任务专用机器网络中作业年龄最小化的抢占式调度

Subhankar Banerjee Sennur Ulukus 马里兰
大学电气与计算机工程系，马里兰州大学公园市
20742 sbanerje@umd.edu ulukus@umd.edu

arXiv:2602.02435v1 [cs.IT] 2 Feb 2026

摘要——我们研究一种时隙作业分配系统，该系统由一台中央服务器、 N 个任务专用机器网络和多个用户组成。每个网络专门执行一种特定类型的任务。用户以随机方式生成各种类型的作业，并将其发送至中央服务器，由服务器将每个作业路由至相应的机器网络。由于资源约束，服务器无法同时处理所有用户的作业，因此需要设计能够进行抢占的调度策略。为评估调度性能，我们提出一种新的及时性指标——作业年龄，其灵感来自广为人知的信息年龄指标。我们研究长期加权平均作业年龄最小化问题。首先，我们通过最小化单步李雅普诺夫漂移提出一种最大权重策略；随后，在机器网络的作业完成时间服从几何分布时，推导出 Whittle 指数（ WI ）策略。对于一般的作业完成时间分布，我们提出带最大权重回退的 Whittle 指数（WIMWF）策略。我们还研究净收益最大化（NGM）策略。数值结果表明，在一般作业完成时间的情形下，所提出的 WIMWF 策略性能最佳。我们还观察到一种规模扩展趋势：在小规模系统中，两种不同的最大权重策略均可能优于 NGM 策略；随着系统规模扩大，NGM 策略的性能有所提升，并渐近地优于最大权重策略。对于几何服务时间，WI 策略在所有考察的系统规模下均能实现最低的作业年龄。

I. 引言

在情报、监视和其他任务关键型控制任务等应用中，将作业卸载到边缘计算设备已成为处理时延敏感型工作负载的一种关键机制 [1]。不同的应用和用户需要不同类型的计算任务，并且具有异构的时延要求。这自然形成了一种系统架构，其中包含多个由边缘计算设备构成的任务专用网络，每个网络专门处理一类特定作业。为确保每项任务都被路由到适当的网络，我们假设存在一台中央服务器，负责收集所有用户或应用的任务请求，并将其转发到相应的边缘网络。

这种架构与近期任务专用大语言模型（LLM）的发展密切契合。训练单个大型通用大语言模型通常需要海量数据集、漫长的训练时间和大量计算资源；尽管这类模型在各种任务上表现出色，但相关的训练和部署成本可能成为瓶颈。相比之下，参数规模较小的任务专用大语言模型可以在更狭窄的领域内更高效地训练，所需的数据和时间更少，而且在其专门设计的特定任务上，往往优于大型通用大语言模型；例如，参见 [2]。我们研究中考虑的模型自然能够描述在任务专用大语言模型网络中路由任务的问题，同时也具有足够的通用性，可涵盖其他任务专用边缘计算服务和架构。

在本文中，我们假设共有 N 个用户组和 M 个用户；第 i 组由 M_i 个用户组成，且同一组中的每位用户都对同一个特定应用感兴趣。每个时隙中，来自第 i 组的用户 j 以 $p_j^i > 0$ 的概率向中央服务器提交作业请求。对于第 i 个用户组，存在相应的第 i 个边缘计算设备网络，专门执行该组 i 感兴趣的任务。我们施加资源约束：第 i 个边缘¯网络最多可并发执行 M_i 个作业，而¯中央服务器最多可并发处理 M 个作业。因此，服务器必须采用一种调度算法，以决定每个时隙服务哪些作业。在给定时隙中，如果服务器决定为第 i 组的用户 j 执行作业，就会将该作业路由到边缘网络 i，随后该网络会按照其服务时间分布执行作业。本文中，边缘计算设备与机器这两个术语可互换使用，边缘网络与机器网络这两个术语也同样如此。我们在图1中以示意图表示该系统。

在本研究中，我们允许以抢占恢复服务的方式抢占正在服务的作业：可以中断正在执行的作业，以服务时延敏感度更高的作业；被中断作业的进度会保存在缓冲区中，之后从中断处继续执行，而不是被丢弃。当用户向中央服务器提交作业时，我们的目标是及时处理该作业。受广为人知的信息年龄指标启发，我们提出一种新颖的及时性指标——作业年龄。为体现不同作业在及时性和时延方面的相对重要性，我们对每个组中每位用户的作业年龄赋予权重。我们的目标是为服务器设计近似最优的调度策略，使系统中的长期平均加权作业年龄最小化。

一个自然的问题是，为什么我们考虑作业年龄型惩罚，而不只是最小化用户提交作业与边缘服务器执行作业之间的延迟。在后续工作中，我们将证明，如果我们只优化延迟，而不采用作业年龄惩罚，那么在最优调度策略下，服务器可能永远不会为某些用户安排作业，这是

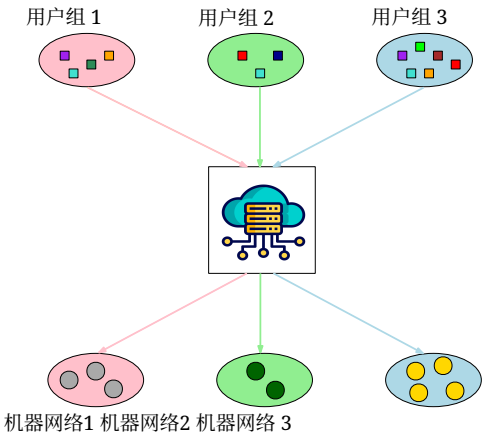

图1. 彩色方块代表用户。顶部的三个椭圆代表用户组，每个组都对应一项特定任务。一个用户可能属于多个组；例如，红色用户同时属于绿色组和蓝色组。用户将请求提交给中央服务器，服务器再将请求路由至相应的任务专用机器网络，如底部椭圆所示，其中机器以圆形表示。例如，灰色机器负责处理与粉色组相关的任务。 [图片来源：中央处理器图标由 RaftelDesign 制作，来自 www.flaticon.com。]

不理想。此外，我们提出的指标“作业年龄”与 [3], 中引入的“作业完成年龄”相似，只有一处细微差别，我们将在第二节中加以说明。因此，与 [3], 类似，我们认为，最小化对应于某个用户的作业年龄，等价于最大化该用户单位时间内完成的作业数。

我们提出了多种调度策略。首先，通过最小化单时隙李雅普诺夫漂移，推导出最大权重策略。随后，我们将调度问题建模为不安分多臂赌博机（RMAB）问题。在开创性论文中， [4],Whittle 提出了一种基于指标的策略，用于求解具有二元动作的 RMAB 问题；此后，许多研究针对各种情形推导了 Whittle 指数策略，例如参见 [5]。应用 Whittle 指数的一项关键挑战在于，需要证明底层问题具有可索引性。对于许多系统，证明可索引性并不容易，因此也有研究考察了无需可索引性的净收益最大化（ NGM）策略 [6]–[9]。对于我们的模型，当边缘计算设备的服务时间服从几何分布时，我们证明了其可索引性，并推导出相应的 Whittle 指数策略。然而，对于一般服务时间分布，我们无法证明其具有可索引性。在这种情况下，受 Whittle 指数框架启发，我们提出了带最大权重回退的 Whittle 指数（ WIMWF ）策略。我们还针对该问题推导了 NGM 策略。我们在第八节通过数值分析比较了所提出策略的性能。

II. 相关工作

近年来，将作业卸载到远程机器的研究已十分广泛；例如，参见 [3], [9]–[14]。这些研究都将每台远程机器建模为马尔可夫机器：当机器未处理服务器分配的作业时，它可能正忙于处理内部作业，也可能处于空闲状态，而其忙碌或空闲状态按照马尔可夫链演变。因此，如果机器正忙于处理内部作业，服务器提交的作业就可能被丢弃，而跟踪机器状态也成了核心难题。在 [10], 中，作者考虑在最小化错误信息年龄的定制版本的同时，兼顾作业丢弃惩罚；而 [9] 则在一个包含多个作业调度器和多台马尔可夫机器的网络中，最小化作业丢弃不确定性。

由于需要跟踪马尔可夫机器， [3], [11], [13], [14] 中的工作假设作业分配实体（中央服务器、作业调度器或资源分配器）会周期性地请求或采样每台机器的状态，以获得对其状态更好的估计。在其中一些工作中，采样成本被显式建模：例如，在 [14], 中，作者优化了成功作业完成与采样成本之间的差分收益，而在 [3], 中，引入了一种新颖的作业完成年龄指标，并在队列稳定条件下将平均作业完成年龄与采样成本一同最小化。其他工作施加了采样约束，并研究了采样率分配问题 [11], [13]。在 [11], 中，资源分配器将作业分配给一个机器网络，并在采样约束下跟踪其状态；为评估跟踪性能，作者使用了众所周知的二元新鲜度指标，并引入了两个额外指标：误接受率和误拒绝率。在 [13],中，作者考虑了一个由耗竭型工作节点组成的网络，其效率取决于其内部状态，资源分配器旨在确定一种最优速率分配方案，以最大化工作节点处于其最高效状态的概率。

相比之下，在我们的模型中，未在执行中央服务器分配的作业的机器只是处于空闲状态，不存在内部作业进程。因此，一旦服务器将作业分配给某台机器，就不存在不确定性：该作业将被执行且不会被丢弃，也无需显式跟踪机器状态。我们还优化了不同的及时性指标：提出了新的指标——作业年龄。与我们的指标最接近的是 [3] 提出的作业完成年龄；然而，与作业完成年龄不同，只有当中央服务器中存在属于该用户的作业时，作业年龄才会增加，而且我们的网络化架构比 [3] 所研究的单台马尔可夫机器更复杂。此外，虽然 [3] 关注的是一类随机化策略，但我们研究了多个策略族，包括已知在某些技术条件下具有渐近最优性的 Whittle 指数策略 [5]。最后， [3], [9]–[14] 中的研究均不允许抢占正在服务的作业，而我们的模型明确纳入了抢占后恢复服务，因此我们的问题与这些研究有本质区别。

在 [15],中，作者考虑了一个系统，其中包含多个面向特定任务的大语言模型网络，以及多个向中央服务器提交二元查询的用户。尽管该系统模型与我们的模型相近，但他们通过优化每个网络中的大语言模型数量来平衡信息准确性与响应延迟，且不允许抢占。在 [16], 中，作者考虑了一种双模式作业卸载系统，其中作业要么

完全在云端处理，要么部分在本地服务器处理、部分在云端处理，目标是最小化延迟；同样，这里也未考虑抢占。

在信息年龄领域，针对各种网络模型的抢占式传输策略已得到研究；例如，参见 [17]–[23]。一些研究在始终抢占或概率性抢占策略下推导出了信息年龄闭式表达式 [17], [18], [22], [23], ，另一些研究则提出优化问题，以确定最优抢占策略 [19]–[21]。我们的工作与这些研究相关，因为我们同样设计并分析抢占策略，但有一个重要区别。在大多数基于信息年龄的抢占研究中，抢占通常意味着丢弃正在服务的数据包；而我们的模型采用抢占后恢复服务，即中断作业时暂停其执行并保存进度，之后从中断处继续执行，而不是将其丢弃。

III. 系统模型与问题表述

我们用集合 $\mathcal{M}_i$ 表示第 i 个用户组。令 $M_i = |\mathcal{M}_i|$。用户集合不必互不相交，即同一用户可能关联到由不同网络执行的多个任务。因此，一般而言，我们有 $\sum_{i=1}^{N} M_i \geq M$。在时刻 t，我们用指示随机变量 $a_j^i(t)$ 表示用户 $j \in \mathcal{M}_i$ 的作业生成事件，其中 $a_j^i(t) = 1$ 表示生成了一个作业，而 $a_j^i(t) = 0$ 则表示没有生成作业。对于用户 $j \in \mathcal{M}_i$，中央服务器维护一个单位容量缓冲区，其在时刻 t 的状态用指示变量 $b_j^i(t)$ 表示，其中 $b_j^i(t) = 1$ 表示在时刻 t，用户 j 在中央服务器上存有一个作业，而 $b_j^i(t) = 0$ 则表示没有作业。我们假设，如果缓冲区中已有一个作业，而同一用户又有新的作业请求到达，则丢弃新请求。此外，如果某个用户属于 k 个不同的集合，中央服务器会为该用户维护 k 个互不相同的单位容量缓冲区，每个用户组对应一个。

在时刻 t，若第 i 个网络为用户 $j \in \mathcal{M}_i$ 服务一个作业，则指示变量 $c_j^i(t)$ 的值为 1，否则为 0。我们假设，在每个机器网络中，完成任何作业至少需要一个时隙。此外，第 i 个网络中的服务（作业完成）时间被建模为一个随机变量，该变量在不同作业请求之间独立同分布，并且在不同机器网络之间相互独立。我们用 Y_i 表示该服务时间，其概率质量函数为 $\boldsymbol{f}^i$，即 $P(Y_i = k) = \boldsymbol{f}^i(k)$。因此，分配给第 i 个网络的作业以 k 个时隙完成的概率为 $\boldsymbol{f}^i(k)$。由于不同作业的处理时间独立同分布，Y_i 不依赖于作业索引。

我们进一步假设，机器在一个时隙结束时完成作业，并用指示随机变量 $d_j^i(t)$ 表示这一事件。具体而言，$d_j^i(t) = 1$ 表示在时隙 t 结束时，对应用户 $j \in \mathcal{M}_i$ 的作业已完成，而 $d_j^i(t) = 0$ 则表示相反情况。于是，$b_j^i(t)$ 的演变可表示为

$$b_j^i(t) = \min\{b_j^i(t-1) - d_j^i(t-1) + a_j^i(t), 1\}. \tag{1}$$

在时刻 t，我们将对应用户 $j \in \mathcal{M}_i$ 的作业的年龄记为 $\Delta_j^i(t)$，其定义为当前作业在对应作业缓冲区中截至时刻 t 已停留的时长。形式化地，

$$u_j^i(t) = \sup\{t' \mid t' < t,\ d_j^i(t') = 1\}, \tag{2}$$

$$\bar{u}_j^i(t) = \inf\{t' \mid u_j^i(t) < t' \leq t,\ a_j^i(t') = 1\}, \tag{3}$$

$$\Delta_j^i(t) = \begin{cases} t - \bar{u}_j^i(t), & \bar{u}_j^i(t) \neq 0, \\ 0, & \bar{u}_j^i(t) = 0. \end{cases} \tag{4}$$

在 (2) 和 (3) 中，我们假设空集的上确界和下确界均为 0。

形式化地，对于每个时刻 t，我们有以下约束：

$$\sum_{j=1}^{M_i} c_j^i(t) \leq \bar{M}_i, \quad \forall i \in \{1, 2, \cdots, N\}, \tag{5}$$

$$\sum_{i=1}^{N} \sum_{j=1}^{M_i} c_j^i(t) \leq \bar{M}. \tag{6}$$

为了使我们的问题具有非平凡性，我们假设

$$\sum_{i=1}^{N} \bar{M}_i > \bar{M}, \quad \bar{M} > \bar{M}_i, \quad \forall i \in \{1, 2, \cdots, N\}. \tag{7}$$

在我们的系统中，不同类型的作业可能具有不同的优先级。为对此进行建模，我们为每个作业的年龄赋予一个权重。对于用户 $j \in \mathcal{M}_i$ 的作业，我们赋予权重 w_j^i，其中权重越大，优先级越高。我们考虑以下优化问题：

$$\inf_{\pi \in \Pi} \limsup_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{N} \sum_{j=1}^{M_i} \mathbb{E}_\pi \left[w_j^i \Delta_j^i(t) \right]$$
$$\text{s.t.} \quad (5),\ (6), \quad \forall t \in \{1, 2, \cdots\}, \tag{8}$$

其中 Π 表示所有因果调度策略的集合。注意，在 (8) 中，我们隐含地假设初始条件 $\Delta_j^i(0) = 1$ 对所有 $j \in \mathcal{M}_i$ 和 $i \in \{1, 2, \cdots, N\}$ 均成立，因此未在 (8) 中明确说明。为方便记号，我们记

$$\Delta^{\text{avg},\pi} = \limsup_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{N} \sum_{j=1}^{M_i} \mathbb{E}_\pi \left[w_j^i \Delta_j^i(t) \right]. \tag{9}$$

由于（5）中的约束，可能发生网络级服务中断；例如，在时刻 t，如果网络 i 正在处理 M_i 个作业时，来自 $\mathcal{M}_i$ 的新作业请求到达服务器，就会出现这种情况。类似地，由于（6）中的约束，当中央服务器在时刻 t 已经在处理 M 个作业，而来自任一 N 用户集合的新作业请求到达时，可能发生服务器级服务中断。为应对此类情况，我们允许采用第一节所述的抢占恢复方式进行服务抢占。当用户 j 的作业被抢占时，中央服务器会将其进度存储在前文介绍的同一作业缓冲区中。抢占决策仅由中央服务器在每个时隙开始时作出。我们将一种调度策略 π 表示为

$$\pi = \left(\left(\left(c_j^i(t) \right)_{j=1}^{M_i} \right)_{i=1}^{N} \right)_{t=1}^{\infty}. \tag{10}$$

假设用户 j 在 $\mathcal{M}_i$ 中的一个作业在过去的 $x>0$ 个时隙内一直得到处理。在时隙 t开始时，服务器抢占该作业并将其存储在缓冲区中。此外，假设服务器在时隙 $t+x_1$开始时恢复处理该作业，其中 $x_1>0$。那么，该作业在时隙 $t+x_1$结束时完成的概率为，

$$\frac{\boldsymbol{f}^i(x+1)}{\sum_{k=x+1}^{\infty}\boldsymbol{f}^i(k)}.$$

我们用 $\bar{\Delta}_j^i(t)$ 表示截至时间 t （不包括t）与用户 $j\in\mathcal{M}_i$ 对应的作业已处理的时间。形式上，

$$\bar{\Delta}_j^i(t)=\sum_{\tau=\bar{u}_j^i(t)}^{t-1}c_j^i(\tau). \tag{11}$$

在时间 t，我们用 $s_j^i(t)=(\Delta_j^i(t)\ \bar{\Delta}_j^i(t))$ 表示与用户 $j\in\mathcal{M}_i$ 对应的状态。同样，网络 i 在时间 t 的状态为 $s_i(t)=(s_j^i(t))_{j=1}^{M_i}$，整个系统的状态为 $s(t)=(s_i(t))_{i=1}^{N}$。在策略 π 下，中央服务器在每个时间 t 观察 $s(t)$ 和 $\left((a_j^i(t))_{j=1}^{M_i}\right)_{i=1}^{N}$，以决定服务哪些作业。

IV. 最大权重策略

在本节中，我们通过最小化一步条件李雅普诺夫漂移来推导最大权重策略，该漂移在时间 t 定义为：

$$\mathbb{E}[\mathcal{L}(s(t+1))-\mathcal{L}(s(t))\,|\,s(t)], \tag{12}$$

其中我们使用李雅普诺夫函数

$$\mathcal{L}(s(t))=\sum_{i=1}^{N}\sum_{j\in\mathcal{I}_i(t)}w_j^i\Delta_j^i(t), \tag{13}$$

其中 $\mathcal{I}_i(t)$表示组 i 中在时刻 t于中央服务器上有作业的用户集合。注意

$$\mathbb{E}[\mathcal{L}(s(t+1))|s(t)]=\left[\sum_{i=1}^{N}\sum_{j\in\mathcal{I}_i(t)}(1-c_j^i(t))w_j^i(\Delta_j^i(t)+1)+c_j^i(t)w_j^i\frac{\sum_{k=2}^{\infty}\boldsymbol{f}^i(\bar{\Delta}_j^i(t)+k)}{\sum_{k=1}^{\infty}\boldsymbol{f}^i(\bar{\Delta}_j^i(t)+k)}(\Delta_j^i(t)+1)\right]. \tag{14}$$

我们定义，

$$\alpha_j^i(t)\triangleq\frac{\sum_{k=2}^{\infty}\boldsymbol{f}^i\big(\bar{\Delta}_j^i(t)+k\big)}{\sum_{k=1}^{\infty}\boldsymbol{f}^i\big(\bar{\Delta}_j^i(t)+k\big)}. \tag{15}$$

则单步李雅普诺夫漂移为，

$$\sum_{i=1}^{N}\sum_{j\in\mathcal{I}_i(t)}w_j^i\Big(1+c_j^i(t)\big(\alpha_j^i(t)-1\big)\big(\Delta_j^i(t)+1\big)\Big). \tag{16}$$

(16) 中的第一项与所有 $i\in\{1,2,\cdots,N\}$ 和 $j\in\mathcal{I}_i(t)$ 的调度决策 $c_j^i(t)$ 无关。因此，最小化单步李雅普诺夫漂移等价于最大化，

$$\sum_{i=1}^{N}\sum_{j\in\mathcal{I}_i(t)}c_j^i(t)w_j^i\big(\Delta_j^i(t)+1\big)\big(1-\alpha_j^i(t)\big). \tag{17}$$

因此，为实现所提出的最大权重策略，我们在每个时刻 t求解如下优化问题：

$$\max_{\left((c_j^i(t))_{j=1}^{M_i}\right)_{i=1}^{N}}\ \sum_{i=1}^{N}\sum_{j\in\mathcal{I}_i(t)}c_j^i(t)w_j^i\big(\Delta_j^i(t)+1\big)\big(1-\alpha_j^i(t)\big)$$
$$\text{s.t.}\quad (5),\ (6). \tag{18}$$

V. NGM Policy

式(8)中的问题构成一个不安分多臂赌博机问题，其中每个用户 $j\in\mathcal{M}_i$ 被视为一条臂，该臂在时刻 t 的状态记为 $s_j^i(t)$。本节中，我们研究NGM策略以求解式(8)中的优化问题。

A. 拉格朗日松弛与对偶分解

(8) 中的约束 (5) 和 (6) 是逐时隙约束，即中央服务器必须在每个时隙都满足这些约束。我们将这些约束松弛为期望时间平均约束，从而得到以下问题：

$$\begin{aligned}&\inf_{\pi\in\Pi}\limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{j=1}^{M_i}\mathbb{E}_\pi[w_j^i\Delta_j^i(t)]\\&\text{s.t.}\limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{M_i}\mathbb{E}_\pi[c_j^i(t)]\le\bar{M}_i,\ \forall i\in1,2,\cdots,N\\&\limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{j=1}^{M_i}\mathbb{E}_\pi[c_j^i(t)]\le\bar{M}.\end{aligned} \tag{19}$$

现在，我们采用 $(N+1)$ 拉格朗日乘子 $\mu_i\ge0$ for $i\in\{1,2,\cdots,N\}$ and $\lambda\ge0$，为问题 (19) 构造如下拉格朗日函数：

$$L(\pi,\boldsymbol{\mu},\lambda)=\limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\left[\sum_{i=1}^{N}\sum_{j=1}^{M_i}\mathbb{E}_\pi[w_j^i\Delta_j^i(t)+\lambda c_j^i(t)]+\sum_{i=1}^{N}\mu_i\sum_{j=1}^{M_i}\mathbb{E}_\pi[c_j^i(t)]\right]-\lambda\bar{M}-\sum_{i=1}^{N}\mu_i\bar{M}_i, \tag{20}$$

其中 $\boldsymbol{\mu}=(\mu_1,\mu_2,\cdots,\mu_N)$。现在，(19) 的对偶问题为：

$$\sup_{\lambda\ge0,\boldsymbol{\mu}\ge\mathbf{0}}g(\lambda,\boldsymbol{\mu}), \tag{21}$$

其中 $g(\lambda,\boldsymbol{\mu})$ 是对偶函数，定义如下：

$$g(\lambda,\boldsymbol{\mu})=\inf_{\pi\in\Pi}L(\pi,\lambda,\boldsymbol{\mu}). \tag{22}$$

对于问题 (22)，项 $-\lambda\bar{M}-\sum_{i=1}^{N}\mu_i\bar{M}_i$ 与 π 无关，因此我们在求解 (22) 时将其省略。

注意到，式 (22) 中的问题在网络和用户之间是线性可分的。因此，对于给定的 $\boldsymbol{\mu}$ 和 λ，我们将

式 (22) 中的问题分解为 $\sum_{i=1}^{N} M_i$ 个子问题。与用户 $j \in \mathcal{M}_i$ 对应的子问题为:

$$\inf_{\pi_j^i \in \Pi_j^i} \limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_\pi [w_j^i \Delta_j^i(t) + (\lambda + \mu_i) c_j^i(t)], \quad (23)$$

其中, $\pi_j^i = (c_j^i(t))_{t=1}^{\infty}$, 而 Π_j^i 是用户 $j \in \mathcal{M}_i$的所有因果调度策略的集合。

B. 式（23）的最优解与式（21）的最优解

在本小节中, 我们首先求解式（23）的最优解 $\bar{\pi}_j^i$。注意, 式（23）是一个平均成本马尔可夫决策过程（MDP）, 我们将其记为 ζ_j^i。下面首先介绍该马尔可夫决策过程的组成部分。ζ_j^i 的一个状态表示为$s_j^i = (\Delta_j^i, \bar{\Delta}_j^i)$, 其中 Δ_j^i 表示与用户 $j \in \mathcal{M}_i$对应的作业年龄, $\bar{\Delta}_j^i$ 表示该作业的总处理时间。如果用户 $j \in \mathcal{M}_i$ 的作业缓冲区为空, 则令 $\bar{\Delta}_j^i = \infty$。我们用 $\mathcal{S}_j^i$表示 ζ_j^i 的状态空间,

$$\mathcal{S}_j^i = \{(\Delta_j^i, \bar{\Delta}_j^i) | (\Delta_j^i, \bar{\Delta}_j^i) \in \mathbb{N}^2\} \cup \{(0, \infty)\}. \quad (24)$$

我们用 $c_j^i \in \{0, 1\}$表示 ζ_j^i中的一个动作, 其中 $c_j^i = 1$表示服务器为用户 $j \in \mathcal{M}_i$提供服务, 而 $c_j^i = 0$表示其他情况。如果 $\bar{\Delta}_j^i$为空, 则只有动作$c_j^i = 0$有效。对于状态 s_j^i, 我们用 $\mathcal{A}_j^i(s_j^i)$ 表示有效动作的集合。状态—动作对 (s_j^i, c_j^i) 的成本记为:

$$C_j^i(s_j^i, c_j^i) = w_j^i \Delta_j^i + (\lambda + \mu_i) c_j^i. \quad (25)$$

我们将动作 c_j^i 下从状态 s_j^i 到状态 $\bar{s}_j^i$ 的状态转移概率记为 $P_j^i(s_j^i, \bar{s}_j^i; c_j^i)$。接下来, 我们描述所有可能的非零状态转移概率,

$$P_j^i((0,\infty),(0,\infty);0) = 1 - p_j^i, \quad (26)$$

$$P_j^i((0,\infty),(0,0);0) = p_j^i, \quad (27)$$

$$P_j^i((\Delta_j^i, \bar{\Delta}_j^i),(\Delta_j^i+1, \bar{\Delta}_j^i);0) = 1, \quad (28)$$

$$P_j^i((\Delta_j^i, \bar{\Delta}_j^i),(0,0);1) = \frac{p_j^i \boldsymbol{f}^i(\bar{\Delta}_j^i + 1)}{\sum_{k=1}^{\infty} \boldsymbol{f}^i(\bar{\Delta}_j^i + k)}, \quad (29)$$

$$P_j^i((\Delta_j^i, \bar{\Delta}_j^i),(0,\infty);1) = \frac{(1-p_j^i)\boldsymbol{f}^i(\bar{\Delta}_j^i + 1)}{\sum_{k=1}^{\infty} \boldsymbol{f}^i(\bar{\Delta}_j^i + k)}, \quad (30)$$

$$P_j^i((\Delta_j^i, \bar{\Delta}_j^i),(\Delta_j^i+1, \bar{\Delta}_j^i+1);1) = \frac{\sum_{k=2}^{\infty} \boldsymbol{f}^i(\bar{\Delta}_j^i + k)}{\sum_{k=1}^{\infty} \boldsymbol{f}^i(\bar{\Delta}_j^i + k)}. \quad (31)$$

现在, 我们采用相对值迭代来求解 ζ_j^i的一个最优解。考虑在 n 上的如下迭代,

$$V_j^{i(n)}(s_j^i; \lambda, \boldsymbol{\mu}) = \min_{c_j^i \in \mathcal{A}_j^i(s_j^i)} \Big\{ C_j^i(s_j^i, c_j^i) + \sum_{\bar{s}_j^i \in \mathcal{S}_j^i} P_j^i(s_j^i, \bar{s}_j^i; c_j^i) V_j^{i(n-1)}(\bar{s}_j^i; \lambda, \boldsymbol{\mu}) \Big\} - V_j^{i(n-1)}((0,0); \lambda, \boldsymbol{\mu}), \quad (32)$$

其中, 对所有 $s_j^i \in \mathcal{S}_j^i$, 均有 $V^{i(0)j}(s_j^i; \lambda, \boldsymbol{\mu}) = 0$。与文献 [21], [24], 中计算（32）的方法类似, 我们将 Δ_j^i和 $\bar{\Delta}_j^i$限制为自然数 $Y_{\max}$, 从而限制状态空间 $\mathcal{S}_j^i$。为清晰起见, 我们将这个有界状态空间记为 $\mathcal{S}_j^i(Y_{\max})$。对于这个有界状态空间, 由 [25], 可知

（32）收敛。我们将收敛时得到的相对价值函数记为 $\bar{V}_j^i(s_j^i; \lambda, \boldsymbol{\mu})$。我们定义

$$\bar{V}_j^i(s_j^i; c_j^i, \lambda, \boldsymbol{\mu}) = C_j^i(s_j^i, c_j^i) + \sum_{\bar{s}_j^i \in \mathcal{S}_j^i} P_j^i(s_j^i, \bar{s}_j^i; c_j^i) \bar{V}_j^i(\bar{s}_j^i; \lambda, \boldsymbol{\mu}). \quad (33)$$

对 (32) 右侧关于 $\bar{V}_j^i$ s_j^i λ $\boldsymbol{\mu}$ $s_j^i \in (;$ $\mathcal{S}_j^i$ $Y_{\max}$ $\bar{c}_j^i$ s_j^i λ $\boldsymbol{\mu}$,) 的最小化问题求解, 可得状态 ( ) 的一个最优动作, 记为 (; ,)。接下来, 我们计算,

$$\sum_{i=1}^{N} \sum_{j=1}^{M_i} \limsup_{T\to\infty} \left[\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[\bar{c}_j^i(s_j^i(t); \lambda, \boldsymbol{\mu})] \right], \quad (34)$$

$$\sum_{j=1}^{M_i} \left[\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[\bar{c}_j^i(s_j^i(t); \lambda, \boldsymbol{\mu})] \right], \ \forall i \in \{1, 2, \cdots, N\}. \quad (35)$$

为此, 我们采用以下相对值迭代:

$$\tilde{V}_j^{i(n)}(s_j^i; \lambda, \boldsymbol{\mu}) = \tilde{C}(s_j^i, \bar{c}_j^i(s_j^i; \lambda, \boldsymbol{\mu})) + \sum_{\bar{s}_j^i \in \mathcal{S}_j^i} P_j^i(s_j^i, \bar{s}_j^i; \bar{c}_j^i(s_j^i; \lambda, \boldsymbol{\mu})) \tilde{V}_j^{i(n-1)}(\bar{s}_j^i; \lambda, \boldsymbol{\mu}) - \tilde{V}_j^{i(n-1)}((0,0); \lambda, \boldsymbol{\mu}), \quad (36)$$

初始值为 $\tilde{V}_j^{i(0)}(s_j^i) = 0$, 对所有 $s_j^i \in \mathcal{S}_j^i(Y_{\max})$ 均成立。在 (36) 中, 我们使用了,

$$\tilde{C}(s_j^i, \bar{c}_j^i(s_j^i, \lambda, \boldsymbol{\mu})) = \begin{cases} 1, & \bar{c}_j^i(s_j^i, \lambda, \boldsymbol{\mu}) = 1, \\ 0, & \bar{c}_j^i(s_j^i, \lambda, \boldsymbol{\mu}) = 0. \end{cases} \quad (37)$$

我们将 (36) 收敛时得到的相对价值函数 _ 记为 $\tilde{V}_j^i(s_j^i; \lambda, \boldsymbol{\mu})$。根据 [25], 我们有,

$$\bar{\tilde{V}}_j^i((0,0); \lambda, \boldsymbol{\mu}) = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[\bar{c}_j^i(s_j^i(t); \lambda, \boldsymbol{\mu})]. \quad (38)$$

因此, 我们得到 (34) 和 (35)。

为求得 (21) 中的最优拉格朗日乘子 $(\lambda^*, \boldsymbol{\mu}^*)$, 我们采用次梯度上升算法。$(\lambda, \boldsymbol{\mu})$ 的第 $(k+1)$ 次更新记为 $(\lambda^{(k+1)}, \boldsymbol{\mu}^{(k+1)})$, 其表达式为

$$\lambda^{(k+1)} = \left[\lambda^{(k)} + \tau(k) \left(\sum_{i=1}^{N} \sum_{j=1}^{M_i} \bar{\tilde{V}}_j^i((0,0); \lambda^{(k)}, \boldsymbol{\mu}^{(k)}) - \bar{M} \right) \right]_+, \quad (39)$$

$$\mu_i^{(k+1)} = \left[\mu_i^{(k)} + \bar{\tau}_i(k) \left(\sum_{j=1}^{M_i} \bar{\tilde{V}}_j^i((0,0); \lambda^{(k)}, \boldsymbol{\mu}^{(k)}) - \bar{M}_i \right) \right]_+ \quad \forall i \in \{1, 2, \cdots, N\}, \quad (40)$$

其中 $[x]_+ = \max\{x, 0\}$。我们选择的步长为:

$$\tau(k) = \frac{h}{\sqrt{k}}, \qquad \bar{\tau}_i(k) = \frac{\bar{h}_i}{\sqrt{k}}, \quad (41)$$

其中 h 和 $(\bar{h}_i)_{i=1}^N$ 均为正参数。

C. 净增益最大化抢占式调度策略

在时刻 t，如果用户 $j \in \mathcal{M}_i$对应的作业缓冲区为空，则服务器不会调度该用户。因此，我们只为满足 $\bar{\Delta}_j^i \neq \infty$ 的状态 $s_j^i \in \mathcal{S}_j^i$ 确定调度决策。对于在 V-B 小节中得到的最优拉格朗日乘子 $(\lambda^*, \boldsymbol{\mu}^*)$，以及所有满足 $\bar{\Delta}_j^i \neq \infty$的 $s_j^i \in \mathcal{S}_j^i(Y_{\max})$，我们计算：

$$\beta_j^i(s_j^i) = \bar{V}_j^i(s_j^i; 0, \lambda^*, \boldsymbol{\mu}^*) - \bar{V}_j^i(s_j^i; 1, \lambda^*, \boldsymbol{\mu}^*). \tag{42}$$

对于状态 s_j^i，量 $\beta_j^i(s_j^i)$ 可解释为采用主动动作（$c_j^i = 1$）相对于被动动作（$c_j^i = 0$）所获得的净增益。现在考虑在时刻 t，对于用户 $j \in \mathcal{M}_i$和 $j' \in \mathcal{M}_{i'}$，我们有 $\beta_j^i(s_j^i) > \beta_{j'}^{i'}(s_{j'}^{i'})$；直观来说，中央服务器应优先调度用户$j \in \mathcal{M}_i$，即采用主动动作，而非用户 $j' \in \mathcal{M}_{i'}$。类似地，对于两个用户 $j, j' \in \mathcal{M}_i$，如果 $\beta_j^i(s_j^i) > \beta_{j'}^i(s_{j'}^i)$，那么直观来说，中央服务器也应优先调度用户 j，而非用户 j'。NGM抢占式调度策略将这一直觉形式化，并根据净增益值系统地选择用户。为实现NGM策略，在每个时隙 t，服务器都求解与式（18）相同的优化问题，只需将式（18）中的目标函数替换为 $\sum_{i=1}^N \sum_{j \in \mathcal{I}_i(t)} c_j^i(t)\beta_j^i(s_j^i(t))$。

请注意，在 V-B 小节中得到的最优拉格朗日乘子 $(\lambda^*, \boldsymbol{\mu}^*)$ 是针对状态空间 $\mathcal{S}_j^i(Y_{\max})$ 计算的。我们选择足够大的 $Y_{\max}$，使得在时刻 t，$s_j^i(t) \notin \mathcal{S}_j^i(Y_{\max})$ 的概率很小。然而，我们无法保证 $s_j^i(t)$ 始终保持在 $\mathcal{S}_j^i(Y_{\max})$ 中。如果在时刻 t，$s_j^i(t) \notin \mathcal{S}_j^i(Y_{\max})$，我们就选择足够大的 $Y'\max$，使得 $s_j^i(t) \in \mathcal{S}_j^i(Y'\max)$。使用此前针对 $\mathcal{S}_j^i(Y_{\max})$ 计算出的相同乘子 (λ^*, μ^*)，我们在 $\mathcal{S}_j^i(Y'\max)$ 上运行式 (32) 中的相对值迭代，并评估相应的相对价值函数。利用更新后的价值函数，我们使用式 (42) 计算状态 $s_j^i(t)$ 的净增益，并求解式 (18) 中的优化问题，将目标函数替换为 $\sum_{i=1}^N \sum_{j \in \mathcal{I}_i(t)} c_j^i(t)\beta_j^i(s_j^i(t))$。

VI. 几何作业完成的 Whittle 指数策略

本节中，我们研究在如下假设下针对该问题的 Whittle 指数策略：对于所有 $i \in \{1, 2, \cdots, N\}$，第 i 个网络 Y_i 的作业完成时间服从成功概率为 q_i 的几何分布。我们首先关注式 (23) 中给出的用户 $j \in \mathcal{M}_i$ 的优化问题。V-B 小节中的马尔可夫决策过程表述对于几何服务时间仍然成立。不过，由于几何分布具有无记忆性，当服务器上有一个属于用户 $j \in \mathcal{M}_i$ 的作业时，式 (23) 中马尔可夫决策过程在状态 s_j^i 下的最优动作不再依赖于状态变量 $\bar{\Delta}_j^i$。因此，我们仅将 $\bar{\Delta}_j^i$ 用作指示变量，以表示服务器上是否存在属于用户 $j \in \mathcal{M}_i$ 的作业。若存在作业，则令 $\bar{\Delta}_j^i = 0$；否则，令 $\bar{\Delta}_j^i = \infty$。

与第 V 节类似，我们将 $c_j^i = 1$ 称为主动动作，将 $c_j^i = 0$ 称为被动动作。由式 (23) 可知，对用户 $j \in \mathcal{M}_i$ 执行主动动作会产生调度成本或激活成本 $\lambda + \mu_i$，而被动动作不会产生此类成本。为方便记号，我们用 $\bar{\lambda}_i$表示 $\lambda + \mu_i$。给定 $\bar{\lambda}_i$，我们将所有被动状态（即被动动作为最优选择的状态）构成的集合定义为 $\mathcal{P}_j^i(\bar{\lambda}_i)$。形式化地，我们将其定义为：

$$\mathcal{P}_j^i(\bar{\lambda}_i) = \left\{ s_j^i \in \mathcal{S}_j^i : \bar{\Delta}_j^i = 0,\ \bar{V}_j^i(s_j^i; 0, \bar{\lambda}_i) \leq \bar{V}_j^i(s_j^i; 1, \bar{\lambda}_i) \right\} \cup \left\{ s_j^i \in \mathcal{S}_j^i : \bar{\Delta}_j^i = \infty \right\}, \tag{43}$$

其中 $\bar{V}_j^i(s_j^i;\ c_j^i, \bar{\lambda}_i)$ 与 $\bar{V}_j^i(s_j^i;\ c_j^i, \lambda, \boldsymbol{\mu})$ 相同，后者由在 (33) 中定义，并用 $\bar{\lambda}_i = \lambda + \mu_i$ 求值。若被动状态集合随 $\bar{\lambda}_i$ 单调扩展，则称用户 $j \in \mathcal{M}_i$是可索引的，即对于 $\bar{\lambda}_{i,1} \leq \bar{\lambda}_{i,2}$，有 $\mathcal{P}_j^i(\bar{\lambda}_{i,1}) \subseteq \mathcal{P}_j^i(\bar{\lambda}_{i,2})$ 且 $\mathcal{P}_j^i(\infty) = \mathcal{S}_j^i$。若对每个网络 $i \in \{1, 2, \cdots, N\}$，所有用户 $j \in \mathcal{M}_i$ 都是可索引的，则称 (8) 中的调度问题是可索引的。

在定理1中，我们证明了 (8) 中所定义的调度问题的可索引性。

定理 1. 式 (8) 中的调度问题是可索引的。

现在，我们求出状态 $(\Delta_j^i, 0)$ 的 Whittle 指数。首先，我们给出 Whittle 指数的正式定义。

定义 1. 对于状态 $s_j^i = (\Delta_j^i, 0)$，Whittle 指数为：

$$W_j^i(s_j^i) = \inf\{\bar{\lambda}_i | s_j^i \in \mathcal{P}_j^i(\bar{\lambda}_i)\}. \tag{44}$$

在定理 2 中，我们给出 $\mathcal{S}_j^i$ 中各状态的 Whittle 指数。

定理 2. 对于状态 $s_j^i = (\Delta_j^i, 0)$，Whittle 指数为：

$$W_j^i(s_j^i) = w_j^i \left(\frac{q_i(\Delta_j^i)^2}{2} + \left(1 - \frac{q_i}{2} + \frac{q_i}{p_j^i}\right) \Delta_j^i + \frac{1}{p_j^i} \right). \tag{45}$$

现在，与第 IV 节类似，为了实现 Whittle 指数策略，在每个时隙 t，服务器求解式 (18) 中的优化问题，并将目标函数替换为 $\sum_{i=1}^N \sum_{j \in \mathcal{I}_i(t)} c_j^i(t) W_j^i(s_j^i(t))$.

VII. WIMWF 策略

在第 VI 节中，我们推导出了几何服务时间下的 Whittle 指数策略。然而，对于一般服务时间，我们没有可索引性。受 Whittle 指数策略的启发，我们因此提出一种不需要可索引性的 Whittle 指数类策略。对于固定状态 $s_j^i = (\Delta_j^i, \bar{\Delta}_j^i) \in \mathcal{S}_j^i(Y_{\max})$，其中 $\bar{\Delta}_j^i \neq \infty$，我们定义：

$$\phi(\bar{\lambda}_i) \triangleq \bar{V}_j^i(s_j^i; 0, \bar{\lambda}_i) - \bar{V}_j^i(s_j^i; 1, \bar{\lambda}_i), \tag{46}$$

与第 VI 节类似，为简洁起见，我们将 $\bar{\lambda}_i = \lambda + \mu_i$ 记作。在引理 1 中，我们证明 $\phi(\lambda_i)$ 是关于 λ_i 的连续函数。

引理 1。函数 $\phi(\lambda_i)$ 是关于 $\lambda_i \in (0, \infty)$ 的凹且连续函数。

采用与 [21], [26], 类似的技术，我们可以证明引理 1。我们的目标是找到一个值 $\bar{\lambda}_i^{\text{op}}$，使得我们有$\phi(\bar{\lambda}_i^{\text{op}}) = 0$。对于每个候选值 $\bar{\lambda}_i$，我们通过相对值迭代 [25]来计算 $\phi(\bar{\lambda}_i)$。如果存在这样的根 λ_i^{op}，我们

通过二分搜索找到它，并将其记为 $\bar{W}_j^i(s_j^i)$。注意，对于状态 s_j^i，如果这样的根不存在或我们无法求得它，则改为定义，

$$\bar{W}_j^i(s_j^i) \triangleq \frac{w_j^i(\Delta_j^i+1)\,\boldsymbol{f}^i(\bar{\Delta}_j^i+1)}{\sum_{k=1}^{\infty}\boldsymbol{f}^i(\bar{\Delta}_j^i+k)}, \tag{47}$$

即，我们使用第 IV 节中通过最小化单时隙 Lyapunov 漂移得到的权重。

由于我们未能证明一般服务时间情形下的可索引性，二分搜索的结果可能取决于初始搜索区间的选择，并可能得到不同的 $\bar{W}_j^i(s_j^i)$ 值。为明确起见，我们现在说明本工作中使用的初始搜索区间 $[\bar{\lambda}_i^L, \bar{\lambda}_i^U]$。

我们首先设置 $\bar{\lambda}_i^L = 0$ 并计算 $\phi(0)$。如果 $\phi(0) \le 0$，则直接设置 $\bar{W}_j^i(s_j^i) = 0$。否则，当 $\phi(0) > 0$ 时，我们为 $\bar{\lambda}_i^U$ 选择一个启发式猜测，

$$\bar{\lambda}_i^U = \max\{1,\, w_j^i(\Delta_j^i - 1)\}, \tag{48}$$

并计算 $\phi(\bar{\lambda}_i^U)$。如果 $\phi(\bar{\lambda}_i^U) > 0$，则我们反复将 $\bar{\lambda}_i^U$ 加倍（即 $\bar{\lambda}_i^U \leftarrow 2\bar{\lambda}_i^U$），并重新计算 $\phi(\bar{\lambda}_i^U)$，直至找到一个满足 $\phi(\bar{\lambda}_i^U) \le 0$ 的值 $\bar{\lambda}_i^U$。最后一个满足 $\phi(\bar{\lambda}_i) > 0$ 的值被取作 $\bar{\lambda}_i^L$，由此得到一个满足 $\phi(\bar{\lambda}_i^L) > 0$ 且 $\phi(\bar{\lambda}_i^U) \le 0$ 的区间。根据引理 1，我们认为二分搜索在区间 $[\bar{\lambda}_i^L, \bar{\lambda}_i^U]$ 上收敛，并得到 $\bar{\lambda}_i^{\mathrm{op}} = \bar{W}_j^i(s_j^i)$。请注意，如果我们无法找到满足 $\phi(\bar{\lambda}_i^U) \le 0$ 的 $\bar{\lambda}_i^U$，就无法使用二分搜索；因此，在这种情况下，我们使用 (47) 作为索引。如果在时刻 t，有 $s_j^i(t) \notin \mathcal{S}_j^i(Y_{\max})$，我们也再次使用 (47) 计算索引。

最后，为了实施该策略，在每个时刻 t，我们求解与式 (18) 相同的优化问题，只需将式 (18) 中的目标函数替换为 $\sum_{i=1}^{N}\sum_{j\in\mathcal{I}_i(t)} c_j^i(t)\,\bar{W}_j^i(s_j^i(t))$。

VIII. 数值结果

本节将考察随着用户数量渐近增长，所提出的策略会如何表现。这里“渐近”一词的含义如下：我们从一个包含 $N = 3$ 个网络的基础系统出发，每个网络服务 $M_i = 3$ 个用户。中央服务器总体上每个时隙最多可调度 $M = 2$ 个作业，而每个网络每个时隙最多可服务 $M_i = 2$ 个作业。作业到达概率为，

$$\boldsymbol{p} = \begin{bmatrix} 0.3 & 0.4 & 0.5 \\ 0.4 & 0.2 & 0.6 \\ 0.1 & 0.3 & 0.2 \end{bmatrix}, \tag{49}$$

其中，(i,j) 项对应于 p_j^i。相应的权重为，

$$\boldsymbol{w} = \begin{bmatrix} 2 & 3 & 1 \\ 1 & 2 & 3 \\ 4 & 1 & 2 \end{bmatrix}, \tag{50}$$

其中，(i,j) 项对应于 w_j^i。

对于整数缩放参数 $r \ge 1$，我们通过创建该基础系统的 r 个独立副本来构造一个更大的系统。对于每个原始网络 i 及其三个用户，我们引入 r 个相同的副本，它们具有相同的到达概率、权重、服务时间分布 $\boldsymbol{f}^i$ 以及每网络容量 $\bar{M}_i$。该中央服务器的全局调度容量线性地随 r 扩展，即变为 rM，而每个单独副本的每网络容量仍保持为 M_i。在此构造下，网络数量与用户总数均随 r 线性增长，而每个网络及用户组的局部结构保持不变。这一渐近情形的概念在本质上类似于确立Whittle 指数渐近最优性的论文中所采用的概念；例如，参见 [5]。

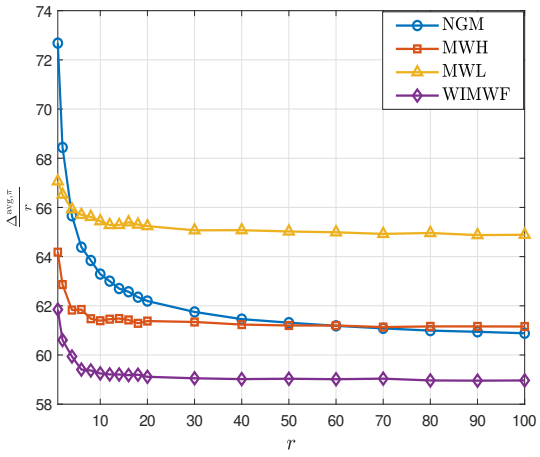

图 2. 一般（非几何）服务时间分布下，归一化加权平均作业年龄随缩放因子 r 的变化。

通过改变 r 并绘制作业归一化长期平均加权年龄，我们可以观察随着用户（及网络）数量增加，各策略 π 的表现。具体而言，我们将 $\Delta^{\mathrm{avg},\pi}$ 除以缩放参数 r，得到作业归一化长期平均加权年龄。

首先，我们考虑一种非几何服务时间分布，

$$\boldsymbol{f} = \begin{bmatrix} 0.1 & 0.2 & 0.1 & 0.1 & 0.4 & 0.1 \\ 0.2 & 0.3 & 0.5 & 0 & 0 & 0 \\ 0.1 & 0.5 & 0.3 & 0.1 & 0 & 0 \end{bmatrix}, \tag{51}$$

其中第 i 行对应于 $\boldsymbol{f}^i$。我们比较四种策略：(i) 第 IV 节提出的基于 Lyapunov 的最大权重（MWL）策略；(ii) 一种启发式最大权重（MWH）策略，采用较简单的权重 $w_j^i\Delta_j^i$；(iii) 第 V 节提出的 NGM策略；以及 (iv) 第 VII 节提出的 WIMWF 策略。对于 NGM策略，我们设定 $Y_{\max} = 10000$，而对于 WIMWF 策略，我们采用 $Y_{\max} = 50$。

在图 2 中可以看到，当 r 的值较小时，MWL 和 MWH 策略的 $\frac{\Delta^{\mathrm{avg},\pi}}{r}$ 低于 NGM 策略，体现了它们在规模相对较小的系统中的良好性能。然而，随着 r 增大，NGM 策略的性能稳步提升，最终超过 MWL 和 MWH。因此，我们可以认为，NGM 策略在渐近意义上优于 MWL 和 MWH 策略。对于 r 的所有取值，WIMWF 策略在四种策略中实现了最低的归一化年龄。最后需要指出，在我们的仿真研究中，无需使用回退权重

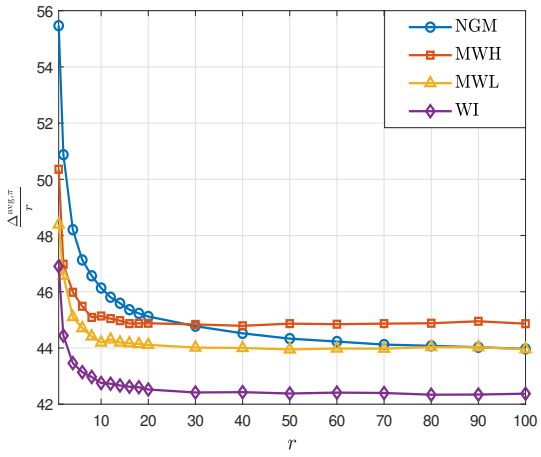

图 3. 几何服务时间下作业的归一化平均加权年龄与缩放因子 r 的关系。

式 (47) 中的回退权重来计算 $\bar{W}_j^i(s_j^i)$：迭代过程始终收敛，并且对于所有 t，我们都有 $s_j^i(t) \in \mathcal{S}_j^i(Y_{\max})$。从图 2 中还可以看到，MWH 的表现优于 MWL。这表明，最小化单时隙 Lyapunov 漂移为构造最大权重策略提供了一种系统化方法；然而，具体取决于系统情况，启发式最大权重策略仍可能优于最小化单时隙漂移的策略。

现在，我们关注几何服务时间，其中对于网络 1、2 和 3，三个基础网络的作业完成概率分别为 0.3、0.5 和 0.7。同样，我们比较四种策略：NGM策略、MWH策略、MWL策略，以及第 VI 节中的 Whittle 指数（WI）策略。对于NGM策略，我们再次设定 $Y_{\max} = 10000$。从图 3 可以看出，当 r 较小时，MWH 和 MWL 策略优于 NGM策略。然而，随着 r 增大，NGM策略逐渐赶上 MWH 和 MWL 策略，并最终超过它们，这再次表明 NGM 渐近优于最大权重策略。结果还表明，对于所有缩放因子 r，Whittle 指数策略始终能获得最低的归一化年龄。

IX. 结论

我们研究了一种可抢占恢复式作业分配系统，其中中央服务器在全局服务器容量约束和各网络计算约束下，将异构用户作业路由至任务专属的机器网络。为刻画作业完成的及时性，我们引入了作业年龄指标，并将问题表述为最小化长期加权平均作业年龄。随后，我们提出并分析了若干调度策略。首先，我们推导了基于李雅普诺夫漂移的最大权重策略。接着，我们将问题表述为不安分多臂赌博机（RMAB），并研究了NGM策略。对于几何作业完成时间，我们证明了可索引性，并推导出闭式Whittle指数，从而得到Whittle指数调度策略。对于无法保证可索引性的一般服务时间分布，我们提出了一种类 Whittle 指数策略 WIMWF，该策略计算主动与被动动作之间的相等点，并在需要时采用基于李雅普诺夫的后备权重。最后，我们通过数值实验评估并比较了所有提出的策略。

参考文献

[1] C. Liu、M. Bennis、M. Debbah 和 H. V. Poor。《面向超可靠低时延边缘计算的动态任务卸载与资源分配》。IEEE 通信汇刊，67(6):4132–4150，2019 年 6 月。[2] Y. Chenghan、Z. Ruiyu、L. Yang 和 J. Ling。《专业化大语言模型综述》。2025 年。在线版见 arXiv:2508.19667。[3] S. Mitrolaris、S. Banerjee 和 S. Ulukus。《稳定队列中的作业完成年龄最小化》。发表于 IEEE ICC，2026 年 5 月。在线版见 arXiv:2511.04630。[4] P. Whittle。《躁动型多臂老虎机：变化世界中的活动分配》。Journal of Applied Probability，25:287–298，1988 年 1 月。[5] I. M. Verloop。《针对可索引和不可索引躁动型多臂老虎机的渐近最优优先级策略》。Annals of Applied Probability，26(4):1947–1995，2016 年 8 月。[6] M. K. C. Shisher、B. Ji、I.-H. Hou 和 Y. Sun。《远程推理系统的学习与通信协同设计：特征长度选择与传输调度》。IEEE Journal on Selected Areas in Information Theory，4:524–538，2023 年 10 月。[7] T. Z. Ornee、M. K. Chowdhury Shisher、C. Kam 和 Y. Sun。《上下文感知状态更新：面向安全关键系统、最大化态势感知的无线调度》。发表于 IEEE Milcom，2023 年 10 月。[8] S. Chakraborty 和 Y. Sun。《接收端具有记忆的及时远程估计》。发表于 Asilomar Conference，2024 年 10 月。[9] S. Chamoun、S. Chakraborty、E. Graves、K. Chan 和 Y. Sun。《用于作业分配的边缘服务器监控》。发表于 IEEE Milcom，2025 年 10 月。[10] S. Banerjee 和 S. Ulukus。《对马尔可夫机器进行跟踪并分配作业》。发表于 IEEE Infocom，2025 年 5 月。[11] S. Liyanaarachchi 和 S. Ulukus。《多台马尔可夫机器的最优监控与作业分配》。发表于 IEEE ISIT，2025 年 6 月。[12] S. Liyanaarachchi 和 S. Ulukus。《估计年龄：何时向马尔可夫机器提交作业以实现收益最大化》。2025 年。在线版见 arXiv:2507.22865。[13] E. B. Sariisik、M. Bastopcu、N. Akar 和 S. Ulukus。《如何提升耗竭型工作节点系统的效率》。发表于 IEEE PIMRC，2025 年 9 月。[14] S. Chamoun、C. McDowell、R. Buchanan、K. Chan、E. Graves 和 Y. Sun。《用于边缘服务器监控的 Mappo》。发表于 IEEE Milcom，2025 年 10 月。[15] Y. Turkmen、B. Buyukates 和 M. Bastopcu。《网络化大语言模型中信息准确性与响应及时性的平衡》。发表于 IEEE Infocom，2026 年 5 月。在线版见 arXiv:2508.02209。[16] D. Jeff 和 E. Modiano。《简单计算卸载系统中的最优服务模式分配》。发表于 Allerton Conference，2025 年 9 月。[17] E. Najm 和 E. Telatar。《多流 M/G/1/1 抢占队列中的状态更新》。发表于 IEEE Infocom，2018 年 4 月。[18] R. D. Yates。《抢占式服务器网络中的信息年龄》。发表于 IEEE Infocom，2018 年 4 月。[19] A. Arafa、R. D. Yates 和 H. V. Poor。《及时云计算：抢占与等待》。发表于 Allerton Conference，2019 年 10 月。[20] B. Wang、S. Feng 和 J. Yang。《何时抢占？链路容量约束下的信息年龄最小化》。Journal of Communications and Networks，21(3):220–232，2019 年 6 月。[21] S. Banerjee 和 S. Ulukus。《状态更新系统中何时抢占？》。发表于 IEEE ISIT，2024 年 6 月。[22] M. Moltafet、H. R. Sadjadpour、Z. Rezki、M. Codreanu 和 R. D. Yates。《具有概率性抢占的多源 M/G/1/1 队列》。发表于 Asilomar Conference，2025 年 10 月。在线版见 arXiv:2512.03241。[23] O. Dogan 和 N. Akar。《具有数据包错误的多源概率性抢占 M/PH/1/1 队列》。IEEE Trans. Commun.，69(11):7297–7308，2021 年 11 月。[24] S. Banerjee、S. Ulukus 和 A. Ephremides。《为保持信息新鲜度，是否重传？》。发表于 IEEE WiOpt，2023 年 8 月。[25] D. Bertsekas。《动态规划与最优控制：第二卷》。Athena Scientific，2012 年。[26] S. Banerjee 和 S. Ulukus。《采用无速率编码的能量采集调度器中的信息年龄最小化》。发表于 IEEE WiOpt，2024 年 10 月。