险强度 $\lambda_i^{(4)}$	[0.03, 0.06]随机值
在线恢复选择概率 p_i^{on}	[0.50, 0.70]随机值
离线恢复选择概率 p_i^{off}	$1 - p_i^{\text{on}}$
在线恢复成功概率 u_i^{on}	[0.70, 0.90]随机值
离线恢复成功概率 u_i^{off}	[0.80, 0.95]随机值
服务等级阈值 τ_{SLO}	0.6
安全收益偏好权重 ω_G	0.4
服务时延偏好权重 ω_τ	0.3
恢复效能偏好权重 ω_R	0.3
不确定性权衡系数 λ	0.3
平滑更新门限 ε	0.015

4.2 对比方法

在仿真实验中, 本文设置静态配置、规则驱动和学习驱动 3 类基线方法。其中, Fixed-Low 和 Fixed-High 并非某一特定算法, 而是静态多余度裁决配置的代表性消融基线, 分别对应低冗余低开销和高冗余高安全两种典型策略^[12], 用于分析固定裁决参数在动态环境下的性能边界。规则驱动方法 (Rule-Based) 参考已有自适应拟态表决器和阈值驱动调整思想^[19]构造, 用于代表工程实践中常见的基于观测阈值的反应式参数调整策略。强化学习驱动方法 (RL-Based) 采用深度 Q 网络 (deep Q-network, DQN)^[30]实现, 用于代表基于交互学习的参数更新策略。上述方法均在相同系统模型、相同动作约束和相同评价指标下运行, 以保证实验比较的公平性。

(1) Fixed-Low。固定采用低冗余裁决参数配置 $(n, k) = (3, 2)$, 即在整个仿真过程中保持较小的并发实例规模和较宽松的一致性阈值。

(2) Fixed-High。固定采用高冗余裁决参数配置 $(n, k) = (7, 5)$, 即在整个仿真过程中保持较大的并发实例规模和较严格的一致性阈值。

(3) Rule-Based。基于规则的方法。当响应一致性下降或恢复反馈变差时, 适当增大并发实例规模或提高一致性阈值; 当服务时延持续升高时, 则适当减小冗余规模。

(4) RL-Based。基于 DQN 的强化学习方法。由于裁决参数 (n, k) 属于离散动作集合, 而系统观测向量 $\mathbf{o}_t = (\delta_t, \hat{\tau}_t, \rho_t)$ 可作为连续状态输入, 因此采用 DQN 作为强化学习基线。DQN 的动作空间与 AIOAO 算法的可行裁决动作集合 Ω_t 保持一致, 状态输入为当前系统观测及上一时刻裁决参数。为保证与本文偏好效用函数 U 在评价维度上等价, RL-Based 的即时奖励函数定义为:

$$r_t = \omega_G \tilde{G}_t - \omega_\tau \tilde{\tau}_t + \omega_R \tilde{R}_t \quad (31)$$

4.3 结果分析

4.3.1 总体性能分析

为验证 AIOAO 算法在动态不确定环境下进行裁决参数更新的整体有效性, 本文在相同仿真环境下, 将 AIOAO 算法与 4 种对比方法进行总体性能比较, 分别给出 500 个时隙下归一化后安全收益 G_t 、服务时延 τ_t 和恢复效能 R_t 的变化曲线, 结果如图 2 所示。

从图 2(a)可以看出, AIOAO 算法在大多数时隙内均保持较高的安全收益水平, 在低负载、平稳负载和高负载阶段的安全收益大致分别稳定在 0.66、0.69 和 0.74 左右, 高负载阶段峰值接近 0.77, 整体高于 Fixed-Low、Rule-Based 和 RL-Based, 并与 Fixed-High 保持相近水平。AIOAO 算法能够保持较高安全收益的原因在于, 其裁决参数更新并非仅依据当前观测进行经验式调整, 而是先通过观测向量对系统隐状态进行后验信念更新, 再利用期望自由能对候选裁决参数进行评价。当响应一致性下降或风险水平升高时, 系统对轻度异常状态、高风险状态或恢复调整状态的置信度随之上升, AIOAO 算法倾向于选择更高冗余规模或更严格一致性阈值的裁决参数, 从而提升异常输出抑制能力。与 Fixed-Low 相比, AIOAO 算法能够避免长期低冗余

导致的安全收益不足;与Fixed-High相比,AIOAO算法并非始终采用高冗余配置,而是根据运行状态变化有针对性地增强裁决能力;与Rule-Based和RL-Based相比,AIOAO算法通过隐状态信念和未来状态预测进行动作选择,因此在动态环境下具有更稳定的安全收益表现。

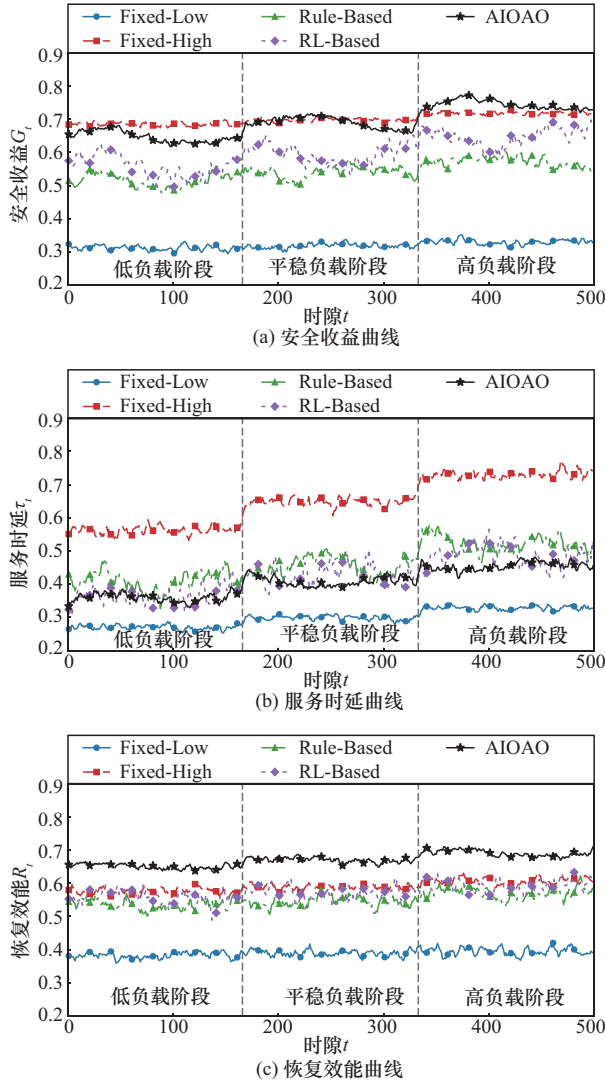


图2 不同方法的总体性能比较

从图2(b)可以看出,AIOAO算法虽然没有在所有时隙内都取得最小服务时延,但能够在绝大多数时隙内将时延控制在较为平稳且可接受的范围内,表现出更好的时延约束满足能力。AIOAO算法的服务时延随负载升高由低负载阶段的约0.36增加至高负载阶段的约0.46,整体明显低于Fixed-High,并与Rule-Based和RL-Based保持相近或更稳定的水平。Fixed-Low虽然具有最低服务时延,

但其低服务时延是以牺牲安全收益和恢复效能为代价取得的。AIOAO算法在服务时延上的优势主要来自其对安全收益和服务时延的联合权衡。较大的并发实例数量 n_t 和一致性阈值 k_t 虽有助于提升异常输出抑制能力,但会增加请求分发、实例等待和裁决比较开销。Fixed-High长期维持高冗余配置,因此在负载升高时服务时延快速增大;Fixed-Low虽然服务时延较低,但其安全收益和恢复效能明显不足。AIOAO算法在动作选择过程中将服务时延作为偏好效用的重要组成部分,并通过服务等级约束限定可行动作集合,因此不会无条件增加冗余规模,而是在安全增强与时延控制之间进行折中。

从图2(c)可以看出,AIOAO算法在多数时隙内均表现出较高的恢复效能,在低负载、平稳负载和高负载阶段的恢复效能分别约为0.65、0.67和0.70,与其他4种方法相比始终保持最高水平。AIOAO算法恢复效能较高的原因在于,其动作选择同时考虑恢复反馈和未来状态不确定性。当系统检测到响应分歧或恢复反馈变差时,AIOAO算法能够将这些观测反映到隐状态后验信念中,并在期望自由能评价中优先选择有利于降低未来状态不确定性、增强有效裁决恢复能力的参数配置。与仅依赖固定阈值的静态方法相比,AIOAO算法能够根据恢复压力变化调整裁决参数;与Rule-Based相比,AIOAO算法不仅根据当前观测作出反应,还能够结合状态转移模型进行前瞻性判断;与RL-Based相比,AIOAO算法不依赖大量试错探索,因此在阶段切换和恢复调整过程中表现出更稳定的恢复效能。

特别地,在第330时隙附近,系统由平稳负载阶段进入高负载阶段,请求到达强度和实例风险强度同步上升,导致响应一致性观测量下降、服务时延上升,并伴随恢复反馈变化。对于AIOAO算法而言,这些观测会通过式(20)和式(21)转化为隐状态后验信念的变化,使系统对高风险状态和恢复调整状态的置信度上升。随后,算法在候选裁决动作集合中重新计算期望自由能,并倾向于选择更大的并发实例规模 n_t 和更严格的一致性阈值 k_t ,以提升异常输出抑制能力和恢复有效性。与之相比,Fixed-Low在该阶段仍保持低冗余配置,难以及时提升安全收益和恢复效能;Fixed-High虽能维持较高冗余,但服务时延随负载上升显著增大;Rule-

Based 主要依赖当前观测阈值进行反应式调整, 在环境突变后存在一定滞后; RL-Based 受探索过程和策略泛化能力影响, 在阶段切换附近容易出现波动。

综合图 2(a)~图 2(c) 可以看出, AIOAO 算法并不是在单一指标上追求极端最优, 而是在 500 个时隙的动态运行过程中实现了更优的综合平衡。

4.3.2 裁决参数动态调整分析

为进一步说明 AIOAO 算法总体性能优势的来源, 本文对 500 个时隙内裁决参数 n_t 和 k_t 的动态变化过程进行分析, 结果如图 3 所示。

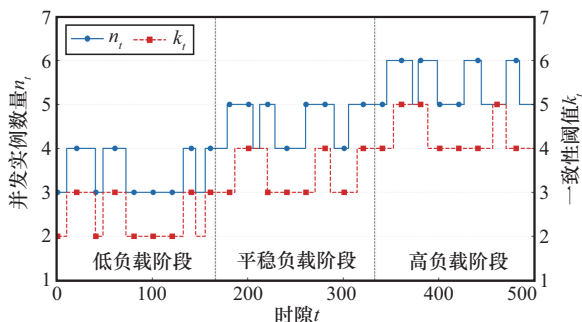


图3 AIOAO 算法裁决参数动态调整曲线

由图 3 可见, AIOAO 算法的裁决参数会随负载与风险水平变化而动态调整。在低负载阶段, n_t 主要分布在 3~4, k_t 主要分布在 2~3, 说明 AIOAO 算法倾向于采用较小的并发实例规模和较宽松的一致性阈值, 以降低额外开销; 与此对应, 图 2 中其服务时延约为 0.36, 而安全收益和恢复效能仍分别保持在 0.66 和 0.65 左右。进入平稳负载阶段后, n_t 逐步提升至 4~5, k_t 主要位于 3~4, 对应图 2 中安全收益由约 0.66 提升至约 0.69, 恢复效能由约 0.65 提升至约 0.67, 而服务时延仅由约 0.36 上升至约 0.41。在高负载阶段, n_t 进一步提升至 5~6, k_t 主要分布在 4~5, 并在局部时隙内出现短时增强, 对应图 2 中安全收益约为 0.74、峰值接近 0.77, 恢复效能稳定在 0.69~0.70, 而服务时延约为 0.46, 虽较前两个阶段有所上升, 但仍明显低于 Fixed-High 的约 0.73。整体来看, AIOAO 算法并未长期固定采用某一组裁决参数, 而是能够随负载与风险水平变化, 对并发实例数量和一致性阈值进行阶段性调整, 在较高冗余与时延可控之间实现动态平衡。

此外, AIOAO 算法的参数更新虽然会随环境变化进行局部调整, 但整体上呈现出清晰的阶段性

抬升特征, 并未出现无界振荡或持续剧烈切换。这说明所设计的平滑更新机制能够在保证参数适应性的同时, 抑制由短时观测波动带来的频繁调整。

4.3.3 AIOAO 算法收敛性分析

为进一步验证 AIOAO 算法在动态不确定环境下的稳定性, 本文对算法在 500 个时隙内期望自由能随时间变化的情况进行分析, 结果如图 4 所示。

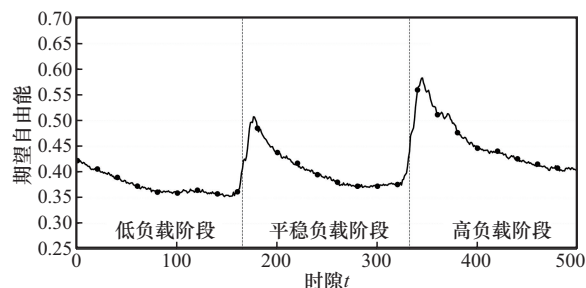


图4 期望自由能随时间变化的收敛曲线

由图 4 可见, 在低负载、平稳负载和高负载 3 个阶段内, 期望自由能整体保持在较低波动范围内, 说明 AIOAO 算法的隐状态信念能够较稳定地解释当前观测。当系统在阶段切换点附近发生负载和风险水平变化时, 期望自由能会出现短暂上升, 表明原有信念与新观测之间出现暂时不匹配; 随后, 算法通过式(20)和式(21)持续更新后验信念, 并通过式(27)和式(28)选择更合适的裁决动作, 使期望自由能逐步下降并重新进入稳定区间。该结果表明, AIOAO 算法能够在动态不确定环境下维持较好的模型匹配能力和在线决策稳定性。

4.3.4 超参数敏感性分析

为验证 AIOAO 算法关键超参数设置的合理性及算法对参数变化的鲁棒性, 本文在其余参数保持不变的条件下, 对不确定性权衡系数 λ 和平滑更新门限 ε 进行敏感性分析, 并分别统计不同取值下 500 个时隙内安全收益、服务时延和恢复效能的平均值, 结果如图 5 和图 6 所示。

从图 5 可以看出, λ 对 AIOAO 算法的 3 项性能指标具有较明显影响。随着 λ 由 0.05 增大到 0.35, 安全收益由约 0.66 提升至约 0.73, 恢复效能由约 0.64 提升至约 0.69, 而服务时延则由约 0.45 下降至约 0.40; 当 λ 继续增大到 0.50 时, 安全收益和恢复效能分别回落至约 0.71 和 0.68, 服务时延回升至约 0.43。这表明, 当 λ 过小时, 算法更偏重即时目标偏好, 容易导致参数更新过于激进; 当 λ 过大时,

又会过度强调未来状态不确定性的抑制,使得参数调整对环境变化的响应不够敏感。综合3项性能指标可知, λ 取0.30~0.35时整体性能较优,说明适中的不确定性权衡更有利于在安全收益、服务时延和恢复效能之间取得平衡。

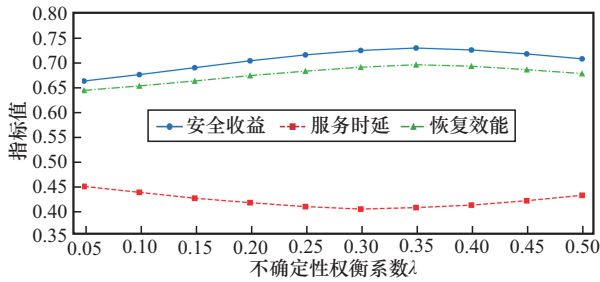


图5 不确定性权衡系数 λ 对3项性能指标的影响

从图6可以看出, ε 的变化主要影响裁决参数更新的灵敏度。当 ε 从0.005增大到0.015时,安全收益由约0.70提升至约0.73,恢复效能由约0.67提升至约0.69,而服务时延则由约0.42下降至约0.40;当 ε 继续增大到0.050时,安全收益下降至约0.66,恢复效能下降至约0.63,而服务时延上升至约0.46。这说明,当 ε 过小时,裁决参数切换过于频繁,容易引入额外开销;当 ε 过大时,AIOAO算法又会因更新门限过高而难以及时响应环境变化。图6表明, $\varepsilon = 0.015$ 附近3项性能指标的综合表现最佳,说明该取值能够在抑制频繁切换和保持动态适应能力之间实现较好的平衡。

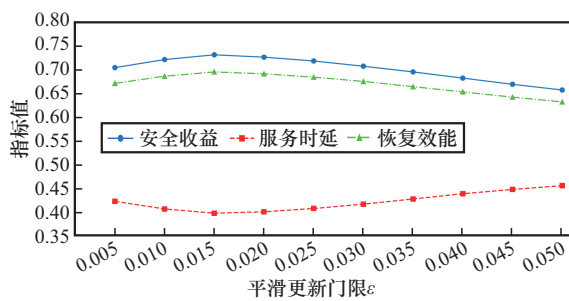


图6 平滑更新门限 ε 对3项性能指标的影响

综上,AIOAO算法能够依据负载与风险变化动态调整裁决参数,在安全收益、服务时延和恢复效能之间取得较优平衡,验证了算法的有效性。其优势并非来自固定高冗余配置,而是源于隐状态推断、期望自由能评价与平滑更新机制的协同作用,体现了本文算法相较于静态配置、规则调整和强化学习基线的创新性。

5 结束语

面向拟态云服务中的多余度裁决,本文AIOAO算法,以解决裁决参数配置难以适应动态不确定环境、难以兼顾安全收益、服务时延与恢复效能的问题。首先,建立拟态云服务多余度裁决过程的系统模型,对安全收益、服务时延和恢复效能等关键指标进行形式化描述,并据此构建面向多目标协调的裁决参数在线优化问题。其次,通过引入主动推理框架,将裁决参数动态更新过程建模为基于隐状态推断和期望自由能最小化的决策过程。最后,仿真实验表明,AIOAO算法能够根据负载与风险水平变化动态调整裁决参数,相比固定裁决参数、规则驱动和强化学习等方法,在安全收益、服务时延和恢复效能之间取得更优的综合平衡。

参考文献:

- [1] 罗军舟,金嘉晖,宋爱波,等.云计算:体系架构与关键技术[J].通信学报,2011,32(7):3-21.
- [2] Luo J Z, Jin J H, Song A B, et al. Cloud computing: architecture and key technologies[J]. Journal on Communications, 2011, 32(7): 3-21.
- [3] Arthurs P, Gillam L, Krause P, et al. A taxonomy and survey of edge cloud computing for intelligent transportation systems and connected vehicles[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(7): 6206-6221.
- [4] Zhu K G, Zhang Z J, Zeadally S, et al. Learning to optimize workflow scheduling for an edge-cloud computing environment[J]. IEEE Transactions on Cloud Computing, 2024, 12(3): 897-912.
- [5] Gannon D, Barga R, Sundaresan N. Cloud-native applications[J]. IEEE Cloud Computing, 2017, 4(5): 16-21.
- [6] Zhou Y Y, Cheng G, Du K. Enhancing microservice security through adaptive moving target defense policies to mitigate DDoS attacks in cloud-native environments[J]. Future Internet, 2025, 17(12): 580.
- [7] 郭江兴. 网络空间拟态防御研究[J]. 信息安全学报, 2016, 1(4): 1-10.
- [8] Wu J X. Research on cyber mimic defense[J]. Journal of Cyber Security, 2016, 1(4): 1-10.
- [9] Hu H C, Wu J X, Wang Z P, et al. Mimic defense: a designed-in cybersecurity defense framework[J]. IET Information Security, 2018, 12(3): 226-237.
- [10] Shao S S, Gu T S, Nie Y J, et al. An active defense adjudication method based on adaptive anomaly sensing for mimic IoT[J]. IEEE Transactions on Services Computing, 2025, 18(1): 57-71.
- [11] 普黎明,卫红权,李星,等.面向云应用的拟态云服务架构[J].网络与信息安全学报,2021,7(1):101-112.

- Pu L M, Wei H Q, Li X, et al. Mimic cloud service architecture for cloud applications[J]. Chinese Journal of Network and Information Security, 2021, 7(1): 101-112.
- [10] Pu L M, Wu J X, Ma H L, et al. MimicCloudSim: an environment for modeling and simulation of mimic cloud service[J]. China Communications, 2021, 18(1): 212-221.
- [11] 吴铤, 胡程楠, 陈庆南, 等. 基于执行体划分的防御增强型动态异构冗余架构[J]. 通信学报, 2021, 42(3): 122-134.
- Wu T, Hu C N, Chen Q N, et al. Defense-enhanced dynamic heterogeneous redundancy architecture based on executor partition[J]. Journal on Communications, 2021, 42(3): 122-134.
- [12] 李卫超, 张铮, 王立群, 等. 基于拟态防御架构的多余度裁决建模与风险分析[J]. 信息安全学报, 2018, 3(5): 64-74.
- Li W C, Zhang Z, Wang L Q, et al. The modeling and risk assessment on redundancy adjudication of mimic defense[J]. Journal of Cyber Security, 2018, 3(5): 64-74.
- [13] Zheng Y X, Zhang B Y, Liu Y. Reliability assessment of reconfigurable k-out-of-n systems with functional dependency[J]. IEEE Transactions on Reliability, 2025, 74(3): 3745-3759.
- [14] Xia H Y, Hao J, Cao R J. Adaptive resource allocation for cloud-native microservice via meta-learning and hyper-heuristic algorithms[J]. Journal of Grid Computing, 2025, 23(4): 27.
- [15] Malekzadeh P, Plataniotis K N. Active inference and reinforcement learning: a unified inference on continuous state and action spaces under partial observability[J]. Neural Computation, 2024, 36(10): 2073-2135.
- [16] Li L S, Wu J X, Hu H C, et al. Secure cloud architecture for 5G core network[J]. Chinese Journal of Electronics, 2021, 30(3): 516-522.
- [17] 张文建, 宋克, 谭力波, 等. 面向拟态判决的可编程语义解析方法[J]. 通信学报, 2020, 41(4): 62-69.
- Zhang W J, Song K, Tan L B, et al. Programmable semantic parsing approach for mimic arbitration[J]. Journal on Communications, 2020, 41(4): 62-69.
- [18] 杨晓晗, 程国振, 刘文彦, 等. 基于深度学习的拟态裁决方法研究[J]. 通信学报, 2024, 45(02): 79-89.
- Yang X H, Cheng G Z, Liu W Y, et al. Research on mimic decision method based on deep learning[J]. Journal on Communications, 2024, 45(2): 79-89.
- [19] 周大成, 陈鸿昶, 程国振, 等. 面向持久性连接的自适应拟态表决器设计与实现[J]. 通信学报, 2022, 43(6): 71-84.
- Zhou D C, Chen H C, Cheng G Z, et al. Design and implementation of adaptive mimic voting device oriented to persistent connection[J]. Journal on Communications, 2022, 43(6): 71-84.
- [20] Xiahou T F, Zheng Y X, Liu Y, et al. Reliability modeling of modular k-out-of-n systems with functional dependency: a case study of radar transmitter systems[J]. Reliability Engineering & System Safety, 2023, 233: 109120.
- [21] Arulkumaran K, Deisenroth M P, Brundage M, et al. Deep reinforcement learning: a brief survey[J]. IEEE Signal Processing Magazine, 2017, 34(6): 26-38.
- [22] Gu S D, Yang L, Du Y L, et al. A review of safe reinforcement learning: methods, theories, and applications[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024, 46(12): 11216-11235.
- [23] Zhu Y H, Cui G C, Liu A B, et al. A reinforcement learning embedded surrogate Lagrangian relaxation method for fast solving unit commitment problems[J]. IEEE Transactions on Power Systems, 2025, 40(5): 3806-3818.
- [24] Friston K, FitzGerald T, Rigoli F, et al. Active inference: a process theory[J]. Neural Computation, 2017, 29(1): 1-49.
- [25] Sajid N, Ball P J, Parr T, et al. Active inference: demystified and compared[J]. Neural Computation, 2021, 33(3): 674-712.
- [26] Dean J, Barroso L A. The tail at scale[J]. Communications of the ACM, 2013, 56(2): 74-80.
- [27] Rockafellar R T, Uryasev S. Optimization of conditional value-at-risk[J]. The Journal of Risk, 2000, 2(3): 21-41.
- [28] Hong Y L. On computing the distribution function for the Poisson binomial distribution[J]. Computational Statistics & Data Analysis, 2013, 59: 41-51.
- [29] Jiang C F, Qiu Y T, Shi W S, et al. Characterizing co-located workloads in Alibaba cloud datacenters[J]. IEEE Transactions on Cloud Computing, 2022, 10(4): 2381-2397.
- [30] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540): 529-533.

[作者简介]



张乾坤 (1995-), 男, 博士, 华中科技大学副研究员, 主要研究方向为网络安全、可信人工智能。



刘冰 (2000-), 男, 华中科技大学博士生, 主要研究方向为网络安全、可信人工智能。



盛章骏 (2002-), 男, 华中科技大学硕士生, 主要研究方向为网络安全。



邓贤君 (1981-), 男, 博士, 华中科技大学教授, 主要研究方向为物联网安全。



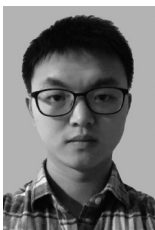
梁浩 (1987-), 男, 博士, 网络空间部队信息工程大学副研究员, 主要研究方向为网络空间主动防御、软件异构化编译。



袁斌 (1990-), 男, 博士, 华中科技大学副教授, 主要研究方向为漏洞检测、网络安全。



程国振 (1986-), 男, 博士, 网络空间部队信息工程大学副教授, 主要研究方向为网络内生安全、AI 安全、新型网络体系结构。



张帅 (1994-), 男, 博士, 网络空间部队信息工程大学助理研究员, 主要研究方向为网络安全、云计算、恶意代码。



丁国栋 (1988-), 男, 嵩山实验室中级工程师, 主要研究方向为网络安全、云原生。