异步联邦学习中的优化权衡：随机网络方法

AbdelkrimAlahyane[1] EMINES，穆罕默德六世理工大学，摩洛哥本盖里尔 LAAS-CNRS，Université de Toulouse，CNRS，法国图卢兹**Céline Comte** LAAS-CNRS，Université de Toulouse，CNRS，法国图卢兹**Matthieu Jonckheere** LAAS-CNRS，Université de Toulouse，CNRS，法国图卢兹

arXiv:2603.26231v1 [cs.LG] 27 Mar 2026

摘要

同步联邦学习因掉队者效应而难以扩展。异步算法通过在更新到达时立即处理来提高更新吞吐量，但也带来两项根本挑战：梯度陈旧性会降低收敛速度；在异构数据分布下，算法会偏向较快客户端。尽管 `AsyncSGD` 和 `GeneralizedAsyncSGD` 等算法通过客户端任务队列缓解了这种偏差，但大多数现有分析都忽略了底层排队动态，也未能以闭式形式刻画更新吞吐量和梯度陈旧度。

为弥补这一差距，我们为 `Generalized AsyncSGD`构建了一个随机排队网络框架，联合建模客户端和中央服务器的随机计算时间，以及随机的上行链路和下行链路通信延迟。借助乘积形式网络理论，我们推导出更新吞吐量的闭式表达式，并给出了通信轮复杂度和达到ϵ-驻点所需期望实际经过时间的闭式上界。这些结果从理论上刻画了梯度陈旧性与实际收敛速度之间的权衡。我们进一步扩展该框架，以量化随机时序下的能耗，揭示了收敛速度与能效之间的另一种权衡。

基于这些分析结果，我们提出了基于梯度的优化策略，以联合优化路由和并发度。在 EMNIST 上进行的实验表明，与 `AsyncSGD`相比，收敛时间减少了 29%–46%，能耗降低了 36%–49%。

2012 年 ACM 学科分类 计算方法 → 机器学习；计算数学 → 随机过程

关键词与短语 异步联邦学习、排队论、Jackson 网络、乘积形式、相对延迟

1 引言

现代机器学习训练依赖随机梯度方法 [50] 及其变体，例如 Adam [28]。为了将这些方法扩展到大型模型和数据集，训练通常分布在多个客户端上，由它们并行计算梯度，同时由中央服务器（CS）聚合更新；这种范式称为联邦学习（FL）。尽管采用了梯度压缩 [3, 57], 去中心化通信 [35], 和本地更新 [45] 等能够降低通信开销的高效通信技术，联邦学习系统通常仍采用同步协议，即中央服务器（CS）会等待收到所有请求的梯度后才更新模型。不幸的是，在异构环境中，同步联邦学习系统会因掉队者效应而减慢。

[1] 通讯作者

异步训练 [60, 10, 61] 通过允许中央服务器（CS）在更新到达时立即处理，从而缓解了这一限制，并提高资源利用率。然而，在异构数据环境中直接引入异步性，会导致偏向较快客户端，也就是更频繁地贡献更新的客户端。为缓解这种偏差，近期研究采用客户端侧队列 和量身定制的采样方案。具体而言， [29]提出了采用均匀采样的 AsyncSGD ，而 [31] 则将其扩展为采用非均匀路由概率的 Generalized AsyncSGD 。然而，这些研究大多忽略了底层的排队动态，要么没有进行明确分析 [29] ，要么依赖渐近近似 [31]。更重要的是，尽管实际运行时间才是采用异步方法的主要动机，它们的收敛分析却以通信轮次而非实际运行时间为衡量标准。

近期有若干研究 [58, 56, 43, 44] 尝试通过推导基于时间的收敛保证来弥补这一不足。然而，这些方法存在若干局限：它们通过时间阈值强制实现部分同步，使学习过程偏向快速客户端而不利于慢速客户端，丢弃即将完成的计算结果，并且通常假设计算时间具有确定性，因而限制了其在现实边缘系统中的适用性。更多详情请见第 1.2 节。

本文在我们早期的会议论文 [2]基础上进行了大幅扩展。在先前的工作中，我们采用随机排队模型，从实际运行时间的角度分析异步联邦学习，揭示了更新陈旧度与更新频率之间的根本权衡。然而，初始模型忽略了通信延迟，也未将并发度视为可控的系统参数，尽管这两者都是决定实际性能的关键因素。

在这一扩展版本中，我们对通信阶段进行显式建模，并将并发度引入为优化变量。我们还通过纳入中央服务器处理速度并分析系统的能耗，进一步丰富了理论框架，从而揭示了收敛速度与能效之间的另一种权衡。这一扩展使我们能够以更贴近现实且更具实际意义的方式刻画异步联邦学习。

1.1 贡献

在 [29, 31, 2], 的基础上，我们为异步联邦学习构建了一个全面的随机框架，明确地对计算、通信和中心服务器侧动态进行建模。我们的主要贡献如下：

- **广义随机网络模型**。我们为 Generalized AsyncSGD （扩展了 AsyncSGD）提出统一的随机排队网络建模框架，可联合描述随机计算时间、上行链路和下行链路通信延迟、CS 处理速度、路由和并发度。该框架扩展了以往的随机模型 [31, 2], 明确纳入了通信和 CS 侧的影响。**延迟与吞吐量的闭式刻画**。借助积形式随机网络，我们推导出平均相对延迟、更新频率及其相对于路由概率的梯度的闭式表达式，从而实现精确的性能分析和基于梯度的优化。**收敛分析**。我们针对 Generalized AsyncSGD, 分别以通信轮次和实际运行时间为衡量标准，建立收敛保证，并明确量化更新频率（速度）与梯度陈旧性（误差）之间的权衡。**能耗感知建模与分析**。我们首次对 Generalized AsyncSGD进行了全面的能耗分析，考察异构数据、硬件和网络条件，并纳入随机时序以及阶段相关的能耗成本（计算、上行传输和下行接收）。

- **路由与并发度的联合优化**。我们构建并求解多目标优化问题，联合优化路由概率和并发度，从而能够以有原则的方式权衡准确率、实际运行时间和能耗。

1.2 相关工作

1.2.1 异步联邦学习

早期联邦学习研究聚焦于同步方法 [59, 49, 40, 42, 57], ，但这类方法在异构环境下会受到掉队客户端和可扩展性差的困扰 [60]。这些局限促使人们引入异步联邦学习算法 [60, 10, 61], ；在简化假设下，其在实际运行时间方面的优势已得到形式化证明 [13, 29]。

异步随机梯度下降的经典分析 [1, 9, 18, 5, 53, 34, 54, 48] 通常以最坏情况（最大）延迟上界为前提，因此过于悲观，且对异常值高度敏感。此外，这些研究往往采用简化的计算模型，掩盖了边缘计算环境中固有的可变性和随机性。例如，虽然 `FedBuff` [48] 引入了客户端侧缓冲以稳定收敛，但其理论保证仍受最大延迟的限制。此外，其分析假设任务完成顺序是均匀随机的；在客户端服务速度各异的真实环境中，这一假设并不成立。

为缓解这种悲观估计，后续研究转向基于平均延迟的分析 [13, 6]。 [13] 通过有选择地丢弃陈旧梯度，获得了更紧的保证，但代价是通信开销翻倍，而且仅适用于同质数据环境。类似地， [6] 推导出了依赖于平均延迟的界，但其分析依赖于方差项，而在最坏情况下，这些方差项可能随最大延迟增长；此外，该方法同样假设数据同质。另一种方法是延迟自适应学习率方法 [46]，它成功消除了对最大延迟的依赖，但在异构数据环境中无法收敛到精确最优解。这些方法根据陈旧度调整步长，使较快客户端获得不成比例的偏重，实际上扭曲了全局目标函数，并收敛到有偏解。

为解决异构数据环境中的偏差问题， [29] 提出了 `AsyncSGD`，通过客户端侧队列和均匀采样确保更新无偏，且其收敛保证取决于平均延迟。 [24] 对这一方法进行了推广，并提出了通过随机打乱实现的改进。然而，这两种分析都假设延迟有界，因此对于具有（更符合实际的）无界排队动态的系统而言，相关性较低。

`Generalized AsyncSGD` [31] 进一步推广了这一框架，在允许非均匀路由的同时，通过适当缩放步长来保持无偏性。 [31] 推导了在处理时间无界时依赖平均延迟的收敛保证；然而，该分析依赖渐近近似，既未提供延迟的闭式表达式，也未就如何选择路由概率和并发度给出指导。

在这一研究脉络的基础上，我们的初步研究 [2] 利用排队论推导出平均延迟的显式闭式表达式，并提出了一种针对路由概率的基于梯度的优化方法。

1.2.2 实际运行时间分析

一些关于异步联邦学习的分析会显式地以时间 [58, 56, 43, 44], 的形式推导收敛保证，但它们通常会采用部分同步机制（例如基于截止时间的更新），这使其又回到了同步联邦学习的框架中。

此外，这些方法往往会使学习偏向快速客户端，丢弃接近完成的计算，并依赖确定性或简化的计算时间模型。

相比之下，第 1.2.1 节引用的大多数异步联邦学习分析都依赖基于轮次的收敛，因此掩盖了真实的实际时间性能。诸如 [17] 和我们的初步工作 [2]等研究强调了一项根本性的权衡：仅针对每轮误差进行优化，可能会显著延长实际收敛时间，而这正是引入异步联邦学习的主要动机。例如， [17] 分析了 `FedBuff` 算法，并表明存在一个最优缓冲区大小，可以平衡这一权衡；然而，该分析假设数据分布和处理速度同质。在 [2]中，我们基于 [31] 的随机框架进一步研究，并证明在 `Generalized AsyncSGD`下，最小化每轮误差会通过使系统的运行节奏与最慢的客户端相匹配来降低陈旧度，但代价是并行性利用不足导致吞吐量下降。

1.2.3 能耗分析

节能联邦学习的研究主要集中于同步场景 [37, 66, 63]。在异步场景中，能量优化通常通过客户端选择策略 [52, 22] 实现，并受到启发式公平性度量的约束。例如， [62] 规定了每个客户端的最低选择概率，而 [11] 则要求最低数据贡献数量。然而，这类约束往往缺乏明确依据，且难以调整。此外，这些方法通常依赖确定性时序模型，无法捕捉真实边缘系统中的随机动态。

1.2.4 既有工作的总结与局限

尽管我们在 [2] 中提出的基础性表述推导出了延迟的闭式表达式，并揭示了陈旧度与吞吐量之间的权衡，但其适用范围仍然有限。具体而言，该表述忽略了通信阶段，假设中央服务器处理是瞬时完成的，并且未将并发度视为可控的优化参数。更广泛地说，现有的异步联邦学习文献尚未提供一个统一的随机框架，能够在现实的系统异质性条件下，同时考虑时序动态和能耗。

为克服这些局限，本文扩展稿提出了一个综合框架，对随机计算、通信延迟、中央服务器处理时间和能耗成本进行联合建模。通过将路由和并发度明确表述为优化变量，并采用基于梯度的方法求解由此得到的问题，我们建立了一种有原则的方法，可在异构边缘环境中权衡实际收敛速度、统计准确性和能效等多个目标。

1.3 符号说明

$\mathbb{Z}$、$\mathbb{N}$、$\mathbb{N}_{>0}$、$\mathbb{R}$、$\mathbb{R}_{\geq 0}$ 和 $\mathbb{R}_{>0}$ 分别表示整数集、非负整数集、正整数集、实数集、非负实数集和正实数集。令 $|\cdot|$ 表示 ℓ_1-范数，令 $\mathbb{1}[\cdot]$ 表示示性函数。对于每个 n、$m \in \mathbb{N}_{>0}$，定义 $\mathbf{X}_{n,m} = \{x \in \mathbb{N}^n : |x| = m\}$ 为由 n 维非负整数项向量组成的集合，其 ℓ_1-范数等于 m。对于每个 $n \in \mathbb{N}_{>0}$，令 $\mathbf{P}_n = \{p \in \mathbb{R}^n : 0 < p_i < 1$ 对于 $i \in \{1, 2, \ldots, n\}$ 且 $|p| = 1\}$。

2 系统模型与学习机制

首先，我们描述异步联邦学习的系统模型。

2.1 问题设置

考虑一个由中央服务器和 n 个客户端组成的异步联邦学习系统，这些客户端的索引为 $\{1, 2, \ldots, n\}$。通过求解以下优化问题，协同训练一个全局模型

$$\min_{w\in\mathbb{R}^d} f(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w),$$

其中，$f_i : w \mapsto \mathbb{E}_{(x,y)\sim\mathcal{D}_i}\left[\ell_i(\mathrm{NN}(x, w), y)\right]$ 是客户端 i 的目标函数。这里，$w \in \mathbb{R}^d$是全局模型的参数向量（例如深度神经网络），d 是可训练参数的数量，$\mathrm{NN}(x, w)$ 是模型的预测函数，ℓ_i 是客户端 i的局部损失函数，而$\mathcal{D}_i$ 是其局部数据分布。每个客户端 i 使用确定性映射 $g_i(w, \zeta_i)$ 近似其梯度，其中随机性完全源于对$\zeta_i \sim \mathcal{D}_i$的采样。这一计算称为任务。

2.2 算法

本文聚焦于 `Generalized AsyncSGD` [31]，这是对 `AsyncSGD` [29, 算法 2] 的扩展，通过引入路由机制来应对异构资源和数据分布。`Generalized AsyncSGD` 过程详见算法 1（中央服务器）和算法 2（客户端 i）。

我们首先从中央服务器的角度进行考察（算法 1）。中央服务器初始化全局模型参数 w_0，并派发一批 m 个初始任务（即梯度评估请求）。初始化时，对这 m 个任务中的每一个，都从总体规模中均匀随机地选择一个接收任务的客户端。正如我们稍后将要展示的，系统的稳态动态对这一初始配置具有鲁棒性。请注意，由于 m 可能大于 n，或者仅仅是因为独立随机选择，同一个客户端可能会收到多个并发任务。收到任务后，客户端立即开始计算梯度。

优化过程以离散的轮次推进，轮次索引为 $k \in \{0, 1, \ldots\ldots, K\}$。我们将第 k 轮定义为全局参数 w 的第 k 次与第 $(k+1)$ 次更新事件之间的时间间隔。流程如下（第 4 行）：当客户端 C_k 完成一项任务并返回梯度估计 $g_{C_k}(w_{I_k}, \zeta_{C_k})$（第 5 行；其中 I_k表示用于计算的模型的轮次索引）时，中央服务器立即更新全局参数（第 6 行）。更新后，中央服务器根据概率 $\mathbb{P}(A_{k+1} = i) = p_i$，独立选择客户端 A_{k+1}，并向其派发新模型 w_{k+1}（第 7–8 行）。

转向客户端视角（算法 2），传入的任务按先进先出（FIFO）方式处理。如果客户端从中央服务器收到模型时处于空闲状态，便立即开始计算；否则，任务会在本地排队，直到客户端可用。任务完成后，结果会传回中央服务器。

最后请注意，更新规则中的步长（算法 1 第 6 行）乘以路由概率 $p_{C_k}^{-1}$ 的倒数，以纠正非均匀路由引入的偏差。

正如我们将在以下各节中展示的，算法的性能由两个关键的系统超参数决定：**(i) 路由概率** 向量 p，它控制计算负载的分布；以及 **(ii) 并发度**，其定义为系统中持续流转的任务数量（无论是在队列中还是正在计算）。

当 p 为均匀分布（即 $p_i = 1/n$）且并发度与网络规模相匹配（$m = n$）时，我们可恢复标准的 `AsyncSGD` 基线 [29, 算法 2]。

算法 1 Generalized AsyncSGD (CS)

1：输入：轮次、客户端和任务的数量 K、 n **和** m **；路由** p**；学习率** η **2：随机初始化参数** w_0 **3：向选定的客户端派发** m **个** w_0 **实例 4：对于** $k = 0$**，.....，** K**，执行 5：中央服务器从客户端** C_k **接收随机梯度** g_{C_k} (w_{I_k}, ζ_{C_k}) **6：更新** $w_{k+1} \leftarrow w_k - \frac{\eta}{np_{C_k}} g_{C_k}$ (w_{I_k}, ζ_{C_k}) **7：根据** p **采样一个新的客户端** A_{k+1} **8：向客户端** A_{k+1} **发送模型参数** w_{k+1} **9：结束循环**

算法 2 Generalized AsyncSGD (Client i)

1: **输入：** 已接收参数队列、本地数据集2: **如果**队列非空，**则**3: 使用先进先出策略从队列中取出已接收的参数 w 4: 从本地数据集中采样一个小批量 ζ_i $\mathcal{D}_i$5: 计算梯度估计 $g_i(w, \zeta_i)$6: 将梯度发送至中央服务器7: 重复 8: **结束如果**

2.3 通过排队确保公平性

Generalized AsyncSGD 中的排队机制对于确保所有客户端持续参与至关重要，无论其处理速度如何。在异质（非独立同分布）环境中，这种包容性至关重要，可防止模型偏向较快客户端的数据分布。

例如，考虑一个包含极慢客户端的系统。在标准异步方法中，为避免排队，任务只分配给空闲客户端；例如 [46],，其中 $m = n$，且每次完成的更新都会立即发送回发起任务的客户端。在这种情况下，掉队者可能只处理一个任务，而较快客户端却完成数千次更新。这种行为会使模型偏向较快客户端的数据分布。相比之下， AsyncSGD 和 Generalized AsyncSGD 会以概率方式分配任务，不论客户端的状态如何。因此，如果某个客户端极慢，任务可能会在其本地队列中不断积累，甚至可能达到该客户端持有所有 m 流转任务的程度。这实际上会限制系统的运行速度，迫使算法等待掉队者提交更新。尽管这会降低吞吐量，却能确保纳入慢速客户端的数据，从而保持所学习模型的统计无偏性。

2.4 相对延迟

异步联邦学习的一个主要缺点是，基于过时的全局模型计算更新会导致模型参数陈旧。我们使用相对延迟来量化这一影响；相对延迟是异步随机梯度下降收敛分析中的一个核心量。在本研究中，我们推导了主要性能指标，即达到 ϵ-近似驻点（即平均平方梯度范数不超过 ϵ）所需的通信轮次、实际运行时间和能量，并表明这些指标显式依赖于平均相对延迟。因此，刻画这种延迟对于评估和优化

系统性能至关重要。

对于每个轮次 $k \in \{1, \ldots, K\}$ 和客户端 $i \in \{1, \ldots, n\}$，相对延迟 $D_{i,k}$ 定义为中央服务器在以下两个时刻之间执行的模型更新次数：(i) 某项任务分配给第 k 轮的客户端 i 时；(ii) 中央服务器应用由此产生的梯度时。例如，如果客户端 i 接收到模型参数 w_k，并且在其计算更新期间恰好应用了来自另一个客户端的一个梯度（因此模型推进至 w_{k+1}），那么$D_{i,k} = 1$。如果第 k 轮没有任务分配给客户端 i（即 $A_k \neq i$），则设 $D_{i,k} = 0$。

2.5 数据模型

与非凸设置下去中心化学习的文献一致 [48, 46, 29, 24]„ 我们作如下假设：

A1 下界有界性：存在一个标量 $f^* \in \mathbb{R}$，使得对所有 $w \in \mathbb{R}^d$，$f(w) \geq f^*$。**A2 L-光滑性：**每个 f_i 均连续可微且 L-光滑。具体而言，存在一个常数 $L > 0$，使得对所有 $i \in \{1, \ldots, n\}$ 以及所有 u，$v \in \mathbb{R}^d$：

$$\|\nabla f_i(u) - \nabla f_i(v)\| \leq L\|u - v\|.$$

A3 无偏随机梯度与有界方差：存在一个常数$\sigma \geq 0$，使得对所有 $i \in \{1, \ldots, n\}$ 以及 $w \in \mathbb{R}^d$：

$$\mathbb{E}_{\zeta_i \sim \mathcal{D}_i}\left[g_i(w, \zeta_i)\right] = \nabla f_i(w),$$
$$\mathbb{E}_{\zeta_i \sim \mathcal{D}_i}\left[\|g_i(w, \zeta_i) - \nabla f_i(w)\|^2\right] \leq \sigma^2.$$

A4 有界梯度差异：存在一个常数 $M \geq 0$，使得对所有客户端 $i \in \{1, \ldots, n\}$ 以及参数 $w \in \mathbb{R}^d$，$\|\nabla f_i(w) - \nabla f(w)\|^2 \leq M^2$。
A5 有界梯度：存在一个常数 $G \geq 0$，使得对所有 $w \in \mathbb{R}^d$ 以及$i \in \{1, \ldots, n\}$，$\|\nabla f_i(w)\| \leq G$。
我们的分析要求梯度有界 **A5**，但与 [46] 相反，随机梯度可能无界。在实践中，常用于拜占庭鲁棒性的梯度裁剪可确保更新范数有界，并限制常数 G [24]。

2.6 计算与通信模型

我们将 [2, 31, 29] 的框架加以推广，明确考虑通信延迟，从而得到图1所示的完整排队网络。在现实的边缘环境中，确定性模型无法捕捉系统资源的随机特性：计算时间会因热节流、动态电压与频率调节（DVFS）而波动，而通信链路则会受到衰落、拥塞和干扰的影响。为了对路由和并发度进行严格分析，既反映这种固有的变化性，又保持数学上的可处理性，我们将这些组成部分建模为随机过程。在此框架下，我们推导出封闭形式的性能指标，从而严格刻画这些系统动态。

沿用先前的工作 [47, 32, 16, 17, 23, 31, 2], ，我们假设客户端 i 执行连续任务（即随机梯度评估）的计算时间相互独立且服从参数为 $\mu_i^{\mathrm{c}} > 0$的指数分布。对于通信，我们假设上行传输时间（客户端 i 向中央服务器发送梯度）和下行传输时间（客户端 i 从中央服务器接收参数）分别相互独立且服从参数为 $\mu_i^{\mathrm{u}} > 0$ 和 $\mu_i^{\mathrm{d}} > 0$的指数分布。

尽管理论分析依赖于计算时间和通信时间服从指数分布这一假设，但我们在补充材料的第5.3节和附录H中通过数值验证发现，其他分布实际上也会带来类似的性能。

从网络的角度来看，这种建模方式有效地将系统映射为一个随机排队网络：每个客户端的计算由单服务器先进先出队列表示，而上行链路和下行链路方向的通信延迟则建模为无限服务器队列（IS）。最初，我们忽略中央服务器（CS）整合更新所需的处理时间，因为其计算资源通常远超客户端；不过，第7节将讨论放宽这一假设后如何扩展我们的分析。

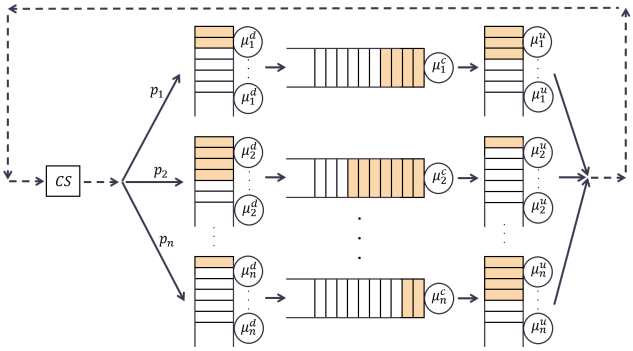


图1 包含**计算**时间和**通信**延迟的排队模型。

3 平稳性分析

如第4、5和6节所述，平均相对延迟是决定所有后续性能指标的关键量，包括轮次复杂度、实际收敛时间和能耗。因此，本节着重分析系统的稳态行为，以明确刻画相对延迟。

我们首先在第3.1节将系统动态建模为一个随机排队网络。随后，第3.2节推导该模型的平稳分布。在此基础上，第3.3节给出平均相对延迟及其梯度的闭式表达式，后者是后文提出的基于梯度的优化策略所必需的。

3.1 随机排队网络

对于每个 $i \in \{1, \ldots, n\}$，中央服务器到客户端 i 的下行链路通信建模为一个无限服务器队列 d_i，客户端 i 的梯度计算建模为一个单服务器队列 c_i，而客户端 i 返回中央服务器的上行链路通信建模为一个无限服务器队列 u_i。

在任意时间 $t \geq 0$，网络状态由以下右连续（右连续且有左极限）随机向量表示：

$$\xi(t) = \left(\xi_i^{\mathrm{d}}(t),\, \xi_i^{\mathrm{c}}(t),\, \xi_i^{\mathrm{u}}(t),\, i \in \{1, \ldots, n\}\right) \in \mathcal{X}_{3n,m},$$

其中，每个分量记录时间 t 对应服务器上存在的任务数量。回顾一下，$\mathcal{X}_{3n,m}$ 是由总和为 m 的非负整数项 $3n$ 维向量构成的集合。

由于相对延迟是在参数更新时刻测量的，因此对于每个$k \in \{0, \ldots, K\}$，我们将嵌入状态向量 X_k 定义为在参数更新后、

派发下一个任务之前立即观测到的系统状态。因此， X_k 属于约化状态空间$\mathcal{X}_{3n,\ m-1}$其定义如下：对于每个 $i \in \{1, \dots, n\}$，

$$\begin{cases} X^{\mathrm{d}}_{i,k} = \xi^{\mathrm{d}}_i(T_k) - \mathbf{1}\{A_k = i\}, \\ X^{\mathrm{c}}_{i,k} = \xi^{\mathrm{c}}_i(T_k), \\ X^{\mathrm{u}}_{i,k} = \xi^{\mathrm{u}}_i(T_k). \end{cases} \tag{1}$$

这里，$\{T_k\}_{k=0}^{K}$ 表示上行服务器$\{\mathrm{u}_i\}_{i=1}^{n}$的服务完成时间序列，其中 $T_0 = 0$。每个时刻 T_k 标志着第 k轮次的开始，该轮次持续 $T_{k+1} - T_k$。此外， A_k 表示在第 k轮次开始时被选中接收新任务的客户端索引，其中 $\mathbb{P}(A_k = i) = p_i$；而 C_k 表示其上行传输在时刻 T_{k+1} 完成、从而标志着第 k轮次结束的客户端。

在整个分析过程中，我们分别用 $\mathbb{P}$ 和 $\mathbb{E}$ 表示系统的平稳概率测度和期望。为分析参数更新间隔时间，我们还引入与点过程 $\{T_k\}_{k\in\mathbb{N}}$相关的 Palm 概率测度 $\mathbb{P}^0$ 及其期望 $\mathbb{E}^0$。在 $\mathbb{P}^0$下，时间原点 $t = 0$ 被条件化为上行服务器集合$\{\mathrm{u}_i\}_{i=1}^{n}$中的一次服务完成时刻。

相对延迟

对于每个 $k \in \{0, \dots, K\}$，$D_{i,\ k}$定义为：在分配给客户端 i 的任务于第 k轮中的逗留期间，所有上行链路节点 $\{u_j\}_{j=1}^{n}$ 发生的服务完成次数。该逗留涵盖整个周期：向服务器 d_i 派发、客户端 i上的本地处理，以及通过服务器 u_i进行传输。如果第 k 轮未向客户端 i 分配任务（$A_k \neq i$），则定义$D_{i,\ k} = 0$。

在本文的其余部分，我们假设系统处于稳态运行。对于足够大的 K，由于 $\xi(t)$ 的分布以指数速度快速收敛到其平稳分布 [36]，这一假设是合理的。因此，我们可以省略时间索引，例如，对于 $\mathbb{E}^0[D_{i,k}]$，可写作 $\mathbb{E}^0[D_i]$ 。

3.2 平稳分布

在第 2.6 节的假设下，我们的系统演化为一个随机**闭式 Jackson 网络** [25, 21]。以下命题刻画了其平稳行为，为后续的延迟分析奠定基础。

▶**命题 1.** 在第 *2.6* 节的设定下，过程 $(\xi(t))_{t\geq 0}$ 和 $(X_k)_{k\in\mathbb{N}}$ 是不可约、正常返的马尔可夫链，且分别具有唯一的平稳分布 $\pi_{n,m}$ 和$\pi_{n,m-1}$。对于任意 $\mathfrak{m} \in \mathbb{N}_{>0}$，分布 $\pi_{n,\mathfrak{m}}$ 具有如下乘积形式：对于任意状态 $x \in \mathcal{X}_{3n,\mathfrak{m}}$

$$\pi_{n,\mathfrak{m}}(x) = \frac{1}{Z_{n,\mathfrak{m}}} \prod_{i=1}^{n} \left(\frac{p_i}{\mu_i^{\mathrm{c}}}\right)^{x_i^{\mathrm{c}}} \frac{1}{x_i^{\mathrm{d}}!} \left(\frac{p_i}{\mu_i^{\mathrm{d}}}\right)^{x_i^{\mathrm{d}}} \frac{1}{x_i^{\mathrm{u}}!} \left(\frac{p_i}{\mu_i^{\mathrm{u}}}\right)^{x_i^{\mathrm{u}}}, \tag{2}$$

其中 $Z_{n,\mathfrak{m}}$ 是归一化常数。

证明. 参见补充材料的附录 C。 ◀

下一节推导的相对时延表达式依赖于平稳分布 $\pi_{n,\ \mathfrak{m}}$ 主要通过相关的归一化常数 $Z_{n,\ \mathfrak{m}}$ 体现。直接

计算这些常数对于大型 n 和 m 而言由于组合增长而不可行。为解决这一问题，我们在补充材料的命题 15 中调整了 *Buzen* 的递归算法 [8], ，使其能够在 $\mathcal{O}(nm^2)$ 时间和$\mathcal{O}(m)$ 内存中完成计算，其中n 表示客户端数量， m 表示并发级别（即任务数量）。这种计算效率至关重要，因为它能够精确计算下一节所介绍的期望相对延迟及其梯度。

3.3 延迟与梯度计算

本节推导平均相对延迟及其相对于路由向量 p 的梯度的闭式表达式，并明确刻画它们对 p、系统并发度 m 以及异构服务率 $\mu = \left(\mu_i^{\mathrm{d}},\ \mu_i^{\mathrm{c}},\ \mu_i^{\mathrm{u}}\right)_{i=1}^{n}$ 的依赖关系。这些结果有助于进行性能敏感性分析，并促进基于梯度的优化。关键的是，它们使我们能够高效地确定最优路由策略和最优并发度 m ，从而最小化任何定义为平均相对延迟的泛函的性能准则。

▶ **定理2.** 在第 *2.6* 节的模型中，对于每个 $i, j \in \{1, \ldots, n\}$，以下恒等式成立：

$$\mathbb{E}^0[D_i] = \sum_{s\in\{\mathrm{c,d,u}\}} \mathbb{E}[X_i^s]\,, \tag{3}$$

$$\frac{\partial}{\partial p_j}\,\mathbb{E}^0[D_i] = \sum_{s\in\{\mathrm{c,d,u}\}} \sum_{r\in\{\mathrm{c,d,u}\}} \frac{1}{p_j}\,\mathrm{Cov}\left(X_i^s,\ X_j^r\right). \tag{4}$$

此外，以下闭式表达式成立：

$$\sum_{s\in\{\mathrm{c,d,u}\}} \mathbb{E}[X_i^s] = \beta_{i,1} + \gamma_i \frac{Z_{n,m-2}}{Z_{n,m-1}}, \tag{5}$$

$$\sum_{s,r\in\{\mathrm{c,d,u}\}} \mathbb{E}[X_i^s X_j^r] = \alpha_{i,j} + \beta_{i,2}\gamma_j + \beta_{j,2}\gamma_i + \psi_{i,j}, \tag{6}$$

其中，系数定义如下：

$$\gamma_i = p_i\left(\frac{1}{\mu_i^{\mathrm{d}}} + \frac{1}{\mu_i^{\mathrm{u}}}\right), \qquad \beta_{i,\ell} = \sum_{k=1}^{m-\ell}\left(\frac{p_i}{\mu_i^{\mathrm{c}}}\right)^k \frac{Z_{n,m-\ell-k}}{Z_{n,m-1}},$$

$$\alpha_{i,j} = \begin{cases} \displaystyle\sum_{k=1}^{m-1}(2k-1)\left(\frac{p_i}{\mu_i^{\mathrm{c}}}\right)^k \frac{Z_{n,m-1-k}}{Z_{n,m-1}} & \text{if } i = j, \\ \displaystyle\sum_{\substack{k,\ell=1 \\ k+\ell\le m-1}}^{m-2} \left(\frac{p_i}{\mu_i^{\mathrm{c}}}\right)^k \left(\frac{p_j}{\mu_j^{\mathrm{c}}}\right)^{\ell} \frac{Z_{n,m-1-k-\ell}}{Z_{n,m-1}} & \text{if } i \neq j, \end{cases}$$

$$\psi_{i,j} = \frac{\gamma_i}{Z_{n,m-1}}\left(\gamma_j Z_{n,m-3} + \mathbf{1}_{\{i=j\}} Z_{n,m-2}\right).$$

而常数 $Z_{n,\ \mathfrak{m}}$ 对于 $\mathfrak{m} \in \{0、1、\ldots, m-1\}$，则使用补充材料附录 *D* 中命题 *15* 的递推关系计算得出。

证明。详细证明见补充材料的第 E 节。直观而言，公式（3）体现了利特尔定律 [26] 的类比：它将分配时定义的预期总延迟成本（$\mathbb{E}^0[D_i]$）等同于服务过程中逐步累积的成本（$\sum_{s\in\{\mathrm{c,d,u}\}} \mathbb{E}[X_i^s]$）。◀

直接计算 $\mathbb{E}^0[D_i]$ 并不可行，因为任务的相对延迟可能取决于数量任意多的未来轮次。通过利用递归常数$Z_{n,k}$ （可在$\mathcal{O}(nm^2)$时间和$\mathcal{O}(m)$内存下计算）来表示延迟及其梯度，定理2保证即使对于大型网络，延迟计算仍然高效且可扩展。

分析

公式（3）意味着以下简单恒等式

$$\sum_{i=1}^{n} \mathbb{E}^0[D_i] = \sum_{i=1}^{n} \sum_{s\in\{\mathrm{c,d,u}\}} \mathbb{E}[X_i^s] = m - 1. \tag{7}$$

这带来两个直接却有悖直觉的结论：（i）总平均相对延迟仅取决于 n 和 m，而 p 和 μ 只决定该延迟在各客户端之间的分配方式；（ii）因此，在 n 和 m固定的情况下，降低某个客户端 i 的延迟（例如，通过减小 p_i 或增大 μ_i^{c}、μ_i^{d}或 μ_i^{u}）必然会以至少一个其他客户端延迟增加为代价。

此外，在路由和服务器速度固定的情况下，将（3）与以下事实结合起来：对于每个 $s \in \{\mathrm{c,u,d}\}$ [55,， $\mathbb{E}[X_i^s]$关于 m 是非递减的，Lemma 2] 意味着 $\mathbb{E}^0[D_i]$ 本身是任务数量 m的非递减函数。

除了能够进行精确推导外，（3）–（4）还可用于通过蒙特卡罗模拟估计期望相对延迟 $\mathbb{E}^0[D_i]$ 及其梯度 $\nabla_p \mathbb{E}^0[D_i]$ 。

梯度下降

如第 4、5 和 6 节所述，所有考虑的系统性能指标都显式依赖于相对延迟。因此，公式（4）使我们能够应用梯度下降，针对这些目标优化路由向量 p 。

4 轮次复杂度

本节分析 `Generalized AsyncSGD` 在异构环境中的收敛行为。首先，第 4.1 节利用 [31] 中确立的性能界，根据平均相对延迟刻画轮次复杂度。接下来，第 4.2 节探讨将基于轮次的收敛速率作为主要性能指标的实际影响与局限性。

4.1 达到 ϵ 准确率所需的轮次

以下定理给出了 `Generalized AsyncSGD` 达到 ϵ 近似驻点所需更新次数 K 的界。重要的是，该结果明确体现了路由概率 p、并发度 m 以及客户端速度异质性的影响。正如我们在补充材料的附录 F 中所述，该结果建立在 Leconte 等人 [31] 的先前结果之上。

▶ **定理 3。** 在假设 ***A1–A5***下，存在 $\epsilon_0 > 0$ ，使得对于任意目标精度 $\epsilon \in (0,\ \epsilon_0]$ 以及满足 $\eta \le \eta_{\max}(p,\ m)$ 的任意学习率，其中

$$\eta_{\max}(p,m) = \min\left\{ \frac{n^2}{8L\sum_{i=1}^n p_i^{-1}},\ \frac{n^2\epsilon}{2LB\sum_{i=1}^n p_i^{-1}},\ \frac{n\sqrt{\epsilon}}{2L}\left(C(m-1)\sum_{i=1}^n \frac{\mathbb{E}^0[D_i]}{p_i^2}\right)^{-1/2}\right\}, \tag{8}$$

期望梯度范数满足

$$\frac{1}{K}\sum_{t=0}^{K-1}\mathbb{E}^0\big[\|\nabla f(w_t)\|^2\big]\le\epsilon$$

只要 $K\ge K_\epsilon(p,m)$，其中：

$$K_\epsilon(p,m)=\frac{24L\Delta}{n\epsilon}\left[\left(4+\frac{B}{\epsilon}\right)\sum_{i=1}^{n}\frac{1}{np_i}+\left(\frac{C(m-1)}{\epsilon}\sum_{i=1}^{n}\frac{\mathbb{E}^0[D_i]}{p_i^2}\right)^{1/2}\right],\tag{9}$$

其中常数为 $\Delta=f(w_0)-f^*$, $B=6(\sigma^2+2M^2)$，以及 $C=6(\sigma^2+G^2)$。利用定理2，可以显式计算稳态相对延迟 $\mathbb{E}^0[D_i]$ 。

证明。 见补充材料的附录 F。 ◀

若放宽假设 **A5** ，则补充材料第 G 节给出了在第 2.6 节的模型下推导出的 $K_\epsilon(p,m)$ 的另一种表达式。

式 (9) 中的收敛率包含两个不同的项。第一项刻画了路由策略和数据异质性的影响。在均匀路由（$p^{\text{uni}}=(\frac{1}{n},\dots,\frac{1}{n})$)和同质数据（$M=0$）下，这一项简化为经典的随机梯度下降收敛率$\mathcal{O}\Big(\frac{L\Delta}{\epsilon}\Big(1+\frac{\sigma^2}{\epsilon}\Big)\Big)$ [19, 4]。第二项考虑了异步性引起的延迟，因此量化了尽管存在梯度陈旧性，为达到与同步随机梯度下降相当的准确率所需的额外迭代次数。对于均匀路由，该惩罚项简化为$\mathcal{O}\Big(\frac{L\Delta\sqrt{\sigma^2+G^2}}{\epsilon\sqrt{\epsilon}}(m-1)\Big)$，其数值完全由系统并发度 m 决定，而与计算或通信速度的大小无关。一般而言，对 G 和 σ 的显式依赖表明，较大的方差或梯度幅值会加剧异步性的影响，因为“旧”梯度与当前真实方向的偏差更大。

补充材料 H 节给出了完全并发（$m=n$）条件下图像分类任务的数值实验，结果表明，轮次优化路由 p^{*K}（即使 K_ϵ 最小的路由）相较于基线方法，能够显著减少通信轮次。

4.2 轮次减少意味着训练更快吗?

在得到相对延迟的闭式表达式（见 (3)）后，我们现在考察轮次复杂度 K_ϵ 对路由向量 p 和并发级别 m 的依赖关系。(9) 中的第一项在均匀路由概率下最小，这体现了无偏客户端参与的优势。第二项通过 $\mathbb{E}^0[D_i]$ 量化陈旧性开销，并且随 m 增大而非减。因此，对于任何固定的路由策略，提高并发度都会加剧陈旧梯度造成的性能损失。

从严格的轮次复杂度角度来看，这意味着最优配置是采用 $m=1$ 的均匀路由。事实上，令 $m=1$ 即可完全消除延迟项，从而实质上恢复串行随机梯度下降。然而，这种平凡的解会严重限制吞吐量，并抵消异步性的主要优势：拖尾者缓解和并行效率。

即使并发度固定为 $m>1$ （例如，像 [29] 中那样采用完全并发 $m=n$），严格最小化轮次复杂度 K_ϵ 实际上会使系统同步到最慢客户端的节奏。如补充材料 H 节和第 5.3.3 节所示，要降低陈旧度，优化器就必须大幅提高分配给较慢客户端的路由概率，导致更新频率急剧下降（在我们的补充材料 H 节实验中，从均匀路由下每个时间单位 41 次更新降至

仅 2.4 次）。尽管这种保守策略减少了通信轮次总数 K_ϵ，但以实际运行时间衡量性能时，更新频率的下降带来的损失超过了这一收益。因此，就实时收敛速度而言，轮次优化路由可能不如简单的均匀基线。

由于延迟惩罚会随梯度上界 G 成比例扩大，激进的梯度裁剪理论上可以缓解这种陈旧度。然而，这种策略同时会阻碍学习进展，并最终减缓收敛速度。

该分析揭示了联邦学习文献中广泛使用的标准基于轮次的收敛指标存在一个根本局限：**它忽略了**实际运行时间**内的性能表现**，而这正是采用异步方法的主要动机。因此，下一节将介绍一种时间感知指标，以更准确地反映异步联邦学习系统的实际效率。

5 时钟时间复杂度

为考虑随机系统延迟，我们使用**达到 ϵ准确率所需的预期实际运行时间**来评估性能。不同于基于轮次的指标，该度量捕捉了训练过程的实际物理持续时间，直接回应了采用异步学习的核心动机。

5.1 达到 ϵ 准确率所需的时间

令 τ_ϵ 表示执行保证达到 ϵ-准确率所需的 $K_\epsilon(p,\ m)$ 轮次所需的随机总实际运行时间。在第 2.6 节的随机网络模型中，以下命题给出了其期望 $\mathbb{E}^0[\tau_\epsilon]$ 的闭式表达式。

▶ **命题 4**。 在假设 ***A1–A5*** 下，存在 $\epsilon_0 > 0$，使得对于任意目标精度 $\epsilon \in (0,\ \epsilon_0]$，以及满足 $\eta \leq \eta_{\max}(p,\ m)$ 的任意学习率，达到 ϵ-准确率所需的期望实际经过时间为：

$$\mathbb{E}^0[\tau_\epsilon] = \frac{K_\epsilon(p,m)}{\lambda(p,m)}, \qquad where: \tag{10}$$

$K_\epsilon(p,\ m)$ 表示达到 ϵ-准确率所需的轮次数，见定理 *3* 的公式 (9)。

$\lambda(p,\ m)$ 表示单位实际运行时间内完成的轮次数的期望，其表达式为

$$\lambda(p,m) = \sum_{i=1}^{n} \mu_i^{\mathrm{u}}\, \mathbb{E}[\xi_i^{\mathrm{u}}] = \frac{Z_{n,m-1}}{Z_{n,m}}, \tag{11}$$

$$\frac{\partial}{\partial p_j}\lambda(p,m) = \frac{1}{p_j}\,\lambda(p,m) \sum_{s\in\{\mathrm{c,d,u}\}} \mathbb{E}[\,X_j^s - \xi_j^s\,], \tag{12}$$

其中 $Z_{n,\ m}$ 和 $Z_{n,\ m-1}$ 是命题 *1* 中定义的归一化常数，$\xi \sim \pi_{n,\ m}$ 和 $X \sim \pi_{n,\ m-1}$。

证明。参见补充材料的附录 I。 ◀

5.2 讨论

为说明所提出的时间感知指标的行为，我们考虑一个简单的双客户端系统，包含两种场景：**(i) 同质资源**，两个客户端的计算速率和通信速率均相同（$\mu_i^{\mathrm{c}} = \mu_i^{\mathrm{u}} = \mu_i^{\mathrm{d}} = 1$ 对于 $i \in \{1,2\}$）；**(ii)异构资源**，客户端 2 的计算速率和通信速率均为客户端 1 的三倍（$\mu_2^{\mathrm{c}} = \mu_2^{\mathrm{u}} = \mu_2^{\mathrm{d}} = 3$），而其他参数保持不变。图 2 展示了期望实际经过时间 $\mathbb{E}^0[\tau_\epsilon]$ 随并发级别 (m) 和客户端 1 的路由概率 $(p_1 = 1 - p_2)$ 变化的情况。

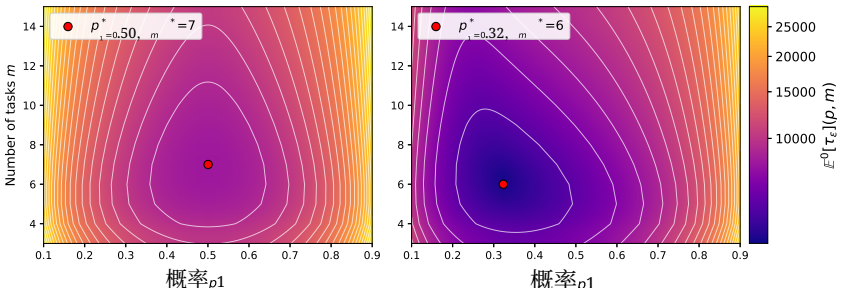


图 2 并发度 m 和路由概率 p_1 对 $\mathbb{E}^0[\tau_\epsilon]$的影响。左：同质资源。右：异构资源，其中客户端 2 的速度是客户端 1 的三倍（$\mu_2^s = 3\mu_1^s$ 对于 $s \in \{c,\ u,\ d\}$）。常数：$\Delta = L = \sigma = 1$、$M = 5$和 $G = 14$。

与轮次复杂度 K_ϵ不同，实际运行时间 $\mathbb{E}^0[\tau_\epsilon]$ 通常不会随 m单调变化。两种场景都表明存在一个能使 $\mathbb{E}^0[\tau_\epsilon]$最小化的最优并发度 $m^* \geq 1$。直观而言，当 $m < m^*$时，系统利用不足，吞吐量受到限制；反之，当 $m > m^*$时，吞吐量持续增加，但由此产生的梯度陈旧性会降低收敛率，其负面影响超过并行化带来的收益。

此外，在异质情形下，优化后的路由更倾向于选择速度较快的客户端，但这种倾向不如单纯以吞吐量最大化为目标的策略明显。

因此，$\mathbb{E}^0[\tau_\epsilon]$ 体现了两个相互竞争目标之间的根本性**权衡**：(i) 降低 K_ϵ 以提高更新质量；(ii) 提高系统吞吐量λ 以增加更新频率。这两个目标本质上相互冲突，因为改善其中一个通常会损害另一个。

通过综合考虑这两种影响，$\mathbb{E}^0[\tau_\epsilon]$提供了一项兼顾陈旧性控制与收敛速度的严谨性能准则，使其成为优化异步联邦学习系统的稳健指标。

5.3 数值结果

本节通过证明联合优化路由向量 p 和并发级别 m 能够显著提高实际收敛速度，验证我们的理论发现。我们将所提方法与三种基线方法进行比较，得到以下四种策略：

1. **时间优化** `Generalized AsyncSGD` **（提出的方法）：** 该策略采用最优参数 $(p^{*\tau}, m^{*\tau})$，这些参数经推导可最小化达到 $\mathbb{E}^0[\tau_\epsilon]$ 精度所需的期望时间ϵ，如命题 4 所述。2. **标准基线（**`AsyncSGD`**）** [29, 算法 2]：表示采用完全并发（$m = n$）和均匀路由（p^{uni}）的传统方法。

3. **轮次优化** `Generalized AsyncSGD` [31, 2]：使用路由向量 p^{*K} 来最小化轮次复杂度 K_ϵ，同时保持完全并发（$m = n$）。这代表了文献中仅专注于优化基于轮次的收敛界的方法。

4. **最大吞吐量** `Generalized AsyncSGD`：一种以完全并发（$p^{*\lambda}$）最大化系统更新频率（$m = n$）的策略。

我们特别纳入最后一个基线，以凸显一个关键权衡：单纯最大化每秒更新次数（吞吐量），可能会损害收敛稳定性和最终准确率。

5.3.1 实验设置

我们模拟了一个由 $n = 100$ 个客户端组成的异构网络，并将其划分为五个集群（A–E 类），涵盖从高性能工作站到资源受限设备的各类设备。为模拟延迟关键型边缘联邦学习环境，我们将总体规模偏向掉队客户端；具体而言，D 类占最大集群（40%），而高性能节点（E 型）仅占 10%，其余 50% 为中端设备。计算（μ^{c}）、上行链路（μ^{u}）和下行链路（μ^{d}）的具体服务速率详见表 1。

表 1 客户端集群及服务速率（速率 μ，单位为任务/秒）。

Type	描述	μ^{c}	μ^{u}	μ^{d}	数量
A	计算快、网络慢	10.0	2.0	2.5	15
B	计算慢、网络快	0.3	9.0	10.0	15
C	均衡	5.0	6.0	7.0	20
D	掉队者	0.15	0.1	0.12	40
E	超级客户端	12.0	10.0	11.0	10

我们在两种不同的分布场景下评估 EMNIST [14] 数据集的性能：

- **同质（独立同分布）：** 数据在各客户端之间以相同方式分布，每个客户端从每个类别中持有相同数量的样本。
- **异质（非独立同分布）：** 我们使用狄利克雷分布模拟特征和标签异质性。对于每个类别 k，分配给客户端 j 的样本比例取自 $q_k \sim \mathrm{Dir}_n(\alpha)$，其中 $\alpha = 0.2$ 是集中参数，如下 [64, 33]。关于 CIFAR-100 [30] 的补充实验见补充材料的 K 节。

5.3.2 优化策略与结果

由于 m具有离散性，我们采用序贯优化方法来最小化 $\mathbb{E}^0[\tau_\epsilon]$。从 $m = 2$开始迭代，对于每个固定的m，使用梯度下降优化路由向量 p。当目标函数 $\mathbb{E}^0[\tau_\epsilon]$ 不再下降时，搜索终止，这表明最优组合 $(m^{*\tau}, p^{*\tau})$ 已被超越。为加速收敛，我们采用热启动策略：在 $m + 1$ 层级进行的优化使用在 m层级找到的最优向量 p^* 进行初始化。p 的优化使用 Adam优化器 [28]，，其梯度 $\mathbb{E}^0[\tau_\epsilon]$ 通过定理2和命题 4以闭式形式计算（更多详情请参见补充材料的附录 J）。

类似地，我们通过 Adam 计算最大吞吐量向量 $p^{*\lambda}$ 和轮次优化向量p^{*K}，并利用方程（12）和（4）中给出的 λ 和 $\mathbb{E}^0[D_i]$ 的梯度表达式。第 2.5 节中引入的常数 σ、 M和 G 根据训练数据进行经验估计。我们将目标梯度范数上界设为 $\epsilon = 1$。

优化后的路由概率和相应的陈旧度指标详见表2。我们特别分析了数量 $\mathbb{E}^0[D_i(p,\ m)]\,p_i^{-2}$，将其作为陈旧性影响因子；该项表示每个客户端对 K_ϵ 中陈旧度项的贡献，有助于识别可能降低算法稳定性的特定集群。

表2 不同客户端集群（A–E 类）中优化后的路由概率和陈旧性影响因子比较 s
不同客户端集群（A–E 类）。

	路由概率 $p \times 100$			**陈旧性影响因子** $\mathbb{E}^0[D_i(p,\ m)]\,p_i^{-2} \times 10^{-2}$			
Type	$p^{*\tau}$	$p^{*\lambda}$	p^{*K}	$(p^{*\tau},\ m^*)$	$(p^{*\lambda},\ n)$	(p^{*K}, n)	$(p^{\text{uni}},\ n)$
A	1.307	0.845	0.526	14.2	181.7	8.6	7.4
B	0.514	0.011	0.627	182.0	49 543.7	28.0	33.9
C	1.752	1.591	0.506	5.5	65.7	4.6	3.8
D	0.34	0.005	1.691	1 615.7	783 209.8	84.6	229.6
E	2.405	5.514	0.496	2.1	12.4	2.5	2.0

关于并发度，优化得出的最优水平为 $m^{*\tau} = 91$，严格低于客户端总数 $n = 100$。这一结果挑战了文献中的标准惯例，因为文献通常假设完全并发（$m = n$）。

吞吐量优化路由（$p^{*\lambda}$）极大地偏向快速客户端，尤其是超级客户端 （E 型），同时大幅降低掉队客户端（D类）的权重。虽然这一策略最大限度地提高了整体更新频率，但也有引入显著偏差的风险。表2显示，D类和B类客户端的陈旧度因子急剧增大（比其他类型高出数个数量级）。这表明它们的更新不仅很少，而且陈旧度极高，从而严重阻碍收敛。

相比之下，轮次优化路由（p^{*K}）优先考虑掉队客户端，以尽量降低全局陈旧度，因此所有集群的陈旧度因子都较低且均衡，但代价是系统吞吐量降低。

所提出的路由（$p^{*\tau}$）实现了平衡：与均匀基线相比，它为快速客户端分配了更高的选择概率，同时确保掉队客户端仍保有不可忽略的选择概率。这种方法有效平衡了更新频率与陈旧度，限制了延迟更新造成的误差，同时保持了高于 p^{*K} 的吞吐量。

从定量结果来看，最大化网络吞吐量可实现 $\lambda(p^{*\lambda},\ n) = 152$ 更新/时间单位，显著高于标准的 `AsyncSGD` ($\lambda(p^{\text{uni}},\ n) = 7.4$)。另一方面，轮次优化路由侧重于降低陈旧度，因此吞吐量最低（$\lambda(p^{*K},\ n) = 4.5$）。最后，我们所提出的时间优化配置实现了有效的折中（$\lambda(p^{*\tau},\ m^*) = 18.7$）。

这些鲜明的对比为分析速度-准确率权衡奠定了基础：吞吐量$8\times$ 的提升是否真的能带来更快的收敛？而仅以降低误差（陈旧度）为目标设计的策略，在实际运行时间方面是否高效？

5.3.3 学习性能与权衡分析

我们采用第 5.3.1 节所述的实验设置来模拟训练过程。为评估系统抵御初始瞬态的鲁棒性，我们将系统初始化在非平衡状态：在 $t = 0$， m 任务被均匀随机地分配给客户端下行服务器，而不是从平稳分布中抽样。此外，为了验证我们的结果并非指数分布假设造成的假象，我们评估了**计算和通信时间** 服从以下三种不同分布时的性能：

(i) 指数分布： 请注意，理论推导特别要求采用指数服务时间。**(ii) 确定性：** 固定服务时间，等于 $1/\mu$ （方差为零）。**(iii) 对数正态分布：** 均值为 $1/\mu$的重尾分布。我们将底层正态分布的方差设为 $\sigma_N^2 = 1$，以反映真实边缘环境的高度变异性。此设置使所有客户端的变异系数保持固定，从而隔离服务速率的影响。

模型采用标准的多类交叉熵损失进行训练，并在未见过的标签均衡测试集上报告性能。学习率通过网格搜索进行调优。实现细节见补充材料 B 节。

表 3 目标测试准确率为 0.6 时相对于基线方法的时间缩减百分比（括号内为 0.75）。

场景		时间缩减（%），相较于		
分布	Data	最大吞吐量	轮次优化	异步随机梯度下降
Exp.	IID	67.5 (66.59)	62.59 (66.71)	46.28 (46.88)
	非独立同分布	79.3	57.04	35.6 (36.56)
对数正态分布	IID	59.08 (70.46)	64.35 (66.1)	41.77 (46.41)
	非独立同分布	79.17	62.44	37.0 (42.38)
Det.	IID	52.35 (64.24)	49.35 (61.38)	29.84 (38.86)
	非独立同分布	N/A	58.88	31.16 (37.81)

图3表明， `Generalized AsyncSGD` 采用优化后的参数（$p^{*\tau}$， $m^{*\tau}$）后，在整个学习过程中始终优于基线方法。表 3 量化了这一优势，报告了达到 0.6 目标精度所需实际运行时间的大幅缩短（括号中的数值表示在可达到更高目标精度 0.75 时的缩短幅度）。值得注意的是，与标准的 `AsyncSGD`以及最大吞吐量和轮次优化策略相比，我们的方法显著加快了速度。此外，在所有模拟服务时间分布下，这些提升都始终如一，证实了我们方法的鲁棒性远不止适用于指数计算时间这一理论假设。

图3和表 3中的结果鲜明地揭示了更新数量 （频率）与更新质量 （陈旧度）之间的根本矛盾。在一个极端，最大吞吐量策略（$p^{*\lambda}$）优先考虑数量。在固定的 400 个时间单位内，它执行了超过 60 000 次参数更新，数量超过了我们所提方法（$(p^{*\tau}, m^{*\tau})$， $\approx 6\,000$）的 $10\times$ 倍。然而，这种不受约束的速度是以牺牲质量为代价的：在非独立同分布确定性设置下，其最终准确率比优化后的方法低 60%。该策略表现出高度不稳定性（损失峰值频繁），并且

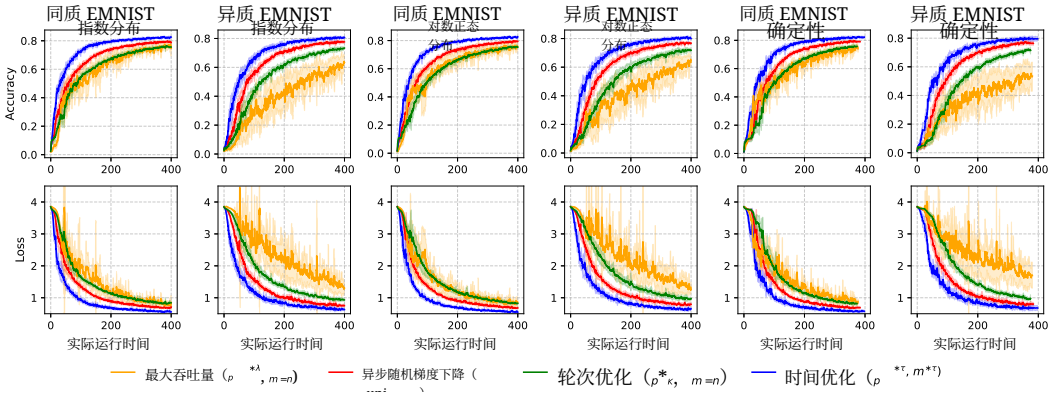


图3 第 5.3.1 节所述场景中中央服务器的测试集性能。上排显示测试准确率，下排显示损失，二者均以实际运行时间为横轴绘制。各列对应同质和异质数据环境下的不同服务时间分布。每个子图比较四种策略。模拟重复进行 10 次，每次持续 400 个实际运行时间单位。实线表示均值，阴影区域表示标准差。

需要采用比其他基线方法更小的学习率 $20\times$，以避免立即发散。这证实，单独最大化更新频率会降低模型质量，并浪费计算资源。

与之相反，轮次优化策略（p^{*K}）优先考虑质量。它积极优先选择掉队客户端以尽量降低陈旧度，从而确保较高的梯度保真度和稳定性。然而，这种侧重严重限制了更新数量，在相同时间内仅完成 1 800 次更新。虽然它在“每轮进展”方面表现有效，但每轮耗时过长，因此按实际运行时间衡量效率较低。

所提出的时间优化策略（$p^{*\tau}$，$m^{*\tau}$）有效弥合了这一差距。它实现了关键的平衡：在保持足够更新频率的同时，将陈旧度限制在适当范围内，以确保稳定收敛。最后，在这一特定设置中，均匀路由表现尚可，因为其概率与时间优化分布非常接近，不过它缺少所提方法所带来的定向加速效果。

6 能量复杂度

在许多实际场景中，设备都在严格的能量约束下运行（例如，电池容量有限）。因此，必须设计能够在保证收敛的同时最大限度降低能量使用量的学习方案。

6.1 能量模型

为了评估系统的能效，我们采用一种状态依赖型功耗模型，明确刻画异质客户端各自不同的硬件特性。

根据文献中既有的模型 [52, 63, 66, 39, 11],，每个客户端 $i \in \{1, \ldots, n\}$均被分配了与其三个活动阶段相对应的功率配置：

(i) 本地计算（$\mathcal{P}_i^{\mathrm{c}}$）： 梯度计算期间消耗的功率。根据 DVFS 定律 [41, 38, 52],，功耗通常与平均 CPU 频率的立方成正比。在我们的模型中，这表示为 $\mathcal{P}_i^{\mathrm{c}} \propto (\mu_i^{\mathrm{c}})^3$。**(ii) 上行传输（$\mathcal{P}_i^{\mathrm{u}}$）：** 为每个分配的任务将计算得到的梯度传输至中央服务器所需的功率。

(iii) 下行接收（$\mathcal{P}_i^{\mathrm{d}}$）： 为每个分配的任务从中央服务器下载模型参数所需的功率。

客户端 i执行特定任务 k 所消耗的总能耗，涵盖从接收参数到发送更新的完整周期，表示为：

$$E_{i,k} = \mathcal{P}_i^{\mathrm{d}} T_{i,k}^{\mathrm{d}} + \mathcal{P}_i^{\mathrm{c}} T_{i,k}^{\mathrm{c}} + \mathcal{P}_i^{\mathrm{u}} T_{i,k}^{\mathrm{u}}, \tag{13}$$

其中 $T_{i,k}^{\mathrm{c}}$, $T_{i,k}^{\mathrm{u}}$ 和 $T_{i,k}^{\mathrm{d}}$ 分别表示客户端 i接收的第 k个任务所对应的计算、上行链路和下行传输的随机持续时间。

我们假设中央服务器连接到固定电网，且不受电池容量限制。因此，我们关注边缘设备的能耗。不过，该模型很容易推广：如果服务器的能耗也需要考虑，则可将其传输功率纳入下行成本 $\mathcal{P}_i^{\mathrm{d}}$。

因此，系统在时间 t 的瞬时功率成本可写为：

$$P(t) = \sum_{i=1}^{n} \mathcal{P}_i^{\mathrm{c}} \, \mathbf{1}\{\xi_i^{\mathrm{c}}(t) > 0\} + \mathcal{P}_i^{\mathrm{u}} \, \xi_i^{\mathrm{u}}(t) + \mathcal{P}_i^{\mathrm{d}} \, \xi_i^{\mathrm{d}}(t). \tag{14}$$

6.2 达到 ϵ-准确率所需的能量

为评估算法能效，我们分析达到 ϵ-准确率所需的预期总能耗，记为 $\mathbb{E}^0[E_\epsilon]$。以下命题给出了该指标的闭式表征。

▶ **命题 5.** 在假设 ***A1–A5***下，存在 $\epsilon_0 > 0$，使得对于任意目标准确率 $\epsilon \in (0,\ \epsilon_0]$，以及满足 $\eta \le \eta_{\max}(p,\ m)$ 的任意学习率，达到 ϵ-准确率所需的期望能耗为：

$$\mathbb{E}^0[E_\epsilon] = K_\epsilon(p,m)\,\frac{\mathbb{E}[P(0)]}{\lambda(p,m)} = K_\epsilon(p,m)\sum_{i=1}^{n} p_i \mathcal{E}_i,$$

其中，$\mathcal{E}_i \triangleq \frac{\mathcal{P}_i^{\mathrm{c}}}{\mu_i^{\mathrm{c}}} + \frac{\mathcal{P}_i^{\mathrm{u}}}{\mu_i^{\mathrm{u}}} + \frac{\mathcal{P}_i^{\mathrm{d}}}{\mu_i^{\mathrm{d}}}$ 是客户端 i 每项任务的平均能耗，K_ϵ 是定理 *3* 的公式 (9) 中定义的轮次复杂度，λ 是 (11) 中定义的系统吞吐量。

证明。 参见补充材料的 L 节。 ◀

量 $\sum_{i=1}^{n} p_i \mathcal{E}_i$ 表示每轮平均能耗。重要的是，该量仅取决于路由向量 p 和客户端的硬件特性，与并发级别 m 无关。

6.3 能耗-延迟权衡

由于轮次复杂度 K_ϵ 关于 m 单调不减（见第 4.2 节），而每轮能耗相对于 m 保持不变，因此当 $m = 1$ 时，预期总能耗 $\mathbb{E}^0[E_\epsilon]$ 严格达到最小。因而，能量最优策略需要设定 $m = 1$，并选择路由向量 p_E^*，使以下乘积最小（而不是在 $m = 1$ 时针对 K_ϵ 的均匀路由最优方案）：

$$\min_{p \in \mathcal{P}_n} \underbrace{\left[\frac{24L\Delta}{n^2\epsilon}\left(4 + \frac{B}{\epsilon}\right)\sum_{j=1}^{n}\frac{1}{p_j}\right]}_{K_\epsilon(p,\,m=1)} \times \underbrace{\left[\sum_{i=1}^{n} p_i \mathcal{E}_i\right]}_{\text{Energy per round}}. \tag{15}$$

通过应用柯西-施瓦茨不等式（见补充材料的第 M 节），该最小化问题的闭式解为：

$$p_i^{*E} \propto \frac{1}{\sqrt{\mathcal{E}_i}}, \quad i \in \{1, \ldots, n\}. \tag{16}$$

在 p^{*E} 处计算式 (15) 中的目标函数，可得最小平均能耗：

$$E^* = \frac{24L\Delta}{n^2\epsilon}\left(4 + \frac{B}{\epsilon}\right)\left(\sum_{i=1}^{n}\sqrt{\mathcal{E}_i}\right)^2. \tag{17}$$

然而，采用 $m = 1$ 会消除并行性和异步性的优势，导致实际训练时间长得难以接受。这揭示了能耗与延迟之间的根本权衡：提高并发级别 m 可加快训练的实际运行时间，但会增加总能耗，因为更高的陈旧度会导致达到目标精度所需的轮次增加。反之，若要尽量降低能耗，则应采用顺序执行，并将路由指向低功耗设备。

硬件异构性进一步加剧了这种矛盾：能效高的设备往往速度最慢（例如，与图形处理器相比的物联网传感器），因此，旨在降低能耗的路由策略通常会带来最大的延迟惩罚。

6.4 联合优化问题

根据 [62, 11, 66, 37, 38],，我们针对路由概率 p 和并发级别 m构建联合优化问题，在确保收敛的同时，解决实际训练时间与总能耗之间固有的权衡。两个目标（期望时间 $\mathbb{E}^0[\tau_\epsilon]$ 和能耗 $\mathbb{E}^0[E_\epsilon]$）的单位和量级不同，因此采用归一化标量化方法将其结合。

具体而言，每个目标都除以其理论最小值进行归一化，该最小值对应于单独优化该指标。由此得到的问题为

$$\min_{m\in\mathbb{N}_{>0},\, p\in\mathcal{P}_n} \quad \rho\,\frac{\mathbb{E}^0[E_\epsilon(p,m)]}{E^*} \;+\; (1-\rho)\,\frac{\mathbb{E}^0[\tau_\epsilon(p,m)]}{\tau^*}, \tag{18}$$

其中， $\tau^* = \mathbb{E}^0[\tau_\epsilon(p_\tau^*, m_\tau^*)]$ 是可实现的最短训练时间（见第5.3节）， $E^* = \mathbb{E}^0[E_\epsilon(p_E^*, 1)]$ 是可实现的最低能耗，而 $\rho \in [0,1]$ 是用户定义的权衡参数。

权重 ρ和 $1-\rho$分别表示能量和时间的相对重要性。等价地，比率 $\rho/(1-\rho)$定义了边际替代率，用于量化为了实现一定程度的能耗相对降低，人们愿意牺牲多少相对训练速度。

6.5 数值结果

6.5.1 实验设置

我们通过在第 5.3.1 节的实验框架中加入明确的功耗模型，来解决联合优化问题 (18)。客户端服务率遵循表 1，而本地计算功率则采用立方动态电压频率调节缩放定律建模，
$P_{\text{comp}} = \kappa(\mu^c)^3$,其中 κ 是硬件特定能量系数。

如表 4 所示，能量系数 κ 反映了我们网络中不同客户端配置之间极端的异质性。虽然 A 型高效的架构（ κ 较低）能够实现

表 4 能量系数和功率配置（归一化单位）。计算功率遵循 $P_{\text{comp}} = \kappa(\mu^{\text{c}})^3$, ，其中较低的 κ 表示更高的硬件效率。

Type	特征配置	系数 κ	相对功率配置 $\mathbf{P}_{\text{comp}}$	$\mathbf{P}_{\text{u}}$	$\mathbf{P}_{\text{d}}$
A	计算高效且网络受限	0.08	80.0	5.0	3.0
B	计算受限且通信优化	200.0	5.4	15.0	10.0
C	速度与功耗均衡	0.25	31.3	4.0	3.0
D	掉队者且效率极低	14400.0	48.6	0.5	0.2
E	高吞吐量与高功耗	1.50	2592.0	50.0	40.0

适中的功率下实现高吞吐量，但由于架构和散热开销，E 型需要消耗大量功率，速度却仅略有提升。最值得注意的是，尽管计算速度极差，D类仍会消耗大量基线计算功耗。由于最快和最慢的客户端都并非严格意义上的能量最优，如何应对这种硬件多样性，使得我们的能量感知优化极具挑战性。

6.5.2 优化策略与结果

对于给定的 ρ，我们通过确定最优组合 $(p^{*\rho},\ m^{*\rho})$ 来求解联合优化问题 (18)。为处理并发度 m 的离散特性，我们采用序贯优化策略：从 $m = 1$ 开始，针对每个固定的 m ，使用梯度下降优化路由向量 p ，并以前一层级得到的解作为热启动初始化。

图 4 总结了联合优化：左图展示了标注有 ρ 的时间–能量帕累托前沿，右图则显示了当目标从严格以时间为中心（$\rho = 0$）转向严格以能量为中心（$\rho = 1$）时，最优路由概率 $p^{*\rho}$ 和并发度 $m^{*\rho}$ 的变化。

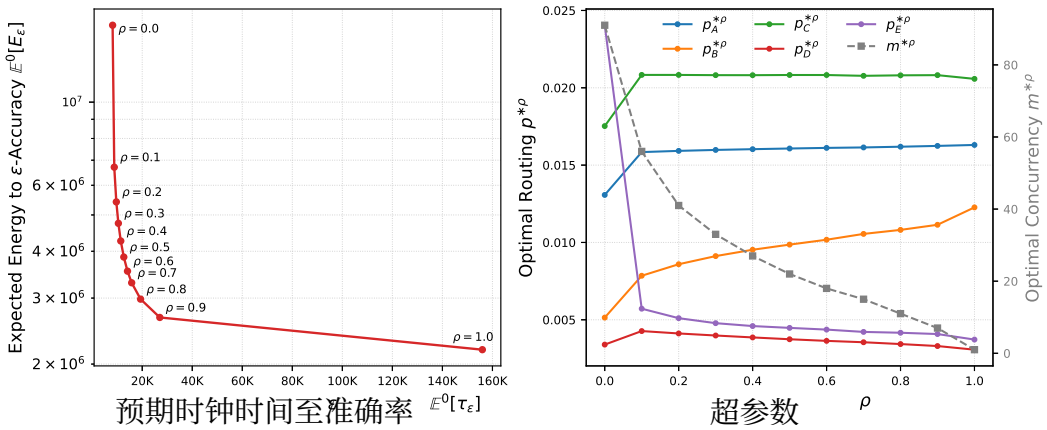


图 4 联合优化分析。**左图：** 标注有 ρ 的时间–能量帕累托前沿。**右图：** 最优路由概率 $p^{*\rho}$ 和最优并发度 $m^{*\rho}$ 随 ρ 的变化，展示了系统性地减少对能量效率低下的 E 型客户端的使用。

引入轻微的能耗惩罚（$\rho = 0.1$）会使并发度从 $m = 91$骤降至 56，从而推动帕累托前沿上的初始能量下降。随着 $\rho \to 1$，系统进一步

通过进一步降低并发度来减少功率浪费，最终收敛至严格的串行执行（$m = 1$）。

同时，路由动态也反映了网络的硬件异构性。以时间为中心的优化（$\rho = 0$）会极大地倾向于速度较快的 E 型客户端。然而，增加 ρ 会迅速限制这一群组，因为其功耗开销巨大，从而将权重转向高效的 C 型和 A 型客户端。相反，在所有机制下，D 类掉队客户端都会受到惩罚，因为它们既没有足够的速度来缩短时间，也不够高效，无法节省能量。

6.5.3 学习性能与权衡分析

为在真实的学习场景中验证这些理论发现，我们在 KMNIST 数据集 [12] 上模拟训练，并使用相同的时间和功率配置。对于每个 ρ，我们采用相应的最优配置（$p^{*\rho}$，$m^{*\rho}$）。我们考虑异构数据分布（狄利克雷分布，参数为 $\alpha = 0.2$）和指数服务时间。将 10 次独立运行的结果取平均，直至达到 0.75 的目标测试准确率，实证结果（图 5）与我们的理论性能趋势一致。结果显示，ρ决定了一种非线性权衡：稍微放宽对收敛速度的要求，就能大幅节省能量。

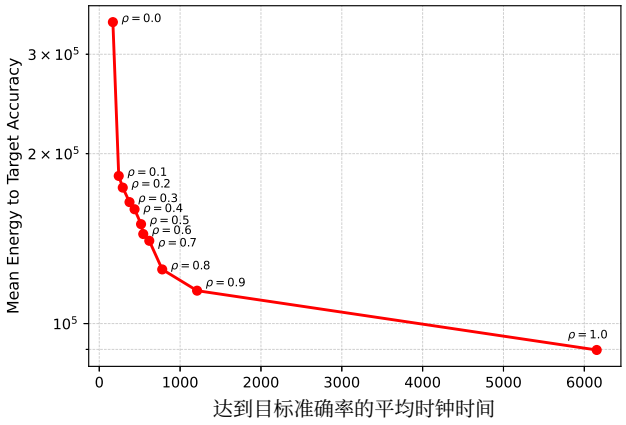


图 5 KMNIST 数据集的实测时间-能量帕累托前沿，展示了在 10 次独立运行中达到 0.75 目标测试准确率所需的平均时钟时间与平均能量。

为进一步进行实证验证，我们在 KMNIST 和 EMNIST 数据集上，将联合优化策略 ($p^{*\rho}$, $m^{*\rho}$) 与 `AsyncSGD` 基线 [29, 算法 2]进行比较。正如先前分析所揭示的，我们设置 $\rho = 0.1$，以实现显著的能量节省，同时尽量减少收敛延迟。模拟采用第 5.3.3 节中的设置，并使用表 4 中的功率配置。表 5 详细列出了所有评估场景中相对于基线的改进。EMNIST 数据集的详细学习轨迹以时钟时间和能量为横轴绘制，见补充材料第 N 节。

通过惩罚能效低下的客户端，并将并发度限制为 $m = 56$，优化器在所有评估过的网络条件下，始终能将总能耗降低 36%–49%。更重要的是，在几乎所有场景中，实际时间收敛也同时提速，最高达 19%。唯一的例外是 EMNIST 确定性非独立同分布设置：该策略以时间仅增加 3.15% 为代价，换来了能耗大幅降低 36.19%。这证实了，朴素的 `AsyncSGD` 在实际时间和能耗方面本质上都不是最优的。相反，我们的联合策略会主动沿帕累托前沿进行权衡，以确保

大幅节省能耗，同时不牺牲整体收敛速度。

表 5 在 $\rho = 0.1$ 时，$(p^{*\rho},\ m^{*\rho})$ 相较于 `AsyncSGD`，在目标准确率为 0.80（KMNIST）和 0.70（EMNIST）时的相对时间和能量降幅（%）。正值表示节省；负值表示增加。

数据集	指标	指数分布		对数正态分布		确定性	
		IID	非独立同分布	IID	非独立同分布	IID	非独立同分布
KMNIST	Time	6.11	10.58	18.95	8.63	15.46	0.53
	能量	46.90	46.10	45.53	40.90	47.19	36.62
EMNIST	Time	15.41	3.90	6.12	6.93	7.12	-3.15
	能量	46.26	41.60	41.83	48.97	39.40	36.19

7 引入 CS 侧缓冲区

虽然异步联邦学习的理论分析通常假设全局更新能够即时完成，但实际的边缘部署需要顺序处理，以确保对共享参数的原子访问。这种串行化造成了瓶颈：短时间内连续到达的更新必须先在中央服务器排队，然后才能纳入 [67]。这种中央服务器侧的延迟源于以下几个因素：

- **模型规模：** 更新大规模模型（例如大语言模型）会产生不可忽略的内存输入输出和向量运算延迟，尤其是在基于 CPU 的服务器上 [27]。**安全与隐私开销：** 为了实现稳健部署，中央服务器必须对每个传入的更新执行计算密集型任务（例如解密载荷、验证数字签名或筛查投毒攻击），确保更新可以安全应用后才能进行处理 [65, 15, 20]。**高并发度：** 在大规模物联网网络中，客户端更新的总到达率可能暂时超过中央服务器的服务能力，导致严重拥塞和不可避免的排队延迟 [31, 48]。

因此，我们将框架扩展为显式地将中央服务器建模为一个排队系统，其中传入的梯度会被缓冲并按顺序处理。

7.1 模型描述

我们在第 2.6 节介绍的网络模型基础上，放宽了中央服务器服务时间可忽略不计的假设。相反，我们将中央服务器建模为一个采用先进先出规则的单服务器队列，并假设模型更新处理时间是速率为 μ^{cs} 的独立同分布指数随机变量。纳入这些动态特性后，得到图6所示的闭合排队网络。

为推导相对延迟的闭式表达式，有必要追踪从任务派发给客户端到相应梯度在中央服务器上应用之间发生的参数更新次数。这需要确定中央服务器队列中每个等待任务的来源。由于标准的排队表示会将中央服务器上的所有任务汇总在一起，因而掩盖其来源，我们将系统重新表述为**多类别杰克逊网络**，并在图6中用不同颜色加以示意。每个任务都属于 n 个类别之一，其类别由被分配到的客户端确定。具体而言，当中央服务器将任务路由至客户端 i 时，该任务会被赋予标签 i，并在整个周期内保持此类别身份，直至其在中央服务器完成服务。

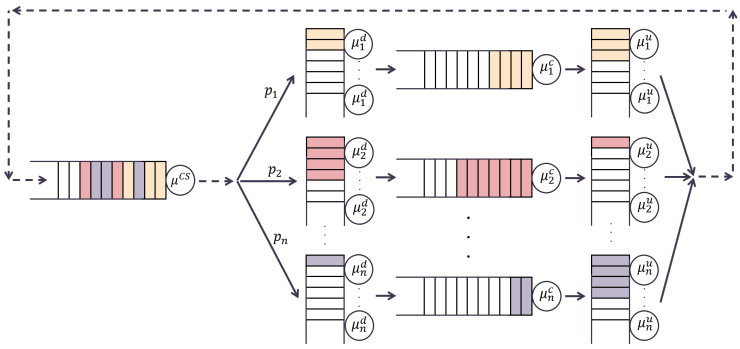


图6 纳入计算、通信和 CS 端更新延迟的扩展排队网络模型。不同颜色对应不同任务类别。

系统动态由定义在扩展状态空间$\mathcal{X}_{4n,\ m}$上的右连续且有左极限（càdlàg）随机过程 $Y = (Y(t))_{t\geq 0}$ 描述。在任意时刻 $t \geq 0$，状态定义为

$$Y(t) = \big(Y_i^{\mathrm{cs}}(t),\, Y_i^{\mathrm{d}}(t),\, Y_i^{\mathrm{c}}(t),\, Y_i^{\mathrm{u}}(t)\ ;\ i \in \{1,\dots,n\}\big),$$

其中， $Y_i^{\mathrm{cs}}(t)$ 表示当前在 CS 处的类别-i 任务数量（包括等待中或正在接受服务的任务）。类似地，对于 $s \in \{\mathrm{d},\mathrm{c},\mathrm{u}\}$， $Y_i^{s}(t)$ 表示客户端 i 对应服务器上的任务数量。

在此背景下，轮次 定义为 CS 上连续两次服务完成之间的时间间隔，对应于连续的模型更新。令 $\{T_k\}_{k\in\mathbb{N}}$ 表示这些完成时间的序列，其中 $T_0 = 0$；因此，区间 $[T_k, T_{k+1})$ 定义了第 k 轮。与前文相同， $\mathbb{P}$ 表示平稳概率测度，而 $\mathbb{P}^0$ （和 $\mathbb{E}^0$）表示与$\{T_k\}_{k\in\mathbb{N}}$相关的 Palm 概率（和期望），其条件是在时间 $t = 0$ 于 CS 发生一次服务完成。

要刻画相对延迟，需要分析模型更新时刻观测到的系统状态。为此，我们定义嵌入过程

$$\tilde{Y}_k = \big(\tilde{Y}_{i,k}^{\mathrm{cs}},\, \tilde{Y}_{i,k}^{\mathrm{d}},\, \tilde{Y}_{i,k}^{\mathrm{c}},\, \tilde{Y}_{i,k}^{\mathrm{u}}\ ;\ i \in \{1,\dots,n\}\big) \in \mathcal{X}_{4n,m-1},$$

它表示中央服务器完成一次服务后、且不包括刚刚完成服务的任务的网络状态。其各分量由跳后状态 $Y(T_k)$ 得出，如下所示：

$$\begin{cases} \tilde{Y}_{i,k}^{\mathrm{cs}} = Y_i^{\mathrm{cs}}(T_k), \\ \tilde{Y}_{i,k}^{\mathrm{d}} = Y_i^{d}(T_k) - \mathbf{1}\{A_k = i\}, \\ \tilde{Y}_{i,k}^{\mathrm{c}} = Y_i^{c}(T_k), \\ \tilde{Y}_{i,k}^{\mathrm{u}} = Y_i^{u}(T_k), \end{cases} \qquad i \in \{1,\dots,n\}, \tag{19}$$

其中， A_k 表示被选中接收在时间 T_k完成的任务的客户端索引，且 $\mathbb{P}(A_k = i) = p_i$。

相对延迟

在这一扩展框架中，相对延迟 $D_{i,\ k}$ 定义为：在第 k 轮分配给客户端 i 的任务逗留期间，中央服务器完成的服务数量。

此逗留涵盖整个周期：派发至服务器 d_i、客户端 i的排队和本地处理、经由服务器 u_i传输，最后在中央服务器排队并应用更新。如果第 k 轮（$A_k \neq i$）没有向客户端 i 分配任务，则设 $D_{i,k} = 0$。

7.2 平稳性分析

尽管过程 $(Y(t))_{t\geq 0}$ 和 $(Y_k)_{k\in\mathbb{N}}$ 通过类别标签追踪任务历史，但在所选状态空间中它们并非马尔可夫性，因为该空间只记录类别计数，而忽略了中央服务器处的先进先出顺序。尽管如此，该系统仍存在乘积形式的边际平稳分布，具体见下述命题；该分布将在后续的时延分析中使用。

▶ **命题 6。** 在第*7.1*节的设定下，过程 $(Y(t))_{t\geq 0}$ 和 $(Y_k)_{k\in\mathbb{N}}$ 分别存在唯一的平稳分布 $\phi_{n,\,m}$ 和 $\phi_{n,\,m-1}$。对于任意 $\mathtt{m} \in \mathbb{N}_{>0}$，分布 $\phi_{n,\,\mathtt{m}}$ 由以下乘积形式给出：

$$\phi_{n,\mathtt{m}}(x) = \frac{1}{W_{n,\mathtt{m}}} \frac{\left(\sum_{j=1}^{n} x_j^{\mathrm{cs}}\right)!}{\prod_{j=1}^{n} x_j^{\mathrm{cs}}!} \prod_{i=1}^{n} \frac{1}{x_i^{\mathrm{d}}!} \frac{1}{x_i^{\mathrm{u}}!} \prod_{\substack{s\in\{\mathrm{CS},\\ \mathrm{c,d,u}\}}} \left(\frac{p_i}{\mu_i^{\mathrm{d}}}\right)^{x_i^{\mathrm{d}}}, \tag{20}$$

对于任意状态 $x \in X_{4n,\,\mathtt{m}}$，其中 $W_{n,\,\mathtt{m}}$ 为归一化常数。

证明。 参见补充材料的 O 节。 ◀

由于该扩展模型的性能指标主要通过归一化常数 $W_{n,\mathtt{m}}$ 依赖于 $\phi_{n,\mathtt{m}}$，因此高效计算这些常数对于避免组合爆炸瓶颈至关重要。为此，我们扩展了 *Buzen* 的递归算法，将 CS 服务率显式纳入其中，从而能够在 $\mathcal{O}(nm^2)$ 时间和 $\mathcal{O}(m)$ 内存中计算这些常数（参见补充材料中的命题19）。这种高效性至关重要，因为它使我们能够精确计算下一节详述的期望相对延迟及其梯度。

7.3 延迟与梯度计算

以下结果将定理2扩展到具有 CS 侧延迟的新设定。

▶ **定理 7。** 在第*7.1*节的模型中，对于每个 $i, j \in \{1, \ldots, n\}$，以下恒等式成立：

$$\mathbb{E}^0[D_i] = \sum_{s\in\{\mathrm{CS,c,d,u}\}} \mathbb{E}\left[\tilde{Y}_i^s\right], \tag{21}$$

$$\frac{\partial}{\partial p_j}\mathbb{E}^0[D_i] = \frac{1}{p_j} \sum_{s,r\in\{\mathrm{CS,c,d,u}\}} \mathrm{Cov}\left(\tilde{Y}_i^s, \tilde{Y}_j^r\right). \tag{22}$$

此外，以下闭式表达式成立：

$$\sum_{s\in\{\mathrm{CS,c,d,u}\}} \mathbb{E}[\tilde{Y}_i^s] = p_i\tilde{\beta}_{\mathrm{CS},1} + \tilde{\beta}_{i,1} + \frac{W_{n,m-2}}{W_{n,m-1}}\gamma_i, \tag{23}$$

$$\sum_{s,r\in\{\mathrm{CS,c,d,u}\}} \mathbb{E}[\tilde{Y}_i^s\tilde{Y}_j^r] = \tilde{\alpha}_{i,j} + \tilde{\beta}_{i,2}\gamma_j + \tilde{\beta}_{j,2}\gamma_i + \tilde{\psi}_{i,j}$$

$$+\tilde{\alpha}_{i,j}^{\mathrm{CS}}+\tilde{\beta}_{\mathrm{CS},2}\left(p_i\gamma_j+p_j\gamma_i\right)+p_i\tilde{\alpha}_{\mathrm{CS},j}+p_j\tilde{\alpha}_{\mathrm{CS},i}. \tag{24}$$

这里，对于每个 $i,j\in\{1,\ldots,n\}$ 和 $\ell\in\{1,2\}$：

$$\gamma_i=p_i\left(\frac{1}{\mu_i^{\mathrm{d}}}+\frac{1}{\mu_i^{\mathrm{u}}}\right),\qquad \tilde{\beta}_{i,\ell}=\sum_{k=1}^{m-\ell}\left(\frac{p_i}{\mu_i^{\mathrm{c}}}\right)^k\frac{W_{n,m-\ell-k}}{W_{n,m-1}},$$

$$\tilde{\alpha}_{i,j}=\begin{cases}\displaystyle\sum_{k=1}^{m-1}(2k-1)\left(\frac{p_i}{\mu_i^{\mathrm{c}}}\right)^k\frac{W_{n,m-1-k}}{W_{n,m-1}} & \text{if } i=j,\\ \displaystyle\sum_{\substack{k,\ell=1\\ k+\ell\le m-1}}^{m-2}\left(\frac{p_i}{\mu_i^{\mathrm{c}}}\right)^k\left(\frac{p_j}{\mu_j^{\mathrm{c}}}\right)^\ell\frac{W_{n,m-1-k-\ell}}{W_{n,m-1}} & \text{if } i\neq j,\end{cases}$$

$$\tilde{\psi}_{i,j}=\frac{\gamma_i}{W_{n,m-1}}\left(\gamma_j W_{n,m-3}+\mathbf{1}_{\{i=j\}}W_{n,m-2}\right),$$

$$\tilde{\alpha}_{i,j}^{\mathrm{CS}}=p_i\sum_{k=1}^{m-1}\left(\frac{1}{\mu^{\mathrm{cs}}}\right)^k\frac{W_{n,m-1-k}}{W_{n,m-1}}\left[2p_j(k-1)+\mathbf{1}_{\{i=j\}}\right],$$

$$\tilde{\alpha}_{\mathrm{CS},i}=\sum_{k=1}^{m-2}\sum_{\ell=1}^{m-1-k}\left(\frac{1}{\mu^{\mathrm{cs}}}\right)^k\left(\frac{p_i}{\mu_i^{\mathrm{c}}}\right)^\ell\frac{W_{n,m-1-k-\ell}}{W_{n,m-1}},$$

$$\tilde{\beta}_{\mathrm{CS},\ell}=\sum_{k=1}^{m-\ell}\left(\frac{1}{\mu^{\mathrm{cs}}}\right)^k\frac{W_{n,m-\ell-k}}{W_{n,m-1}},$$

且常数 $W_{n,\,\mathfrak{m}}$ 对于 $\mathfrak{m}\in\{0,1,\ldots,m-1\}$，是利用补充材料附录*P*中的命题*19*递推计算得出的。

证明。 参见补充材料的第Q节。 ◀

在公式（23）和（24）中，红色 标出的项量化了由CS排队过程特有地引入的额外组成部分。相反，其余项（黑色部分）与不含CS侧拥塞的原始模型中的交互项完全一致。注意，当 $\mu^{\mathrm{cs}}\to\infty$ 时，我们便得到定理2的精确闭式表达式。这是因为对于所有 $n,\ m\in\mathbb{N}_{>0}$，$\lim_{\mu^{\mathrm{cs}}\to\infty}W_{n,\,m}=Z_{n,\,m}$，而CS特有项则趋于零（即 $\lim_{\mu^{\mathrm{cs}}\to\infty}\tilde{\alpha}_{\mathrm{CS},\,i}=\lim_{\mu^{\mathrm{cs}}\to\infty}\tilde{\alpha}_{i,\,j}^{\mathrm{CS}}=\lim_{\mu^{\mathrm{cs}}\to\infty}\tilde{\beta}_{\mathrm{CS},\,\ell}=0$）。最后，第3.3节中关于时延的分析和见解仍适用于这一扩展模型。

在本节剩余部分中，我们将建立实际运行时间与能量复杂度的等价性质，并明确考虑中央服务器的服务率。

7.4 时钟时间复杂度

定理 3 仍适用于第7.1节的模型。因此，K_ϵ 的表达式（9）仍然有效；唯一的修改涉及期望相对延迟 $\mathbb{E}^0[D_i]$的闭式公式，该公式现在如公式（23）所示。

▶ **命题 8**（达到时间 ϵ-准确率）。 在假设 ***A1–A5***下，存在 $\epsilon_0>0$，使得对于任意目标准确率 $\epsilon\in(0,\ \epsilon_0]$ 以及满足条件 $\eta\le\eta_{\max}(p,\ m)$ 的任意学习率，达到 ϵ-准确率所需的期望实际经过时间为：

$$\mathbb{E}^0[\tau_\epsilon]=\frac{K_\epsilon(p,m)}{\tilde{\lambda}(p,m)},\quad \textit{where:} \tag{25}$$

$K_\epsilon(p,\ m)$ 表示达到 ϵ-准确率所需的轮次数，其定义见定理 *3* 的公式 （9）。

$\lambda(p,m)$ 表示单位实际运行时间内完成的预期轮次数，定义如下：

$$\tilde{\lambda}(p,m) = \mu^{\mathrm{cs}}\,\mathbb{P}\left(\sum_{i=1}^{n} Y_i^{\mathrm{cs}} > 0\right) = \frac{W_{n,m-1}}{W_{n,m}}, \tag{26}$$

$$\frac{\partial}{\partial p_j}\,\tilde{\lambda}(p,m) = \frac{1}{p_j}\,\tilde{\lambda}(p,m) \sum_{s\in\{\mathrm{CS,c,d,u}\}} \mathbb{E}[\,\tilde{Y}_j^s - Y_j^s\,], \tag{27}$$

其中，$W_{n,m}$ 和 $W_{n,m-1}$ 如命题 *1* 中所定义，$Y \sim \phi_{n,m}$ 和 $Y \sim \phi_{n,m-1}$。

证明。 参见补充材料中的 R 节。 ◀

7.5 能量复杂度

我们通过放宽中央服务器由电网供电这一假设来扩展框架；相反，我们将其建模为能源受限边缘设备。令 $\mathcal{P}^{\mathrm{cs}}$ 表示中央服务器在本地处理期间消耗的功率，包括解密和算术更新。为简化能量核算，我们在链路级汇总通信功率成本。具体而言，$\mathcal{P}_i^{\mathrm{d}}$ 表示向客户端进行下行传输期间消耗的总功率，其中综合了中央服务器的传输功率和客户端的接收功率；而 $\mathcal{P}_i^{\mathrm{u}}$ 表示上行传输期间消耗的总功率，其中综合了客户端 i 的传输功率和中央服务器的接收功率。在此模型下，系统 $P(t)$ 的瞬时功耗为：

$$P(t) = \mathcal{P}^{\mathrm{cs}}\,\mathbf{1}\{\sum_{i=1}^{n} Y_i^{\mathrm{cs}}(t) > 0\} + \sum_{i=1}^{n} \left(\mathcal{P}_i^{\mathrm{c}}\,\mathbf{1}\{Y_i^{\mathrm{c}}(t) > 0\} + \mathcal{P}_i^{\mathrm{u}}\,Y_i^{\mathrm{u}}(t) + \mathcal{P}_i^{\mathrm{d}}\,Y_i^{\mathrm{d}}(t)\right).$$

算法的平均能量复杂度推导如下：

▶ **命题 9** (达到能量 ϵ-准确率)。 在假设 ***A1–A5*** 下，存在 $\epsilon_0 > 0$，使得对于任意目标精度 $\epsilon \in (0,\ \epsilon_0]$ 以及满足 $\eta \le \eta_{\max}(p,\ m)$ 的任意学习率，达到 ϵ-准确率所需的期望能耗为：

$$\mathbb{E}^0[E_\epsilon] = K_\epsilon(p,m)\,\frac{\mathbb{E}[P(0)]}{\tilde{\lambda}(p,m)} = K_\epsilon(p,m)\left(\frac{\mathcal{P}^{\mathrm{cs}}}{\mu^{\mathrm{cs}}} + \sum_{i=1}^{n} p_i \mathcal{E}_i\right),$$

其中 $E_i \triangleq \frac{\mathcal{P}_i^{\mathrm{c}}}{\mu_i^{\mathrm{c}}} + \frac{\mathcal{P}_i^{\mathrm{u}}}{\mu_i^{\mathrm{u}}} + \frac{\mathcal{P}_i^{\mathrm{d}}}{\mu_i^{\mathrm{d}}}$，$K_\epsilon$ 是定理 *3* 的公式 (9) 中定义的轮次复杂度，而 $\tilde{\lambda}$ 是(26)中定义的系统吞吐量。

证明。 参见补充材料的 S 节。 ◀

命题 9 展现了第 6.3 节讨论的相同能耗-延迟权衡；因此，其中得出的结论仍然适用。若要严格最小化能耗而不考虑训练延迟，则应将并发级别设为 $m = 1$，并按以下方式分配路由概率：

$$p_i^{*E} \propto \left(\sqrt{\frac{\mathcal{P}^{\mathrm{cs}}}{\mu^{\mathrm{cs}}} + \mathcal{E}_i}\right)^{-1}, \quad i \in \{1,\dots,n\}. \tag{28}$$

此配置可实现全局最小平均能耗，记为 E^* ：

$$E^* = \frac{24L\Delta}{n^2\epsilon}\left(4+\frac{B}{\epsilon}\right)\left(\sum_{i=1}^{n}\sqrt{\frac{\mathcal{P}^{\mathrm{cs}}}{\mu^{\mathrm{cs}}}+\mathcal{E}_i}\right)^2. \tag{29}$$

然而，这种策略会导致训练时长过长。因此，必须通过求解式（18）中定义的联合优化问题，在实际应用中取得平衡。

8 结论

客户端侧缓冲已成为缓解异步联邦学习中数据异质性的关键机制， `AsyncSGD` 和 `Generalized AsyncSGD`等算法便是例证。然而，现有分析大多忽视了决定这些机制实际表现的底层排队动态。为弥补这一不足，我们基于 Jackson 网络模型推导出关键性能指标的可优化闭式上界，该模型将随机网络理论与异步联邦学习理论联系起来。

不同于以往聚焦于基于轮次的收敛、假设处理时间模型具有确定性或有界，或忽视通信延迟的研究，我们提出了实际时间收敛分析，以刻画真实的边缘系统异质性。我们的模型在异构数据分布下，综合考虑随机计算、通信、中央服务器侧处理时间以及能量约束。

在此框架下，我们刻画了模型参数陈旧度与更新频率之间的权衡，并推导出关键性能指标的闭式表达式，表明孤立地优化单一目标会降低整体性能。

通过将基于梯度的优化与 Buzen 递归算法的扩展相结合，我们能够高效计算最优路由和并发参数，从而使大规模精确性能优化变得可行。

我们进一步将分析扩展到能耗，揭示了训练延迟与能效之间相辅相成的权衡关系。结果表明，常见策略（例如均匀路由）的表现可能远非最优。为此，我们提出了一种联合优化策略，通过可调参数 ρ在实际运行时间与能耗之间的帕累托前沿上进行权衡，在保持实用训练速度的同时节能高达 36%–49%。

总体而言，我们的研究结果表明，实现可持续且高效的边缘智能，需要超越迭代次数或单纯的速度指标，转向全面的、系统感知的性能目标。未来的工作例如可以将这一框架扩展到客户端参与动态变化的不可靠环境中。

参考文献

1 Alekh Agarwal 和 John C Duchi。分布式延迟随机优化。神经信息处理系统进展，24，2011。2 Abdelkrim Alahyane、Céline Comte、Matthieu Jonckheere 和 Éric Moulines。异步联邦学习优化：模型参数陈旧度与更新频率之间的微妙权衡。载于欧洲人工智能会议（*ECAI*），第 2826–2833 页。IOS Press，2025。 doi:10.3233/FAIA251139。3 Dan Alistarh、Demjan Grubic、Jerry Li、Ryota Tomioka 和 Milan Vojnovic。Qsgd：通过梯度量化与编码实现通信高效的随机梯度下降。神经信息处理系统进展，30，2017。

4 Yossi Arjevani、Yair Carmon、John C Duchi、Dylan J Foster、Nathan Srebro 和 Blake Woodworth。非凸随机优化的下界。《数学规划》，199(1)：165–214，2023。5 Yossi Arjevani、Ohad Shamir 和 Nathan Srebro。具有延迟更新的随机梯度下降的紧致收敛分析。载于《算法学习理论》，第 111–132 页。PMLR，2020。6 Rotem Zamir Aviv、Ido Hakimi、Assaf Schuster 和 Kfir Y Levy。延迟反馈下的学习：隐式适应梯度延迟。arXiv 预印本 arXiv:2106.12261，2021。7 Pierre Brémaud。《马尔可夫链：吉布斯场、蒙特卡洛模拟与排队》。应用数学丛书。Springer Cham，2020。8 Jeffrey P Buzen。具有指数分布服务器的封闭排队网络的计算算法。《ACM 通讯》，16(9)：527–531，1973。9 Sorathan Chaturapruek、John C Duchi 和 Christopher Ré。异步随机凸优化：噪声就在噪声之中，随机梯度下降并不在意。《神经信息处理系统进展》，28，2015。10 Yujing Chen、Yue Ning、Martin Slawski 和 Huzefa Rangwala。面向非独立同分布数据边缘设备的异步在线联邦学习。载于《2020 IEEE大数据国际会议》，第 15–24 页。IEEE，2020。11 Shunfeng Chu、Jun Li、Jianxin Wang、Yiyang Ni、Kang Wei、Wen Chen 和 Shi Jin。面向数字孪生赋能物联网网络的资源高效异步联邦学习。《IEEE绿色通信与网络汇刊》，2025。12 Tarin Clanuwat、Mikel Bober-Irizar、Asanobu Kitamoto、Alex Lamb、Kazuaki Yamamoto 和 David Ha。日本古典文学的深度学习。arXiv 预印本 arXiv:1812.01718，2018。13 Alon Cohen、Amit Daniely、Yoel Drori、Tomer Koren 和 Mariano Schain。对任意延迟具有鲁棒性的异步随机优化。《神经信息处理系统进展》，34：9024–9035，2021。14 Gregory Cohen、Saeed Afshar、Jonathan Tapson 和 Andre Van Schaik。EMNIST：将 MNIST 扩展至手写字母。载于《2017 年国际联合神经网络会议（IJCNN）》，第 2921–2926 页。IEEE，2017。15 Romain de Laage、Peterson Yuhala、François-Xavier Wicht、Pascal Felber、Christian Cachin 和 Valerio Schiavoni。结合密码学与可信执行环境实现实用的安全聚合。载于《ACM 分布式与事件驱动系统国际会议》，第 152–163 页，2025。16 Sanghamitra Dutta、Viveck Cadambe 和 Pulkit Grover。Short-dot：利用编码短点积分布式计算大型线性变换。《神经信息处理系统进展》，29，2016。17 Sanghamitra Dutta、Gauri Joshi、Soumyadip Ghosh、Parijat Dube 和 Priya Nagpurkar。缓慢且陈旧的梯度也能赢得竞赛：分布式随机梯度下降中的误差—运行时间权衡。载于《人工智能与统计国际会议》，第 803–812 页。PMLR，2018。18 Hamid Reza Feyzmahdavian、Arda Aytekin 和 Mikael Johansson。用于正则化随机优化的异步小批量算法。《IEEE 自动控制汇刊》，61：3740–3754，2016。19 Saeed Ghadimi 和 Guanghui Lan。用于非凸随机规划的随机一阶与零阶方法。《SIAM 优化杂志》，23(4)：2341–2368，2013。20 Sahar Ghoflsaz Ghinani 和 Elaheh Sadredini。Fusefl：完全安全且可扩展的跨机构联邦学习。arXiv 预印本 arXiv:2507.13591，2025。21 William J Gordon 和 Gordon F Newell。具有指数分布服务器的封闭排队系统。《运筹学》，15(2)：254–265，1967。22 Ala Gouissem、Zina Chkirbene 和 Ridha Hamila。联邦学习能效的全面综述：策略与挑战。载于《2024 IEEE 第八届能源会议》，第 1–6 页。IEEE，2024。

23 Robert Hannah 和 Wotao Yin。每秒迭代次数更多，质量不变——异步算法为何可能大幅胜过传统算法。arXiv预印本 arXiv:1708.05136，2017。24 Rustem Islamov、Mher Safaryan 和 Dan Alistarh。Asgrad：异步随机梯度下降算法的精确统一分析。载于《人工智能与统计国际会议》，第649–657页。PMLR，2024。25 James R Jackson。排队网络。《运筹学》，5(4):518–521，1957。26 DC John。Little。利特尔定律：值其问世50周年之际的回顾。《运筹学》，59(3):536–549，2011。27 Ahmad Faraz Khan、Yuze Li、Xinran Wang、Sabaat Haroon、Haider Ali、Yue Cheng、Ali R Butt 和 Ali Anwar。面向联邦学习的经济高效、资源感知型边缘聚合。载于《2023 IEEE大数据国际会议（BigData）》，第690–699页。IEEE，2023。28 Diederik P Kingma 和 Jimmy Ba。Adam：一种随机优化方法。arXiv预印本 arXiv:1412.6980，2014。29 Anastasiia Koloskova、Sebastian U Stich 和 Martin Jaggi。面向分布式学习和联邦学习的异步随机梯度下降：更精确的收敛保证。《神经信息处理系统进展》，35:17202–17215，2022。30 Alex Krizhevsky、Geoffrey Hinton 等。学习来自微型图像的多层特征。2009。31 Louis Leconte、Matthieu Jonckheere、Sergey Samsonov 和 Eric Moulines。异步联邦学习的排队动态。载于《人工智能与统计国际会议》，第1711–1719页。PMLR，2024。32 Kangwook Lee、Maximilian Lam、Ramtin Pedarsani、Dimitris Papailiopoulos 和 Kannan Ramchandran。利用编码加速分布式机器学习。《IEEE信息论汇刊》，64(3):1514–1529，2017。33 Qinbin Li、Yiqun Diao、Quan Chen 和 Bingsheng He。非独立同分布数据孤岛上的联邦学习：实验研究。载于《2022 IEEE第38届数据工程国际会议（ICDE）》，第965–978页。IEEE，2022。34 Xiangru Lian、Yijun Huang、Yuncheng Li 和 Ji Liu。面向非凸优化的异步并行随机梯度。《神经信息处理系统进展》，28，2015。35 Xiangru Lian、Ce Zhang、Huan Zhang、Cho-Jui Hsieh、Wei Zhang 和 Ji Liu。去中心化算法能否胜过集中式算法？去中心化并行随机梯度下降的案例研究。《神经信息处理系统进展》，30，2017。36 Paweł Lorek 和 Ryszard Szekli。通过谱隙实现收敛至平稳态的速度：存在故障与修复的排队网络。arXiv预印本 arXiv:1101.0332，2011。37 Bing Luo、Xiang Li、Shiqiang Wang、Jianwei Huang 和 Leandros Tassiulas。移动边缘网络中的经济高效联邦学习。《IEEE通信选定领域期刊》，39(12):3606–3621，2021。38 Siqi Luo、Xu Chen、Qiong Wu、Zhi Zhou 和 Shuai Yu。Hfel：面向经济高效分层联邦边缘学习的联合边缘关联与资源分配。《IEEE 无线通信汇刊》，19(10):6535–6548，2020。39 Minh Ngoc Luu、Minh-Duong Nguyen、Ebrahim Bedeer、Van Duc Nguyen、Dinh Thai Hoang、Diep N Nguyen 和 Quoc-Viet Pham。通过样本驱动控制实现联邦持续学习的能效与实时感知。《IEEE移动计算汇刊》，2025。40 Maksim Makarenko、Elnur Gasanov、Rustem Islamov、Abdurakhmon Sadiev 和 Peter Richtarik。面向通信高效分布式训练的自适应压缩。arXiv预印本 arXiv:2211.00188，2022。41 Yuyi Mao、Jun Zhang、SH Song 和 Khaled B Letaief。面向多用户移动边缘计算系统的随机联合无线电与计算资源管理。《IEEE无线通信汇刊》，16(9):5994–6009，2017。

42 Yuzhu Mao、Zihao Zhao、Guangfeng Yan、Yang Liu、Tian Lan、Linqi Song 和 Wenbo Ding。采用自适应量化的通信高效联邦学习。《ACM智能系统与技术汇刊》（TIST），13(4):1–26，2022。43 Artavazd Maranjyan、Omar Shaikh Omar 和 Peter Richtárik。Mindflayer：在工作节点计算时间异质且随机的情况下实现高效异步并行随机梯度下降。arXiv预印本 arXiv:2410.04285，2024。44 Artavazd Maranjyan、Alexander Tyurin 和 Peter Richtárik。Ringmaster异步随机梯度下降：首个具有最优时间复杂度的异步随机梯度下降算法。arXiv预印本 arXiv:2501.16168，2025。45 Brendan McMahan、Eider Moore、Daniel Ramage、Seth Hampson 和 Blaise Aguera y Arcas。利用去中心化数据进行深度网络的通信高效学习。载于《人工智能与统计》，第1273–1282页。PMLR，2017。46 Konstantin Mishchenko、Francis Bach、Mathieu Even 和 Blake E Woodworth。任意延迟下，异步随机梯度下降优于小批量随机梯度下降。《神经信息处理系统进展》，35:420–433，2022。47 Ioannis Mitliagkas、Ce Zhang、Stefan Hadjis 和 Christopher Ré。异步性带来动量，并将其应用于深度学习。载于《2016年第54届Allerton通信、控制与计算年会（Allerton）》，第997–1004页。IEEE，2016。48 John Nguyen、Kshitiz Malik、Hongyuan Zhan、Ashkan Yousefpour、Mike Rabbat、Mani Malek 和 Dzmitry Huba。采用缓冲异步聚合的联邦学习。载于《人工智能与统计国际会议》，第3581–3607页。PMLR，2022。49 Linping Qu、Shenghui Song 和 Chi-Ying Tsui。FedDQ：采用递降量化的通信高效联邦学习。载于《GLOBECOM 2022 IEEE全球通信大会》，第281–286页。IEEE，2022。50 Herbert Robbins 和 Sutton Monro。一种随机逼近方法。《数学统计年刊》，第400–407页，1951。51 Richard Serfozo。《随机网络导论》，第44卷《随机建模与应用概率》。Springer，纽约，1999。 doi:10.1007/978-1-4612-1482-3。52 Selman Sezgin、Kahina Mokrani、Julien Jacques 和 Sylvain Allio。联邦学习中的能效：关于模型、策略与展望的综述。2025。53 Suvrit Sra、Adams Wei Yu、Mu Li 和 Alex Smola。AdaDelay：延迟自适应分布式随机优化。载于《人工智能与统计》，第957–965页。PMLR，2016。54 Sebastian U Stich 和 Sai Praneeth Karimireddy。误差反馈框架：带延迟梯度的随机梯度下降。《机器学习研究期刊》，21(237):1–36，2020。55 Rajan Suri。单调性概念及其在封闭排队网络中的刻画。《运筹学》，33:606–624，1985。 doi:10.1287/opre.33.3.606。56 Alexander Tyurin、Marta Pozzi、Ivan Ilin 和 Peter Richtárik。Shadowheart随机梯度下降：在任意计算与通信异质性条件下具有最优时间复杂度的分布式异步随机梯度下降。《神经信息处理系统进展》，37:3717–3780，2024。57 Alexander Tyurin 和 Peter Richtárik。Dasha：具有通信压缩、最优预言机复杂度且无需客户端同步的分布式非凸优化。arXiv预印本 arXiv:2202.01268，2022。58 Alexander Tyurin 和 Peter Richtárik。固定计算模型下并行随机优化方法的最优时间复杂度。《神经信息处理系统进展》，36:16515–16577，2023。59 Jianyu Wang、Qinghua Liu、Hao Liang、Gauri Joshi 和 H Vincent Poor。应对异质联邦优化中的目标不一致问题。《神经信息处理系统进展》，33:7611–7623，2020。60 Cong Xie、Sanmi Koyejo 和 Indranil Gupta。异步联邦优化。arXiv预印本 arXiv:1903.03934，2019。61 Chenhao Xu、Youyang Qu、Yong Xiang 和 Longxiang Gao。异构设备上的异步联邦学习：综述。《计算机科学评论》，50:100595，2023。

62 Jiarong Yang、Yuan Liu、Fangjiong Chen、Wen Chen 和 Changle Li. 基于概率客户端选择的异步无线联邦学习。*IEEE 无线通信汇刊*，23(7):7144–7158，2023。63 Zhaohui Yang、Mingzhe Chen、Walid Saad、Choong Seon Hong 和 Mohammad Shikh-Bahaei. 无线通信网络中的节能联邦学习。*IEEE 无线通信汇刊*，20(3):1935–1949，2020。64 Mikhail Yurochkin、Mayank Agarwal、Soumya Ghosh、Kristjan Greenewald、Nghia Hoang 和 Yasaman Khazaeni. 神经网络的贝叶斯非参数联邦学习。载于国际机器学习会议，第7252–7261页。PMLR，2019。65 Jianxiang Zhao、Xiangman Li 和 Jianbing Ni. 异步联邦学习中的隐私保护模型聚合。载于 *2023*年 *IEEE/CIC* 中国通信国际会议，第1–6页。IEEE，2023。66 Xinyu Zhou、Jun Zhao、Huimei Han 和 Claude Guet. 联邦学习中能量消耗与完成时间的联合优化。载于 *2022*年第*42*届 *IEEE* 分布式计算系统国际会议 *(ICDCS)*，第1005–1017页。IEEE，2022。67 Yuncong Zuo、Bart Cox、Lydia Y Chen 和 Jérémie Decouchant. 面向地理分布式客户端的异步多服务器联邦学习。*arXiv* 预印本 *arXiv:2406.01439*，2024。

异步联邦学习中的优化权衡：随机网络方法补充材料

A 一般引理 34 B 实验细节 35 C 命题 1 的证明 36 D Buzen 递归算法 38 E 定理 2 的证明 39 F 定理 3 的证明 46 G 放宽定理 3 的有界梯度假设 (A5) 50 H 优化轮次复杂度：实验结果 53 I 命题 4 的证明 57 J 优化时间复杂度：优化结果 58 K 优化时间复杂度：补充实验 58 L 命题 5 的证明 60 M 公式 16 的证明 61 N 优化能耗复杂度：补充实验 62 O 命题 6 的证明 63 P 带 CS 端队列的 Buzen 递归算法 69 Q 定理 7 的证明 69 R 命题 8 的证明 77 S 命题 9 的证明 79

A 一般引理

我们首先陈述一些关于平稳马尔可夫链的结果，这些结果将在主要结果的证明中发挥重要作用。

A.1 离散时间马尔可夫链

▶ 引理 *10*。考虑一个具有离散状态空间 X、转移概率 $p(x,y)$、$x,y \in X$ 和不变分布 $\pi(x)$、$x \in X$ 的不可约正常返离散时间马尔可夫链 $(X_t, t \in \mathbb{N})$。过程 $((X_t, X_{t+1})\ t \in \mathbb{N})$ 也是一个不可约正常返马尔可夫链，其状态空间为 $X_2 = \{(x,y) \in X \times X : p(x,y) > 0\}$，不变分布为 $\pi_2(x,y) = \pi(x)p(x,y)$，$(x,y) \in X_2$。

证明。 留给读者验证 $((X_t, X_{t+1})\ t \in \mathbb{N})$ 确实是一个不可约马尔可夫链，其状态空间为 $\mathcal{X}_2 = \{(x,y) \in \mathcal{X} \times \mathcal{X} : \ p(x,y) > 0\}$。该马尔可夫链的转移概率为

$$\mathbb{P}\left((X_{t+1}, X_{t+2}) = (y,z) \mid (X_t, X_{t+1}) = (x,y)\right) = p(y,z), \quad t \in \mathbb{N},$$

对于每个 $x,y,z \in \mathcal{X}$，使得 $(x,y) \in \mathcal{X}_2$ 且 $(y,z) \in \mathcal{X}_2$。因此，该马尔可夫链的平衡方程为

$$\sum_{x \in \mathcal{X}:\, p(x,y)>0} \pi_2(x,y)p(y,z) = \pi_2(y,z), \quad (y,z) \in \mathcal{X}_2. \tag{30}$$

将引理正文中给出的 π_2 定义代入该平衡方程的右侧，得到

$$\sum_{x \in \mathcal{X}:\, p(x,y)>0} \pi_2(x,y)p(y,z) = \sum_{x \in \mathcal{X}} \pi(x)p(x,y)p(y,z) = \pi(y)p(y,z),$$

其中，第二个等式由应用马尔可夫链 $(X_t, t \in \mathbb{N})$ 的平衡方程得到。最后，只需注意到 $\pi(y)p(y,z) = \pi_2(y,z)$。通过写出以下表达式，即可得出 π_2 确实是一个概率分布：

$$\sum_{(x,y) \in \mathcal{X}_2} \pi_2(x,y) = \sum_{x \in \mathcal{X}} \pi(x) \sum_{y \in \mathcal{X}:p(x,y)>0} p(x,y) = \sum_{x \in \mathcal{X}} \pi(x) = 1.$$

◀

▶**推论 11.** 考虑一个不可约正常返离散时间马尔可夫链 $(X_t, t \in \mathbb{N})$，其状态空间为 X，转移概率为 $p(x,y)$，$x,y \in X$，不变分布为 $\pi(x)$，$x \in X$。另设 (X,Y) 表示一个服从引理 *10* 中定义的分布 π_2 的随机二元组。给定 $H \subseteq X \times X$ 和 $g: H \to \mathbb{R}^+$，有

$$\mathbb{E}\left[g(X,Y) | (X,Y) \in H\right] = \frac{\sum_{(x,y) \in H} g(x,y)\pi(x)p(x,y)}{\sum_{(x,y) \in H} \pi(x)p(x,y)}.$$

A.2 连续时间马尔可夫链

我们调整结果的表述，以说明初始马尔可夫链是用转移率而非转移概率定义的。

▶引理 *12.* 考虑一个不可约正常返连续时间马尔可夫链 $(X_\tau, \tau \in [0,+\infty))$，其离散状态空间为 X，转移率为 $q(x,y)$，$x,y \in X$，且不变分布为 $\pi(x)$，$x \in X$。令 $(\hat{X}_t, t \in \mathbb{N})$ 表示其嵌入离散时间马尔可夫链。过程 $((\hat{X}_t, \hat{X}_{t+1})\ t \in \mathbb{N})$ 是一个不可约正常返离散时间马尔可夫链，其状态空间为 $X_2 = \{(x,y) \in X \times X : q(x,y) > 0\}$，不变测度为$\pi_2(x,y) = \pi(x)q(x,y)$，$(x,y) \in X_2$。

证明。 注意到马尔可夫链 $(\hat{X}_t, t \in \mathbb{N})$ 的转移概率为 $q(x,y)/\sum_{z\in\mathcal{X}} q(x,z)$，$x,y \in \mathcal{X}$，不变测度为$\pi(x)\sum_{y\in\mathcal{X}} q(x,y)$，$x \in \mathcal{X}$，即可由引理 10 得出此结果。◀

▶注 13. 与第 13.3 节 [7, 中对嵌入马尔可夫链的定义不同，2],这里，连续时间马尔可夫链 $(X_\tau, \tau \in [0,+\infty))$ 及其嵌入马尔可夫链 $(\hat{X}_t, t \in \mathbb{N})$ 都可能出现从某个状态跳转回自身的情况。

▶**推论14.** 设有一个不可约正常返连续时间马尔可夫链$(X_t, t \in \mathbb{N})$，其状态空间为 X，转移率为 $q(x,y)$，$x,y \in X$，不变分布为 $\pi(x)$，$x \in X$。此外，设(X,Y) 表示一个服从引理 *10* 中所定义分布的随机二元组 π_2。给定 $H \subseteq X \times X$ 和 $g: H \to \mathbb{R}^+$，有

$$\mathbb{E}\left[g(X,Y)|(X,Y) \in H\right] = \frac{\sum_{(x,y)\in H} g(x,y)\pi(x)q(x,y)}{\sum_{(x,y)\in H} \pi(x)q(x,y)}.$$

B 实验细节

B.1 数据集和神经网络架构

我们在三个标准均衡数据集上评估所提出的方法：

- **KMNIST：** 包含 70,000 张灰度图像（28×28 像素），均匀分布于 10 个类别中，并划分为 60,000 个训练样本和 10,000 个测试样本。**EMNIST：** 包含 131,600 张灰度图像（28×28 像素），均匀分布于 47 个类别中，并划分为 112,800 个训练样本和 18,800 个测试样本。**CIFAR-100：** 包含 60,000 张 RGB 图像（32×32 像素），涵盖 100 个类别，并划分为 50,000 个训练样本和 10,000 个测试样本。

KMNIST 和 EMNIST 架构： 对于这些数据集，我们采用如下结构的卷积神经网络（CNN）：

- 两个卷积层，包含 7×7 个滤波器并采用 ReLU 激活。第一层有 20 个通道，第二层有 40 个通道。一个 2×2 最大池化层。最后是一个包含 10 个神经元并采用 Softmax 激活函数的全连接层。

CIFAR-100 架构：对于 CIFAR-100，我们采用了一个更深的 CNN 架构，包含：

- **三个连续的卷积块：** 每个卷积块由两个 3×3 卷积层组成，后接 ReLU 激活和组归一化。三个卷积块的通道深度分别为 32、64 和 128。每个卷积块最后都接一个 2×2 最大池化层和一个 Dropout 层（$p = 0.25$）。**分类头：** 该模块包括一个展平层、一个含 128 个神经元的全连接层、一个 Dropout 层（$p = 0.25$），以及一个对应 100 个类别并采用 softmax 激活的最终输出层。

所有实验均使用 PyTorch 实现，并在 NVIDIA Tesla P100 GPU 上运行。除非另有说明，每个任务的随机梯度均使用批量大小为 128 的数据进行计算。

B.2 约束优化的Softmax重参数化

为了在满足 p 的单纯形约束（即 $p_i > 0$ 和 $\sum p_i = 1$）的同时，通过梯度下降优化通用的可微路由目标函数 $h(p)$，我们采用 Softmax 重参数化。我们不直接优化 p，而是引入无约束的辅助参数$\Theta = (\theta_1, \ldots\ldots, \theta_n) \in \mathbb{R}^n$，并将路由概率定义为:

$$p_j(\Theta) = \frac{e^{\theta_j}}{\sum_{i=1}^{n} e^{\theta_i}}, \quad j \in 1, \ldots, n. \tag{31}$$

这种变换能够确保所得向量 p 在整个优化过程中始终是有效的概率分布。

因此，关于辅助参数 Θ 的梯度通过链式法则计算:

$$\frac{\partial h}{\partial \theta_j} = \left\langle \nabla_p h, \frac{\partial p}{\partial \theta_j} \right\rangle. \tag{32}$$

这里，雅可比矩阵项为 $\frac{\partial p}{\partial \theta_j} = p_j(e_j - p)$，其中 e_j 表示 $\mathbb{R}^n$ 中的标准基向量（在索引 j 处为 1，其余位置为 0），而 $\langle \cdot, \cdot \rangle$ 表示欧几里得点积。

C 命题 1 的证明

我们首先证明，连续时间过程 $\xi = (\xi(t))_{t \geq 0}$ 是一个具有平稳分布$\pi_{n,\ m}$的遍历马尔可夫链。随后，我们证明离散过程 $(X_k)_{k \in \mathbb{N}}$也是一个遍历马尔可夫链，但其平稳分布为 $\pi_{n,\ m-1}$。

C.1 连续时间过程 ξ

由于对于所有 $i \in \{1, \ldots, n\}$，路由概率满足 $p_i > 0$，因此底层路由图是强连通的，这意味着序列 $\xi(t)$ 是不可约的 [51, 命题 1.10]。此外，由于状态空间$\mathcal{X}_{n,\ m}$ 是有限的，该过程是正常返的。

根据模型假设，无穷小生成元$Q = (q(x,\ y))_{x,\ y \in \mathcal{X}_{n,\ m}}$ 的非零元素如下:

$$q(x,y) = \begin{cases} \mu_i^{\mathrm{d}}\, x_i^{\mathrm{d}}, & \text{if } y = x - \mathbf{e}_i^{\mathrm{d}} + \mathbf{e}_i^{\mathrm{c}}, \quad i \in \{1, \ldots, n\}, \\ \mu_i^{\mathrm{c}}\, \mathbf{1}(x_i^{\mathrm{c}} \geq 1), & \text{if } y = x - \mathbf{e}_i^{\mathrm{c}} + \mathbf{e}_i^{\mathrm{u}}, \quad i \in \{1, \ldots, n\}, \\ \mu_i^{\mathrm{u}}\, x_i^{\mathrm{u}}\, p_j, & \text{if } y = x - \mathbf{e}_i^{\mathrm{u}} + \mathbf{e}_j^{\mathrm{d}}, \quad i, j \in \{1, \ldots, n\}, \\ -\sum_{z \neq x} q(x,z), & \text{if } y = x, \\ 0, & \text{otherwise.} \end{cases} \tag{33}$$

这里，$\mathbf{e}_i^{\mathrm{d}}$、$\mathbf{e}_i^{\mathrm{c}}$和$\mathbf{e}_i^{\mathrm{u}} \in \{0、1\}^{3n}$ 分别表示客户端 i 的下行链路队列、计算队列和上行队列对应的标准单位向量。具体而言，如果状态的索引为

as $x = (x_1^{\mathrm{d}}, x_1^{\mathrm{c}}, x_1^{\mathrm{u}}, \ldots\ldots, x_n^{\mathrm{d}}, x_n^{\mathrm{c}}, x_n^{\mathrm{u}})$，那么 $\mathbf{e}_i^{\mathrm{d}}$在索引 $3(i-1)$ 处取值为 $1+1$，其余位置均为 0（$\mathbf{e}_i^{\mathrm{c}}$、$\mathbf{e}_i^{\mathrm{u}}$ 也同理）。

容易验证，分布 $\pi_{n,m}$ 满足全局平衡方程：

$$\pi_{n,m}(x)\sum_{y\neq x} q(x,y) = \sum_{y\neq x} \pi_{n,m}(y)q(y,x), \quad x \in \mathcal{X}_{n,m}.$$

由此可证明它是唯一的平稳分布。

C.2 离散时间过程 X

回顾一下，$(T_k)_{k\in\mathbb{N}}$ 表示服务器 $(u_i)^{i\in\{1,\ldots,n\}}$ 上发生服务完成的时刻序列。显然，事件 $\{T_k \le t\}$ 关于历史$\{\xi(s): 0 \le s \le t\}$ 可测；因此，对所有 $k \in \mathbb{N}$， T_k 都是一个停时。

根据强马尔可夫性质，序列 $(\xi(T_k))_{k\in\mathbb{N}}$ 构成一个离散时间齐次马尔可夫链，$(X_k)_{k\in\mathbb{N}}$ 也是如此。由于 ξ 不可约且状态空间有限，$(X_k)_{k\in\mathbb{N}}$ 因此不可约且正常返。接下来推导其平稳分布。

令 $(\hat{\xi}_k)^{k\in\mathbb{N}_{\ge 0}}$ 表示连续时间马尔可夫链 $(\xi(t))^{t\in\mathbb{R}_{\ge 0}}$ 的跳链。根据引理 12，以下成对序列

$$\{(\hat{\xi}_k, \hat{\xi}_{k+1}) : k \in \mathbb{N}_{\ge 0}\}$$

构成一个不可约离散时间齐次马尔可夫链，其不变测度为

$$\hat{\pi}(x,y) = \pi_{n,m}(x)\, q(x,y), \quad x, y \in \mathcal{X}_{3n,m},$$

其中， q 表示在（33）中定义的 $(\xi(t))^{t\in\mathbb{R}_{\ge 0}}$ 的无穷小生成元。对于所有 $x \in \mathcal{X}_{3n,m-1}$，$(X_k)^{k\in\mathbb{N}_{\ge 0}}$ 的平稳分布满足：

$$\mathbb{P}(X = x) = \mathbb{E}\left[\sum_{i=1}^{n}\sum_{j=1}^{n} \mathbf{1}\{\hat{\xi}_k = x + \mathbf{e}_j^{\mathrm{u}},\, \hat{\xi}_{k+1} = x + \mathbf{e}_i^{\mathrm{d}}\} \,\middle|\, (\hat{\xi}_k, \hat{\xi}_{k+1}) \in H\right], \tag{34}$$

其中，有效转移集合 H 定义为

$$H = \{(y,z) \in \mathcal{X}_{3n,m} \times \mathcal{X}_{3n,m} : z = y - \mathbf{e}_i^{\mathrm{u}} + \mathbf{e}_j^{\mathrm{d}} \text{ for some } i,j \in \{1,\ldots,n\}\}.$$

将推论14应用于方程（34），其中

$$g(y,z) = \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbf{1}\{y = x + \mathbf{e}_j^{\mathrm{u}},\, z = x + \mathbf{e}_i^{\mathrm{d}}\},$$

对于每个 $x \in \mathcal{X}_{3n,\ m-1}$，平稳分布可表示为

$$\mathbb{P}(X = x) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} \pi_{n,m}(x + \mathbf{e}_j^{\mathrm{u}})\, \mu_j^{\mathrm{u}}\, (x_j^{\mathrm{u}} + 1)\, p_i}{\sum_{y\in\mathcal{X}_{n,m}}\sum_{i=1}^{n}\sum_{j=1}^{n} \pi_{n,m}(y)\, \mu_j^{\mathrm{u}}\, y_j^{\mathrm{u}}\, p_i} \tag{35}$$

$$= \frac{\sum_{j=1}^{n} \pi_{n,m}(x + \mathbf{e}_j^{\mathrm{u}})\, \mu_j^{\mathrm{u}}\, (x_j^{\mathrm{u}} + 1)}{\sum_{j=1}^{n} \mu_j^{\mathrm{u}}\, \mathbb{E}[\xi_j^{\mathrm{u}}]}. \tag{36}$$

我们分别化简分子和分母。对于分母，注意到对于每个 $j \in \{1, \dots, n\}$，

$$
\begin{aligned}
\mathbb{E}[\xi_j^{\mathrm{u}}] &= \sum_{x \in \mathcal{X}_{n,m}} x_j^{\mathrm{u}}\, \pi_{n,m}(x) \\
&= \frac{1}{Z_{n,m}} \sum_{x \in \mathcal{X}_{n,m}} x_j^{\mathrm{u}} \prod_{i=1}^{n} \left(\frac{p_i}{\mu_i^{\mathrm{c}}}\right)^{x_i^{\mathrm{c}}} \frac{1}{x_i^{\mathrm{d}}!} \left(\frac{p_i}{\mu_i^{\mathrm{d}}}\right)^{x_i^{\mathrm{d}}} \frac{1}{x_i^{\mathrm{u}}!} \left(\frac{p_i}{\mu_i^{\mathrm{u}}}\right)^{x_i^{\mathrm{u}}} \\
&= \frac{1}{Z_{n,m}} \frac{p_j}{\mu_j^{\mathrm{u}}} \underbrace{\sum_{\substack{x \in \mathcal{X}_{n,m} \\ x_j^{\mathrm{u}} \geq 1}} \prod_{i=1}^{n} \left(\frac{p_i}{\mu_i^{\mathrm{c}}}\right)^{x_i^{\mathrm{c}}} \frac{1}{x_i^{\mathrm{d}}!} \left(\frac{p_i}{\mu_i^{\mathrm{d}}}\right)^{x_i^{\mathrm{d}}} \frac{1}{(x_i^{\mathrm{u}} - \mathbf{1}\{i=j\})!} \left(\frac{p_i}{\mu_i^{\mathrm{u}}}\right)^{x_i^{\mathrm{u}} - \mathbf{1}\{i=j\}}}_{= Z_{n,m-1}} \\
&= \frac{p_j}{\mu_j^{\mathrm{u}}} \frac{Z_{n,m-1}}{Z_{n,m}}.
\end{aligned}
$$

第二个等号由方程（2）中$\pi_{n,m}$ 的定义得出，第三个等号通过提取公因子 $\frac{p_j}{\mu_j^{\mathrm{u}}}$ 得出，最后一个等号则通过进行变量替换 $y = x - \mathbf{e}_j^{\mathrm{u}}$，并结合归一化常数 $Z_{n,m-1}$ 的定义得出。注意，当 x 遍历$\{x \in \mathcal{X}_{n,m} : x_j^{\mathrm{u}} \geq 1\}$时，变量 $y = x - \mathbf{e}_j^{\mathrm{u}}$遍历整个状态空间 $\mathcal{X}_{n,m-1}$。因此，对所有 j 求和后，分母变为：

$$
\sum_{j=1}^{n} \mu_j^{\mathrm{u}}\, \mathbb{E}[\xi_j^{\mathrm{u}}] = \sum_{j=1}^{n} \mu_j^{\mathrm{u}} \frac{Z_{n,m-1}}{Z_{n,m}} \frac{p_j}{\mu_j^{\mathrm{u}}} = \frac{Z_{n,m-1}}{Z_{n,m}}. \tag{37}
$$

对于分子，我们针对每个 $j \in \{1, \dots, n\}$进行如下计算：

$$
\begin{aligned}
&\pi_{n,m}(x + \mathbf{e}_j^{\mathrm{u}})\, \mu_j^{\mathrm{u}}\, (x_j^{\mathrm{u}} + 1) \\
&= \frac{\mu_j^{\mathrm{u}}(x_j^{\mathrm{u}} + 1)}{Z_{n,m}} \prod_{i=1}^{n} \left(\frac{p_i}{\mu_i^{\mathrm{c}}}\right)^{x_i^{\mathrm{c}}} \frac{1}{x_i^{\mathrm{d}}!} \left(\frac{p_i}{\mu_i^{\mathrm{d}}}\right)^{x_i^{\mathrm{d}}} \frac{1}{(x_i^{\mathrm{u}} + \mathbf{1}\{i=j\})!} \left(\frac{p_i}{\mu_i^{\mathrm{u}}}\right)^{x_i^{\mathrm{u}} + \mathbf{1}\{i=j\}} \\
&= \frac{\mu_j^{\mathrm{u}}}{Z_{n,m}} \cdot \frac{p_j}{\mu_j^{\mathrm{u}}} \underbrace{\prod_{i=1}^{n} \left(\frac{p_i}{\mu_i^{\mathrm{c}}}\right)^{x_i^{\mathrm{c}}} \frac{1}{x_i^{\mathrm{d}}!} \left(\frac{p_i}{\mu_i^{\mathrm{d}}}\right)^{x_i^{\mathrm{d}}} \frac{1}{x_i^{\mathrm{u}}!} \left(\frac{p_i}{\mu_i^{\mathrm{u}}}\right)^{x_i^{\mathrm{u}}}}_{= Z_{n,m-1}\, \pi_{n,m-1}(x)} \\
&= \frac{Z_{n,m-1}}{Z_{n,m}}\, p_j\, \pi_{n,m-1}(x).
\end{aligned}
$$

因此，对 j 求和可得：

$$
\sum_{j=1}^{n} \pi_{n,m}(x + \mathbf{e}_j^{\mathrm{u}})\, \mu_j^{\mathrm{u}}\, (x_j^{\mathrm{u}} + 1) = \frac{Z_{n,m-1}}{Z_{n,m}}\, \pi_{n,m-1}(x). \tag{38}
$$

最后，将 (37) 和 (38) 代回 (36)，可得：

$\mathbb{P}(X = x) = \pi_{n,m-1}(x)$, 对每个 $x \in \mathcal{X}_{n,m-1}$.

D Buzen 递归算法

第 2.6 节中为该模型推导出的所有性能指标，都依赖于平稳分布$\phi_{n,\,\mathrm{m}}$ 主要通过其归一化常数 $Z_{n,\,\mathrm{m}}$。直接计算

这些常数需要枚举全部可容许状态，共有 $\binom{n+m-1}{m-1}$ 个；其数量随 n 和 m 呈组合式增长，很快便变得难以处理。

为克服这一局限，我们对 Buzen 递归算法进行调整 [8], ，使其能够在$\mathcal{O}(nm^2)$ 时间内完成归一化常数的计算，且仅需$\mathcal{O}(m)$ 的内存。此外，递归结构使得所有常数 $Z_{n,\,\mathfrak{m}}$（其中 $\mathfrak{m} \in \{0,\ 1,\ \ldots\ldots,\ m\}$）都能在一次遍历中同时计算出来。所得算法在以下命题中形式化给出。

▶ **命题 15**。 对于所有 $\mathfrak{m} \in \{0, 1, \ldots, m\}$，归一化常数 $Z_{n,\,\mathfrak{m}}$ 可通过 *Buzen* 递归算法在 $O(nm^2)$ 时间内计算，且需要 $O(m)$ 内存。特别地，对于所有 $\mathfrak{m} \in \{0, 1, \ldots, m\}$，$Z_{n,\,\mathfrak{m}} = U_{3n,\,\mathfrak{m}}$，其中各量 $U_{\mathfrak{n},\,\mathfrak{m}}$ 按以下方式递归定义：

- $U_{\mathfrak{n},0} = 1,$ $\qquad$ *for* $\mathfrak{n} \in \{1, \ldots, 3n\}.$
- $U_{1,\mathfrak{m}} = \left(\dfrac{p_1}{\mu_1^{\mathrm{c}}}\right)^{\mathfrak{m}},$ $\qquad$ *for* $\mathfrak{m} \in \{0, \ldots, m\}.$
- $U_{\mathfrak{n},\mathfrak{m}} = U_{\mathfrak{n}-1,\mathfrak{m}} + \dfrac{p_{\mathfrak{n}}}{\mu_{\mathfrak{n}}^{\mathrm{c}}}\, U_{\mathfrak{n},\mathfrak{m}-1},$ $\qquad$ *for* $\mathfrak{n} \in \{2, \ldots, n\}$ *and* $\mathfrak{m} \in \{1, \ldots, m\}.$
- $U_{\mathfrak{n},\mathfrak{m}} = \displaystyle\sum_{k=0}^{\mathfrak{m}} \frac{1}{k!} \left(\frac{p_{\mathfrak{n}-n}}{\mu_{\mathfrak{n}-n}^{\mathrm{d}}}\right)^k U_{\mathfrak{n}-1,\mathfrak{m}-k},$ $\qquad$ *for* $\mathfrak{n} \in \{n+1, \ldots, 2n\}$ *and* $\mathfrak{m} \in \{1, \ldots, m\}.$
- $U_{\mathfrak{n},\mathfrak{m}} = \displaystyle\sum_{k=0}^{\mathfrak{m}} \frac{1}{k!} \left(\frac{p_{\mathfrak{n}-2n}}{\mu_{\mathfrak{n}-2n}^{\mathrm{u}}}\right)^k U_{\mathfrak{n}-1,\mathfrak{m}-k},$ $\qquad$ *for* $\mathfrak{n} \in \{2n+1, \ldots, 3n\}$ *and* $\mathfrak{m} \in \{1, \ldots, m\}.$

E 定理2的证明

我们依次证明该定理中的每个等式。

E.1 公式（3）的证明

固定 $i \in \{1, \ldots, n\}$ 和 $k \in \mathbb{N}$。在稳态下，我们有

$$\mathbb{E}^0[D_i] = \mathbb{E}^0[D_{i,k}] = \sum_{j=1}^{n} \mathbb{E}^0[D_{i,k} \mid A_k = j]\, \mathbb{P}^0(A_k = j) = p_i\, \mathbb{E}^0[D_{i,k} \mid A_k = i] = p_i\, \mathbb{E}^0[R_i],$$

其中，$R_{i,l}$ 表示从中央服务器向客户端 i 发送第 l 个任务，到相应梯度被应用之间发生的模型参数更新次数。这对应于该任务在由服务器 d_i、c_i 和 u_i 组成的子系统中的逗留时间（以“任务数”计）。我们用 $\mathbb{E}^0[R_i]$ 表示在平稳性下 $R_{i,l}$的期望。

因此，证明公式（3）等价于证明，对于每个 $i \in \{1, \ldots, n\}$，

$$\mathbb{E}^0[R_i] = \frac{\mathbb{E}[X_i^{\mathrm{d}} + X_i^{\mathrm{c}} + X_i^{\mathrm{u}}]}{p_i}, \tag{39}$$

这可以解释为在上行队列 $\mathsf{u}_{i=1}^n$ 的服务完成离散时刻应用的利特尔定律。

步骤 1：均匀化构造。
为方便技术处理，我们引入一个辅助的连续时间系统，称为均匀（连续时间）系统。在该系统中，任意两次连续的模型参数更新（轮次）之间的时间服从均值为 1 的指数分布。

尽管这一假设引入了一个虚构的时间尺度，但它使我们能够应用经典连续时间利特尔法则。随后，我们会将结果映射回原始的离散时间过程 $(X_k)_{k\in\mathbb{N}}$。

回顾一下，$(T_k)_{k\in\mathbb{N}}$ 表示服务器 $(u_i)_{i\in\{1,\dots,n\}}$ 上发生服务完成的时刻序列。如附录 C 所示，$(\xi(T_k))_{k\in\mathbb{N}}$ 构成一个遍历的离散时间齐次马尔可夫链。我们通过将 $(\xi(T_k))_{k\in\mathbb{N}}$ 从属于速率为 1 的泊松过程 $(N(t))_{t\geq 0}$，构造出均匀化连续时间马尔可夫链 $(\bar{\xi}(t))_{t\in\mathbb{R}_{\geq 0}}$：

$$\bar{\xi}(t) = \xi(T_{N(t)}), \quad t \geq 0.$$

步骤 2：到达与逗留时间。

对于每个 i，令 $T_{i,k}$ 表示分配给客户端 i 的第 k 个任务在均匀系统中的到达时间，并令 $N_i(t)$ 表示相应的计数过程。

在这个简化模型中，$(T_{i,k})_{k\in\mathbb{N}}$ 构成一个速率为 p_i 的齐次泊松点过程。事实上，由于轮次时长独立同分布且服从均值为 1 的指数分布，中央服务器发出的全局任务流构成一个速率为 1 的泊松过程。根据泊松稀疏化的性质，由于每个任务以 p_i 的概率独立地路由至客户端 i，所得过程是一个速率为 p_i 的泊松过程。

分配给客户端 i 的第 k 个任务（依次经过服务器 d_i、c_i、u_i）的逗留时间为

$$\bar{R}_{i,k} = \sum_{l=1}^{R_{i,k}+1} E_l,$$

其中，$(E_l)_{l\in\mathbb{N}}$ 是均值为 1 的独立同分布的指数随机变量。给定这些序列，在时间 t，客户端 i 的子系统（由服务器 d_i、c_i 和 u_i 组成）中的任务总数为：

$$\bar{\xi}_i^{\mathrm{d}}(t) + \bar{\xi}_i^{\mathrm{c}}(t) + \bar{\xi}_i^{\mathrm{u}}(t) = \sum_{k\in\mathbb{N}} \mathbf{1}\{\bar{T}_{i,k} \leq t < \bar{T}_{i,k} + \bar{R}_{i,k}\}. \tag{40}$$

步骤 3：应用利特尔定律。

所构造的系统满足利特尔定律 [51, 定理 5.2] 的标准条件：$(\bar{\xi}(t))_{t\geq 0}$ 是遍历马尔可夫链；(ii) 到达过程 $(T_{i,k})_{k\in\mathbb{N}}$ 是泊松过程（因此是平稳且简单的）；(iii) 逗留时间 $R_{i,k}$ 仅取决于过程 $(\bar{\xi}(t))_{t\geq T_{i,k}}$，以及 (iv) $\mathbb{E}[\bar{\xi}_i^{\mathrm{d}} + \bar{\xi}_i^{\mathrm{c}} + \bar{\xi}_i^{\mathrm{u}}] < \infty$ 和 $\mathbb{E}[N_i(1)] = p_i$。

因此，公式（40）得出：

$$\mathbb{E}[\bar{\xi}_i^{\mathrm{d}} + \bar{\xi}_i^{\mathrm{c}} + \bar{\xi}_i^{\mathrm{u}}] = p_i\, \mathbb{E}^0[\bar{R}_i]. \tag{41}$$

步骤 4：映射回离散模型。

根据均匀化的性质，ξ 的平稳分布与 $(\bar{\xi}(T_k))_{k\in\mathbb{N}}$（从属链）的平稳分布相同。因此，对于每个分量 $s \in \{\mathsf{d}, \mathsf{c}, \mathsf{u}\}$ 和客户端 i，$\mathbb{E}[\xi_i^s] = \mathbb{E}[\xi_i^s(T_k)]$。此外，由于变量 $(E_l)_{l\in\mathbb{N}}$ 是均值为 1 的独立同分布变量，且与 R_i 独立，根据沃尔德恒等式可得 $\mathbb{E}[R_i] = \mathbb{E}[R_i + 1]$。将这些代入公式（41）：

$$\mathbb{E}[\xi_i^{\mathrm{d}}(T_k) + \xi_i^{\mathrm{c}}(T_k) + \xi_i^{\mathrm{u}}(T_k)] = p_i\, \mathbb{E}^0[R_i + 1] = p_i\, \mathbb{E}^0[R_i] + p_i.$$

利用 $\mathbb{E}[\mathbf{1}\{A_k = i\}] = p_i$ 这一事实，我们将其重新整理为：

$$\mathbb{E}[\xi_i^{\mathrm{d}}(T_k) - \mathbf{1}\{A_k = i\} + \xi_i^{\mathrm{c}}(T_k) + \xi_i^{\mathrm{u}}(T_k)] = p_i\, \mathbb{E}^0[R_i].$$

最后，回顾离散状态的定义 X （即新任务 (1) 派发之前 参数更新时刻的状态），可得关系式：$X_k^d{}_{i=k}+X_{i,}^c$ $_k+X_i^u{}_{,}$ $\xi_i^{\mathrm{d}}\ T_k\ -\mathbf{1}\ A_k=i\ +\xi_i^{\mathrm{c}}\ T_k\ +\xi_i^{\mathrm{u}}\ T_k$ ( ) { } ( ) ( )。取期望可得：

$$\mathbb{E}[X_i^{\mathrm{d}}+X_i^{\mathrm{c}}+X_i^{\mathrm{u}}]=p_i\,\mathbb{E}^0[R_i],$$

这恰好就是（39）。至此，公式（3）的证明完成。

E.2 方程（4）的证明

令 $i,j\in\{1,2,\dots,n\}$。我们的目标是证明（4）。根据（3）以及协方差的**双线性**，这等价于证明：

$$\frac{\partial\mathbb{E}[X_i^{\mathrm{d}}+X_i^{\mathrm{c}}+X_i^{\mathrm{u}}]}{\partial(\log p_j)}=\mathrm{Cov}[X_i^{\mathrm{d}}+X_i^{\mathrm{c}}+X_i^{\mathrm{u}},\ X_j^{\mathrm{d}}+X_j^{\mathrm{c}}+X_j^{\mathrm{u}}].$$

回忆一下，随机向量 X 服从平稳分布 $\pi_{n,m-1}$ （见方程（2）），我们可以将其改写为

$$\log\pi_{n,m-1}(x)=-\log Z_{n,m-1}+\sum_{i=1}^{n}\left(x_i^{\mathrm{d}}+x_i^{\mathrm{c}}+x_i^{\mathrm{u}}\right)\log p_i \\ -\log\left((\mu_i^{\mathrm{c}})^{x_i^{\mathrm{c}}}(\mu_i^{\mathrm{d}})^{x_i^{\mathrm{d}}}x_i^{\mathrm{d}}!(\mu_i^{\mathrm{u}})^{x_i^{\mathrm{u}}}x_i^{\mathrm{u}}!\right),\quad x\in\mathcal{X}_{3n,m-1},\quad (42)$$

其中 $Z_{n,m-1}$ 由归一化得到：

$$Z_{n,m-1}=\sum_{x\in\mathcal{X}_{n,m-1}}\exp\left(\sum_{i=1}^{n}\left(x_i^{\mathrm{d}}+x_i^{\mathrm{c}}+x_i^{\mathrm{u}}\right)\log p_i-\log\left((\mu_i^{\mathrm{c}})^{x_i^{\mathrm{c}}}(\mu_i^{\mathrm{d}})^{x_i^{\mathrm{d}}}x_i^{\mathrm{d}}!(\mu_i^{\mathrm{u}})^{x_i^{\mathrm{u}}}x_i^{\mathrm{u}}!\right)\right).\quad (43)$$

我们先证明以下中间结论：

$$\frac{\partial\log Z_{n,m-1}}{\partial(\log p_j)}=\mathbb{E}[X_j^{\mathrm{d}}+X_j^{\mathrm{c}}+X_j^{\mathrm{u}}],\\ \frac{\partial\log\pi_{n,m-1}(x)}{\partial(\log p_j)}=\left(x_j^{\mathrm{d}}+x_j^{\mathrm{c}}+x_j^{\mathrm{u}}\right)-\mathbb{E}[X_j^{\mathrm{d}}+X_j^{\mathrm{c}}+X_j^{\mathrm{u}}],\quad x\in\mathcal{X}_{3n,m-1}.\quad (44)$$

（44）的第一部分可通过对（43）求偏导数并重新排列各项，以得出 $\pi_{n,\ m-1}$ 的定义：

$$\begin{aligned}&\frac{\partial\log(Z_{n,m-1})}{\partial(\log p_j)}\\&=\frac{1}{Z_{n,m-1}}\frac{\partial Z_{n,m-1}}{\partial(\log p_j)},\\&=\frac{1}{Z_{n,m-1}}\sum_{x\in\mathcal{X}_{n,m-1}}\left(x_j^{\mathrm{d}}+x_j^{\mathrm{c}}+x_j^{\mathrm{u}}\right)\exp\left(\sum_{i=1}^{n}\left(x_j^{\mathrm{d}}+x_j^{\mathrm{c}}+x_j^{\mathrm{u}}\right)\log p_i-\log\left((\mu_i^{\mathrm{c}})^{x_i^{\mathrm{c}}}(\mu_i^{\mathrm{d}})^{x_i^{\mathrm{d}}}x_i^{\mathrm{d}}!(\mu_i^{\mathrm{u}})^{x_i^{\mathrm{u}}}x_i^{\mathrm{u}}!\right)\right),\\&=\sum_{x\in\mathcal{X}_{n,m-1}}\left(x_j^{\mathrm{d}}+x_j^{\mathrm{c}}+x_j^{\mathrm{u}}\right)\exp\left(\sum_{i=1}^{n}\left(x_j^{\mathrm{d}}+x_j^{\mathrm{c}}+x_j^{\mathrm{u}}\right)\log p_i-\log\left((\mu_i^{\mathrm{c}})^{x_i^{\mathrm{c}}}(\mu_i^{\mathrm{d}})^{x_i^{\mathrm{d}}}x_i^{\mathrm{d}}!(\mu_i^{\mathrm{u}})^{x_i^{\mathrm{u}}}x_i^{\mathrm{u}}!\right)-\log Z_{n,m-1}\right),\\&=\sum_{x\in\mathcal{X}_{n,m-1}}\left(x_j^{\mathrm{d}}+x_j^{\mathrm{c}}+x_j^{\mathrm{u}}\right)\pi_{n,m-1}(x),\end{aligned}$$

$$= \mathbb{E}[X_j^{\mathrm{d}} + X_j^{\mathrm{c}} + X_j^{\mathrm{u}}].$$

现在，（44）的第二部分可通过对（42）求偏导数并代入前面的结果得出：

$$\frac{\partial \log \pi_{n,m-1}(x)}{\partial(\log p_j)} = x_j^{\mathrm{d}} + x_j^{\mathrm{c}} + x_j^{\mathrm{u}} - \frac{\partial \log Z_{n,m-1}}{\partial(\log p_j)} = x_j^{\mathrm{d}} + x_j^{\mathrm{c}} + x_j^{\mathrm{u}} - \mathbb{E}[X_j^{\mathrm{d}} + X_j^{\mathrm{c}} + X_j^{\mathrm{u}}], \quad x \in \mathcal{X}_{3n,m-1}.$$

总之，只需将（44）的第二部分代入期望的定义即可：

$$\begin{aligned}
&\frac{\partial \mathbb{E}[X_i^{\mathrm{d}} + X_i^{\mathrm{c}} + X_i^{\mathrm{u}}]}{\partial(\log p_j)} \\
&= \sum_{x \in \mathcal{X}_{3n,m-1}} \left(x_i^{\mathrm{c}} + x_i^{\mathrm{d}} + x_i^{\mathrm{u}}\right) \frac{\partial \pi_{n,m-1}(x)}{\partial(\log p_j)}, \\
&= \sum_{x \in \mathcal{X}_{3n,m-1}} \pi_{n,m-1}(x) \left(x_i^{\mathrm{c}} + x_i^{\mathrm{d}} + x_i^{\mathrm{u}}\right) \frac{\partial \log \pi_{n,m-1}(x)}{\partial(\log p_j)}, \\
&= \sum_{x \in \mathcal{X}_{3n,m-1}} \pi_{n,m-1}(x) \left(x_i^{\mathrm{c}} + x_i^{\mathrm{d}} + x_i^{\mathrm{u}}\right) \left(x_j^{\mathrm{d}} + x_j^{\mathrm{c}} + x_j^{\mathrm{u}} - \mathbb{E}[X_j^{\mathrm{d}} + X_j^{\mathrm{c}} + X_j^{\mathrm{u}}]\right), \\
&= \mathrm{Cov}[X_i^{\mathrm{d}} + X_i^{\mathrm{c}} + X_i^{\mathrm{u}}, X_j^{\mathrm{d}} + X_j^{\mathrm{c}} + X_j^{\mathrm{u}}].
\end{aligned}$$

E.3 公式（5）的证明

为推导公式（5），我们明确计算任意客户端 $i \in \{1, \ldots, n\}$ 的期望 $\mathbb{E}[X_i^{\mathrm{c}}]$、$\mathbb{E}[X_i^{\mathrm{d}}]$ 和 $\mathbb{E}[X_i^{\mathrm{u}}]$，然后将它们相加，得到最终结果。

我们首先计算计算队列中的任务期望数量 $\mathbb{E}[X_i^{\mathrm{c}}]$。利用非负整数值随机变量期望的尾和公式，我们有 $\mathbb{E}[X_i^{\mathrm{c}}] = \sum_{k=1}^{m-1} \mathbb{P}(X_i^{\mathrm{c}} \geq k)$。首先，对于任意 $k \in \{0, \ldots, m-1\}$，我们通过对客户端 i 至少有 k 个计算任务的所有状态的平稳分布 $\pi_{n,m-1}$ 求和，推导出概率 $\mathbb{P}(X_i^{\mathrm{c}} \geq k)$ 的表达式：

$$\begin{aligned}
\mathbb{P}(X_i^{\mathrm{c}} \geq k) &= \sum_{\substack{x \in \mathcal{X}_{n,m-1} \\ x_i^{\mathrm{c}} \geq k}} \pi_{n,m-1}(x) \\
&= \frac{1}{Z_{n,m-1}} \sum_{\substack{x \in \mathcal{X}_{n,m-1} \\ x_i^{\mathrm{c}} \geq k}} \left[\left(\frac{p_i}{\mu_i^{\mathrm{c}}}\right)^{x_i^{\mathrm{c}}} \prod_{j \neq i} \left(\frac{p_j}{\mu_j^{\mathrm{c}}}\right)^{x_j^{\mathrm{c}}}\right] \prod_{l=1}^{n} \frac{1}{x_l^{\mathrm{d}}!} \left(\frac{p_l}{\mu_l^{\mathrm{d}}}\right)^{x_l^{\mathrm{d}}} \frac{1}{x_l^{\mathrm{u}}!} \left(\frac{p_l}{\mu_l^{\mathrm{u}}}\right)^{x_l^{\mathrm{u}}} \\
&= \frac{1}{Z_{n,m-1}} \left(\frac{p_i}{\mu_i^{\mathrm{c}}}\right)^{k} \underbrace{\sum_{\substack{x \in \mathcal{X}_{n,m-1} \\ x_i^{\mathrm{c}} \geq k}} \left(\frac{p_i}{\mu_i^{\mathrm{c}}}\right)^{x_i^{\mathrm{c}}-k} \left[\prod_{j \neq i} \left(\frac{p_j}{\mu_j^{\mathrm{c}}}\right)^{x_j^{\mathrm{c}}}\right] \prod_{l=1}^{n} \frac{1}{x_l^{\mathrm{d}}!} \left(\frac{p_l}{\mu_l^{\mathrm{d}}}\right)^{x_l^{\mathrm{d}}} \frac{1}{x_l^{\mathrm{u}}!} \left(\frac{p_l}{\mu_l^{\mathrm{u}}}\right)^{x_l^{\mathrm{u}}}}_{= Z_{n,m-1-k}} \\
&= \left(\frac{p_i}{\mu_i^{\mathrm{c}}}\right)^{k} \frac{Z_{n,m-1-k}}{Z_{n,m-1}}.
\end{aligned} \tag{45}$$

这里，第二个等号根据方程（2）中$\pi_{n,\ m-1}$ 的定义得出，而第三个等号则通过提出 $(p_i/\mu_i^{\mathrm{c}})^{k}$ 得到。通过进行变量替换 $y = x - k\mathbf{e}_i^{\mathrm{c}}$，剩余的求和可识别为归一化常数 $Z_{n,\ m-1-k}$。回想一下，$\mathbf{e}_i^{\mathrm{d}}$，$\mathbf{e}_i^{\mathrm{c}}$，$\mathbf{e}_i^{\mathrm{u}}$表示状态向量中与分量 x_i^{d}，x_i^{c}，x_i^{u}对应的标准单位向量

状态向量 x。我们注意到，当 x 遍历$\{x \in \mathcal{X}_{n,\ m-1} : x_i^{\mathrm{c}} \geq k\}$时，平移后的变量$y$ 与整个状态空间$\mathcal{X}_{n,\ m-1-k}$之间构成双射。对 k求和，即可得到 $\beta_{i,\ 1}$ 的定义：

$$\mathbb{E}[X_i^{\mathrm{c}}] = \sum_{k=1}^{m-1} \left(\frac{p_i}{\mu_i^{\mathrm{c}}}\right)^k \frac{Z_{n,m-1-k}}{Z_{n,m-1}} = \beta_{i,1}. \tag{46}$$

接下来，我们直接根据期望的定义计算下行链路时延 $\mathbb{E}[X_i^{\mathrm{d}}]$ ：

$$\begin{aligned}
\mathbb{E}[X_i^{\mathrm{d}}] &= \sum_{x \in \mathcal{X}_{n,m-1}} x_i^{\mathrm{d}}\, \pi_{n,m-1}(x) \\
&= \frac{1}{Z_{n,m-1}} \sum_{x \in \mathcal{X}_{n,m-1}} x_i^{\mathrm{d}} \prod_{j=1}^{n} \left(\frac{p_j}{\mu_j^{\mathrm{c}}}\right)^{x_j^{\mathrm{c}}} \frac{1}{x_j^{\mathrm{d}}!} \left(\frac{p_j}{\mu_j^{\mathrm{d}}}\right)^{x_j^{\mathrm{d}}} \frac{1}{x_j^{\mathrm{d}}!} \left(\frac{p_j}{\mu_j^{\mathrm{u}}}\right)^{x_j^{\mathrm{d}}} \\
&= \frac{1}{Z_{n,m-1}} \frac{p_i}{\mu_i^{\mathrm{d}}} \underbrace{\sum_{\substack{x \in \mathcal{X}_{n,m-1} \\ x_i^{\mathrm{d}} \geq 1}} \prod_{j=1}^{n} \left(\frac{p_j}{\mu_j^{\mathrm{c}}}\right)^{x_j^{\mathrm{c}}} \frac{1}{(x_j^{\mathrm{d}} - \mathbf{1}\{j=i\})!} \left(\frac{p_j}{\mu_j^{\mathrm{u}}}\right)^{x_j^{\mathrm{d}} - \mathbf{1}\{j=i\}} \frac{1}{x_j^{\mathrm{u}}!} \left(\frac{p_j}{\mu_j^{\mathrm{u}}}\right)^{x_j^{\mathrm{u}}}}_{= Z_{n,m-2}} \\
&= \frac{p_i}{\mu_i^{\mathrm{d}}} \frac{Z_{n,m-2}}{Z_{n,m-1}}.
\end{aligned} \tag{47}$$

第三步的化简用到了恒等式 $x_i^{\mathrm{d}} \cdot \frac{1}{x_i^{\mathrm{d}}!} = \frac{1}{(x_i^{\mathrm{d}}-1)!}$，并提取出因子 p_i/μ_i^{d}。剩余的求和对应于总体规模为 $m-1-1 = m-2$ 的系统的归一化常数，这通过变量替换 $y = x - \mathbf{e}_i^{\mathrm{d}}$ 得到。

由于乘积形式解中上行链路与下行链路过程具有对称性，上行链路时延 $\mathbb{E}[X_i^{\mathrm{u}}]$ 的推导与 X_i^{d} 完全相同。因此，直接可得：

$$\mathbb{E}[X_i^{\mathrm{u}}] = \frac{p_i}{\mu_i^{\mathrm{u}}} \frac{Z_{n,m-2}}{Z_{n,m-1}}. \tag{48}$$

最后，将（46）、（47）和（48）中推导出的各项相加，得到：

$$\sum_{s \in \{\mathrm{c},\mathrm{d},\mathrm{u}\}} \mathbb{E}[X_i^s] = \beta_{i,1} + \left(\frac{p_i}{\mu_i^{\mathrm{d}}} + \frac{p_i}{\mu_i^{\mathrm{u}}}\right) \frac{Z_{n,m-2}}{Z_{n,m-1}} = \beta_{i,1} + \frac{Z_{n,m-2}}{Z_{n,m-1}} \gamma_i,$$

其中，我们代入了 $\gamma_i = p_i(1/\mu_i^{\mathrm{d}} + 1/\mu_i^{\mathrm{u}})$。至此，公式（5）的证明完成。

E.4 公式（6）的证明

为证明公式（6），我们将左侧的求和分解为三类不同的交互：队列-队列相关性、队列-延迟相关性和延迟-延迟相关性。我们分别推导每一类的闭式表达式，然后将它们合并。

E.4.1 队列-队列相关性（$\mathbb{E}[X_i^{\mathrm{c}} X_j^{\mathrm{c}}]$）

我们首先计算任意一对客户端 i, j的队列长度 $\mathbb{E}[X_i^{\mathrm{c}} X_j^{\mathrm{c}}]$ 的联合期望。先考虑两个客户端不同的情况（$i \neq j$）。利用恒等式

$x_i^{\mathrm{c}} x_j^{\mathrm{c}} = \sum_{k=1}^{x_i^{\mathrm{c}}} \sum_{\ell=1}^{x_j^{\mathrm{c}}} 1$，我们可以将期望改写为尾概率之和：

$$\begin{aligned}\mathbb{E}[X_i^{\mathrm{c}} X_j^{\mathrm{c}}] &= \sum_{x \in \mathcal{X}_{n,m-1}} x_i^{\mathrm{c}} x_j^{\mathrm{c}} \pi_{n,m-1}(x) \\ &= \sum_{x \in \mathcal{X}_{n,m-1}} \sum_{k=1}^{x_i^{\mathrm{c}}} \sum_{\ell=1}^{x_j^{\mathrm{c}}} \pi_{n,m-1}(x) = \sum_{\substack{k,\ell=1 \\ k+\ell \le m-1}}^{m-1} \sum_{\substack{x \in \mathcal{X}_{n,m-1} \\ x_i^{\mathrm{c}} \ge k, x_j^{\mathrm{c}} \ge \ell}} \pi_{n,m-1}(x).\end{aligned}$$

将方程（2）中的乘积形式解 $\pi_{n,m-1}(x)$ 代入此求和式，可得：

$$\begin{aligned}\mathbb{E}[X_i^{\mathrm{c}} X_j^{\mathrm{c}}] &= \frac{1}{Z_{n,m-1}} \sum_{\substack{k,\ell=1 \\ k+\ell \le m-1}}^{m-2} \sum_{\substack{x \in \mathcal{X}_{n,m-1} \\ x_i^{\mathrm{c}} \ge k, x_j^{\mathrm{c}} \ge \ell}} \prod_{r=1}^{n} \left(\frac{p_r}{\mu_r^{\mathrm{c}}}\right)^{x_r^{\mathrm{c}}} \frac{1}{x_r^{\mathrm{d}}!} \left(\frac{p_r}{\mu_r^{\mathrm{d}}}\right)^{x_r^{\mathrm{d}}} \frac{1}{x_r^{\mathrm{u}}!} \left(\frac{p_r}{\mu_r^{\mathrm{u}}}\right)^{x_r^{\mathrm{u}}} \\ &\overset{\text{(a)}}{=} \frac{1}{Z_{n,m-1}} \sum_{\substack{k,\ell=1 \\ k+\ell \le m-1}}^{m-2} \left(\frac{p_i}{\mu_i^{\mathrm{c}}}\right)^{k} \left(\frac{p_j}{\mu_j^{\mathrm{c}}}\right)^{\ell} \underbrace{\sum_{y \in \mathcal{X}_{n,m-1-k-\ell}} \prod_{r=1}^{n} \left(\frac{p_r}{\mu_r^{\mathrm{c}}}\right)^{y_r^{\mathrm{c}}} \frac{1}{y_r^{\mathrm{d}}!} \left(\frac{p_r}{\mu_r^{\mathrm{d}}}\right)^{y_r^{\mathrm{d}}} \frac{1}{y_r^{\mathrm{u}}!} \left(\frac{p_r}{\mu_r^{\mathrm{u}}}\right)^{y_r^{\mathrm{u}}}}_{=Z_{n,m-1-k-\ell}} \\ &= \sum_{k=1}^{m-2} \sum_{\ell=1}^{m-1-k} \left(\frac{p_i}{\mu_i^{\mathrm{c}}}\right)^{k} \left(\frac{p_j}{\mu_j^{\mathrm{c}}}\right)^{\ell} \frac{Z_{n,m-1-k-\ell}}{Z_{n,m-1}} = \alpha_{i,j}. \end{aligned} \tag{49}$$

步骤 (a) 是通过进行变量替换 $y = x - k\mathbf{e}_i^{\mathrm{c}} - \ell \mathbf{e}_j^{\mathrm{c}}$ 得出的，其中 $\mathbf{e}_i^{\mathrm{c}}$ 表示状态向量 x 中与分量 x_i^{c} 对应的 $3n$ 维标准单位向量。内层求和等于包含 $m-1-k-\ell$ 个任务的系统的归一化常数。

相反，对于单个队列的二阶矩（$i = j$），我们利用恒等式 $(x_i^{\mathrm{c}})^2 = \sum_{k=1}^{x_i^{\mathrm{c}}} (2k-1)$。据此，我们可以将期望改写为尾概率的加权和：

$$\begin{aligned}\mathbb{E}[(X_i^{\mathrm{c}})^2] &= \sum_{x \in \mathcal{X}_{n,m-1}} (x_i^{\mathrm{c}})^2 \pi_{n,m-1}(x) \\ &= \sum_{x \in \mathcal{X}_{n,m-1}} \sum_{k=1}^{x_i^{\mathrm{c}}} (2k-1) \pi_{n,m-1}(x) = \sum_{k=1}^{m-1} (2k-1) \sum_{\substack{x \in \mathcal{X}_{n,m-1} \\ x_i^{\mathrm{c}} \ge k}} \pi_{n,m-1}(x).\end{aligned}$$

再次应用方程（2）中的乘积形式解，可得：

$$\begin{aligned}\mathbb{E}[(X_i^{\mathrm{c}})^2] &= \frac{1}{Z_{n,m-1}} \sum_{k=1}^{m-1} (2k-1) \sum_{\substack{x \in \mathcal{X}_{n,m-1} \\ x_i^{\mathrm{c}} \ge k}} \prod_{r=1}^{n} \left(\frac{p_r}{\mu_r^{\mathrm{c}}}\right)^{x_r^{\mathrm{c}}} \frac{1}{x_r^{\mathrm{d}}!} \left(\frac{p_r}{\mu_r^{\mathrm{d}}}\right)^{x_r^{\mathrm{d}}} \frac{1}{x_r^{\mathrm{u}}!} \left(\frac{p_r}{\mu_r^{\mathrm{u}}}\right)^{x_r^{\mathrm{u}}} \\ &\overset{\text{(b)}}{=} \frac{1}{Z_{n,m-1}} \sum_{k=1}^{m-1} (2k-1) \left(\frac{p_i}{\mu_i^{\mathrm{c}}}\right)^{k} \underbrace{\sum_{y \in \mathcal{X}_{n,m-1-k}} \prod_{r=1}^{n} \left(\frac{p_r}{\mu_r^{\mathrm{c}}}\right)^{y_r^{\mathrm{c}}} \frac{1}{y_r^{\mathrm{d}}!} \left(\frac{p_r}{\mu_r^{\mathrm{d}}}\right)^{y_r^{\mathrm{d}}} \frac{1}{y_r^{\mathrm{u}}!} \left(\frac{p_r}{\mu_r^{\mathrm{u}}}\right)^{y_r^{\mathrm{u}}}}_{=Z_{n,m-1-k}} \\ &= \sum_{k=1}^{m-1} (2k-1) \left(\frac{p_i}{\mu_i^{\mathrm{c}}}\right)^{k} \frac{Z_{n,m-1-k}}{Z_{n,m-1}} = \alpha_{i,i}. \end{aligned} \tag{50}$$

步骤 (b) 可由变量替换 $y = x - k\mathbf{e}_i^{\mathrm{c}}$ 得出。

E.4.2 队列-延迟相关性（$\mathbb{E}[X_i^{\mathrm{c}} X_j^{\mathrm{d}}]$ 和 $\mathbb{E}[X_i^{\mathrm{c}} X_j^{\mathrm{u}}]$）

接下来，我们评估客户端 i 的计算队列与客户端 j 的通信延迟之间的相互作用。我们关注以下项 $\mathbb{E}[X_i^{\mathrm{c}} X_j^{\mathrm{d}}]$:

$$\begin{aligned}
\mathbb{E}[X_i^{\mathrm{c}} X_j^{\mathrm{d}}] &= \sum_{x \in \mathcal{X}_{n,m-1}} \sum_{k=1}^{x_i^{\mathrm{c}}} x_j^{\mathrm{d}} \pi_{n,m-1}(x) = \sum_{k=1}^{m-2} \sum_{\substack{x \in \mathcal{X}_{n,m-1} \\ x_i^{\mathrm{c}} \geq k}} x_j^{\mathrm{d}} \pi_{n,m-1}(x) \\
&= \frac{1}{Z_{n,m-1}} \sum_{k=1}^{m-2} \sum_{\substack{x \in \mathcal{X}_{n,m-1} \\ x_i^{\mathrm{c}} \geq k, x_j^{\mathrm{d}} \geq 1}} \prod_{r=1}^{n} \left(\frac{p_r}{\mu_r^{\mathrm{c}}}\right)^{x_r^{\mathrm{c}}} \frac{1}{(x_r^{\mathrm{d}} - \mathbf{1}\{r=j\})!} \left(\frac{p_r}{\mu_r^{\mathrm{d}}}\right)^{x_r^{\mathrm{d}}} \frac{1}{x_r^{\mathrm{u}}!} \left(\frac{p_r}{\mu_r^{\mathrm{u}}}\right)^{x_r^{\mathrm{u}}} \\
&\overset{(c)}{=} \frac{1}{Z_{n,m-1}} \sum_{k=1}^{m-2} \left(\frac{p_i}{\mu_i^{\mathrm{c}}}\right)^{k} \frac{p_j}{\mu_j^{\mathrm{d}}} \underbrace{\sum_{y \in \mathcal{X}_{n,m-2-k}} \prod_{r=1}^{n} \left(\frac{p_r}{\mu_r^{\mathrm{c}}}\right)^{y_r^{\mathrm{c}}} \frac{1}{y_r^{\mathrm{d}}!} \left(\frac{p_r}{\mu_r^{\mathrm{d}}}\right)^{y_r^{\mathrm{d}}} \frac{1}{y_r^{\mathrm{u}}!} \left(\frac{p_r}{\mu_r^{\mathrm{u}}}\right)^{y_r^{\mathrm{u}}}}_{= Z_{n,m-2-k}} \\
&= \frac{p_j}{\mu_j^{\mathrm{d}}} \sum_{k=1}^{m-2} \left(\frac{p_i}{\mu_i^{\mathrm{c}}}\right)^{k} \frac{Z_{n,m-2-k}}{Z_{n,m-1}} = \beta_{i,2} \frac{p_j}{\mu_j^{\mathrm{d}}}.
\end{aligned}$$

这里，步骤 (c) 使用变量替换 $y = x - k\mathbf{e}_i^{\mathrm{c}} - \mathbf{e}_j^{\mathrm{d}}$，并利用了 $x_j^{\mathrm{d}} \geq 1$ 这一事实。根据对称性，上行链路相关性为 $\mathbb{E}[X_i^{\mathrm{c}} X_j^{\mathrm{u}}] = \beta_{i,\ 2} \frac{p_j}{\mu_j^{\mathrm{u}}}$。将这些分量相加，得到客户端 i 的计算与客户端 j 的通信之间的总交互:

$$\mathbb{E}[X_i^{\mathrm{c}}(X_j^{\mathrm{d}} + X_j^{\mathrm{u}})] = \beta_{i,2} p_j \left(\frac{1}{\mu_j^{\mathrm{d}}} + \frac{1}{\mu_j^{\mathrm{u}}}\right) = \beta_{i,2} \gamma_j, \quad i,j \in \{1,\ldots,n\}. \tag{51}$$

同样，交换 i 和 j 的角色即可得到对称交互:

$$\mathbb{E}[(X_i^{\mathrm{d}} + X_i^{\mathrm{u}}) X_j^{\mathrm{c}}] = \beta_{j,2} \gamma_i, \quad i,j \in \{1,\ldots,n\}. \tag{52}$$

E.4.3 延迟-延迟相关性

最后，我们来讨论延迟节点之间的相关性。首先计算不同队列对的期望。对于交叉交互 $\mathbb{E}[X_i^{\mathrm{d}} X_j^{\mathrm{u}}]$，对于所有 i，$j \in \{1, \ldots\ldots, n\}$，下行队列和上行队列都是不同的实体。计算过程如下:

$$\begin{aligned}
\mathbb{E}[X_i^{\mathrm{d}} X_j^{\mathrm{u}}] &= \frac{1}{Z_{n,m-1}} \sum_{\substack{x \in \mathcal{X}_{n,m-1} \\ x_i^{\mathrm{d}} \geq 1, x_j^{\mathrm{u}} \geq 1}} x_i^{\mathrm{d}} x_j^{\mathrm{u}} \pi_{n,m-1}(x) \\
&\overset{(d)}{=} \frac{1}{Z_{n,m-1}} \frac{p_i}{\mu_i^{\mathrm{d}}} \frac{p_j}{\mu_j^{\mathrm{u}}} \underbrace{\sum_{y \in \mathcal{X}_{n,m-3}} \prod_{r=1}^{n} \left(\frac{p_r}{\mu_r^{\mathrm{c}}}\right)^{y_r^{\mathrm{c}}} \frac{1}{y_r^{\mathrm{d}}!} \left(\frac{p_r}{\mu_r^{\mathrm{d}}}\right)^{y_r^{\mathrm{d}}} \frac{1}{y_r^{\mathrm{u}}!} \left(\frac{p_r}{\mu_r^{\mathrm{u}}}\right)^{y_r^{\mathrm{u}}}}_{= Z_{n,m-3}} \\
&= \frac{Z_{n,m-3}}{Z_{n,m-1}} \frac{p_i p_j}{\mu_i^{\mathrm{d}} \mu_j^{\mathrm{u}}},
\end{aligned}$$

其中，(d)由变量替换 $y = x - \mathbf{e}_i^{\mathrm{d}} - \mathbf{e}_j^{\mathrm{u}}$得出。同理，对于不同的客户端（$i \neq j$），同类型延迟节点之间的相关性遵循相同的模式:

$$\mathbb{E}[X_i^{\mathrm{d}} X_j^{\mathrm{d}}] = \frac{Z_{n,m-3}}{Z_{n,m-1}} \frac{p_i p_j}{\mu_i^{\mathrm{d}} \mu_j^{\mathrm{d}}}, \quad \text{and } \mathbb{E}[X_i^{\mathrm{u}} X_j^{\mathrm{u}}] = \frac{Z_{n,m-3}}{Z_{n,m-1}} \frac{p_i p_j}{\mu_i^{\mathrm{u}} \mu_j^{\mathrm{u}}}.$$

然而，在计算单个队列（例如， X_i^{d}）的二阶矩时，适用阶乘矩展开：

$$\begin{aligned}
\mathbb{E}[(X_i^{\mathrm{d}})^2] &= \frac{1}{Z_{n,m-1}} \sum_{x\in\mathcal{X}_{n,m-1}x_i^{\mathrm{d}}\geq 1} (x_i^{\mathrm{d}})^2 \pi_{n,m-1}(x) \\
&= \frac{1}{Z_{n,m-1}} \sum_{\substack{x\in\mathcal{X}_{n,m-1} \\ x_i^{\mathrm{d}}\geq 2}} x_i^{\mathrm{d}}(x_i^{\mathrm{d}}-1)\pi_{n,m-1}(x) + \frac{1}{Z_{n,m-1}} \sum_{\substack{x\in\mathcal{X}_{n,m-1} \\ x_i^{\mathrm{d}}\geq 1}} x_i^{\mathrm{d}} \pi_{n,m-1}(x) \\
&\overset{\text{(e)}}{=} \frac{1}{Z_{n,m-1}} (\frac{p_i}{\mu_i^{\mathrm{d}}})^2 \underbrace{\sum_{y\in\mathcal{X}_{n,m-3}} \prod_{r=1}^{n} \left(\frac{p_r}{\mu_r^{\mathrm{c}}}\right)^{y_r^{\mathrm{c}}} \frac{1}{y_r^{\mathrm{d}}!} \left(\frac{p_r}{\mu_r^{\mathrm{d}}}\right)^{y_r^{\mathrm{d}}} \frac{1}{y_r^{\mathrm{u}}!} \left(\frac{p_r}{\mu_r^{\mathrm{u}}}\right)^{y_r^{\mathrm{u}}}}_{=Z_{n,m-3}} + \mathbb{E}[X_i^{\mathrm{d}}] \\
&\overset{\text{(f)}}{=} \frac{Z_{n,m-3}}{Z_{n,m-1}} (\frac{p_i}{\mu_i^{\mathrm{d}}})^2 + \frac{Z_{n,m-2}}{Z_{n,m-1}} \frac{p_i}{\mu_i^{\mathrm{d}}} \\
&= \frac{1}{Z_{n,m-1}} \frac{p_i}{\mu_i^{\mathrm{d}}} (Z_{n,m-3} \frac{p_i}{\mu_i^{\mathrm{d}}} + Z_{n,m-2}),
\end{aligned}$$

其中，(e)由变量替换 $y = x - 2\mathbf{e}_i^{\mathrm{d}}$得到，(f)则由（47）中$\mathbb{E}[X_i^{\mathrm{d}}]$ 的表达式得到。

将其推广到所有延迟对，总的延迟—延迟贡献为：

$$\sum_{s\in\{\mathrm{d},\mathrm{u}\}} \sum_{r\in\{\mathrm{d},\mathrm{u}\}} \mathbb{E}[X_i^s X_j^r] = \frac{Z_{n,m-3}}{Z_{n,m-1}} p_i p_j \left(\frac{1}{\mu_i^{\mathrm{d}}} + \frac{1}{\mu_i^{\mathrm{u}}}\right)\left(\frac{1}{\mu_j^{\mathrm{d}}} + \frac{1}{\mu_j^{\mathrm{u}}}\right) = \frac{Z_{n,m-3}}{Z_{n,m-1}} \gamma_i \gamma_j,$$

若 $i \neq j$。否则，若 $i = j$，则有：

$$\begin{aligned}
\sum_{s\in\{\mathrm{d},\mathrm{u}\}} \sum_{r\in\{\mathrm{d},\mathrm{u}\}} \mathbb{E}[X_i^s X_i^r] &= \frac{Z_{n,m-3}}{Z_{n,m-1}} (p_i)^2 \left(\frac{1}{\mu_i^{\mathrm{d}}} + \frac{1}{\mu_i^{\mathrm{u}}}\right)^2 + \frac{Z_{n,m-2}}{Z_{n,m-1}} \left(\frac{1}{\mu_j^{\mathrm{d}}} + \frac{1}{\mu_i^{\mathrm{u}}}\right) \\
&= \frac{\gamma_i}{Z_{n,m-1}} (\gamma_i Z_{n,m-3} + Z_{n,m-2}).
\end{aligned}$$

在这两种场景下，都可以简化为统一项：

$$\sum_{s\in\{\mathrm{d},\mathrm{u}\}} \sum_{r\in\{\mathrm{d},\mathrm{u}\}} \mathbb{E}[X_i^s X_j^r] = \frac{Z_{n,m-3}}{Z_{n,m-1}} \gamma_i \gamma_j + \mathbf{1}_{\{i=j\}} \frac{Z_{n,m-2}}{Z_{n,m-1}} \gamma_i = \psi_{i,j}. \tag{53}$$

将三个组成部分的贡献相加：来自（49）和（50）的队列—队列项$\alpha_{i,\ j}$，来自（51）和（52）的队列-时延交互项$\beta_{i,\ 2}\gamma_j + \beta_{j,\ 2}\gamma_i$，以及来自（53）的延迟—延迟项$\psi_{i,\ j}$，即可得到公式（6）中的最终结果。

F 定理 3 的证明

为了证明定理 3，我们采用 [29] 中提出并在 [31] 中进一步发展的虚拟迭代框架。虽然我们遵循 [31, 定理 1] 的一般推理来建立轮次复杂度 K_ϵ 的表达式，但我们引入了一项关键修改。具体而言，我们指出原证明中存在一个技术性缺陷：作者隐含地假设，对于所有 k，相对延迟 $D_{i,\ k}$ 与 $\nabla f(w_k)$ 相互独立。在第 2.6 节（以及第7.1节）所述的闭合杰克逊网络动力学下，这一假设不一定成立。我们的证明严格地考虑了这种依赖关系。

令 S_0 表示最初选定的客户端多重集，其中每个客户端出现的次数等于其最初被分配的任务数。根据构造， S_0 的基数为 m。为简化记号，我们将基于本地数据样本 ζ_i 计算得到的随机梯度记为 $g_i(w)$，而不记为 $g_i(w, \zeta_i)$。对于 $k \in \mathbb{N}_{>0}$，虚拟迭代 v_k 递归定义如下：

$$\begin{cases} v_0 = w_0, \\ v_1 = v_0 - \eta \sum_{i \in S_0} \frac{1}{np_i} g_i(w_0), \\ v_{k+1} = v_k - \frac{\eta}{np_{A_k}} g_{A_k}(w_k), \qquad k \geq 1. \end{cases}$$

v_k 与 w_k 之间的差值表示中央服务器派发的在途计算任务。这些任务产生的梯度在步骤 $k-1$ 结束时尚未收到（即刚收到来自客户端 C_{k-1} 的梯度、尚未向客户端 A_k 派发新任务之后）。

我们首先回顾在我们的精确问题设定下成立的 [31, 引理 4], :

▶**引理 16.** 在第 *2.6* 节的框架下，对于满足 $\eta \leq_{8L \sum_{i=1}^{n^2} p_i^{-1}}^{n}$ 的任意学习率，以下不等式成立：

$$\frac{1}{K} \sum_{k=0}^{K-1} \mathbb{E}^0 \left[\|\nabla f(w_k)\|^2 \right] \leq \frac{4\Delta}{\eta K} + \frac{4\eta L (2M^2 + \sigma^2)}{n^2} \sum_{i=1}^{n} \frac{1}{p_i} + \frac{2L^2}{K} \sum_{k=0}^{K-1} \mathbb{E}^0 \left[\|v_k - w_k\|^2 \right].$$

确立该引理后，我们继续推导陈旧误差 $\|v_k - w_k\|^2$ 的精确界，以揭示 K_ϵ 的表达式。

令 $\mathcal{Q}_k$ 表示由元组 (i, j) 构成的多重集，其中 i 是在轮次 k 开始时其梯度计算任务尚未被中央服务器应用的客户端索引（即该任务当前正在计算、排队或传输中），而 j 是该任务最初被派发的轮次。

多重集$\mathcal{Q}_k$ 可递归定义如下：

$$\begin{aligned} \mathcal{Q}_1 &= \{(i, 0) \mid i \in S_0, i \neq C_0\}, \\ \mathcal{Q}_{k+1} &= (\mathcal{Q}_k \setminus \{(C_k, I_k)\}) \cup \{(A_k, k)\} \quad \text{for all } k \geq 1. \end{aligned}$$

回顾一下， I_k 表示在第 k轮结束时收到的梯度最初被派发给客户端 C_k的轮次。

利用 [31, 引理 9] 表示差值 $v_k - w_k$，随后应用柯西-施瓦茨不等式，对于每个 $k \in \{1, \ldots, K\}$，我们有：

$$\|v_k - w_k\|^2 = \left\| -\eta \sum_{(i,j) \in \mathcal{Q}_k} \frac{1}{np_i} g_i(w_j) \right\|^2 \leq \eta^2 |\mathcal{Q}_k| \sum_{(i,j) \in \mathcal{Q}_k} \frac{1}{n^2 p_i^2} \|g_i(w_j)\|^2,$$

其中， $|\mathcal{Q}_k|$ 表示多重集$\mathcal{Q}_k$的基数。值得注意的是，对于每个 $k \in \mathbb{N}_{>0}$，在第 k次模型参数更新之后，正在执行中的任务数量严格为 $|\mathcal{Q}_k| = m - 1$。将该误差在全部 K 轮次上取平均，并重新排列求和指标后，我们得到：

$$\begin{aligned} \frac{1}{K} \sum_{k=0}^{K-1} \|v_k - w_k\|^2 &\leq \frac{\eta^2 (m-1)}{K} \sum_{k=1}^{K-1} \sum_{(i,j) \in \mathcal{Q}_k} \frac{1}{n^2 p_i^2} \|g_i(w_j)\|^2, \\ &= \frac{\eta^2 (m-1)}{K} \sum_{k=1}^{K-1} \sum_{i=1}^{n} \frac{1}{n^2 p_i^2} \sum_{j=0}^{k-1} \|g_i(w_j)\|^2 \mathbb{1}[(i,j) \in \mathcal{Q}_k], \end{aligned}$$

$$= \frac{\eta^2(m-1)}{K} \sum_{i=1}^{n} \frac{1}{n^2 p_i^2} \sum_{j=0}^{K-2} \|g_i(w_j)\|^2 \sum_{k=j+1}^{K-1} \mathbb{1}[(i,j) \in \mathcal{Q}_k],$$

其中，第一个等式是通过重新排列求和符号得到的，第二个等式则是通过交换关于 k 和 j 的求和得到的。

我们观察到，对于所有 $j \in \{1、2、\ldots\ldots、K-1\}$ 和 $i \in \{1、2、\ldots\ldots、n\}$，示性函数的内层求和由相对延迟界定：

$$\sum_{k=j+1}^{K-1} \mathbb{1}[(i,j) \in \mathcal{Q}_k] \leq \sum_{k=j+1}^{\infty} \mathbb{1}[(i,j) \in \mathcal{Q}_k] = D_{i,j},$$

其中，$D_{i,\,j}$ 定义为以下两者之间中央服务器执行的模型更新次数：(i) 在第 j 轮开始时将任务分配给客户端 i 的时间，以及 (ii) 中央服务器应用相应梯度的时间（若 $A_j = i$；否则，$D_{i,\,j} = 0$）。

类似地，对于 $j = 0$ 处的初始任务，我们有：

$$\sum_{k=1}^{K-1} \mathbb{1}[(i,0) \in \mathcal{Q}_k] \leq \sum_{k=1}^{\infty} \mathbb{1}[(i,0) \in \mathcal{Q}_k] = D_{i,0}.$$

稍微滥用一下符号，$D_{i,\,0}$ 表示从时间 0 到中央服务器应用最初分配给客户端 $i \in S_0$ 的所有任务所产生的梯度之间执行的更新次数（否则，$D_{i,\,0} = 0$）。

对 Palm 测度取期望后，我们得到：

$$\mathbb{E}^0\left[\frac{1}{K}\sum_{k=0}^{K-1}\|v_k - w_k\|^2\right] \leq \frac{\eta^2(m-1)}{K} \sum_{i=1}^{n} \frac{1}{n^2 p_i^2} \sum_{j=0}^{K-1} \mathbb{E}^0\left[\|g_i(w_j)\|^2 D_{i,j}\right]. \tag{54}$$

现在我们来考察项 $\mathbb{E}^0[\|g_i(w_j)\|^2 D_{i,\,j}]$。关键在于，给定 w_j 后，随机梯度$g_i(w_j)$仅取决于客户端 i的本地数据，而相对延迟 $D_{i,\,j}$ 完全取决于网络动态，且网络动态的运行独立于客户端对本地数据的采样。因此，给定 w_j 后，$D_{i,\,j}$ 与 $g_i(w_j)$条件独立。运用全期望定律，我们可以安全地将各项[2]拆分：

$$\begin{aligned}\mathbb{E}^0\left[\|g_i(w_j)\|^2 D_{i,j}\right] &= \mathbb{E}^0\left(\mathbb{E}^0\left[\|g_i(w_j)\|^2 D_{i,j} \mid w_j\right]\right) \\ &= \mathbb{E}^0\left(\mathbb{E}^0\left[\|g_i(w_j)\|^2 \mid w_j\right]\mathbb{E}^0\left[D_{i,j} \mid w_j\right]\right).\end{aligned} \tag{55}$$

为了界定梯度范数，我们结合模型假设使用不等式 $\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$：

$$\begin{aligned}\mathbb{E}^0\left[\|g_i(w_j)\|^2 \mid w_j\right] &= \mathbb{E}^0\left[\|g_i(w_j) - \nabla f_i(w_j) + \nabla f_i(w_j)\|^2 \mid w_j\right] \\ &\leq 2\mathbb{E}^0\left[\|g_i(w_j) - \nabla f_i(w_j)\|^2 \mid w_j\right] + 2\mathbb{E}^0\left[\|\nabla f_i(w_j)\|^2 \mid w_j\right] \\ &\leq 2\sigma^2 + 2\mathbb{E}^0\left[\|\nabla f_i(w_j)\|^2 \mid w_j\right] \quad \text{(using Assumption } \mathbf{A3}) \\ &\leq 2\sigma^2 + 2G^2 \quad \text{(using Assumption } \mathbf{A5}).\end{aligned}$$

将此界代回式（55），可得：

$$\mathbb{E}^0\left[\|g_i(w_j)\|^2 D_{i,j}\right] \leq 2(\sigma^2 + G^2)\mathbb{E}^0\left[D_{i,j}\right].$$

[2] 虽然 [31]隐含地假设 w_j 和 $D_{i,\,j}$ 相互独立，但这种依赖关系通常成立；因此，我们采用另一种方法，严格地界定期望 $\mathbb{E}^0\left[\|g_i(w_j)\|^2 D_{i,\,j}\right]$。

将此结果代入不等式（54），即可得到在途陈旧误差的显式界：

$$\mathbb{E}^0\left[\frac{1}{K}\sum_{k=0}^{K-1}\|v_k - w_k\|^2\right] \leq \frac{2\eta^2(m-1)}{K}\sum_{i=1}^{n}\frac{\sigma^2+G^2}{n^2 p_i^2}\sum_{j=0}^{K-1}\mathbb{E}^0\left[D_{i,j}\right]. \tag{56}$$

将不等式 (56) 代入引理 16 中得到的初始界，我们得到：

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}^0\left[\|\nabla f(w_k)\|^2\right] \leq \frac{4\Delta}{\eta K} + \frac{4\eta L(2M^2+\sigma^2)}{n^2}\sum_{i=1}^{n}\frac{1}{p_i} + \frac{4\eta^2L^2(m-1)(\sigma^2+G^2)}{n^2}\sum_{i=1}^{n}\frac{1}{p_i^2}\left(\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}^0\left[D_{i,k}\right]\right).$$

由于序列 $D_{i,\,k}$ 的平稳性，对于所有 $k \in\{1，2，\ldots\ldots，K-1\}$，都有 $\mathbb{E}^0\left[D_{i,k}\right] = \mathbb{E}^0\left[D_i\right]$。因此，上述不等式可简化为：

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}^0\left[\|\nabla f(w_k)\|^2\right] \leq \frac{4\Delta}{\eta K} + \frac{4\eta L(2M^2+\sigma^2)}{n^2}\sum_{i=1}^{n}\frac{1}{p_i} + \frac{4\eta^2L^2(m-1)(\sigma^2+G^2)}{n^2}\sum_{i=1}^{n}\frac{1}{p_i^2}\left(\frac{\mathbb{E}^0[D_{i,0}]}{K} + \mathbb{E}^0[D_i]\right).$$

在下文中，我们假设学习率 η 满足

$$\eta \leq \frac{n^2}{8L\sum_{i=1}^{n}p_i^{-1}}, \tag{57}$$

这样即可满足引理 16 的假设，且最后一个不等式也成立。

为保证达到 ϵ 准确率，我们将右侧的三个项分别界定为 $\epsilon/3$。暂且假设总轮数 K 足够大，从而初始延迟可忽略不计，相较于稳态延迟（即 $\frac{\mathbb{E}^0[D_{i,0}]}{K} \leq \mathbb{E}^0[D_i]$），则可将期望延迟项 $\left(\frac{\mathbb{E}^0[D_{i,0}]}{K} + \mathbb{E}^0[D_i]\right)$ 上界为 $2\mathbb{E}^0[D_i]$。应用这一简化，并将第二项和第三项界定为 $\epsilon/3$，可得到关于学习率 η 的以下约束：

$$\eta \leq \frac{n^2\epsilon}{12L(2M^2+\sigma^2)\sum_{i=1}^{n}\frac{1}{p_i}}, \quad \text{and} \quad \eta \leq \frac{n}{2L}\sqrt{\frac{\epsilon}{6(m-1)(\sigma^2+G^2)\sum_{i=1}^{n}\frac{\mathbb{E}^0[D_i]}{p_i^2}}}.$$

回到第一项，条件 $\frac{4\Delta}{\eta K} \leq \frac{\epsilon}{3}$ 规定了 $K \geq \frac{12\Delta}{\eta\epsilon}$。由于既定的界要求 η 至少以$\mathcal{O}(\sqrt{\epsilon})$的速率缩小，因此，当 $\epsilon \to 0$时，所需的轮次数量 K将以$\bar{\mathcal{O}}(1/\epsilon^{3/2})$的速率无限增长。这证实了我们暂时的假设：严格来说，存在一个精度阈值 $\epsilon_0 > 0$，使得对于所有 $0 < \epsilon \leq \epsilon_0$，$K$ 都足够大，从而严格满足 $K \geq \max_i \frac{\mathbb{E}^0[D_{i,0}]}{\mathbb{E}^0[D_i]}$。

现在，将推导出的关于 η 的上界以及式 (57) 中的初始学习率条件代入要求 $K \geq \frac{12\Delta}{\eta\epsilon}$，即可得出总轮数的显式下界：

$$K \geq \frac{12\Delta}{\epsilon}\frac{1}{\eta} \geq \frac{12\Delta}{\epsilon}\max\left\{\frac{8L}{n^2}\sum_{i=1}^{n}\frac{1}{p_i}, \frac{12L(2M^2+\sigma^2)}{n^2\epsilon}\sum_{i=1}^{n}\frac{1}{p_i},\right.$$

$$\frac{2L\sqrt{6(m-1)(\sigma^2+G^2)}}{n\sqrt{\epsilon}}\sqrt{\sum_{i=1}^{n}\frac{\mathbb{E}^0[D_i]}{p_i^2}}\Bigg\}.$$

因此，只需满足 $K \geq K_\epsilon(p,\ m)$，即可使收敛准则 $\frac{1}{K}\sum_{t=0}^{K-1}\mathbb{E}[\|\nabla f(w_t)\|^2] \leq \epsilon$ 对任意 $0<\epsilon\leq\epsilon_0$ 严格成立，其中：

$$K_\epsilon(p,m)=\frac{24L\Delta}{n\epsilon}\left[\left(4+\frac{B}{\epsilon}\right)\sum_{i=1}^{n}\frac{1}{np_i}+\left(\frac{C(m-1)}{\epsilon}\sum_{i=1}^{n}\frac{\mathbb{E}^0[D_i]}{p_i^2}\right)^{1/2}\right].$$

$K_\epsilon(p,\ m)$ 的表达式是通过将前述不等式中的最大值以上界替换为各项之和，进行化简，并定义常数 $B=6(\sigma^2+2M^2)$ 和 $C=6(\sigma^2+G^2)$ 得到的。证明至此完成。

G 放宽定理 3 的有界梯度假设 (A5)

即使去除了有界梯度假设（**A5**），第 2.6 节中详细介绍的排队网络模型仍能为轮次复杂度提供严格的收敛保证。我们在以下定理中对此进行形式化表述。

▶**定理 17**。 定义系统级陈旧因子 S_{sys} 如下：

$$S_{\text{sys}}=(m-1)|\mu^{\text{u}}|\sum_{i=1}^{n}\left(\frac{1}{\mu_i^{\text{d}}}+\frac{1}{\mu_i^{\text{u}}}+\frac{m}{\mu_i^{\text{c}}}\right)p_i^{-2},\quad where\quad |\mu^{\text{u}}|=\sum_{i=1}^{n}\mu_i^{\text{u}}. \tag{58}$$

那么，在假设 ***A1–A4*** 以及第 *2.6* 节所述模型成立的条件下，存在 $\epsilon_0>0$，使得对于任意目标精度 $\epsilon\in(0,\ \epsilon_0]$ 以及满足以下条件的任意学习率

$$\eta\leq\min\left\{\frac{n^2}{8L\sum_{i=1}^{n}p_i^{-1}},\frac{n^2\epsilon}{4LB\sum_{i=1}^{n}p_i^{-1}},\frac{n\sqrt{\epsilon}}{2L}\left(2B(m-1)\sum_{i=1}^{n}\frac{\mathbb{E}^0[D_i]}{p_i^2}\right)^{-1/2},\frac{n}{4L\sqrt{(m-1)S_{\text{sys}}}}\right\},$$

期望梯度范数满足

$$\frac{1}{K}\sum_{t=0}^{K-1}\mathbb{E}^0\left[\|\nabla f(w_t)\|^2\right]\leq\epsilon$$

当 $K\geq K_\epsilon(p,m)$ 时，所需的轮次复杂度 $K_\epsilon(p,m)$ 为：

$$K_\epsilon(p,m)=\frac{96L\Delta}{n\epsilon}\left[\left(2+\frac{B}{\epsilon}\right)\sum_{i=1}^{n}\frac{1}{np_i}+\sqrt{(m-1)S_{\text{sys}}}+\left(\frac{B(m-1)}{2\epsilon}\sum_{i=1}^{n}\frac{\mathbb{E}^0[D_i]}{p_i^2}\right)^{1/2}\right].$$

这里，$B=6(\sigma^2+2M^2)$。

该定理的意义有三方面。首先，它在不依赖限制性有界梯度假设的情况下，建立了严格的收敛保证。其次，它引入了 S_{sys}，以从物理层面刻画网络中的聚合陈旧惩罚。最后，它通过将最大学习率 η 界定为与 S_{sys}成反比，正式确立了硬件—算法协同设计，并证明严重的网络拥塞在数学上要求采用更小的学习率，以确保优化稳定性。

证明。定理 17 的证明采用了与定理 3 的证明相同的技巧。直到公式 (55) 为止，推导过程仍然有效。我们必须以不同方式考察期望项$\mathbb{E}^0\left[\|g_i(w_j)\|^2 D_{i,\,j}\right]$。具体而言，我们不依赖有界梯度假设（**A5**），并且必须明确考虑相对延迟与模型参数之间的依赖关系。

我们结合模型假设使用不等式 $\|a+b\|^2 \le 2\|a\|^2 + 2\|b\|^2$：

$$\begin{aligned}
\mathbb{E}^0\left[\|g_i(w_j)\|^2 \mid w_j\right] &= \mathbb{E}^0\left[\|g_i(w_j) - \nabla f_i(w_j) + \nabla f_i(w_j)\|^2 \mid w_j\right] \\
&\le 2\mathbb{E}^0\left[\|g_i(w_j) - \nabla f_i(w_j)\|^2 \mid w_j\right] + 2\mathbb{E}^0\left[\|\nabla f_i(w_j)\|^2 \mid w_j\right] \\
&\le 2\sigma^2 + 2\mathbb{E}^0\left[\|\nabla f_i(w_j)\|^2 \mid w_j\right] \\
&\le 2\sigma^2 + 4\mathbb{E}^0\left[\|\nabla f_i(w_j) - \nabla f(w_j)\|^2 \mid w_j\right] + 4\mathbb{E}^0\left[\|\nabla f(w_j)\|^2 \mid w_j\right] \\
&\le 2\sigma^2 + 4M^2 + 4\mathbb{E}^0\left[\|\nabla f(w_j)\|^2 \mid w_j\right].
\end{aligned}$$

其中，第二个不等式由 **A3** 得出，第三个不等式由经典不等式 $\|a+b\|^2 \le 2\|a\|^2 + 2\|b\|^2$ 得出，第四个不等式则由 **A4** 得出。

将其代回公式 (55)，可得：

$$\mathbb{E}^0\left[\|g_i(w_j)\|^2 D_{i,j}\right] \le 2(\sigma^2 + 2M^2)\mathbb{E}^0\left[D_{i,j}\right] + 4\mathbb{E}^0\left[\|\nabla f(w_j)\|^2 D_{i,j}\right]. \tag{59}$$

请注意，相对延迟 $D_{i,\,j}$ 与 $\nabla f(w_j)$ 并不独立。因此，我们使用一个与 $\nabla f(w_j)$ 独立的新随机变量，对相对延迟$D_{i,\,j}$ 进行随机上界估计。具体而言，我们考虑发送给客户端 i、在轮次 j到达的任务所经历的最坏情况逗留时间场景：该任务完全下载后，发现客户端 i 的计算队列中已有 $m-1$ 个任务排队。由于指数服务时间具有无记忆性，该任务在客户端 i（包括服务器 d_i、c_i 和 u_i）的逗留时间 S_i 服从若干相互独立的指数随机变量之和的分布：

$$S_i = E_i^{\mathrm{d}} + \sum_{k=1}^{m} E_{i,k}^{\mathrm{c}} + E_i^{\mathrm{u}},$$

其中，E_i^{d}是参数为 μ_i^{d}的指数随机变量，$(E_{i,k}^{\mathrm{c}})_{k\ge 1}$ 是参数为 μ_i^{c}的独立同分布指数随机变量，E_i^{u} 是参数为 μ_i^{u}的指数随机变量。数量$D_{i,j}$表示该任务逗留期间的参数更新次数，可由独立泊松过程 N 在时间区间内生成的事件数随机界定。该泊松过程的强度为$(m-1)|\mu^{\mathrm{u}}| = (m-1)\sum_{k=1}^{n}\mu_k^{\mathrm{u}}$，时间区间的分布为 S_i。

因此，对所有 $j \in \{1, \dots, K\}$：

$$\begin{aligned}
\mathbb{E}^0\left[D_{i,j}\|\nabla f(w_j)\|^2\right] &\le \mathbb{E}^0\left[N(S_i)\|\nabla f(w_j)\|^2\right] \\
&= \mathbb{E}^0\left[N(S_i)\right] \cdot \mathbb{E}^0\left[\|\nabla f(w_j)\|^2\right] \quad \text{(by independence)} \\
&= \mathbb{E}^0\left[\|\nabla f(w_j)\|^2\right] \cdot \left(\frac{1}{\mu_i^{\mathrm{d}}} + \frac{m}{\mu_i^{\mathrm{c}}} + \frac{1}{\mu_i^{\mathrm{u}}}\right)(m-1)\sum_{k=1}^{n}\mu_k^{\mathrm{u}}.
\end{aligned}$$

将此界代入（59），可得：

$$\mathbb{E}^0\left[\|g_i(w_j)\|^2 D_{i,j}\right] \le 2(\sigma^2+2M^2)\mathbb{E}^0\left[D_{i,j}\right] + 4(m-1)|\mu^{\mathrm{u}}|\left(\frac{1}{\mu_i^{\mathrm{d}}} + \frac{m}{\mu_i^{\mathrm{c}}} + \frac{1}{\mu_i^{\mathrm{u}}}\right)\mathbb{E}^0\left[\|\nabla f(w_j)\|^2\right].$$

将此结果代入不等式（54），即可得到在途陈旧度的显式界

$^{\mathrm{e}}$错误：

$$\mathbb{E}^0\left[\frac{1}{K}\sum_{k=0}^{K-1}\|v_k-w_k\|^2\right]\le\frac{2\eta^2(m-1)}{K}\sum_{i=1}^{n}\frac{1}{n^2p_i^2}\sum_{j=0}^{K-1}\Bigg((\sigma^2+2M^2)\mathbb{E}^0\left[D_{i,j}\right]$$
$$+2(m-1)|\mu^{\mathrm{u}}|\left(\frac{1}{\mu_i^{\mathrm{d}}}+\frac{m}{\mu_i^{\mathrm{c}}}+\frac{1}{\mu_i^{\mathrm{u}}}\right)\mathbb{E}^0\left[\|\nabla f(w_j)\|^2\right]\Bigg).$$

将后一个不等式代入引理 16 中建立的初始界，并将系统级陈旧因子定义为 $S_{\mathrm{sys}}=(m-1)|\mu^{\mathrm{u}}|\sum_{i=1}^{n}\left(\frac{1}{\mu_i^{\mathrm{d}}}+\frac{1}{\mu_i^{\mathrm{u}}}+\frac{m}{\mu_i^{\mathrm{c}}}\right)p_i^{-2}$，我们得到：

$$\frac{1}{K}\sum_{k=0}^{K-1}\left(1-\frac{8\eta^2L^2(m-1)}{n^2}S_{\mathrm{sys}}\right)\mathbb{E}^0\left[\|\nabla f(w_k)\|^2\right]$$
$$\le\frac{4\Delta}{\eta K}+\frac{4\eta L(2M^2+\sigma^2)}{n^2}\sum_{i=1}^{n}\frac{1}{p_i}+\frac{4\eta^2L^2(m-1)(\sigma^2+2M^2)}{n^2}\sum_{i=1}^{n}\frac{1}{p_i^2}\left(\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}^0\left[D_{i,k}\right]\right).$$

通过对学习率施加以下约束：

$$\eta\le\frac{n}{4L}\frac{1}{\sqrt{(m-1)S_{\mathrm{sys}}}},\tag{60}$$

左侧乘以期望梯度范数的因子有下界1/2。将整个不等式乘以 2，得到：

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}^0\left[\|\nabla f(w_k)\|^2\right]\le\frac{8\Delta}{\eta K}+\frac{8\eta L(2M^2+\sigma^2)}{n^2}\sum_{i=1}^{n}\frac{1}{p_i}$$
$$+\frac{8\eta^2L^2(m-1)(\sigma^2+2M^2)}{n^2}\sum_{i=1}^{n}\frac{1}{p_i^2}\left(\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}^0\left[D_{i,k}\right]\right).$$

由于序列 $D_{i,k}$ 的平稳性，对于所有 $k\ge1$，我们有 $\mathbb{E}^0\left[D_{i,k}\right]=\mathbb{E}^0\left[D_i\right]$。因此，不等式可简化为：

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}^0\left[\|\nabla f(w_k)\|^2\right]\le\frac{8\Delta}{\eta K}+\frac{8\eta L(2M^2+\sigma^2)}{n^2}\sum_{i=1}^{n}\frac{1}{p_i}$$
$$+\frac{8\eta^2L^2(m-1)(\sigma^2+2M^2)}{n^2}\sum_{i=1}^{n}\frac{1}{p_i^2}\left(\frac{\mathbb{E}^0[D_{i,0}]}{K}+\mathbb{E}^0[D_i]\right).$$

为了保证 ϵ-准确率，我们将右侧的三个项分别界定为$\epsilon/3$。如果暂时假设总轮数 K 足够大，能够确保初始延迟被稳态延迟所主导（即 $\frac{\mathbb{E}^0[D_{i,0}]}{K}\le\mathbb{E}^0[D_i]$），则可将期望延迟项 $\left(\frac{\mathbb{E}^0[D_{i,0}]}{K}+\mathbb{E}^0[D_i]\right)$ 上界为 $2\mathbb{E}^0[D_i]$。应用这一简化，并将第二项和第三项界定为 $\epsilon/3$，可得学习率 η需满足以下约束：

$$\eta\le\frac{n^2\epsilon}{24L(2M^2+\sigma^2)\sum_{i=1}^{n}\frac{1}{p_i}},\quad\text{and}\quad\eta\le\frac{n}{4L}\sqrt{\frac{\epsilon}{3(m-1)(\sigma^2+2M^2)\sum_{i=1}^{n}\frac{\mathbb{E}^0[D_i]}{p_i^2}}}.$$

回到第一项，条件 $\frac{8\Delta}{\eta K}\le\frac{\epsilon}{3}$ 规定了 $K\ge\frac{24\Delta}{\eta\epsilon}$。由于已得出的界要求 ~~$\eta$~~ 至少以 $\mathcal{O}(\sqrt{\epsilon})$ 的速度缩小，所需轮次数 K

会在 $\epsilon \to 0$ 时无界增长，达到 $\mathcal{O}(1/\epsilon^{3/2})$。这证实了我们暂定的假设：严格来说，存在一个精度阈值 $\epsilon_0 > 0$，使得对于所有 $0 < \epsilon \leq \epsilon_0$，$K$ 都足够大，能够严格满足 $K \geq \max_i \frac{\mathbb{E}^0[D_{i,0}]}{\mathbb{E}^0[D_i]}$。

现在，将推导出的关于 η 的上界，以及初始约束 $\eta \leq \frac{1}{8L\sum_{i=1}^{n^2} p_i^{-1}}$ 和 (60)，代入条件 $K \geq \frac{24\Delta}{\eta\epsilon}$，即可得到总轮数的显式下界：

$$K \geq \frac{24\Delta}{\epsilon\eta} \geq \frac{24\Delta}{\epsilon} \max\left\{ \frac{8L}{n^2}\sum_{i=1}^{n}\frac{1}{p_i}, \quad \frac{4L}{n}\sqrt{(m-1)S_{\text{sys}}}, \quad \frac{24L(2M^2+\sigma^2)}{n^2\epsilon}\sum_{i=1}^{n}\frac{1}{p_i}, \right.$$
$$\left. \frac{4L\sqrt{3(m-1)(\sigma^2+2M^2)}}{n\sqrt{\epsilon}}\sqrt{\sum_{i=1}^{n}\frac{\mathbb{E}^0[D_i]}{p_i^2}} \right\}.$$

因此，只要要求 $K \geq K_\epsilon(p, m)$，对于任意 $0 < \epsilon \leq \epsilon_0$，收敛准则 $\frac{1}{K}\sum_{t=0}^{K-1}\mathbb{E}[\|\nabla f(w_t)\|^2] \leq \epsilon$ 都严格成立，其中：

$$K_\epsilon(p,m) = \frac{96L\Delta}{n\epsilon}\left[\left(2+\frac{B}{\epsilon}\right)\sum_{i=1}^{n}\frac{1}{np_i} + \sqrt{(m-1)S_{\text{sys}}} + \left(\frac{B(m-1)}{2\epsilon}\sum_{i=1}^{n}\frac{\mathbb{E}^0[D_i]}{p_i^2}\right)^{1/2}\right].$$

$K_\epsilon(p, m)$ 的表达式可通过用各项之和对前述不等式中的最大值取上界、化简并定义常数 $B = 6(\sigma^2 + 2M^2)$ 得到。至此，证明完毕。◀

H 优化轮次复杂度：实验结果

在本节中，我们通过实证验证理论结果，证明优化路由概率向量 p 能够显著降低轮次复杂度。遵循 [29]，中确立的完全并发设置（$m = n$），我们将提出的策略与两种标准基线方法进行比较：

1. 轮次优化 `Generalized AsyncSGD`： 我们提出的方法采用最优路由向量 p^{*K}，该向量旨在最小化理论轮次复杂度上界 K_ϵ，并在完全并发条件下（$m = n$）。**2. 标准基线（`AsyncSGD`）** [29, 算法 2]：传统异步方法在所有客户端中采用均匀路由概率（p^{uni}），并实现完全并发（$m = n$）。**3. 最大吞吐量 `Generalized AsyncSGD`：** 一种启发式策略，通过最大化全局更新频率（$p^{*\lambda}$），在完全并发条件下（$m = n$）优先提升系统速度，而不考虑数据异质性对收敛的影响。

H.1 实验设置

为了评估我们提出的策略相对于基线方法的效率，我们模拟了一个吞吐量多样化场景，其中包含 $n = 100$ 个客户端，并将其划分为五个不同的集群（A–E 类）。此配置体现了现实异构边缘环境中常见的多样化硬件和网络条件，涵盖从高性能超级客户端到资源受限的掉队客户端。计算（μ^{c}）、上行链路（μ^{u}）和下行链路（μ^{d}）的具体服务速率详见表6。

我们在三种不同的数据分布场景下评估 EMNIST [14] 数据集的性能：

表6 客户端集群、代表性硬件配置及服务速率（速率 μ 任务/秒）

类型	配置	μ^{c}	μ^{u}	μ^{d}	数量
A	计算快、网络慢 10.0 2.0			2.5	15
B	计算慢、网络快 2.5		8.0	9.0	35
C	均衡	5.0	5.0	6.0	30
D	掉队者	0.5	0.8	1.1	15
E	高性能	15.0	10.0	11.0	5

- **同质（独立同分布）：** 训练数据经过均匀打乱后分配给各客户端，确保每个客户端都具有相同的类别分布和相同数量的样本。
- **异质（非独立同分布）：** 我们使用狄利克雷分布模拟现实中的特征异质性和标签异质性。对于每个类别 k，分配给客户端 j的样本比例取自向量 $q_k \sim \mathrm{Dir}_n(\alpha)$。我们将集中参数设为 $\alpha = 0.2$，遵循 [64, 33]。
- **高度异质（病态非独立同分布）：** 我们施加极端标签偏斜，使每个客户端仅持有来自 3 个不同类别的数据，这些类别从总标签空间中均匀随机选取。所有客户端的样本总数均衡。

H.2 优化：策略与结果

我们通过梯度下降最小化轮次复杂度 K_ϵ，从而确定最优路由向量 p^{*K}。优化采用Adam优化器 [28], ，其中 K_ϵ 的梯度通过其闭式表达式（9）结合定理2中推导出的延迟梯度（4）进行精确计算。我们注意到，目标函数 K_ϵ（p，m）关于 p是非凸的；因此，为降低收敛到次优局部极小值的风险，我们采用多次随机初始化。如果只关注寻找优于均匀路由的解决方案，则均匀分布可以作为优化的有效起点。

同样地，在基线比较中，我们通过最大化系统更新频率 λ来计算最大吞吐量路由向量 $p^{*\lambda}$。具体而言，我们使用 Adam 执行梯度上升，并利用公式（12）中给出的更新频率闭式梯度。数据相关常数 σ、 M和 G （在第 2.5 节中引入）根据训练数据集进行经验估计，并将目标梯度范数上界设为 $\epsilon = 1$。

优化路由概率及相应的性能指标列于表7。为解读优化行为，我们明确分析项 $\mathbb{E}^0[D_i(p, m=n)]\, p_i^{-2}$，并将其定义为陈旧性影响因子。该指标量化了每个客户端对 K_ϵ（见公式(9)）中陈旧性分量的单独贡献，使我们能够识别出对算法稳定性造成不成比例损害的特定集群。

吞吐量优化路由（$p^{*\lambda}$）对快速客户端表现出极端偏向，大力优待超级客户端（E 型），同时大幅降低掉队客户端（D 类）的权重。尽管该策略成功将系统更新频率提升至最高（$\lambda = 151$ 次更新/秒），但它对 D 类客户端造成了极端的陈旧性影响（$\mathbb{E}^0[D_i]\, p_i^{-2} \approx 1.5 \times 10^6$）。因此，当掉队客户端终于贡献一次更新时，其梯度已经极度陈旧。这种严重影响会降低收敛效果，并完全抵消快速客户端频繁更新带来的益处。

表7 不同客户端集群（A–E 类）中优化路由概率与陈旧性影响因子的比较。

	路由概率 $p \times 100$		**陈旧性影响因子** $\mathbb{E}^0[D_i]\, p_i^{-2} \times 10^{-2}$		
Type	$p^{*\lambda}$	p^{*K}	$p^{*\lambda}$	p^{uni}	p^{*K}
A	0.405	0.788	373.9	40.9	3.0
B	0.422	0.754	274.8	29.1	2.0
C	1.421	0.748	75.5	23.8	1.8
D	0.045	1.033	14,869.5	500.2	9.7
E	7.173	0.718	8.7	10.6	0.8

标准均匀路由（p^{uni}）未能考虑这种系统异质性。它将 D 类掉队客户端与 E 类超级客户端一视同仁，因此给掉队客户端带来的陈旧度因子（约为 5×10^4）虽处于 $p^{*\lambda}$ 两种极端之间，仍远非最优。我们所提方法 p^{*K} 在此基线基础上提升了 $50\times$ 倍（将陈旧性影响降至 $\approx 10^3$），这表明，对高度异质的客户端一视同仁，效率极其低下。

相比之下，轮次优化路由（p^{*K}）采用了一种反直觉但极其有效的策略：优先选择最慢的客户端。它为 D 类掉队客户端分配了最高的路由概率（几乎是 E 型的 $1.4\times$ 倍），因此牺牲了原始更新频率（$\lambda = 2.4$，而 p^{uni} 为 41），以使网络中的陈旧性影响趋于均衡。通过迫使慢速客户端更频繁地更新，优化器确保与掉队客户端相关的陈旧度项不会主导收敛界。

H.3 学习性能

为验证路由优化分析所得的见解并佐证理论结果，我们在图像分类任务上模拟训练过程。我们采用 H.1 节所述的实验设置来模拟学习动态。

为评估系统对初始瞬态的鲁棒性，我们将系统初始化在非平衡状态：在 $t = 0$，$m = n$ 任务以均匀随机的方式分配给客户端下行服务器，而非从平稳分布中抽取。此外，为验证我们的结论并非指数服务时间假设所致，我们评估了计算时间和通信时间在以下三种不同分布下的性能：

(i) 指数分布： 服务时间服从指数分布，这是理论分析所要求的。**(ii) 确定性：** 服务时间为固定值，等于 $1/\mu$，对应于零方差。**(iii) 对数正态分布：** 服务时间服从均值为 $1/\mu$ 的重尾分布。我们将底层正态分布的方差设为 $\sigma_N^2 = 1$，以反映现实边缘系统中观察到的高变异性，同时确保各客户端的变异系数固定。

模型采用标准的多类交叉熵损失进行训练，并在未见过的标签均衡测试集上评估性能。学习率通过网格搜索选取，更多实现细节见 B 节。

图7所示的实验结果表明，在所考察的各个场景中， `Generalized AsyncSGD`配备优化路由向量 p^{*K} 后始终优于所有基线方法。在整个训练过程中，即使面对高度异构的数据分布， p^{*K} 也能取得更高的准确率和更好的稳定性，从而大幅减少收敛所需的通信轮次。

值得注意的是，尽管直觉上，将更高的路由概率分配给较慢的客户端似乎会使模型偏向其本地数据，但我们的模拟显示，实际效果恰恰相反。优化路由有效缓解了梯度陈旧性，从而实现平滑且稳定的收敛。相比之下，均匀路由策略，尤其是最大吞吐量策略，表现出显著更高的损失方差，这证实了盲目提高更新频率会导致严重的学习不稳定性。这些观察结果验证了我们的框架能够准确捕捉并抵消由数据和服务时间异质性所引发的陈旧度不利影响。

此外，在所有评估过的服务时间分布中， p^{*K} 所带来的性能提升始终如一，并且在确定性情形下尤为显著。这表明，所提出方法的鲁棒性远不止适用于理论分析所依据的指数服务时间假设。

最后，需要结合所选性能指标来理解这些结果。展望第 4.2 节， p^{*K} 的优势仅适用于轮次复杂度指标 K_ϵ。按照设计，最小化 K_ϵ 会优先考虑速度较慢的客户端，从而降低陈旧度，但这也必然会降低系统吞吐量。因此，如果以实际运行时间而非通信轮次来评估性能，各策略的相对排名可能会有所不同。

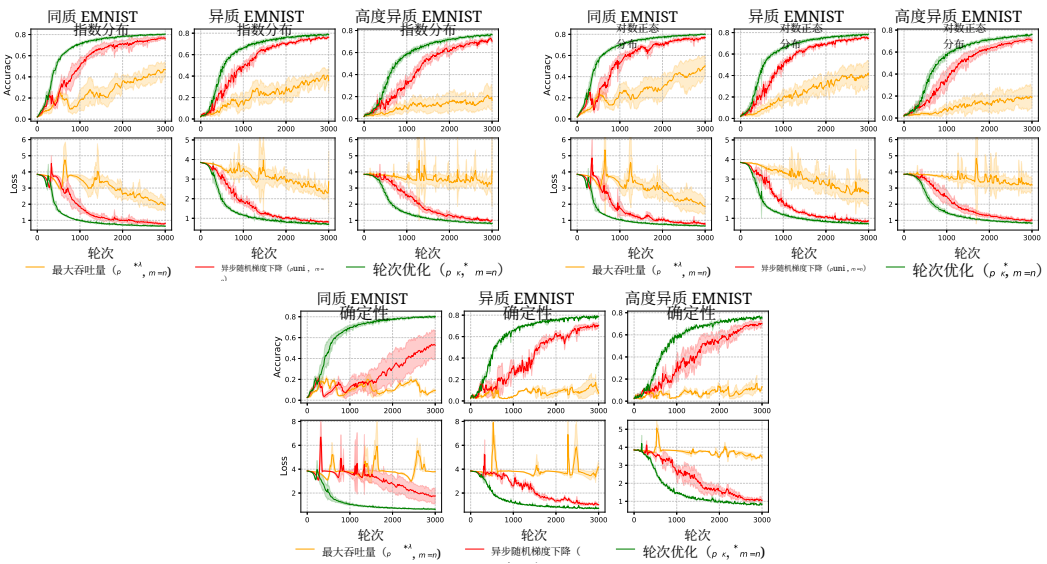


图7 使用表6中定义的客户端配置，在 EMNIST 数据集（$n = 100$、 $m = n$）上测试集准确率和损失随轮次的变化。子图评估三种服务时间分布下的性能：**（左上）**指数分布、**（右上）**对数正态分布，以及**（底部）**确定性。每个图比较三种数据设置下的收敛情况：同质（独立同分布）、异质（狄利克雷 $\alpha = 0.2$）和高度异质（每个客户端3个类别）。

I 命题 4 的证明

证明分为三个不同阶段。首先，我们推导K_ϵ 个全局轮次的预期耗时（10）。接下来，我们给出系统吞吐量的显式表达式（11）。最后，我们计算该吞吐量的精确解析梯度（12）。

I.1 方程 (10) 的证明

回顾一下，$(T_k)_{k\in\mathbb{N}}$ 表示服务器 $(\mathrm{u}_i)_{i=1}^n$ 上的服务完成时间序列，其中$T_0 = 0$。在此背景下，T_k 表示第 k 轮的开始，因此该轮持续时间为 $T_{k+1} - T_k$。根据定义，K_ϵ 轮的总时间为：

$$\tau_\epsilon = \sum_{k=1}^{K_\epsilon(p,m)} (T_k - T_{k-1}). \tag{61}$$

设 $N = (N(t))_{t\geq 0}$ 为与点过程 $(T_k)_{k\in\mathbb{N}}$ 相关联的计数过程。该过程记录服务完成的总次数，可表示为上行队列大小正向递减量之和：

$$N(t) = \sum_{0<s\leq t} \sum_{i=1}^{n} \left(\xi_i^{\mathrm{u}}(s^-) - \xi_i^{\mathrm{u}}(s)\right)^+.$$

由于 ξ 是平稳的，N 是平稳点过程。因此，在 Palm 概率测度 $\mathbb{P}^0$ 下，事件间隔时间同分布。于是，对于所有 $k \geq 1$，$\mathbb{E}^0[T_k - T_{k-1}] = \mathbb{E}^0[T_1]$。

应用平稳点过程的反演公式（参见 [51, 推论 6.16],取 $X_t = 1$），我们得到：

$$\lambda\, \mathbb{E}^0[T_1] = 1,$$

其中，λ 表示 $(N(t))_{t\geq 0}$ 在 $\mathbb{P}$下的平均强度。因此，在方程 (61) 中对 $\mathbb{P}^0$ 取期望可得：

$$\mathbb{E}^0[\tau_\epsilon] = \sum_{k=1}^{K_\epsilon(p,m)} \mathbb{E}^0[T_k - T_{k-1}] = \frac{K_\epsilon(p,m)}{\lambda(p,m)}.$$

这就证明了方程 (10)。

I.2 方程 (11) 的证明

在第 2.6 节的网络模型中，强度 $\lambda(p,m)$ 定义为所有上行服务器 $(\mathrm{u}_i)_{i=1}^n$ 的瞬时服务完成率之和的期望。根据方程 (33) 中给出的生成元，服务器 u_i 的瞬时速率为 $\mu_i^{\mathrm{u}}\xi_i^{\mathrm{u}}$。对所有 i 求和，并在平稳测度 $\mathbb{P}$ 下取期望，可得：

$$\lambda(p,m) = \sum_{i=1}^{n} \mu_i^{\mathrm{u}}\, \mathbb{E}[\xi_i^{\mathrm{u}}] = \frac{Z_{n,m-1}}{Z_{n,m}}.$$

第二个等式在 C 节中得以证明，并且可直接由公式（37）推出。由此证明了公式（11）。

I.3 公式（12）的证明

利用公式（44）的第一部分，我们得到：

$$\frac{\partial Z_{n,m-1}}{\partial p_j} = \frac{Z_{n,m-1}}{p_j}\mathbb{E}[X_j^{\mathrm{d}} + X_j^{\mathrm{c}} + X_j^{\mathrm{u}}].$$

遵循与 E.2节相同的逻辑（具体而言，应用平稳分布 $\pi_{n,\,m}$ 的推导），我们可以类似地得到：

$$\frac{\partial Z_{n,m}}{\partial p_j} = \frac{Z_{n,m}}{p_j}\mathbb{E}[\xi_j^{\mathrm{d}} + \xi_j^{\mathrm{c}} + \xi_j^{\mathrm{u}}].$$

对吞吐量 $\lambda(p,\ m) = Z_{n,\ m-1}/Z_{n,\ m}$ 的公式（11）求导可得：

$$\begin{aligned}\frac{\partial}{\partial p_j}\lambda(p,m) &= \frac{\partial}{\partial p_j}\left(\frac{Z_{n,m-1}}{Z_{n,m}}\right)\\ &= \frac{Z_{n,m-1}}{Z_{n,m}}\left(\frac{1}{Z_{n,m-1}}\frac{\partial Z_{n,m-1}}{\partial p_j} - \frac{1}{Z_{n,m}}\frac{\partial Z_{n,m}}{\partial p_j}\right)\\ &= \frac{\lambda(p,m)}{p_j}\left(\mathbb{E}[X_j^{\mathrm{d}} + X_j^{\mathrm{c}} + X_j^{\mathrm{u}}] - \mathbb{E}[\xi_j^{\mathrm{d}} + \xi_j^{\mathrm{c}} + \xi_j^{\mathrm{u}}]\right),\end{aligned}$$

至此，证明完毕。

J 优化时间复杂度：优化结果

如第5.3.2节所述，基于定理2和命题 4，我们推导出达到 ϵ准确率所需期望时间 $\mathbb{E}^0[\tau_\epsilon]$的闭式表达式，该表达式是路由向量$p$ 和并发级别 m的函数。此外，我们还可以计算其关于路由概率的精确梯度。为找到最优参数对（$p^{*\tau}$，$m^{*\tau}$），我们采用序贯优化方法最小化$\mathbb{E}^0[\tau_\epsilon]$，以考虑 m的离散特性。

从 $m = 2$开始，我们使用 Adam优化器 [28]，针对每个固定的 m 确定最优路由向量 p^*。为加快收敛，我们采用热启动策略：在级别 $m+1$ 上的优化以级别 m处找到的最优向量 p^* 作为初始值。当目标函数 $\mathbb{E}^0[\tau_\epsilon]$ 不再下降时，序贯搜索便终止，这表明已找到最优并发度 $m^{*\tau}$。

图8展示了序贯优化过程中，优化后的期望时间 $\mathbb{E}^0[\tau_\epsilon](p^*,\ m)$ 随 m 的变化。宏观曲线呈现出初期急剧下降的趋势，表明引入适度的并发度，借助并行处理的优势，可以大幅缩短达到 ϵ-平稳驻点所需的时间。然而，随着 m 持续增加，系统的边际收益递减，曲线也趋于平缓。放大后的插图展示了这一渐近区域的精确变化：期望时间在 $m^{*\tau} = 91$ 处达到严格的绝对最小值。超过这一最优并发度后，曲线开始上升，证实进一步提高并行性带来的计算收益最终会被陈旧效应抵消。搜索恰好在这一最小值处停止，从而得到使系统效率最大化的最优配置（$m^{*\tau}$，$p^{*\tau}$）。

K 优化时间复杂度：补充实验

为进一步验证第5.3节得出的见解，我们将实验评估扩展到难度更高的CIFAR-100数据集。我们保留了所述的实验设置

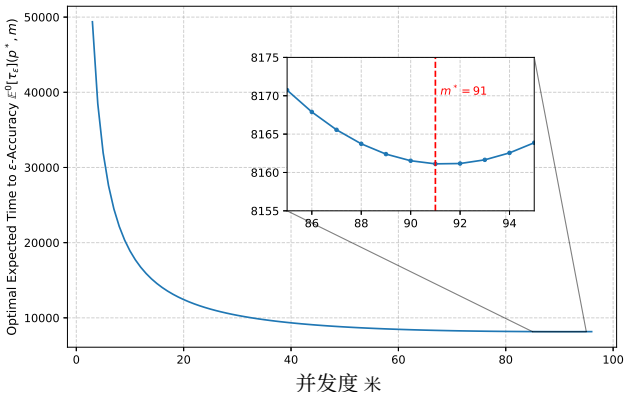


图8 达到 ϵ-准确率的优化期望时间，$\mathbb{E}^0[\tau_\epsilon](p^*,\ m)$，随并发级别 m 变化的函数曲线。该曲线展示了并行加速与梯度陈旧性之间的基本权衡，插图突出显示了在最优并发度 $m^* = 91$ 下达到的最小值。

第 5.3.1 节，采用表 1 中相同的网络动态和客户端配置。与第5.3节一样，我们考虑独立同分布和非独立同分布数据分布（使用狄利克雷分布建模，其中 $\alpha = 0.2$），并在三种服务时间分布下评估性能：指数分布、对数正态分布和确定性分布。

我们比较三种策略：(i) **时间优化**配置（$p^{*\tau}$，$m^{*\tau}$），(ii)**轮次优化**配置（p^{*K}，$m = n$），以及 (iii) 采用均匀路由的基线 `AsyncSGD`（p^{uni}，$m = n$），与表2中的设置一致。

图9展示了所有配置的测试损失和准确率随实际运行时间的变化。此外，表8报告了相对于基线方法，达到 0.5 目标准确率所需实际运行时间的缩减百分比。

总体而言，这些结果与第5.3节中的观察结果高度一致。时间优化策略始终实现最佳的实际时间性能，这证实了在将缩短训练时间作为首要目标时，该策略具有鲁棒性和有效性。

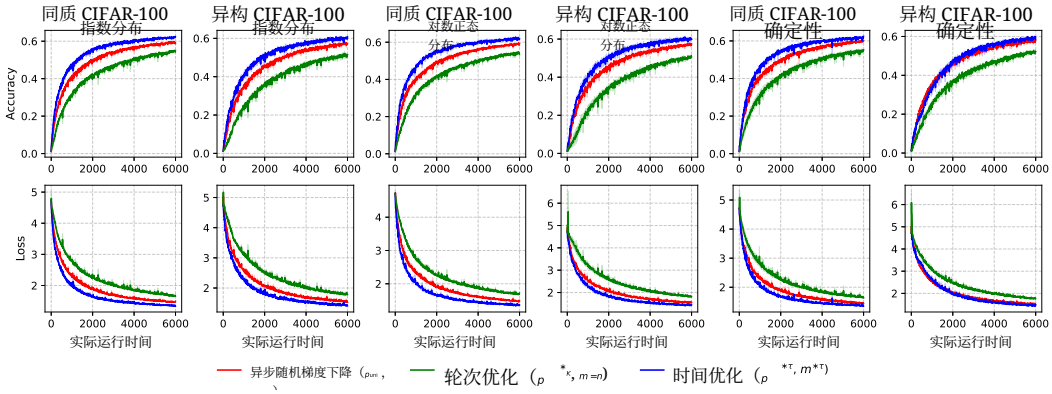


图9 第 5.3.1 节所述 CIFAR-100 场景下的测试集性能。上排显示测试准确率，下排显示损失，两者均以实际运行时间为横轴绘制。各列对应同质（独立同分布）和异质（非独立同分布）数据设置下的不同服务时间分布。每个子图均比较三种策略。模拟重复进行 3 次，每次持续 6000 个实际运行时间单位。实线表示均值，阴影区域表示标准差。

表8 在 CIFAR-100 上达到目标测试准确率 0.5 时，时间优化策略相较于基线方法的时间缩减百分比。

场景		时间缩减（%），相较于	
分布。	Data	轮次优化	异步随机梯度下降
Exp.	IID	67.95	40.04
	非独立同分布	60.52	31.26
对数正态分布	IID	67.31	40.01
	非独立同分布	65.46	28.86
Det.	IID	66.23	38.05
	非独立同分布	51.79	8.83

L 命题 5 的证明

请注意， $P(t)$表示系统在实际运行时间 t时的瞬时功耗。与第6.1节中引入的能量模型一致，瞬时功率定义如下：

$$P(t) = \sum_{i=1}^{n} \left(\mathcal{P}_i^{\mathrm{c}} \, \mathbf{1}\{\xi_i^{\mathrm{c}}(t) > 0\} + \mathcal{P}_i^{\mathrm{u}} \, \xi_i^{\mathrm{u}}(t) + \mathcal{P}_i^{\mathrm{d}} \, \xi_i^{\mathrm{d}}(t) \right). \tag{62}$$

令 τ_ϵ 和 K_ϵ 分别表示达到 ϵ 准确率所需的实际运行时间和全局轮次数。注意， K_ϵ 是由路由向量 p、并发级别 m 和服务速率 μ 确定的确定性整数，而 τ_ϵ 是一个随机变量。达到 ϵ 准确率之前的总能耗为

$$E_\epsilon = \int_0^{\tau_\epsilon} P(t)\, dt = \sum_{k=1}^{K_\epsilon(p,m)} \int_{T_{k-1}}^{T_k} P(t)\, dt,$$

其中，$(T_k)_{k\in\mathbb{N}}$ 表示模型参数更新时间序列（即上行服务器 $(\mathrm{u}_i)_{i=1}^n$ 上的服务完成时间），且 $T_0 = 0$。

由于系统状态过程 $(\xi(t))_{t\geq 0}$ 平稳且遍历，更新区间 $[T_{k-1}, T_k)$ 上的能量增量序列构成一个平稳序列。因此，在与 $\{T_k\}$ 关联的 Palm 概率测度下，每轮能耗的期望为常数：

$$\mathbb{E}^0\left[\int_{T_{k-1}}^{T_k} P(t)\, dt\right] = \mathbb{E}^0\left[\int_0^{T_1} P(t)\, dt\right].$$

应用平稳点过程的反演公式（见 [51, 推论 6.16]，取 $X_t = P(t)$）可得

$$\mathbb{E}^0\left[\int_0^{T_1} P(t)\, dt\right] = \frac{\mathbb{E}[P(0)]}{\lambda(p,m)},$$

其中 λ 表示公式（11）中的系统吞吐量（更新频率）。将此表达式代入 E_ϵ 的求和式可得

$$\mathbb{E}^0[E_\epsilon] = \sum_{k=1}^{K_\epsilon(p,m)} \frac{\mathbb{E}[P(0)]}{\lambda} = \frac{K_\epsilon(p,m)}{\lambda(p,m)}\, \mathbb{E}[P(0)]. \tag{63}$$

接下来需要计算稳态期望功率 $\mathbb{E}[P(0)]$。对（62）关于平稳分布$\pi_{n,\,m}$ 取期望，并利用线性性质可得

$$\mathbb{E}[P(0)] = \sum_{i=1}^{n} \left(\mathcal{P}_i^{\mathrm{c}}\, \mathbb{P}(\xi_i^{\mathrm{c}} > 0) + \mathcal{P}_i^{\mathrm{d}}\, \mathbb{E}[\xi_i^{\mathrm{d}}] + \mathcal{P}_i^{\mathrm{u}}\, \mathbb{E}[\xi_i^{\mathrm{u}}] \right).$$

现在，我们利用前文确立的结果来计算这些项。对于计算队列，将 (45) 应用于总体规模为 m 的系统，其中 $k = 1$ （注意，(45) 适用于$X \sim \pi_{n,\,m-1}$，而此处为 $\xi \sim \pi_{n,\,m}$），可得

$$\mathbb{P}(\xi_i^{\mathrm{c}} > 0) = \frac{Z_{n,m-1}}{Z_{n,m}} \frac{p_i}{\mu_i^{\mathrm{c}}}.$$

S类似地，利用 (47) 中推导出的平稳表达式及其上行链路对应式，可得

$$\mathbb{E}[\xi_i^{\mathrm{d}}] = \frac{Z_{n,m-1}}{Z_{n,m}} \frac{p_i}{\mu_i^{\mathrm{d}}}, \qquad \mathbb{E}[\xi_i^{\mathrm{u}}] = \frac{Z_{n,m-1}}{Z_{n,m}} \frac{p_i}{\mu_i^{\mathrm{u}}}.$$

将这些表达式代入 $\mathbb{E}[P(0)]$，并回顾 (11) 中的 $\lambda(p,\ m) = {}^{Z_{n,}\, Z_{n\ m}^{\,m-1}}$，归一化常数与 (63) 中的因子 $1/\lambda(p,\ m)$ 相消。因此得到

$$\mathbb{E}^0[E_\epsilon] = K_\epsilon(p, m) \sum_{i=1}^{n} p_i \left(\frac{\mathcal{P}_i^{\mathrm{c}}}{\mu_i^{\mathrm{c}}} + \frac{\mathcal{P}_i^{\mathrm{d}}}{\mu_i^{\mathrm{d}}} + \frac{\mathcal{P}_i^{\mathrm{u}}}{\mu_i^{\mathrm{u}}} \right),$$

由此完成证明。

M 公式 16 的证明

我们要寻找使目标函数最小化的最优概率分布 p^{*E}：

$$\left(\sum_{j=1}^{n} \frac{1}{p_j} \right) \left(\sum_{i=1}^{n} p_i \mathcal{E}_i \right).$$

回顾柯西-施瓦茨不等式：对于任意实数序列（a_j）和（b_j），都有：

$$\left(\sum_{j=1}^{n} a_j^2 \right) \left(\sum_{j=1}^{n} b_j^2 \right) \geq \left(\sum_{j=1}^{n} a_j b_j \right)^2.$$

通过定义序列 $a_j = {}^{1}_{\sqrt{p_j}}$ 和 $b_j = \sqrt{p_j \mathcal{E}_j}$，我们可以得到目标函数的如下下界：

$$\left(\sum_{j=1}^{n} \frac{1}{p_j} \right) \left(\sum_{j=1}^{n} p_j \mathcal{E}_j \right) \geq \left(\sum_{j=1}^{n} \frac{1}{\sqrt{p_j}} \sqrt{p_j \mathcal{E}_j} \right)^2 = \left(\sum_{j=1}^{n} \sqrt{\mathcal{E}_j} \right)^2.$$

因此，目标函数的下界为常数 $\left(\sum_{j=1}^{n} \sqrt{\mathcal{E}_j} \right)^2$。 —

当柯西-施瓦茨等式成立时，最小值即可达到；当且仅当序列（a_j）和（b_j）线性成比例（$a_j \propto b_j$）时，该等式成立。代入我们的定义，要求：

$$\frac{1}{\sqrt{p_j}} \propto \sqrt{p_j \mathcal{E}_j}.$$

对 p_j 进行整理，可得最优路由概率（相差一个比例常数）：

$$p_i^{*E} \propto \frac{1}{\sqrt{\mathcal{E}_i}}.$$

由于 p^{*E} 必须是有效的概率分布（$\sum_{i=1}^n p_i^{*E} = 1$），我们对其进行归一化，得到精确的闭式解：

$$p_i^{*E} = \frac{\mathcal{E}_i^{-1/2}}{\sum_{j=1}^n \mathcal{E}_j^{-1/2}}, \quad i \in \{1, \dots, n\},$$

由此得证。

N 能耗复杂度优化：补充实验

在第 6.5.3 节所述结果的基础上，我们通过提供 EMNIST 数据集的详细学习轨迹，进一步展示 `Generalized AsyncSGD` 中的时间与能耗权衡。我们针对不同的服务时间分布（指数分布、对数正态分布和确定性分布），评估同质和异质（狄利克雷分布，$\alpha = 0.2$）数据分区。客户端速度和功率配置分别按照表 1 和表 4 进行设置。沿用第 5.3 节的整体实验设置，我们评估以下五种不同策略的收敛动态：

- **最大吞吐量：** 在完全并发（$m = n$）的情况下最大化系统更新频率（$p^{*\lambda}$），而不考虑梯度陈旧性对收敛质量的影响。**时间优化：** 使用最优参数（$p^{*\tau}, m^{*\tau}$），将梯度陈旧性与更新频率结合起来。如命题 4 所述，无论最终能耗如何，该策略都能严格最小化达到 ϵ 准确率所需的期望实际经过时间 $\mathbb{E}^0[\tau_\epsilon]$ 。**能耗优化：** 采用最优参数（$p^{*E}, m = 1$），严格最小化达到 ϵ准确率所需的期望能耗 $\mathbb{E}^0[E_\epsilon]$ （命题 5），而不考虑收敛速度。**时间与能耗协同优化：** 采用最优配置（$p^{*\rho}, m^{*\rho}$），最小化式（18）中定义的联合目标。我们设定 $\rho = 0.1$；如图 4 中的帕累托前沿所示，该策略通过以极小的收敛速度损失换取显著的能耗节省，在权衡中实现了策略性平衡。**标准基线（`AsyncSGD`）** [29, 算法 2]: 代表传统方法，采用完全并发（ $m = n$ ）和均匀路由概率（ p^{uni} ）。

图 10 展示了测试准确率相对于累计能耗（上排）和实际运行时间（下排）的变化过程。我们分析了每种路由策略的收敛动态，重点说明其不同的数学目标如何决定学习速度与能效之间的基本权衡。

尽管 **最大吞吐量** 策略实现了理论上最高的更新频率，但在两项指标上的表现都很差。由于过度偏向速度最快、能耗最高的客户端（E 型），该策略付出了巨大的能耗成本。此外，忽略梯度陈旧性严重阻碍了收敛。在异质（非独立同分布）场景下，这一问题尤其不利：较慢客户端的延迟更新会引发严重的客户端漂移，使优化朝相互冲突的方向推进。因此，最大吞吐量策略极不稳定，表现出严重的性能尖峰和较高的方差（图 10 中的橙色区域）。

时间优化 策略通过联合优化更新频率和梯度陈旧性，缓解了这种不稳定性。如图 10 底行所示，该策略在所有评估场景中始终实现最快收敛。然而，由于其目标函数完全不考虑功耗，因此会受到显著的能耗惩罚，表现明显落后于考虑能耗的策略。

相反，**能耗优化** 策略能够严格实现总能耗最小化。在能耗指标上（上排），它大幅优于所有其他策略，以最低功率实现了较高的准确率。然而，为避免并发通信带来的能耗开销，该策略强制采用严格的顺序处理（$m = 1$）。这造成了严重的时间代价，使其实际时间收敛慢得难以接受（在下排中几乎呈平坦状）。

标准 `AsyncSGD` 基线代表传统的、未经优化的方法。它依赖均匀路由概率和固定的完全并发，且不进行自适应调优，因此在两个方面的表现都平平。它始终比时间优化策略慢，同时消耗的能量也显著高于能耗感知策略。

最后，**时间与能耗协同优化** 策略有效地弥合了这两个极端。通过沿帕累托前沿（$\rho = 0.1$）寻找合适位置，它实现了高效的运行状态：其能效仅次于严格的能耗优化策略，而时间收敛表现仅次于严格的时间优化策略。尤为重要的是，这种联合策略严格优于 `AsyncSGD` 基线，在所有数据分布和服务时间分布下，都能以更高的准确率更快地收敛，同时显著降低能耗。

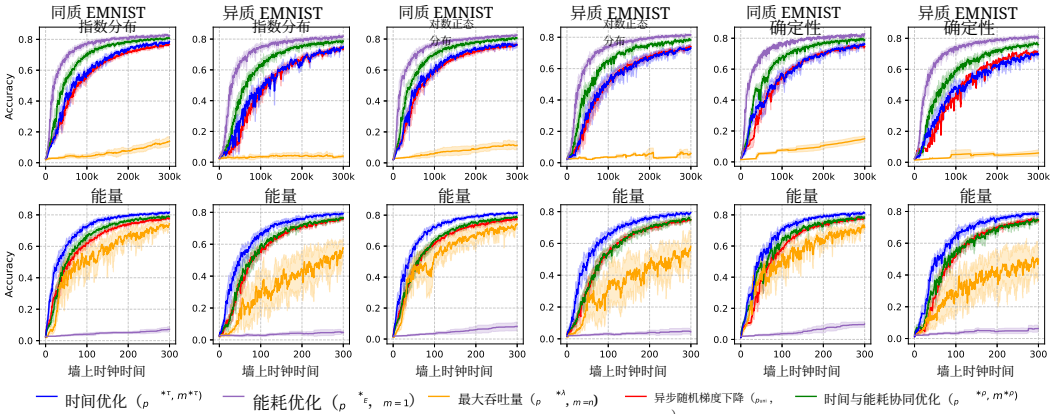


图 10 在第 5.3.1 节所述实验场景及表 4 中功率配置下，EMNIST 数据集上的学习轨迹。顶部行和底部行分别展示测试准确率随总能耗和实际运行时间的变化。各列表示同质（独立同分布）和异质（非独立同分布）数据分区下不同的服务时间分布。每个面板比较五种路由策略的收敛动态。结果取自 10 次独立运行的平均值，运行范围涵盖 300,000 个能量单位和 300 个时间单位；实线表示均值，阴影区域表示标准差。

O 命题 6 的证明

~虽然（Y（t））$_{t\geq 0}$ 和（Y_k）$_{k\in\mathbb{N}}$ 通过类别标签跟踪任务历史，但它们并非严格意义上的马尔可夫链。由于只记录每个类别中的任务总数，这些状态向量舍弃了刻画中央服务器处先进先出动态所需的具体到达顺序。

为了确定平稳分布并便于应用利特尔定律，我们必须首先扩展状态描述，以恢复马尔可夫性质。本证明分为三个步骤。首先，我们引入一个详细的马尔可夫状态空间，明确纳入中央服务器队列中任务的顺序，并推导其平稳分布。其次，我们对该详细分布求边缘分布，以证明连续时间过程（$Y(t)$）$_{t\geq 0}$~服从$\phi_{n,\,m}$。最后，我们证明离散时间嵌入链（Y_k）$_{k\in\mathbb{N}}$ 以$\phi_{n,\,m-1}$ 为其平稳分布。

O.1 马尔可夫描述

为了恢复对系统动态的完整马尔可夫性描述，必须显式跟踪中央服务器（CS）处任务的排序。事实上，聚合状态表示会丢弃中央服务器处的先进先出顺序，而这对于正确刻画未来转移至关重要。因此，我们引入一个增广状态过程，同时记录任务计数和排序信息。具体而言，我们定义如下右连续（右连续且有左极限）随机过程

$$Z(t) = \big(Z^{\mathrm{cs}}(t),\, Z_1^{\mathrm{d}}(t),\, Z_1^{\mathrm{c}}(t),\, Z_1^{\mathrm{u}}(t), \ldots, Z_n^{\mathrm{d}}(t),\, Z_n^{\mathrm{c}}(t),\, Z_n^{\mathrm{u}}(t)\big).$$

对于每个 $t \geq 0$和 $i \in \{1, \ldots, n\}$，分量 $Z_i^{\mathrm{d}}(t)$、$Z_i^{\mathrm{c}}(t)$ 和 $Z_i^{\mathrm{u}}(t)$ 分别表示服务器 d_i、c_i 和 u_i 上的任务数量。分量 $Z^{\mathrm{cs}}(t)$记录 CS队列中任务类别的有序序列，该队列遵循先进先出规则。形式上，

$$Z^{\mathrm{cs}}(t) = \big(Z_1^{\mathrm{cs}}(t), Z_2^{\mathrm{cs}}(t), \ldots, Z_{\ell(Z^{\mathrm{cs}}(t))}^{\mathrm{cs}}(t)\big),$$

其中，$Z_j^{\mathrm{cs}}(t) \in \{1, \ldots, n\}$ 表示位于位置 j 的任务类别（其中 $j = 1$ 对应正在服务的任务），而 $\ell(Z^{\mathrm{cs}}(t))$ 表示当前位于中央服务器（CS）的任务总数（即该序列的长度）。

过程 $Z(t)$在一个满足任务总数守恒的状态空间上演化。具体而言，其取值为

$$\mathcal{S}_m = \Bigg\{ z = \big(z^{\mathrm{cs}},\, z_1^{\mathrm{d}},\, z_1^{\mathrm{c}},\, z_1^{\mathrm{u}}, \ldots, z_n^{\mathrm{d}},\, z_n^{\mathrm{c}},\, z_n^{\mathrm{u}}\big) \in \mathcal{Z}_m \times \mathcal{X}_{3n,\leq m} \;: \\ \ell(z^{\mathrm{cs}}) + \sum_{i=1}^{n} (z_i^{\mathrm{d}} + z_i^{\mathrm{c}} + z_i^{\mathrm{u}}) = m \Bigg\}. \quad (64)$$

其中，$\mathcal{Z}_m$ 表示长度至多为 m的有限类别序列集合，

$$\mathcal{Z}_m = \{(z_1, \ldots, z_\ell) : 0 \leq \ell \leq m,\, z_j \in \{1, \ldots, n\} \text{ for all } j\},$$

而$\mathcal{X}_{3n,\leq m}$ 表示由 $3n$维非负整数项向量组成、且各项之和至多为 m的集合：

$$\mathcal{X}_{3n,\leq m} = \{x \in \mathbb{N}^{3n} : |x| \leq m\}.$$

通过这一扩充的状态空间，系统具有一种易于处理平稳行为的马尔可夫性描述，具体表述如下引理。

▶**引理18**。 在第*7.1*节的模型中，过程$(Z(t))_{t\geq 0}$ 是一个不可约、正常返的连续时间马尔可夫链。其平稳分布为

$$\zeta_{n,m}(z) = \frac{1}{W_{n,m}} \prod_{i=1}^{n} \left(\frac{p_i}{\mu^{\mathrm{CS}}}\right)^{|z^{\mathrm{cs}}|_i} \left(\frac{p_i}{\mu_i^{\mathrm{c}}}\right)^{z_i^{\mathrm{c}}} \frac{1}{z_i^{\mathrm{d}}!} \left(\frac{p_i}{\mu_i^{\mathrm{d}}}\right)^{z_i^{\mathrm{d}}} \frac{1}{z_i^{\mathrm{u}}!} \left(\frac{p_i}{\mu_i^{\mathrm{u}}}\right)^{z_i^{\mathrm{u}}}, \quad (65)$$

对于所有 $z \in \mathcal{S}_m$，其中 $W_{n,\,m}$ 是归一化常数。这里，$|z^{\mathrm{cs}}|_i$ 表示序列 z^{cs} 中类别-i 任务的数量，定义为 $|z^{\mathrm{cs}}|_i = \sum_{j=1}^{\ell(z^{\mathrm{cs}})} \mathbf{1}\{z_j^{\mathrm{cs}} = i\}$。

证明。 我们首先证明不可约性和正常返性。由于所有路由概率p_i 都严格为正，且网络拓扑强连通，因此 $\mathcal{S}_m$ 中的任意状态都可以从其他任意状态到达。由于 $\mathcal{S}_m$ 是有限的，该马尔可夫链不可约且正常返。

令 r 表示 $(Z(t))_{t\geq 0}$ 的转移率。对于任意状态 z 和类别索引 $i \in \{1,\dots,n\}$，令 $\ell = \ell(z^{\mathrm{cs}})$ 表示 CS队列中的当前任务数。我们定义以下状态变换：

- z^{+i}：将一个类别为 i 的任务追加到序列 z^{cs} 的末尾（即，新序列为 $(z_1^{\mathrm{cs}},\dots,z_\ell^{\mathrm{cs}},i)$）。当 $\ell < m$ 时有效。z_{+i}：将一个类别为 i 的任务插入到 z^{cs} 的开头（即，新序列为 $(i, z_1^{\mathrm{cs}},\dots,z_\ell^{\mathrm{cs}})$）。当 $\ell < m$ 时有效。z^{-i}：移除 z^{cs} 中的最后一个任务。当 $\ell > 0$ 且 $z_\ell^{\mathrm{cs}} = i$ 时有效。z_{-i}：移除 z^{cs} 中的第一个任务。当 $\ell > 0$ 且 $z_1^{\mathrm{cs}} = i$ 时有效。

$(Z(t))_{t\geq 0}$ 的非零转移率为：

$$\begin{cases} r(z,\, z - \mathbf{e}_i^{\mathrm{d}} + \mathbf{e}_i^{\mathrm{c}}) = \mu_i^{\mathrm{d}}\, z_i^{\mathrm{d}}, \\ r(z,\, z - \mathbf{e}_i^{\mathrm{c}} + \mathbf{e}_i^{\mathrm{u}}) = \mu_i^{\mathrm{c}}\, \mathbf{1}\{z_i^{\mathrm{c}} > 0\}, \\ r(z,\, z^{+i} - \mathbf{e}_i^{\mathrm{u}}) = \mu_i^{\mathrm{u}}\, z_i^{\mathrm{u}}, \\ r(z,\, z_{-i} + \mathbf{e}_j^{\mathrm{d}}) = \mu^{\mathrm{CS}}\, \mathbf{1}\{z_1^{\mathrm{cs}} = i\}\, p_j, \end{cases} \qquad i,j \in \{1,\dots,n\}. \tag{66}$$

从状态 z 离开的总速率为：

$$r(z) = \sum_{y\in\mathcal{S}_m} r(z,y) = \sum_{i=1}^{n} \mu_i^{\mathrm{d}} z_i^{\mathrm{d}} + \mu_i^{\mathrm{c}} \mathbf{1}\{z_i^{\mathrm{c}} > 0\} + \mu_i^{\mathrm{u}} z_i^{\mathrm{u}} + \mu^{\mathrm{cs}} \mathbf{1}\{z_1^{\mathrm{cs}} = i\} \underbrace{\sum_{j=1}^{n} p_j}_{=1}$$

$$= \mu^{\mathrm{CS}} \mathbf{1}\{\ell(z^{\mathrm{cs}}) \neq 0\} + \sum_{i=1}^{n} \left(\mu_i^{\mathrm{d}} z_i^{\mathrm{d}} + \mu_i^{\mathrm{c}} \mathbf{1}\{z_i^{\mathrm{c}} > 0\} + \mu_i^{\mathrm{u}} z_i^{\mathrm{u}}\right).$$

为证明引理18，我们调用反转检验 [7, 定理13.4.13]。我们根据候选测度 $\zeta_{n,m}$ 构造反向转移率 $\tilde{r}$，并验证总速率方程 $r(z) = \sum_y \tilde{r}(z,y)$是否成立。若成立，$\zeta_{n,m}$ 即为不变测度。

对于每种转移类型，我们推导反向速率 ~~$\tilde{r}(z,y)$~~ $= \frac{\zeta_{n,m}(y)}{\zeta_{n,m}(z)} r(y,z)$：

1. **计算 $\to$ 上行链路：**

$$\begin{aligned} \tilde{r}(z,\, z - \mathbf{e}_i^{\mathrm{u}} + \mathbf{e}_i^{\mathrm{c}}) &= \frac{\zeta_{n,m}(z - \mathbf{e}_i^{\mathrm{u}} + \mathbf{e}_i^{\mathrm{c}})}{\zeta_{n,m}(z)}\, r(z - \mathbf{e}_i^{\mathrm{u}} + \mathbf{e}_i^{\mathrm{c}},\, z) \\ &= \frac{p_i/\mu_i^{\mathrm{c}}}{(1/z_i^{\mathrm{u}})(p_i/\mu_i^{\mathrm{u}})}\, \mu_i^{\mathrm{c}} \mathbf{1}\{z_i^{\mathrm{u}} \geq 1\} \\ &= \mu_i^{\mathrm{u}} z_i^{\mathrm{u}}. \end{aligned}$$

2. **下行链路 $\to$计算：**

$$\begin{aligned} \tilde{r}(z,\, z - \mathbf{e}_i^{\mathrm{c}} + \mathbf{e}_i^{\mathrm{d}}) &= \frac{\zeta_{n,m}(z - \mathbf{e}_i^{\mathrm{c}} + \mathbf{e}_i^{\mathrm{d}})}{\zeta_{n,m}(z)}\, r(z - \mathbf{e}_i^{\mathrm{c}} + \mathbf{e}_i^{\mathrm{d}},\, z) \\ &= \frac{(1/(z_i^{\mathrm{d}}+1))(p_i/\mu_i^{\mathrm{d}})}{p_i/\mu_i^{\mathrm{c}}}\, \mu_i^{\mathrm{d}}(z_i^{\mathrm{d}}+1) \mathbf{1}\{z_i^{\mathrm{c}} \geq 1\} \\ &= \mu_i^{\mathrm{c}} \mathbf{1}\{z_i^{\mathrm{c}} > 0\}. \end{aligned}$$

3. **到达中央服务器：**

$$\begin{aligned}\tilde{r}(z,\, z^{-i}+\mathbf{e}_i^{\mathrm{u}}) &= \frac{\zeta_{n,m}(z^{-i}+\mathbf{e}_i^{\mathrm{u}})}{\zeta_{n,m}(z)}\, r(z^{-i}+\mathbf{e}_i^{\mathrm{u}},\, z)\\ &= \frac{(1/(z_i^{\mathrm{u}}+1))(p_i/\mu_i^{\mathrm{u}})}{(p_i/\mu^{\mathrm{CS}})}\, \mu_i^{\mathrm{u}}(z_i^{\mathrm{u}}+1)\mathbf{1}\{z^{\mathrm{cs}}_{|z^{\mathrm{cs}}|}=i\}\\ &= \mu^{\mathrm{CS}}\mathbf{1}\{z^{\mathrm{cs}}_{\ell(z^{\mathrm{cs}})}=i\}.\end{aligned}$$

4. **离开中央服务器：**

$$\begin{aligned}\tilde{r}(z,\, z_{+i}-\mathbf{e}_j^{\mathrm{d}}) &= \frac{\zeta_{n,m}(z_{+i}-\mathbf{e}_j^{\mathrm{d}})}{\zeta_{n,m}(z)}\, r(z_{+i}-\mathbf{e}_j^{\mathrm{d}},\, z)\\ &= \frac{(p_i/\mu^{\mathrm{CS}})}{(1/z_j^{\mathrm{d}})(p_j/\mu_j^{\mathrm{d}})}\, \mu^{\mathrm{CS}}\, p_j\, \mathbf{1}\{z_j^{\mathrm{d}}\geq 1\}\\ &= p_i\, \mu_j^{\mathrm{d}} z_j^{\mathrm{d}}.\end{aligned}$$

接下来，对所有可能的转移求和，得到总反向速率：

$$\begin{aligned}\tilde{r}(z) &= \sum_{y\in\mathcal{S}_m}\tilde{r}(z,y) = \sum_{i=1}^{n}\Big(\mu_i^{\mathrm{c}}\,\mathbf{1}\{z_i^{\mathrm{c}}>0\}+\mu_i^{\mathrm{u}}z_i^{\mathrm{u}}+\mu^{\mathrm{cs}}\,\mathbf{1}\{z^{\mathrm{cs}}_{\ell(z^{\mathrm{cs}})}=i\}+p_i\sum_{j=1}^{n}\mu_j^{\mathrm{d}}z_j^{\mathrm{d}}\Big)\\ &= \mu^{\mathrm{cs}}\sum_{i=1}^{n}\mathbf{1}\{z^{\mathrm{cs}}_{\ell(z^{\mathrm{cs}})}=i\}+\sum_{i=1}^{n}\Big(\mu_i^{\mathrm{c}}\,\mathbf{1}\{z_i^{\mathrm{c}}>0\}+\mu_i^{\mathrm{u}}z_i^{\mathrm{u}}+\mu_i^{\mathrm{d}}z_i^{\mathrm{d}}\sum_{j=1}^{n}p_j\Big)\\ &= \mu^{\mathrm{cs}}\,\mathbf{1}\{\ell(z^{\mathrm{cs}})\neq 0\}\;+\;\sum_{i=1}^{n}\mu_i^{\mathrm{d}}z_i^{\mathrm{d}}+\mu_i^{\mathrm{c}}\,\mathbf{1}\{z_i^{\mathrm{c}}>0\}+\mu_i^{\mathrm{u}}z_i^{\mathrm{u}}=r(z).\end{aligned}$$

由于对所有 $z\in\mathcal{S}_m$，$r(z)=\tilde{r}(z)$，反转检验定理 [7, 定理 13.4.13] 表明，$\zeta_{n,m}$ 确实是马尔可夫链 $(Z(t))_{t\geq 0}$ 的平稳分布。

◀

O.2 连续时间过程 $(Y(t))_{t\geq 0}$ 的平稳分布

利用引理18中给出的详细马尔可夫过程 $(Z(t))_{t\geq 0}$ 的平稳分布，通过边缘化推导出聚合过程 $(Y(t))_{t\geq 0}$ 的平稳分布。

对于任意可容许状态

$$x=\big(x_j^{\mathrm{cs}},\, x_j^{\mathrm{d}},\, x_j^{\mathrm{c}},\, x_j^{\mathrm{u}}\,;\; j\in\{1,\dots,n\}\big)\in\mathcal{X}_{4n,m},$$

稳态概率 $\mathbb{P}(Y=x)$ 可通过对所有投影到 x 的详细状态 $z\in\mathcal{S}_m$ 的 $\zeta_{n,m}(z)$ 求和得到。等价地，这对应于对所有类别序列 z^{cs} 求和，其中每个类别 $i\in\{1,\dots,n\}$ 恰好包含 x_i^{cs} 个该类别的任务 i。因此：

$$\begin{aligned}\mathbb{P}(Y=x) &= \sum_{\substack{z^{\mathrm{cs}}\in\mathcal{Z}_m\\ |z^{\mathrm{cs}}|_i=x_i^{\mathrm{cs}},\,\forall i}}\zeta_{n,m}\big(z^{\mathrm{cs}},\, x_j^{\mathrm{d}},\, x_j^{\mathrm{c}},\, x_j^{\mathrm{u}}\,;\; j\in\{1,\dots,n\}\big)\\ &= \frac{1}{W_{n,m}}\left[\prod_{j=1}^{n}\left(\frac{p_j}{\mu^{\mathrm{CS}}}\right)^{x_j^{\mathrm{cs}}}\left(\frac{p_j}{\mu_j^{\mathrm{c}}}\right)^{x_j^{\mathrm{c}}}\frac{1}{x_j^{\mathrm{d}}!}\left(\frac{p_j}{\mu_j^{\mathrm{d}}}\right)^{x_j^{\mathrm{d}}}\frac{1}{x_j^{\mathrm{u}}!}\left(\frac{p_j}{\mu_j^{\mathrm{u}}}\right)^{x_j^{\mathrm{u}}}\right]\underbrace{\sum_{\substack{z^{\mathrm{cs}}\in\mathcal{Z}_m\\ |z^{\mathrm{cs}}|_i=x_i^{\mathrm{cs}},\;\forall i}}1}_{\text{相容序列的数量}}\end{aligned}$$

$$= \frac{1}{W_{n,m}} \left[\prod_{j=1}^{n} \left(\frac{p_j}{\mu^{\mathrm{CS}}} \right)^{x_j^{\mathrm{cs}}} \left(\frac{p_j}{\mu_j^{\mathrm{c}}} \right)^{x_j^{\mathrm{c}}} \frac{1}{x_j^{\mathrm{d}}!} \left(\frac{p_j}{\mu_j^{\mathrm{d}}} \right)^{x_j^{\mathrm{d}}} \frac{1}{x_j^{\mathrm{u}}!} \left(\frac{p_j}{\mu_j^{\mathrm{u}}} \right)^{x_j^{\mathrm{u}}} \right] \frac{(\sum_{i=1}^{n} x_i^{\mathrm{cs}})!}{\prod_{i=1}^{n} x_i^{\mathrm{cs}}!}$$

$$= \phi_{n,m}(x).$$

第二个等式根据 (65) 中 $\zeta_{n,m}$ 的定义得出；注意，概率质量仅取决于序列 z^{cs} 中各类别的计数，而与其具体顺序无关。因此，方括号中的项可从求和式中提出。第三个等式计算了带下划线的求和项，它统计长度为 $\sum_i x_i^{\mathrm{cs}}$ 的不同序列 z^{cs} 的数量，其中每个类别 i 恰好包含 x_i^{cs} 个元素。这个数量正好由多项式系数给出：

$$\binom{\sum_{i=1}^{n} x_i^{\mathrm{cs}}}{x_1^{\mathrm{cs}}, x_2^{\mathrm{cs}}, \ldots, x_n^{\mathrm{cs}}} = \frac{(\sum_{i=1}^{n} x_i^{\mathrm{cs}})!}{\prod_{i=1}^{n} x_i^{\mathrm{cs}}!}.$$

至此，便完成了命题 6 中所述 $(Y(t))_{t\geq 0}$ 的乘积形式推导。

˜O.3 离散时间过程 $(Y_k)_{k\in\mathbb{N}}$ 的平稳分布

现在，我们确定参数更新˜时刻观测到的系统状态的平稳分布，即离散时间过程 $(Y_k)_{k\in\mathbb{N}}$ 的平稳分布，并证明其不变分布为 $\phi_{n,m-1}$ 。

设 $(\hat{}Z_k)_{k\in\mathbb{N}}$ 表示与遍历连续时间马尔可夫过程 $(Z(t))_{t\geq 0}$ 相关联的跳链。根据引理 12，转移序列

$$\{(\hat{Z}_k, \hat{Z}_{k+1}) : k \in \mathbb{N}\}$$

构成一个不可约的离散时间齐次马尔可夫链，其不变测度为

$$\hat{\zeta}(u, v) = \zeta_{n,m}(u)\, r(u, v), \qquad u, v \in \mathcal{S}_m,$$

其中$\mathcal{S}_m$ 是过程 $(Z(t))_{t\geq 0}$ 的状态空间，如式（64）所定义； r 表示其转移率，如式（66）所给出；$\zeta_{n,m}$ 表示其平稳分布，由式（65）推导得出。

我们的平稳分布对任意状态

$$x = \left(x_i^{\mathrm{cs}},\, x_i^{\mathrm{d}},\, x_i^{\mathrm{c}},\, x_i^{\mathrm{u}}\,;\; i \in \{1, \ldots, n\}\right) \in \mathcal{X}_{4n,m-1},$$

˜，稳态概率 $\mathbb{P}(Y = x)$ 可表示为

$$\mathbb{P}(\tilde{Y} = x) = \mathbb{E}\left[\sum_{\substack{z^{\mathrm{cs}} \in \mathcal{Z}_{m-1} \\ |z^{\mathrm{cs}}|_l = x_l^{\mathrm{cs}}, \forall l}} \sum_{i=1}^{n} \sum_{j=1}^{n} \mathbf{1}\{\hat{Z}_k = z_{+i},\, \hat{Z}_{k+1} = z + \mathbf{e}_j^{\mathrm{d}}\} \;\middle|\; (\hat{Z}_k, \hat{Z}_{k+1}) \in H \right], \quad (67)$$

其中， z 是与 x 对应的详细状态（扩展加入序列 z^{cs}），即

$$z = \left(z^{\mathrm{cs}},\, x_i^{\mathrm{d}},\, x_i^{\mathrm{c}},\, x_i^{\mathrm{u}}\,;\; i \in \{1, \ldots, n\}\right) \in \mathcal{S}_{m-1},$$

且 H 表示对应于中央服务器（CS）服务完成的跳转链转移集合：

$$H = \left\{(u, v) \in \mathcal{S}_m \times \mathcal{S}_m : v = u_{-i} + \mathbf{e}_j^{\mathrm{d}} \text{ for some } i, j \in \{1, \ldots, n\}\right\}.$$

将推论14应用于方程（67），其中

$$g(u,v)=\sum_{\substack{z^{\mathrm{cs}}\in\mathcal{Z}_{m-1}\\ |z^{\mathrm{cs}}|_l=x_l^{\mathrm{cs}},\,\forall l}}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbf{1}\{u=z_{+i},\,v=z+\mathbf{e}_j^{\mathrm{d}}\},$$

对于每个 $x\in\mathcal{X}_{4n,m-1}$，我们得到：

$$\begin{aligned}\mathbb{P}(\tilde{Y}=x)&=\sum_{\substack{z^{\mathrm{cs}}\in\mathcal{Z}_{m-1}\\ |z^{\mathrm{cs}}|_l=x_l^{\mathrm{cs}},\,\forall l}}\frac{\displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n}\zeta_{n,m}(z_{+i})\,\mu^{\mathrm{cs}}p_j}{\displaystyle\sum_{v\in\mathcal{S}_m}\sum_{i=1}^{n}\sum_{j=1}^{n}\zeta_{n,m}(v)\,\mu^{\mathrm{cs}}\,\mathbf{1}\{v_1^{\mathrm{cs}}=i\}\,p_j}\\&=\frac{\left(\sum_{j=1}^{n}x_j^{\mathrm{cs}}\right)!}{\prod_{j=1}^{n}x_j^{\mathrm{cs}}!}\frac{\displaystyle\sum_{i=1}^{n}\zeta_{n,m}(z_{+i})\sum_{j=1}^{n}p_j}{\displaystyle\sum_{v\in\mathcal{S}_m}\sum_{i=1}^{n}\zeta_{n,m}(v)\,\mathbf{1}\{v_1^{\mathrm{cs}}=i\}\sum_{j=1}^{n}p_j}\\&=\frac{\left(\sum_{j=1}^{n}x_j^{\mathrm{cs}}\right)!}{\prod_{j=1}^{n}x_j^{\mathrm{cs}}!}\frac{\displaystyle\sum_{i=1}^{n}\zeta_{n,m}(z_{+i})}{\displaystyle\sum_{v\in\mathcal{S}_m}\zeta_{n,m}(v)\sum_{i=1}^{n}\mathbf{1}\{v_1^{\mathrm{cs}}=i\}}.\end{aligned}\tag{68}$$

第二个等式由 (65) 中 $\zeta_{n,m}$ 的定义得出。由于分子中的项仅取决于类别计数，而不取决于具体的序列 z^{cs}，对 z^{cs} 的求和可化简为计算不同排列的数量，该数量由多项式系数给出

$$\frac{\left(\sum_{i=1}^{n}x_i^{\mathrm{cs}}\right)!}{\prod_{i=1}^{n}x_i^{\mathrm{cs}}!}.$$

此外，因子 μ^{cs} 在分子和分母中相互抵消。第三个等式通过重新排列求和并利用 $\sum_{j=1}^{n}p_j=1$ 这一事实得出。注意，我们用 v_1^{cs} 表示队列的队首（正在服务的任务）。

现在我们来简化剩余项。首先，分析分母。对所有类别 i 的示性函数求和，等价于检查 CS队列是否非空：

$$\sum_{v\in\mathcal{S}_m}\zeta_{n,m}(v)\sum_{i=1}^{n}\mathbf{1}\{v_1^{\mathrm{cs}}=i\}=\sum_{v\in\mathcal{S}_m}\zeta_{n,m}(v)\,\mathbf{1}\{\ell(v^{\mathrm{cs}})\neq 0\}=\mathbb{P}\Big(\sum_{i=1}^{n}Y_i^{\mathrm{cs}}>0\Big),\tag{69}$$

其中 $Y\sim\phi_{n,m}$。利用 (20) 中 $\phi_{n,m}$ 的显式形式，CS 忙碌的概率为：

$$\mathbb{P}\Big(\sum_{i=1}^{n}Y_i^{\mathrm{cs}}>0\Big)=\sum_{x\in\mathcal{X}_{4n,m}}\phi_{n,m}(x)\,\mathbf{1}\Big\{\sum_{i=1}^{n}x_i^{\mathrm{cs}}>0\Big\}=\frac{1}{\mu^{\mathrm{cs}}}\frac{W_{n,m-1}}{W_{n,m}}.\tag{70}$$

另一方面，对于涉及$\zeta_{n,m}(z_{+i})$的分子项，我们建立总体规模为 m 和 $m-1$的平稳分布之间的关系：

$$\sum_{i=1}^{n}\zeta_{n,m}(z_{+i})=\sum_{i=1}^{n}\frac{p_i}{\mu^{\mathrm{cs}}}\frac{W_{n,m-1}}{W_{n,m}}\zeta_{n,m-1}(z)=\frac{1}{\mu^{\mathrm{cs}}}\frac{W_{n,m-1}}{W_{n,m}}\zeta_{n,m-1}(z).\tag{71}$$

第一个等式是将 $\frac{p_i}{\mu^{\mathrm{cs}}}$ 提取出来，并乘以再除以$W_{n,m-1}$，从而还原出$\zeta_{n,m-1}(z)$的表达式。第二个等式则依据$\sum_{i=1}^{n} p_i = 1$这一事实得出。

将 (69)、(70) 和 (71) 代回 (68) 后，项 $\frac{1}{\mu^{\mathrm{cs}}} W_{n,} W_{n\ m}^{m-1}$ 相互抵消，得到：

$$\mathbb{P}(\tilde{Y} = x) = \frac{(\sum_{j=1}^{n} x_j^{\mathrm{cs}})!}{\prod_{j=1}^{n} x_j^{\mathrm{cs}}!} \zeta_{n,m-1}(z) = \phi_{n,m-1}(x), \quad x \in \mathcal{X}_{4n,m-1}.$$

最后一个等式可通过代入 $\zeta_{n,\ m-1}$ 的定义得到。证明至此完成。

P Buzen 的递归算法与中央服务器端队列

我们将*Buzen*的递归算法扩展为纳入中央服务器服务率，从而能够以$\mathcal{O}(nm^2)$ 的时间计算第 7.1 节模型的归一化常数。

▶ **命题19**。 对于所有 $\mathfrak{m} \in \{0, 1, \ldots, m\}$，归一化常数 $W_{n,\ \mathfrak{m}}$ 可使用 *Buzen* 递归算法在 $O(nm^2)$ 时间内计算，并占用 $O(m)$ 内存。特别地，对于所有 $\mathfrak{m} \in \{0, 1, \ldots, m\}$，有 $W_{n,\ \mathfrak{m}} = V_{3n,\ \mathfrak{m}}$，其中各量 $V_{\mathfrak{n},\ \mathfrak{m}}$ 由以下递归关系定义

- $V_{\mathfrak{n},0} = 1,$ *for* $\mathfrak{n} \in \{1, \ldots, 3n+1\}$.
- $V_{1,\mathfrak{m}} = \left(\frac{1}{\mu^{\mathrm{cs}}}\right)^{\mathfrak{m}},$ *for* $\mathfrak{m} \in \{0, \ldots, m\}$.
- $V_{\mathfrak{n}+1,\mathfrak{m}} = V_{\mathfrak{n},\mathfrak{m}} + \frac{p_{\mathfrak{n}}}{\mu^{\mathrm{c}}_{\mathfrak{n}}} V_{\mathfrak{n}+1,\mathfrak{m}-1},$ *for* $\mathfrak{n} \in \{1, \ldots, n\}, \mathfrak{m} \in \{1, \ldots, m\}$.
- $V_{\mathfrak{n}+1,\mathfrak{m}} = \sum_{k=0}^{\mathfrak{m}} \frac{1}{k!} \left(\frac{p_{\mathfrak{n}-n}}{\mu^{\mathrm{d}}_{\mathfrak{n}-n}}\right)^k V_{\mathfrak{n},\mathfrak{m}-k},$ *for* $\mathfrak{n} \in \{n+1, \ldots, 2n\}, \mathfrak{m} \in \{1, \ldots, m\}$.
- $V_{\mathfrak{n}+1,\mathfrak{m}} = \sum_{k=0}^{\mathfrak{m}} \frac{1}{k!} \left(\frac{p_{\mathfrak{n}-2n}}{\mu^{\mathrm{u}}_{\mathfrak{n}-2n}}\right)^k V_{\mathfrak{n},\mathfrak{m}-k},$ *for* $\mathfrak{n} \in \{2n+1, \ldots, 3n\}, \mathfrak{m} \in \{1, \ldots, m\}$.

Q 定理 7 的证明

定理 7 的证明沿用了第一个模型（定理 2）所采用的方法。我们采用相同的均匀化 技术，以弥合系统的离散时间操作语义与利特尔定律的连续时间要求之间的差距。接下来，我们将依次证明该定理的各项断言。

Q.1 方程(21)的证明

固定一个客户端索引 $i \in \{1, \ldots, n\}$ 以及一个轮次索引 $k \in \mathbb{N}$。在稳态下，期望延迟为

$$\mathbb{E}^0[D_i] = \mathbb{E}^0[D_{i,k}] = \sum_{j=1}^{n} \mathbb{E}^0[D_{i,k} \mid A_k = j]\, \mathbb{P}^0(A_k = j) = p_i\, \mathbb{E}^0[D_{i,k} \mid A_k = i] = p_i\, \mathbb{E}^0[R_i],$$

其中 $R_{i,k}$ 表示从将第 k 个任务派发给客户端 i 到应用其产生的梯度期间发生的全局模型更新次数。该量表示作业在由服务器路径 $\mathsf{d}_i \to \mathsf{c}_i \to \mathsf{u}_i$ 和 CS队列构成的子系统中的逗留时间（以“任务数”计）。我们用 $\mathbb{E}^0[R_i]$ 表示这一离散逗留时间的 Palm稳态期望。

因此，证明公式（21）归结为证明：对于每个 $i \in \{1, \dots, n\}$，

$$\mathbb{E}^0[R_i] = \frac{\mathbb{E}[\tilde{Y}_i^{\mathrm{cs}} + \tilde{Y}_i^{\mathrm{d}} + \tilde{Y}_i^{\mathrm{c}} + \tilde{Y}_i^{\mathrm{u}}]}{p_i}. \tag{72}$$

这一关系可解释为将利特尔定律应用于中央服务器处的服务完成离散时刻。

步骤 1：均匀化构造 .

为了应用标准排队论定理，我们引入一个辅助的均匀连续时间系统。在这个代理模型中，每一轮的持续时间（参数更新之间的时间）均为独立同分布的指数分布，均值为 1。尽管这创造了一个虚构的时间尺度，但它满足经典连续时间利特尔法则的前提，使我们能够推导出结果，再将其映射回原始离散时间过程。

回顾一下，$(T_k)_{k\in\mathbb{N}}$ 表示中央服务器处的服务完成时间序列。由于每个 T_k 相对于连续时间马尔可夫过程 $(Z(t))_{t\geq 0}$都是一个停时，因此，采样序列 $(Z(T_k))_{k\in\mathbb{N}}$ 构成一个遍历离散时间齐次马尔可夫链。我们通过将这一离散链从属于速率为 1 的泊松过程 $(N(t))_{t\geq 0}$ ，定义均匀化连续时间马尔可夫链 $(\bar{Z}(t))_{t\geq 0}$：

$$\bar{Z}(t) = Z(T_{N(t)}), \quad t \geq 0.$$

步骤 2：到达过程与逗留时间。

令 $\bar{T}_{i,k}$ 表示分配给客户端 i 的第 k个任务的到达时间，并令 $N_i(t)$ 表示相应的计数过程。

在这一转换后的设定中，到达流 $(\bar{T}_{i,k})_{k\in\mathbb{N}}$ 构成强度为 p_i的齐次泊松点过程。这是泊松稀疏化的结果：离开 CS 的全局任务流是速率为 1 的泊松过程（这是因为轮次时长服从指数分布），且每个任务都以概率 p_i独立地路由至客户端 i 。

分配给客户端 i 的第 k个任务的逗留时间，对应于 $R_{i,k}+1$个更新间隔之和。具体而言，

$$\bar{R}_{i,k} = \sum_{l=1}^{R_{i,k}+1} E_l,$$

其中 $(E_l)_{l\in\mathbb{N}}$ 是均值为 1 的独立同分布指数随机变量序列。因此，时刻 t 与客户端 i相关的子系统（服务器 $\mathrm{d}_i, \mathrm{c}_i, \mathrm{u}_i$ 及 CS队列）的总占用量为:

$$h_i(\bar{Z}(t)) = \sum_{k\in\mathbb{N}} \mathbf{1}\{\bar{T}_{i,k} \leq t < \bar{T}_{i,k} + \bar{R}_{i,k}\} = \bar{Z}_i^{\mathrm{d}}(t) + \bar{Z}_i^{\mathrm{c}}(t) + \bar{Z}_i^{\mathrm{u}}(t) + \sum_{k=1}^{\ell(\bar{Z}^{\mathrm{cs}}(t))} \mathbf{1}\{\bar{Z}_k^{\mathrm{cs}}(t) = i\}. \tag{73}$$

步骤 3：利特尔定律的应用。

均匀系统满足定理 5.2] 所述平稳版利特尔定律的必要条件： (i) 系统状态 $(Z(t))_{t\geq 0}$ 是遍历马尔可夫链；(ii) 到达过程是泊松过程（因此是平稳且简单的）；(iii) 逗留时间 $R_{i,\,k}$ 仅由到达后系统的演化决定 $(Z(t)t \geq T_{i,\,k})$；(iv) 一阶矩有限： $\mathbb{E}[h_i(Z)] < \infty$ 和 $\mathbb{E}[N_i(1)] = p_i$。

将定理应用于 (73)，得到:

$$\mathbb{E}[h_i(\bar{Z})] = p_i\, \mathbb{E}^0[\bar{R}_i]. \tag{74}$$

步骤 4：映射回离散模型。

现在，我们将 (74) 转换回原始系统的运行参数。根据均匀化的性质， Z 的平稳分布与嵌入链 $(Z(T_k))_{k\in\mathbb{N}}$的平稳分布相同。因此， $\mathbb{E}[f_i(Z)] = \mathbb{E}[h_i(Z(T_k))]$。展开状态函数$h_i$，得到：

$$h_i(Z(T_k)) = Y_i^{\mathrm{d}}(T_k) + Y_i^{\mathrm{c}}(T_k) + Y_i^{\mathrm{u}}(T_k) + Y_i^{\mathrm{cs}}(T_k).$$

此外，由于虚拟间隔（E_l）与离散计数 R_i 相互独立，¯沃尔德恒等式意味着 $\mathbb{E}[R_i] = \mathbb{E}[R_i + 1]$。将这些关系代入（74）可得：

$$\mathbb{E}[Y_i^{\mathrm{cs}}(T_k) + Y_i^{\mathrm{d}}(T_k) + Y_i^{\mathrm{c}}(T_k) + Y_i^{\mathrm{u}}(T_k)] = p_i(\mathbb{E}^0[R_i] + 1) = p_i\,\mathbb{E}^0[R_i] + p_i.$$

利用路由概率满足 $\mathbb{E}[\mathbf{1}\{A_k = i\}] = p_i$ 这一事实，我们可以将常数项 p_i 吸收到左侧的期望中：

$$\mathbb{E}[Y_i^{\mathrm{cs}}(T_k) + Y_i^{\mathrm{c}}(T_k) + Y_i^{\mathrm{u}}(T_k) + (Y_i^{\mathrm{d}}(T_k) - \mathbf{1}\{A_k = i\})] = p_i\,\mathbb{E}^0[R_i].$$

¯最后，我们注意到括号中的项正是 Y 的状态定义（更新时刻的系统状态不包括 新路由的任务，如（19）中所定义）。具体而言，$\tilde{Y}_{i,\,k}^{\mathrm{d}} = Y_{\mathrm{d}i}(T_k) - \mathbf{1}\{A_k = i\}$。因此：

$$\mathbb{E}[\tilde{Y}_i^{\mathrm{cs}} + \tilde{Y}_i^{\mathrm{d}} + \tilde{Y}_i^{\mathrm{c}} + \tilde{Y}_i^{\mathrm{u}}] = p_i\,\mathbb{E}^0[R_i].$$

这证实了（72），并完成了公式（21）的证明。

Q.2 公式（22）的证明

令 $i, j \in \{1, \dots, n\}$。我们的目标是推导期望延迟关于路由参数的梯度。应用公式（21），并利用协方差算子的双线性性质，证明公式（22）等价于证明以下恒等式：

$$\frac{\partial \mathbb{E}[\tilde{Y}_i^{\mathrm{cs}} + \tilde{Y}_i^{\mathrm{d}} + \tilde{Y}_i^{\mathrm{c}} + \tilde{Y}_i^{\mathrm{u}}]}{\partial(\log p_j)} = \mathrm{Cov}\Big[\tilde{Y}_i^{\mathrm{cs}} + \tilde{Y}_i^{\mathrm{d}} + \tilde{Y}_i^{\mathrm{c}} + \tilde{Y}_i^{\mathrm{u}},\ \tilde{Y}_j^{\mathrm{cs}} + \tilde{Y}_j^{\mathrm{d}} + \tilde{Y}_j^{\mathrm{c}} + \tilde{Y}_j^{\mathrm{u}}\Big].$$

为简洁起见，令 $S_k(x) = x_k^{\mathrm{cs}} + x_k^{\mathrm{d}} + x_k^{\mathrm{c}} + x_k^{\mathrm{u}}$ 表示类别 k 在˜状态 x中的任务总数。回顾一下，随机向量 Y服从式（20）中给出的平稳分布 $\phi_{n,m-1}$ 。对该概率质量函数取对数可知，它关于规范参数 $\log p_k$属于指数族：

$$\log \phi_{n,m-1}(x) = -\log W_{n,m-1} + \sum_{k=1}^{n} S_k(x)\log p_k + \underbrace{\log\left(\left(\sum_{k=1}^{n} x_k^{\mathrm{cs}}\right)!\right) - \log\left((\mu^{\mathrm{CS}})^{\sum_k x_k^{\mathrm{cs}}} \prod_{k=1}^{n} x_k^{\mathrm{cs}}!(\mu_k^{\mathrm{c}})^{x_k^{\mathrm{c}}}(\mu_k^{\mathrm{d}})^{x_k^{\mathrm{d}}} x_k^{\mathrm{d}}!(\mu_k^{\mathrm{u}})^{x_k^{\mathrm{u}}} x_k^{\mathrm{u}}!\right)}_{=H(x)}, \tag{75}$$

对所有 $x \in \mathcal{X}_{4n,\,m-1}$。归一化常数$W_{n,\,m-1}$ 定义为对状态空间上的指数分布项求和：

$$W_{n,m-1} = \sum_{x\in\mathcal{X}_{4n,m-1}} \exp\big(H(x) + \sum_{k=1}^{n} S_k(x)\log p_k\big), \tag{76}$$

其中 $H(x)$ 汇集了所有与 $\{p_k\}_{k=1}^n$ 无关的项。

我们接下来推导两个中间微分结果。首先，计算函数 $\log W_{n,m-1}$ 的梯度。对 (76) 关于 $\log p_j$ 求微分，得到：

$$\begin{aligned}\frac{\partial W_{n,m-1}}{\partial(\log p_j)} &= \sum_{x\in\mathcal{X}_{4n,m-1}} S_j(x)\,\exp\big(H(x)+\sum_{k=1}^n S_k(x)\log p_k\big)\\ &= \sum_{x\in\mathcal{X}_{4n,m-1}} S_j(x)\,W_{n,m-1}\,\phi_{n,m-1}(x).\end{aligned}$$

~两边除以 $W_{n,m-1}$，即可得到 $S_j(Y)$ 的期望：

$$\frac{\partial \log W_{n,m-1}}{\partial(\log p_j)} = \frac{1}{W_{n,m-1}}\frac{\partial W_{n,m-1}}{\partial(\log p_j)} = \mathbb{E}[S_j(\tilde{Y})] = \mathbb{E}[\tilde{Y}_j^{\mathrm{cs}}+\tilde{Y}_j^{\mathrm{d}}+\tilde{Y}_j^{\mathrm{c}}+\tilde{Y}_j^{\mathrm{u}}]. \tag{77}$$

其次，我们对对数概率本身求微分。使用 (75) 并代入 (77)：

$$\begin{aligned}\frac{\partial \log \phi_{n,m-1}(x)}{\partial(\log p_j)} &= S_j(x) - \frac{\partial \log W_{n,m-1}}{\partial(\log p_j)}\\ &= (x_j^{\mathrm{cs}}+x_j^{\mathrm{d}}+x_j^{\mathrm{c}}+x_j^{\mathrm{u}}) - \mathbb{E}[\tilde{Y}_j^{\mathrm{cs}}+\tilde{Y}_j^{\mathrm{d}}+\tilde{Y}_j^{\mathrm{c}}+\tilde{Y}_j^{\mathrm{u}}].\end{aligned} \tag{78}$$

最后，为了求得期望的梯度，我们对期望值的定义求微分：

$$\begin{aligned}\frac{\partial \mathbb{E}[S_i(\tilde{Y})]}{\partial(\log p_j)} &= \sum_{x\in\mathcal{X}_{4n,m-1}} S_i(x)\frac{\partial \phi_{n,m-1}(x)}{\partial(\log p_j)}\\ &= \sum_{x\in\mathcal{X}_{4n,m-1}} S_i(x)\,\phi_{n,m-1}(x)\frac{\partial \log \phi_{n,m-1}(x)}{\partial(\log p_j)}.\end{aligned}$$

将 (78) 代入此表达式，得到：

$$\begin{aligned}\frac{\partial \mathbb{E}[S_i(\tilde{Y})]}{\partial(\log p_j)} &= \sum_{x\in\mathcal{X}_{4n,m-1}} \phi_{n,m-1}(x)\,S_i(x)\,\Big(S_j(x)-\mathbb{E}[S_j(\tilde{Y})]\Big)\\ &= \mathbb{E}\Big[S_i(\tilde{Y})\big(S_j(\tilde{Y})-\mathbb{E}[S_j(\tilde{Y})]\big)\Big]\\ &= \mathrm{Cov}\Big[S_i(\tilde{Y}),S_j(\tilde{Y})\Big].\end{aligned}$$

将 S_i 和 S_j 展开为各自的分量，即完成证明。

Q.3 方程 (23) 的证明

为推导式 (23) 中的闭式表达式，我们首先刻画任务计数在中央服务器处聚合时系统的平稳分布。

▶**引理20**。 考虑聚合过程

$$\Big(\sum_{j=1}^n \tilde{Y}_j^{\mathrm{cs}},\,\tilde{Y}_i^{\mathrm{d}},\,\tilde{Y}_i^{\mathrm{c}},\,\tilde{Y}_i^{\mathrm{u}};\ i=1,\dots,n\Big),$$

其属于状态空间 $\boldsymbol{X}_{3n+1,\ m-1}$。该过程具有以下平稳分布：

$$\nu_{n,m-1}(x) = \frac{1}{W_{n,m-1}}\left(\frac{1}{\mu^{\mathrm{cs}}}\right)^{x^{\mathrm{cs}}}\prod_{i=1}^n\left[\left(\frac{p_i}{\mu_i^{\mathrm{c}}}\right)^{x_i^{\mathrm{c}}}\frac{1}{x_i^{\mathrm{d}}!}\left(\frac{p_i}{\mu_i^{\mathrm{d}}}\right)^{x_i^{\mathrm{d}}}\frac{1}{x_i^{\mathrm{u}}!}\left(\frac{p_i}{\mu_i^{\mathrm{u}}}\right)^{x_i^{\mathrm{u}}}\right], \tag{79}$$

对于 $x = (x^{\mathrm{cs}}, x_1^{\mathrm{d}}, \ldots, x_n^{\mathrm{d}}, x_1^{\mathrm{c}}, \ldots, x_n^{\mathrm{c}}, x_1^{\mathrm{u}}, \ldots, x_n^{\mathrm{u}}) \in \boldsymbol{X}_{3n+1,\ m-1}$，其中 $W_{n,\ m-1}$ 是与命题 *6* 中相同的归一化常数。

证明。详细系统状态 $y = \left(y_1^{\mathrm{cs}}, \ldots, y_n^{\mathrm{cs}}, y_1^{\mathrm{d}}, \ldots, y_n^{\mathrm{d}}, y_1^{\mathrm{c}}, \ldots, y_n^{\mathrm{c}}, y_1^{\mathrm{u}}, \ldots, y_n^{\mathrm{u}}\right)$ 的稳态概率由乘积形式解 $\phi_{n,m-1}(y)$ 给出。为了得到聚合状态 x 的稳态概率（其中 CS 处各类别的个体计数合并为一个总数 $x^{\mathrm{cs}} = \sum_{j=1}^{n} y_j^{\mathrm{cs}}$），我们对满足这一求和约束的 CS队列所有详细配置的 $\phi_{n,\ m-1}(y)$ 求和。

因此，对于 $x = (x^{\mathrm{cs}}, x_1^{\mathrm{d}}, \ldots, x_n^{\mathrm{d}}, x_1^{\mathrm{c}}, \ldots, x_n^{\mathrm{c}}, x_1^{\mathrm{u}}, \ldots, x_n^{\mathrm{u}}) \in \mathcal{X}_{3n+1,m-1}$，稳态概率为：

$$
\begin{aligned}
&\mathbb{P}\left(\sum_{j=1}^{n} \tilde{Y}_j^{\mathrm{cs}} = x^{\mathrm{cs}},\ \tilde{Y}_i^{\mathrm{d}} = x_i^{\mathrm{d}},\ \tilde{Y}_i^{\mathrm{c}} = x_i^{\mathrm{c}},\ \tilde{Y}_i^{\mathrm{u}} = x_i^{\mathrm{u}}\right) \\
&= \sum_{\substack{y \in \mathbb{N}^n \\ \sum_{k=1}^{n} y_k = x^{\mathrm{cs}}}} \phi_{n,m-1}\left(y_i,\ x_i^{\mathrm{d}},\ x_i^{\mathrm{c}},\ x_i^{\mathrm{u}}; i = 1, \ldots, n\right) \\
&= \frac{1}{W_{n,m-1}} \prod_{i=1}^{n} \left(\frac{p_i}{\mu_i^{\mathrm{c}}}\right)^{x_i^{\mathrm{c}}} \frac{1}{x_i^{\mathrm{d}}!} \left(\frac{p_i}{\mu_i^{\mathrm{d}}}\right)^{x_i^{\mathrm{d}}} \frac{1}{x_i^{\mathrm{u}}!} \left(\frac{p_i}{\mu_i^{\mathrm{u}}}\right)^{x_i^{\mathrm{u}}} \times \underbrace{\sum_{\substack{y \in \mathbb{N}^n \\ \sum y_k = x^{\mathrm{cs}}}} \frac{x^{\mathrm{cs}}!}{\prod_{k=1}^{n} y_k!} \prod_{i=1}^{n} \left(\frac{p_i}{\mu^{\mathrm{cs}}}\right)^{y_i}}_{\text{Multinomial Expansion}}.
\end{aligned}
$$

对带下划线的项应用多项式定理，可得：

$$
\sum_{\substack{y \in \mathbb{N}^n \\ \sum y_k = x^{\mathrm{cs}}}} \frac{x^{\mathrm{cs}}!}{\prod_{k=1}^{n} y_k!} \prod_{i=1}^{n} \left(\frac{p_i}{\mu^{\mathrm{cs}}}\right)^{y_i} = \left(\sum_{i=1}^{n} \frac{p_i}{\mu^{\mathrm{cs}}}\right)^{x^{\mathrm{cs}}} = \left(\frac{1}{\mu^{\mathrm{cs}}} \underbrace{\sum_{i=1}^{n} p_i}_{=1}\right)^{x^{\mathrm{cs}}} = \left(\frac{1}{\mu^{\mathrm{cs}}}\right)^{x^{\mathrm{cs}}}.
$$

将其代回，即得式 (79) 中 $\nu_{n,m-1}$ 的表达式。 ◀

有了平稳分布$\nu_{n,m-1}$后，我们接下来计算期望队列长度，首先从无限服务器节点开始。对于下行服务器 d_j，其期望定义如下：

$$
\begin{aligned}
\mathbb{E}[\tilde{Y}_j^{\mathrm{d}}] &= \sum_{x \in \mathcal{X}_{3n+1,m-1}} x_j^{\mathrm{d}}\, \nu_{n,m-1}(x) \\
&= \frac{p_j}{\mu_j^{\mathrm{d}}} \frac{1}{W_{n,m-1}} \sum_{\substack{x \in \mathcal{X}_{3n+1,m-1} \\ x_j^{\mathrm{d}} > 0}} \left(\frac{1}{\mu^{\mathrm{cs}}}\right)^{x^{\mathrm{cs}}} \prod_{i=1}^{n} \left(\frac{p_i}{\mu_i^{\mathrm{c}}}\right)^{x_i^{\mathrm{c}}} \frac{1}{(x_i^{\mathrm{d}} - \mathbf{1}_{\{i=j\}})!} \left(\frac{p_i}{\mu_i^{\mathrm{d}}}\right)^{x_i^{\mathrm{d}} - \mathbf{1}_{\{i=j\}}} \frac{1}{x_i^{\mathrm{u}}!} \left(\frac{p_i}{\mu_i^{\mathrm{u}}}\right)^{x_i^{\mathrm{u}}}.
\end{aligned}
$$

令 $x' = x - \mathbf{e}_j^{\mathrm{d}}$，求和范围为 $\mathcal{X}_{3n+1,m-2}$，其和为 $W_{n,m-2}$。因此，我们得到 $\mathbb{E}[\tilde{Y}_{\mathrm{d}}$ $_j] = \frac{p_j}{\mu_j^{\mathrm{d}}} \frac{W_{n,W_n\ m-1,\circ}}{}$

根据对称性，同样的推导适用于上行服务器 u_j，得到 $\mathbb{E}[Y_j^{\mathrm{u}}] = \frac{p_j}{\mu_j^{\mathrm{u}}} W_{n,}\, W_n^{\,m-2}{}_{m-1}{}_{,\circ}$。将这两部分相加，即可得到总通信延迟贡献：

$$
\mathbb{E}[\tilde{Y}_j^{\mathrm{d}}] + \mathbb{E}[\tilde{Y}_j^{\mathrm{u}}] = p_j \left(\frac{1}{\mu_j^{\mathrm{d}}} + \frac{1}{\mu_j^{\mathrm{u}}}\right) \frac{W_{n,m-2}}{W_{n,m-1}} = \frac{W_{n,m-2}}{W_{n,m-1}} \gamma_j. \tag{80}
$$

接下来，我们分析单服务器节点，首先从中央服务器开始。我们先使用尾和公式 cs(cs)计算任务的期望总 数量 $M_{\mathrm{cs}} = \sum_{i=1}^{n} \tilde{Y}_i$ $\mathbb{E}[\tilde{Y}\ ] = \mathrm{cs}_{\circ}$ 队列长度超过的概率为: $\sum_{l=1}^{m-1} \mathbb{P}\ \tilde{Y}\ \geq l$ l

$$\mathbb{P}(M_{\mathrm{cs}} \geq l) = \sum_{\substack{x \in \mathcal{X}_{3n+1,m-1} \\ x^{\mathrm{cs}} \geq l}} \nu_{n,m-1}(x)$$

$$= \left(\frac{1}{\mu^{\mathrm{cs}}}\right)^{l} \frac{1}{W_{n,m-1}} \sum_{\substack{x \in \mathcal{X}_{3n+1,m-1} \\ x^{\mathrm{cs}} \geq l}} \left(\frac{1}{\mu^{\mathrm{cs}}}\right)^{x^{\mathrm{cs}}-l} \prod_{i=1}^{n} \left(\frac{p_i}{\mu_i^{\mathrm{c}}}\right)^{x_i^{\mathrm{c}}} \frac{1}{x_i^{\mathrm{d}}!} \left(\frac{p_i}{\mu_i^{\mathrm{d}}}\right)^{x_i^{\mathrm{d}}} \frac{1}{x_i^{\mathrm{u}}!} \left(\frac{p_i}{\mu_i^{\mathrm{u}}}\right)^{x_i^{\mathrm{u}}}.$$

通过变量替换 $x' = x - l \cdot \mathbf{e}^{\mathrm{cs}}$，该求和可化简为 $W_{n,\ m-1-l}$。因此，任务总数的期望为

$$\mathbb{E}[M_{\mathrm{cs}}] = \sum_{l=1}^{m-1} \left(\frac{1}{\mu^{\mathrm{cs}}}\right)^{l} \frac{W_{n,m-1-l}}{W_{n,m-1}} = \tilde{\beta}_{\mathrm{CS},1}. \tag{81}$$

~ ~在给定总数 M_{cs} 的条件下，类别计数 $(Y_{\mathrm{cs}1}, \ldots, Y_{\mathrm{cs}n})$ 服从多项分布~，其概率为 $(p_1, \ldots, p_n)$。因此， $\mathbb{E}[Y_{\mathrm{cs}_j}\ \mid M_{\mathrm{cs}}] = p_j M_{\mathrm{cs}}$，取期望可得:

$$\mathbb{E}[\tilde{Y}_j^{\mathrm{cs}}] = p_j \mathbb{E}[M_{\mathrm{cs}}] = p_j \tilde{\beta}_{\mathrm{CS},1}. \tag{82}$$

类似的逻辑也适用于本地计算节点 cj，它表现为一个相对负载为 p_j/μ_j^{c}的单服务器。应用相同的尾和论证可得:

$$\mathbb{E}[\tilde{Y}_j^{\mathrm{c}}] = \sum_{l=1}^{m-1} \left(\frac{p_j}{\mu_j^{\mathrm{c}}}\right)^{l} \frac{W_{n,m-1-l}}{W_{n,m-1}} = \tilde{\beta}_{j,1}. \tag{83}$$

最后，结合式 (80)、(82) 和 (83) 中推导出的各项，可得所需的闭式表达式:

$$\sum_{s \in \{\mathrm{CS},\mathrm{c},\mathrm{d},\mathrm{u}\}} \mathbb{E}[\tilde{Y}_j^{s}] = p_j \tilde{\beta}_{\mathrm{CS},1} + \tilde{\beta}_{j,1} + \frac{W_{n,m-2}}{W_{n,m-1}} \gamma_j.$$

Q.4 式(24)的证明

为证明公式（24），我们将左侧的求和分解为三类不同的交互：队列-队列相关性、队列-延迟相关性和延迟-延迟相关性。我们分别推导每一类的闭式表达式，然后将它们合并。

~ ~ ~ ~ ~ ~**1. 队列-队列相关性 (**$\mathbb{E}[Y_i^{\mathrm{cs}} Y_{\mathrm{c}_j}]$**,** $\mathbb{E}[Y_{\mathrm{cs}_i}\ Y_j^{\mathrm{cs}}]$**, 以及** $\mathbb{E}[Y_i^{\mathrm{c}} Y_{\mathrm{c}_j}]$**)**

~ ~我们首先计算任意一对客户端 $i,\ j$的 CS 队列与计算队列之间的联合期望 $\mathbb{E}[Y_{\mathrm{cs}}$ $_i\ Y_{\mathrm{c}_j}]$。令 $M_{\mathrm{cs}} = \sum_{l=1}^{n} \tilde{Y}_{\mathrm{cs}l}$ 表示 CS 中的任务总数。根据迭代期望定律，我们有:

$$\mathbb{E}[\tilde{Y}_i^{\mathrm{cs}} \tilde{Y}_j^{\mathrm{c}}] = \mathbb{E}\big[\mathbb{E}[\tilde{Y}_i^{\mathrm{cs}} \tilde{Y}_j^{\mathrm{c}} \mid M_{\mathrm{cs}}, \tilde{Y}_j^{\mathrm{c}}]\big] = \mathbb{E}\big[\mathbb{E}[\tilde{Y}_i^{\mathrm{cs}} \mid M_{\mathrm{cs}}] \tilde{Y}_j^{\mathrm{c}}\big].$$

~ ~在给定总负载 M_{cs} 的条件下，向量 $(Y_{\text{中央服务器}1}, \ldots, Y_{\text{中央服务器}n})$ 服从概率参数为 $p_1, \ldots, p_n$、试验次数为 M_{cs} 的多项分布。因此， $\mathbb{E}[Y_{\text{中央服务器}_i}\ \mid M_{\mathrm{cs}}] = p_i M_{\mathrm{cs}\circ}$ 将其代入前一个方程可得:

$$\mathbb{E}[\tilde{Y}_i^{\mathrm{cs}} \tilde{Y}_j^{\mathrm{c}}] = p_i \mathbb{E}[M_{\mathrm{cs}} \tilde{Y}_j^{\mathrm{c}}]. \tag{84}$$

要计算 $\mathbb{E}[M_{\mathrm{cs}}\tilde{Y}_{\mathrm{c}_i}]$，我们利用恒等式 $x^{\mathrm{cs}}x_i^{\mathrm{c}} = \sum_{k=1}^{x^{\mathrm{cs}}}\sum_{\ell=1}^{x_i^{\mathrm{c}}} 1$ 将期望重写为尾概率之和：

$$\mathbb{E}[M_{\mathrm{cs}}\tilde{Y}_j^{\mathrm{c}}] = \sum_{x\in\mathcal{X}_{n,m-1}}\sum_{k=1}^{x^{\mathrm{cs}}}\sum_{\ell=1}^{x_j^{\mathrm{c}}}\nu_{n,m-1}(x) = \sum_{\substack{k,\ell=1\\ k+\ell\le m-1}}^{m-2}\sum_{\substack{x\in\mathcal{X}_{n,m-1}\\ x^{\mathrm{cs}}\ge k, x_j^{\mathrm{c}}\ge\ell}}\nu_{n,m-1}(x).$$

将方程（79）中的乘积形式解 $\nu_{n,\ m-1}(x)$ 代入此求和式，得到：

$$\begin{aligned}
&\mathbb{E}[M_{\mathrm{cs}}\tilde{Y}_j^{\mathrm{c}}]\\
&= \frac{1}{W_{n,m-1}}\sum_{\substack{k,\ell=1\\ k+\ell\le m-1}}^{m-2}\sum_{\substack{x\in\mathcal{X}_{n,m-1}\\ x^{\mathrm{cs}}\ge k, x_j^{\mathrm{c}}\ge\ell}}\prod_{r=1}^{n}\left(\frac{1}{\mu^{\mathrm{cs}}}\right)^{x^{\mathrm{cs}}}\left(\frac{p_r}{\mu_r^{\mathrm{c}}}\right)^{x_r^{\mathrm{c}}}\frac{1}{x_r^{\mathrm{d}}!}\left(\frac{p_r}{\mu_r^{\mathrm{d}}}\right)^{x_r^{\mathrm{d}}}\frac{1}{x_r^{\mathrm{u}}!}\left(\frac{p_r}{\mu_r^{\mathrm{u}}}\right)^{x_r^{\mathrm{u}}}\\
&\overset{(a)}{=} \frac{1}{W_{n,m-1}}\sum_{\substack{k,\ell=1\\ k+\ell\le m-1}}^{m-2}\left(\frac{1}{\mu^{\mathrm{cs}}}\right)^{k}\left(\frac{p_j}{\mu_j^{\mathrm{c}}}\right)^{\ell}\underbrace{\sum_{y\in\mathcal{X}_{n,m-1-k-\ell}}\prod_{r=1}^{n}\left(\frac{1}{\mu^{\mathrm{cs}}}\right)^{y^{\mathrm{cs}}}\left(\frac{p_r}{\mu_r^{\mathrm{c}}}\right)^{y_r^{\mathrm{c}}}\frac{1}{y_r^{\mathrm{d}}!}\left(\frac{p_r}{\mu_r^{\mathrm{d}}}\right)^{y_r^{\mathrm{d}}}\frac{1}{y_r^{\mathrm{u}}!}\left(\frac{p_r}{\mu_r^{\mathrm{u}}}\right)^{y_r^{\mathrm{u}}}}_{=W_{n,m-1-k-\ell}}\\
&= \sum_{\substack{k,\ell=1\\ k+\ell\le m-1}}^{m-2}\left(\frac{1}{\mu^{\mathrm{cs}}}\right)^{k}\left(\frac{p_j}{\mu_j^{\mathrm{c}}}\right)^{\ell}\frac{W_{n,m-1-k-\ell}}{W_{n,m-1}} = \tilde{\alpha}_{\mathrm{CS},j}.
\end{aligned}$$

步骤 (a) 通过进行变量替换 $y = x - k\mathbf{e}^{\mathrm{cs}} - \ell\mathbf{e}_i^{\mathrm{c}}$ 得出，其中 $\mathbf{e}^{\mathrm{cs}}$ 和 $\mathbf{e}_i^{\mathrm{c}}$ 分别表示对应于分量 x^{cs} 和 x_i^{c} 的 $3n$ 维标准单位向量。内层求和恰好等于一个含有 $m-1-k-\ell$ 个任务的系统的归一化常数。将此结果代回初始表达式（84），得到：

$$\mathbb{E}[\tilde{Y}_i^{\mathrm{cs}}\tilde{Y}_j^{\mathrm{c}}] = p_i\tilde{\alpha}_{\mathrm{CS},j}\quad i,j\in\{1,\dots,n\}. \tag{85}$$

接下来，我们计算内部中央服务器相关性 $\mathbb{E}[Y_{\mathrm{cs}_i}\ Y_j^{\mathrm{cs}}]$。首先考虑类别互不相同的情况（$i\neq j$）。对 M_{cs} 采用相同的条件化处理，可得：

$$\mathbb{E}[\tilde{Y}_i^{\mathrm{cs}}\tilde{Y}_j^{\mathrm{cs}}] = \mathbb{E}\big[\mathbb{E}[\tilde{Y}_i^{\mathrm{cs}}\tilde{Y}_j^{\mathrm{cs}}\mid M_{\mathrm{cs}}]\big] = p_ip_j\mathbb{E}[M_{\mathrm{cs}}(M_{\mathrm{cs}}-1)], \tag{86}$$

其中，第二个等式由多项分布的交叉矩性质得出。利用恒等式 $x^{\mathrm{cs}}(x^{\mathrm{cs}}-1) = \sum_{k=1}^{x^{\mathrm{cs}}-1}2k$，我们可以将阶乘矩 $\mathbb{E}[M_{\mathrm{cs}}(M_{\mathrm{cs}}-1)]$ 表示为尾概率之和：

$$\begin{aligned}
&\mathbb{E}[M_{\mathrm{cs}}(M_{\mathrm{cs}}-1)]\\
&= \sum_{x\in\mathcal{X}_{n,m-1}}x^{\mathrm{cs}}(x^{\mathrm{cs}}-1)\nu_{n,m-1}(x) = \sum_{k=1}^{m-2}2k\sum_{\substack{x\in\mathcal{X}_{n,m-1}\\ x^{\mathrm{cs}}\ge k+1}}\nu_{n,m-1}(x)\\
&\overset{(b)}{=} \frac{1}{W_{n,m-1}}\sum_{k=1}^{m-2}2k\left(\frac{1}{\mu^{\mathrm{cs}}}\right)^{k+1}\underbrace{\sum_{y\in\mathcal{X}_{n,m-2-k}}\prod_{r=1}^{n}\left(\frac{1}{\mu^{\mathrm{cs}}}\right)^{y^{\mathrm{cs}}}\left(\frac{p_r}{\mu_r^{\mathrm{c}}}\right)^{y_r^{\mathrm{c}}}\frac{1}{y_r^{\mathrm{d}}!}\left(\frac{p_r}{\mu_r^{\mathrm{d}}}\right)^{y_r^{\mathrm{d}}}\frac{1}{y_r^{\mathrm{u}}!}\left(\frac{p_r}{\mu_r^{\mathrm{u}}}\right)^{y_r^{\mathrm{u}}}}_{=W_{n,m-2-k}}\\
&= \sum_{k=1}^{m-1}2(k-1)\left(\frac{1}{\mu^{\mathrm{cs}}}\right)^{k}\frac{W_{n,m-1-k}}{W_{n,m-1}}.
\end{aligned} \tag{87}$$

这里，步骤 (b) 由变量替换 $y = x - (k+1)\mathbf{e}^{\mathrm{cs}}$ 得出。将 (87) 代入 (86)，即可得到交叉项：

$$\mathbb{E}[\tilde{Y}_i^{\mathrm{cs}}\tilde{Y}_j^{\mathrm{cs}}] = p_i p_j \sum_{k=1}^{m-1} 2(k-1)\left(\frac{1}{\mu^{\mathrm{cs}}}\right)^k \frac{W_{n,m-1-k}}{W_{n,m-1}} = \tilde{\alpha}_{i,j}^{\mathrm{CS}}, \quad \text{if } i \neq j. \tag{88}$$

相反，对于中央服务器中单个类别的二阶矩（$i = j$），多项分布方差性质给出：

$$\mathbb{E}[(\tilde{Y}_i^{\mathrm{cs}})^2] = \mathbb{E}\big[\mathbb{E}[(\tilde{Y}_i^{\mathrm{cs}})^2 \mid M_{\mathrm{cs}}]\big] = p_i\mathbb{E}[M_{\mathrm{cs}}] + p_i^2\mathbb{E}[M_{\mathrm{cs}}(M_{\mathrm{cs}}-1)].$$

将（81）中已知的 $\mathbb{E}[M_{\mathrm{cs}}]$ 和 $\mathbb{E}[M_{\mathrm{cs}}（M_{\mathrm{cs}}-1）]$ 的表达式分别代入，并重新整理 (87)
各项，可得：

$$\mathbb{E}[(\tilde{Y}_i^{\mathrm{cs}})^2] = p_i \sum_{k=1}^{m-1}\left(\frac{1}{\mu^{\mathrm{cs}}}\right)^k \frac{W_{n,m-1-k}}{W_{n,m-1}}\left[1 + 2(k-1)p_i\right] = \tilde{\alpha}_{i,i}^{\mathrm{CS}}. \tag{89}$$

最后，对本地计算节点应用相同的尾和论证（类似于（50）和（49）中的推导，但使用分布 $\nu_{n,\ m-1}$ 而非 $\pi_{n,\ m-1}$），即可得到最终的队列-队列项：

$$\mathbb{E}[\tilde{Y}_i^{\mathrm{c}}\tilde{Y}_j^{\mathrm{c}}] = \sum_{k=1}^{m-2}\sum_{\ell=1}^{m-1-k}\left(\frac{p_i}{\mu_i^{\mathrm{c}}}\right)^k\left(\frac{p_j}{\mu_j^{\mathrm{c}}}\right)^\ell \frac{W_{n,m-1-k-\ell}}{W_{n,m-1}} = \tilde{\alpha}_{i,j}, \quad \text{if } i \neq j, \tag{90}$$

$$\mathbb{E}[(\tilde{Y}_i^{\mathrm{c}})^2] = \sum_{k=1}^{m-1}(2k-1)\left(\frac{p_i}{\mu_i^{\mathrm{c}}}\right)^k \frac{W_{n,m-1-k}}{W_{n,m-1}} = \tilde{\alpha}_{i,i}. \tag{91}$$

2. 队列-延迟相关性（$\mathbb{E}[\tilde{Y}^{\mathrm{cs}i}\tilde{Y}_j^{\mathrm{d}}]$，$\mathbb{E}[\tilde{Y}^{\mathrm{cs}i}\tilde{Y}_j^{\mathrm{u}}]$，$\mathbb{E}[\tilde{Y}_i^{\mathrm{c}}\tilde{Y}_j^{\mathrm{d}}]$ 和 $\mathbb{E}[\tilde{Y}_i^{\mathrm{c}}\tilde{Y}_j^{\mathrm{u}}]$）

接下来，我们评估客户端i 的 CS 队列（或计算队列）与客户端 j的通信延迟之间的相互作用。我们从互相关 $\mathbb{E}[\tilde{Y}_{\mathrm{cs}_i}\ \tilde{Y}^{\mathrm{d}}{}_j]$ 开始。根据迭代期望定律，我们有：

$$\mathbb{E}[\tilde{Y}_i^{\mathrm{cs}}\tilde{Y}_j^{\mathrm{d}}] = \mathbb{E}\big[\mathbb{E}[\tilde{Y}_i^{\mathrm{cs}}\tilde{Y}_j^{\mathrm{d}} \mid M_{\mathrm{cs}}, \tilde{Y}_j^{\mathrm{d}}]\big] = \mathbb{E}\big[\mathbb{E}[\tilde{Y}_i^{\mathrm{cs}} \mid M_{\mathrm{cs}}]\tilde{Y}_j^{\mathrm{d}}\big] = p_i\mathbb{E}[M_{\mathrm{cs}}\tilde{Y}_j^{\mathrm{d}}].$$

为了计算 $\mathbb{E}[M_{\mathrm{cs}}\tilde{Y}^{\mathrm{d}}{}_j\ ]$，我们利用聚合过程 $\nu_{n,\ m-1}$ 的平稳分布。将期望展开为尾概率之和，可得：

$$\begin{aligned}
\mathbb{E}[M_{\mathrm{cs}}\tilde{Y}_j^{\mathrm{d}}] &= \sum_{x\in\mathcal{X}_{n,m-1}}\sum_{k=1}^{x^{\mathrm{cs}}} x_j^{\mathrm{d}}\nu_{n,m-1}(x) = \sum_{k=1}^{m-2}\sum_{\substack{x\in\mathcal{X}_{n,m-1}\\ x^{\mathrm{cs}}\geq k}} x_j^{\mathrm{d}}\nu_{n,m-1}(x) \\
&= \frac{1}{W_{n,m-1}}\sum_{k=1}^{m-2}\sum_{\substack{x\in\mathcal{X}_{n,m-1}\\ x^{\mathrm{cs}}\geq k,\, x_j^{\mathrm{d}}\geq 1}}\left(\frac{1}{\mu^{\mathrm{cs}}}\right)^{x^{\mathrm{cs}}}\prod_{r=1}^{n}\left(\frac{p_r}{\mu_r^{\mathrm{c}}}\right)^{x_r^{\mathrm{c}}}\frac{1}{(x_r^{\mathrm{d}}-\mathbf{1}_{\{r=j\}})!}\left(\frac{p_r}{\mu_r^{\mathrm{d}}}\right)^{x_r^{\mathrm{d}}}\frac{1}{x_r^{\mathrm{u}}!}\left(\frac{p_r}{\mu_r^{\mathrm{u}}}\right)^{x_r^{\mathrm{u}}} \\
&\overset{(c)}{=} \frac{1}{W_{n,m-1}}\sum_{k=1}^{m-2}\left(\frac{1}{\mu^{\mathrm{cs}}}\right)^k\frac{p_j}{\mu_j^{\mathrm{d}}}\underbrace{\sum_{y\in\mathcal{X}_{n,m-2-k}}\left(\frac{1}{\mu^{\mathrm{cs}}}\right)^{y^{\mathrm{cs}}}\prod_{r=1}^{n}\left(\frac{p_r}{\mu_r^{\mathrm{c}}}\right)^{y_r^{\mathrm{c}}}\frac{1}{y_r^{\mathrm{d}}!}\left(\frac{p_r}{\mu_r^{\mathrm{d}}}\right)^{y_r^{\mathrm{d}}}\frac{1}{y_r^{\mathrm{u}}!}\left(\frac{p_r}{\mu_r^{\mathrm{u}}}\right)^{y_r^{\mathrm{u}}}}_{=W_{n,m-2-k}} \\
&= \frac{p_j}{\mu_j^{\mathrm{d}}}\sum_{k=1}^{m-2}\left(\frac{1}{\mu^{\mathrm{cs}}}\right)^k\frac{W_{n,m-2-k}}{W_{n,m-1}} = \tilde{\beta}_{\mathrm{CS},2}\frac{p_j}{\mu_j^{\mathrm{d}}}.
\end{aligned}$$

这里，步骤 (c) 使用变量替换 $y = x - k\mathbf{e}^{\mathrm{cs}} - \mathbf{e}_j^{\mathrm{d}}$，并利用了 $x_j^{\mathrm{d}} \geq 1$ 这一事实。根据对称性，上行链路相关性为 $\mathbb{E}[\tilde{Y}_{\mathrm{cs}}_i \tilde{Y}^{\mathrm{u}}_j] = p_i\tilde{\beta}_{\mathrm{CS},2}\frac{p_j}{\mu_j^{\mathrm{u}}}$。将这些分量相加，得到类别 i 在中央服务器上的计数与客户端 j 的总体通信延迟之间的总交互：

$$\mathbb{E}[\tilde{Y}_i^{\mathrm{cs}}(\tilde{Y}_j^{\mathrm{d}} + \tilde{Y}_j^{\mathrm{u}})] = p_i\tilde{\beta}_{\mathrm{CS},2}\, p_j\left(\frac{1}{\mu_j^{\mathrm{d}}} + \frac{1}{\mu_j^{\mathrm{u}}}\right) = p_i\tilde{\beta}_{\mathrm{CS},2}\gamma_j, \quad i,j \in \{1,\dots,n\}. \tag{92}$$

同样，交换 i 和 j 的角色，可得对称的交互项：

$$\mathbb{E}[\tilde{Y}_j^{\mathrm{cs}}(\tilde{Y}_i^{\mathrm{d}} + \tilde{Y}_i^{\mathrm{u}})] = p_j\tilde{\beta}_{\mathrm{CS},2}\gamma_i, \quad i,j \in \{1,\dots,n\}. \tag{93}$$

最后，按照在（51）中针对本地计算节点所采用的完全相同的推理（但使用分布 $\nu_{n,\,m-1}$ 而非 $\pi_{n,\,m-1}$），我们得到客户端 i 的计算队列与客户端 j的通信延迟之间的相关性：

$$\mathbb{E}[\tilde{Y}_i^{\mathrm{c}}(\tilde{Y}_j^{\mathrm{d}} + \tilde{Y}_j^{\mathrm{u}})] = \tilde{\beta}_{i,2}\gamma_j, \quad i,j \in \{1,\dots,n\}. \tag{94}$$

3. 延迟-延迟相关性

延迟-延迟相关性（$\mathbb{E}[\tilde{Y}_{\mathrm{d}}_i \tilde{Y}^{\mathrm{u}}_j]$、$\mathbb{E}[\tilde{Y}_{\mathrm{d}}_i \tilde{Y}^{\mathrm{d}}_j]$等）仅涉及无限服务器节点。其推导与推导（53）时所用的方法完全相同，由此得到该项$\psi_{i,\,j}$。

将所有推导出的贡献相加，即得公式（24）中的最终结果：来自（85）的中央服务器-计算队列项 $p_i\tilde{\alpha}_{\mathrm{CS},\,j} + p_j\tilde{\alpha}_{\mathrm{CS},\,i}$；来自（88）和（87）的中央服务器-类别项 $\tilde{\alpha}_i^{\mathrm{CS}}{}_{,\,j}$；来自（90）和（91）的计算队列-队列项 $\tilde{\alpha}_{i,\,j}$；来自（92）、（93）和（94）的队列-时延交互项 $\beta_{\mathrm{CS},\,2}(p_i\gamma_j + p_j\gamma_i) + \beta_{i,\,2}\gamma_j + \beta_{j,\,2}\gamma_i$；以及延迟—延迟项 $\psi_{i,\,j}$。

R 命题 8 的证明

我们依次推导这些公式：首先是公式 (25)，接着是 (26)，最后是 (27)。

R.1 公式 (25) 的证明

设 $T_0 = 0$，并令 $(T_k)_{k\geq 1}$ 表示中央服务器上服务完成的连续时刻。因此，第 k 个全局轮次的持续时间恰好为 $T_{k+1} - T_k$。执行 K_ϵ 轮次所需的实际运行时间，自然可表示为以下望远镜求和：

$$\tau_\epsilon = \sum_{k=1}^{K_\epsilon(p,m)} (T_k - T_{k-1}). \tag{95}$$

为分析期望持续时间，考虑计数过程 $N = (N(t))_{t\geq 0}$，它跟踪截至时间 t 的中央服务器离开的累计次数。该过程由 CS 队列的状态转移决定：

$$N(t) = \sum_{0<s\leq t}\sum_{i=1}^{n} \left(Y_i^{\mathrm{cs}}(s^-) - Y_i^{\mathrm{cs}}(s)\right)^+.$$

由于底层状态过程 Y 是平稳的，点过程 N 也继承了这一平稳性。因此，在相应的 Palm 概率测度 $\mathbb{P}^0$下，离开间隔服从相同的分布，这意味着对于所有 $k \geq 1$，均有 $\mathbb{E}^0[T_k - T_{k-1}] = \mathbb{E}^0[T_1]$。

运用平稳点过程的基本反演公式（例如，参见 [51,推论 6.16]，将其应用于常数函数 $X_t = 1$），可得关系式：

$$\tilde{\lambda}(p, m)\, \mathbb{E}^0[T_1] = 1,$$

其中，$\tilde{\lambda}$ 表示在标准概率测度 $\mathbb{P}$ 下 $(N(t))_{t\geq 0}$ 的稳态离开强度。将此关系代回 (95) 的期望值，可直接得到：

$$\mathbb{E}^0[\tau_\epsilon] = \sum_{k=1}^{K_\epsilon(p,m)} \mathbb{E}^0[T_k - T_{k-1}] = \frac{K_\epsilon(p,m)}{\tilde{\lambda}(p,m)},$$

由此即可得到公式 (25)。

R.2 公式 (26) 的证明

为严格推导 (26) 中 $\lambda(p,m)$ 的表达式，我们可以通过聚合中央服务器（CS）上所有类别的任务来分析系统的行为。

▶**引理 21**。 定义连续时间聚合状态过程如下

$$\xi(t) = \Big(\sum_{j=1}^{n} Y_j^{\mathrm{cs}}(t),\, Y_i^{\mathrm{d}}(t),\, Y_i^{\mathrm{c}}(t),\, Y_i^{\mathrm{u}}(t);\ i = 1, \ldots, n\Big),$$

该过程在状态空间 $\boldsymbol{X}_{3n+1,\ m}$上演化。此过程构成一条遍历的齐次马尔可夫链。其唯一的不变概率测度为：

$$\nu_{n,m}(x) = \frac{1}{W_{n,m}} \left(\frac{1}{\mu^{\mathrm{cs}}}\right)^{x^{\mathrm{cs}}} \prod_{i=1}^{n} \left[\left(\frac{p_i}{\mu_i^{\mathrm{c}}}\right)^{x_i^{\mathrm{c}}} \frac{1}{x_i^{\mathrm{d}}!} \left(\frac{p_i}{\mu_i^{\mathrm{d}}}\right)^{x_i^{\mathrm{d}}} \frac{1}{x_i^{\mathrm{u}}!} \left(\frac{p_i}{\mu_i^{\mathrm{u}}}\right)^{x_i^{\mathrm{u}}} \right], \tag{96}$$

对于任意状态 $x = (x^{\mathrm{cs}}, x_1^{\mathrm{d}}, \ldots, x_n^{\mathrm{d}}, x_1^{\mathrm{c}}, \ldots, x_n^{\mathrm{c}}, x_1^{\mathrm{u}}, \ldots, x_n^{\mathrm{u}}) \in \boldsymbol{X}_{3n+1,\ m}$，其中 $W_{n,\ m}$ 是命题 6 中定义的相同归一化常数。

证明。 通过应用引理20中使用的相同边缘化论证，我们可以直接将完整过程 Y 的详细平稳分布 $\phi_{n,\ m}$ 化简为 $\nu_{n,\ m}$。◀

在我们所提出的架构（第7.1节）中，中央服务器仅在非空闲时以速率 μ^{cs} 处理任务。因此，期望瞬时吞吐量为 $\tilde{\lambda}(p,\ m) = \mu^{\mathrm{cs}} \mathbb{P}\big(\sum_{i=1}^{n} Y_{\text{中央服务器}_i} > 0\big)$。将此期望对聚合不变测度 $\nu_{n,\ m}$ 展开，可得：

$$\begin{aligned}
\tilde{\lambda}(p,m) &= \mu^{\mathrm{cs}} \sum_{\substack{x \in \mathcal{X}_{3n+1,m} \\ x^{\mathrm{cs}} \geq 1}} \nu_{n,m}(x) \\
&= \frac{\mu^{\mathrm{cs}}}{W_{n,m}} \sum_{\substack{x \in \mathcal{X}_{3n+1,m} \\ x^{\mathrm{cs}} \geq 1}} \left(\frac{1}{\mu^{\mathrm{cs}}}\right)^{x^{\mathrm{cs}}} \prod_{l=1}^{n} \left(\frac{p_l}{\mu_l^{\mathrm{c}}}\right)^{x_l^{\mathrm{c}}} \frac{1}{x_l^{\mathrm{d}}!} \left(\frac{p_l}{\mu_l^{\mathrm{d}}}\right)^{x_l^{\mathrm{d}}} \frac{1}{x_l^{\mathrm{u}}!} \left(\frac{p_l}{\mu_l^{\mathrm{u}}}\right)^{x_l^{\mathrm{u}}} \\
&= \frac{1}{W_{n,m}} \underbrace{\sum_{\substack{x \in \mathcal{X}_{3n+1,m} \\ x^{\mathrm{cs}} \geq 1}} \left(\frac{1}{\mu^{\mathrm{cs}}}\right)^{x^{\mathrm{cs}}-1} \prod_{l=1}^{n} \left(\frac{p_l}{\mu_l^{\mathrm{c}}}\right)^{x_l^{\mathrm{c}}} \frac{1}{x_l^{\mathrm{d}}!} \left(\frac{p_l}{\mu_l^{\mathrm{d}}}\right)^{x_l^{\mathrm{d}}} \frac{1}{x_l^{\mathrm{u}}!} \left(\frac{p_l}{\mu_l^{\mathrm{u}}}\right)^{x_l^{\mathrm{u}}}}_{=W_{n,m-1}} \\
&= \frac{W_{n,m-1}}{W_{n,m}}.
\end{aligned} \tag{97}$$

将求和映射到 $W_{n,\ m-1}$ 这一步，是通过提出公因子 $1/\mu^{\mathrm{cs}}$ 并进行几何状态转移 $y = x - \mathbf{e}^{\mathrm{cs}}$ 实现的（其中 $\mathbf{e}^{\mathrm{cs}}$ 表示对应于中央服务器分量的标准单位向量）。由此完成公式 (26) 的证明。

R.3 方程 (27) 的证明

现在，我们计算吞吐量对路由决策的敏感度。根据此前推导方程 (77) 时的结果，归一化常数 $W_{n,m-1}$ 的偏导数为：

$$\frac{\partial W_{n,m-1}}{\partial p_j} = \frac{W_{n,m-1}}{p_j}\mathbb{E}[\tilde{Y}_j^{\mathrm{cs}} + \tilde{Y}_j^{\mathrm{d}} + \tilde{Y}_j^{\mathrm{c}} + \tilde{Y}_j^{\mathrm{u}}].$$

同样，对完整的 m 任务系统应用相同的概率逻辑，可得：

$$\frac{\partial W_{n,m}}{\partial p_j} = \frac{W_{n,m}}{p_j}\mathbb{E}[Y_j^{\mathrm{cs}} + Y_j^{\mathrm{d}} + Y_j^{\mathrm{c}} + Y_j^{\mathrm{u}}].$$

˜要计算吞吐量 $\lambda(p,m) = W_{n,m-1}/W_{n,m}$ 的梯度，只需应用商法则并代入上述导数：

$$\begin{aligned}\frac{\partial}{\partial p_j}\tilde{\lambda}(p,m) &= \frac{1}{W_{n,m}^2}\left(W_{n,m}\frac{\partial W_{n,m-1}}{\partial p_j} - W_{n,m-1}\frac{\partial W_{n,m}}{\partial p_j}\right)\\ &= \frac{W_{n,m-1}}{W_{n,m}}\left(\frac{1}{W_{n,m-1}}\frac{\partial W_{n,m-1}}{\partial p_j} - \frac{1}{W_{n,m}}\frac{\partial W_{n,m}}{\partial p_j}\right)\\ &= \frac{\tilde{\lambda}(p,m)}{p_j}\left(\mathbb{E}[\tilde{Y}_j^{\mathrm{cs}} + \tilde{Y}_j^{\mathrm{d}} + \tilde{Y}_j^{\mathrm{c}} + \tilde{Y}_j^{\mathrm{u}}] - \mathbb{E}[Y_j^{\mathrm{cs}} + Y_j^{\mathrm{d}} + Y_j^{\mathrm{c}} + Y_j^{\mathrm{u}}]\right).\end{aligned}$$

由此可得精确的梯度，证明至此完成。

S 命题 9 的证明

我们首先形式化网络在任意实际运行时间 t 下的瞬时功耗，记为 $P(t)$。根据第7.5节确立的物理能耗模型，总功率是网络所有组件活跃状态的总和：

$$P(t) = \mathcal{P}^{\mathrm{cs}}\,\mathbf{1}\Big\{\sum_{i=1}^{n} Y_i^{\mathrm{cs}}(t) > 0\Big\} + \sum_{i=1}^{n}\Big(\mathcal{P}_i^{\mathrm{c}}\,\mathbf{1}\{Y_i^{\mathrm{c}}(t) > 0\} + \mathcal{P}_i^{\mathrm{u}}\,Y_i^{\mathrm{u}}(t) + \mathcal{P}_i^{\mathrm{d}}\,Y_i^{\mathrm{d}}(t)\Big). \tag{98}$$

回顾一下，K_ϵ 表示达到 ϵ-准确率所需的确定性全局轮次数（完全取决于路由向量 p、并发级别 m 和硬件服务率），而 τ_ϵ 表示相应的随机实际运行时间。达到此准确率阈值之前消耗的累计能耗为功率随时间的积分：

$$E_\epsilon = \int_0^{\tau_\epsilon} P(t)\,dt = \sum_{k=1}^{K_\epsilon(p,m)} \int_{T_{k-1}}^{T_k} P(t)\,dt,$$

其中，$T_0 = 0$ 和 $(T_k)_{k\geq 1}$ 是中央服务器上的更新完成时间序列。

由于聚合的连续时间状态过程 $(\xi(t))_{t\geq 0}$ 严格平稳且遍历（如引理20所证），更新区间 $[T_{k-1}, T_k)$ 内的能量增量序列也平稳。因此，在与完成时刻 $\{t_{\geq 0}\}$ 相关联的 Palm 概率测度 $\mathbb{P}^0$ 下，

任意单轮次所消耗的期望能耗保持不变：

$$\mathbb{E}^0\left[\int_{T_{k-1}}^{T_k} P(t)\,dt\right] = \mathbb{E}^0\left[\int_0^{T_1} P(t)\,dt\right].$$

通过调用平稳点过程的基本反演公式（参见 [51,推论 6.16], 针对函数 $X_t = P(t)$ 的求值），我们将 Palm 期望映射回标准的时间平均期望：

$$\mathbb{E}^0\left[\int_0^{T_1} P(t)\,dt\right] = \frac{\mathbb{E}[P(0)]}{\tilde{\lambda}(p,m)},$$

其中，$\tilde{\lambda}$ 是由公式 (26) 得出的稳态系统吞吐量。将这一关系代入累积能量求和可得：

$$\mathbb{E}^0[E_\epsilon] = \sum_{k=1}^{K_\epsilon(p,m)} \frac{\mathbb{E}[P(0)]}{\tilde{\lambda}(p,m)} = \frac{K_\epsilon(p,m)}{\tilde{\lambda}(p,m)}\,\mathbb{E}[P(0)]. \tag{99}$$

下一步是计算稳态期望功率 $\mathbb{E}[P(0)]$。根据不变测度 $\nu_{n,m}$ （来自公式 (96)）对 (98) 取期望，并利用期望的线性性，将总功率分解为：

$$\mathbb{E}[P(0)] = \mathcal{P}^{\mathrm{cs}}\,\mathbb{P}\Big(\sum_{i=1}^n Y_i^{\mathrm{cs}} > 0\Big) + \sum_{i=1}^n \Big(\mathcal{P}_i^{\mathrm{c}}\,\mathbb{P}(Y_i^{\mathrm{c}} > 0) + \mathcal{P}_i^{\mathrm{d}}\,\mathbb{E}[Y_i^{\mathrm{d}}] + \mathcal{P}_i^{\mathrm{u}}\,\mathbb{E}[Y_i^{\mathrm{u}}]\Big).$$

我们利用此前得出的排队指标来计算各项。对于 CS 活动概率，我们使用公式 (97) 中的恒等式：

$$\mathbb{P}\Big(\sum_{i=1}^n Y_i^{\mathrm{cs}} > 0\Big) = \frac{W_{n,m-1}}{W_{n,m}}\,\frac{1}{\mu^{\mathrm{cs}}}.$$

对本地计算节点应用相同的边缘化逻辑可得：

$$\mathbb{P}(Y_i^{\mathrm{c}} > 0) = \frac{W_{n,m-1}}{W_{n,m}}\,\frac{p_i}{\mu_i^{\mathrm{c}}}.$$

类似地，对于无限服务器通信阶段（下行链路和上行链路），我们有：

$$\mathbb{E}[Y_i^{\mathrm{d}}] = \frac{W_{n,m-1}}{W_{n,m}}\,\frac{p_i}{\mu_i^{\mathrm{d}}}, \qquad \mathbb{E}[Y_i^{\mathrm{u}}] = \frac{W_{n,m-1}}{W_{n,m}}\,\frac{p_i}{\mu_i^{\mathrm{u}}}.$$

最后，将这些稳态项代回 $\mathbb{E}[P(0)]$, ，我们发现每一项都含有公因子 $W_n W_{n\,m}^{\,m-1}$，这正是吞吐量 $\tilde{\lambda}(p,m)$。当代入（99）后，该因子会抵消乘数 $1/\lambda(p,m)$。抵消后即可得到最终的闭式表达式：

$$\mathbb{E}^0[E_\epsilon] = K_\epsilon(p,m)\left(\frac{\mathcal{P}^{\mathrm{cs}}}{\mu^{\mathrm{cs}}} + \sum_{i=1}^n p_i\left(\frac{\mathcal{P}_i^{\mathrm{c}}}{\mu_i^{\mathrm{c}}} + \frac{\mathcal{P}_i^{\mathrm{d}}}{\mu_i^{\mathrm{d}}} + \frac{\mathcal{P}_i^{\mathrm{u}}}{\mu_i^{\mathrm{u}}}\right)\right).$$

这表明，每轮的期望能耗严格取决于服务容量和路由决策，完全不受排队延迟影响，由此完成证明。