

UNISCHED: A Unified Scheduler for Deep Learning Training Jobs With Different User Demands

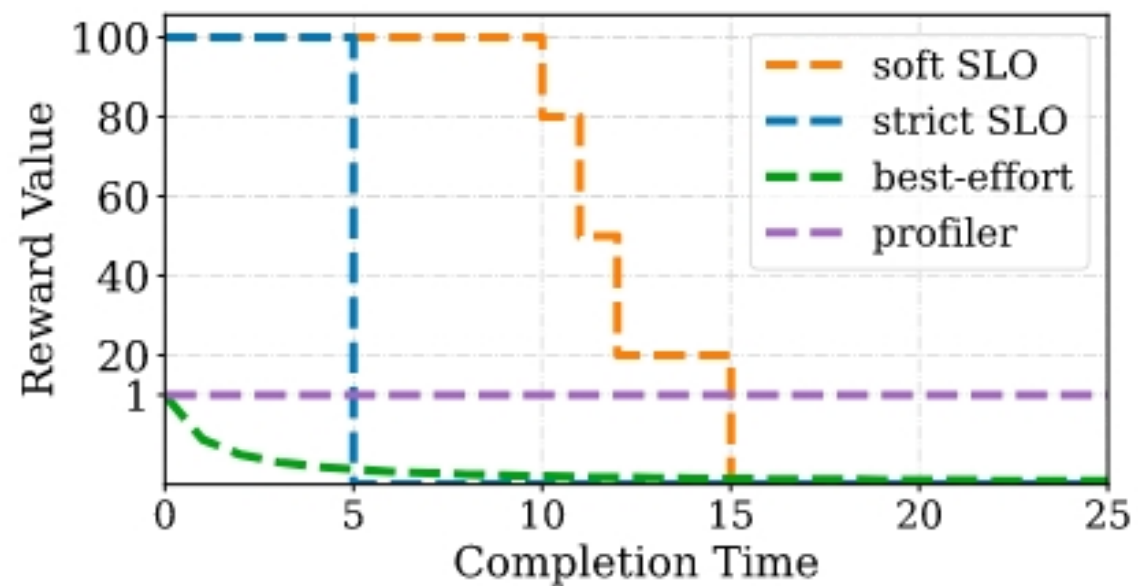
Wei Gao , Zhisheng Ye , Peng Sun , Tianwei Zhang , *Member, IEEE*, and Yonggang Wen , *Fellow, IEEE*

背景：深度学习（DL）技术的飞速发展，公司和机构通常会建设大规模 GPU 集群，以满足不同用户对 DLT 作业的高强度需求。为了管理 DLT 作业的执行并为其分配资源，就需要一个调度器。

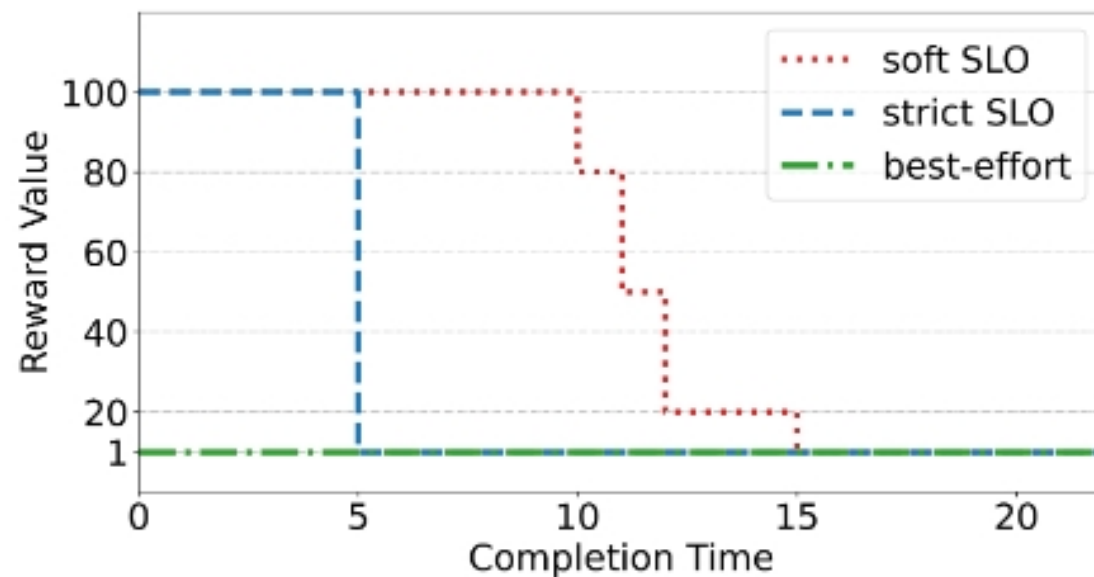
论文中的研究现状：首先，用户对调度时延可能有不同预期（SLO作业和尽力而为作业）。同时由于 DLT 是一个迭代过程，用户完成训练作业的停止条件也可能不同（迭代次数或者目标性能）。现有工作大多只关注某些特定需求，无法同时覆盖所有类型。

研究方向和贡献：本文提出 UNISCHED，这是一种新型调度系统，能够以统一方式满足所有调度时延需求和停止条件。

Selector



(a) Reward





每个作业会经过两个阶段：profiling（橙色），执行阶段（黑色）。

调度器两个主要组件组成：Estimator（用于预测作业持续时间）和 Selector（用于选择作业并为其分配资源以执行）。

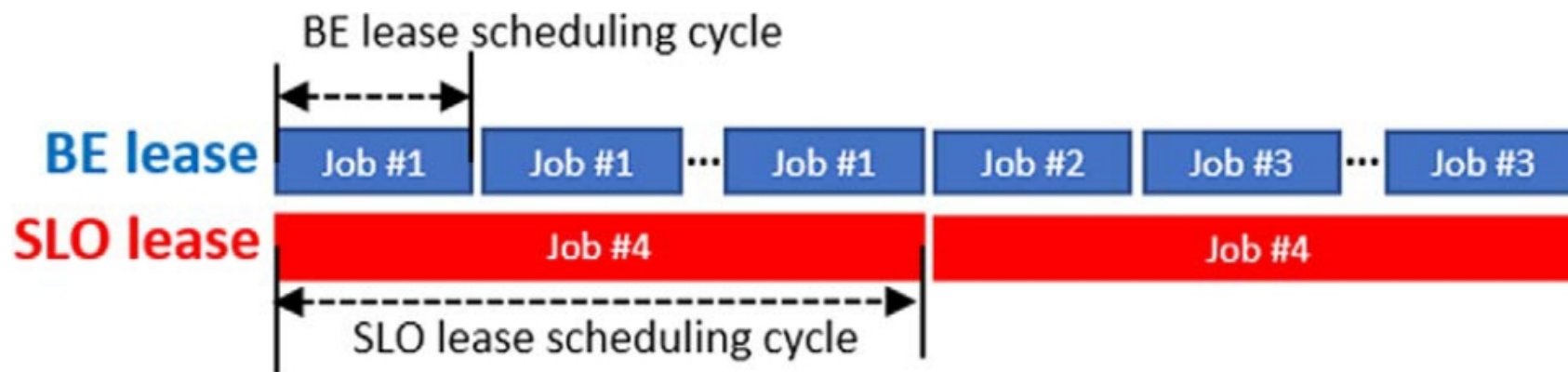
Estimator

$$T_i^{\text{dur}} = T_i^{\text{sr}} + N_i^{\text{iter}} \cdot T_i^{\text{iter}}$$

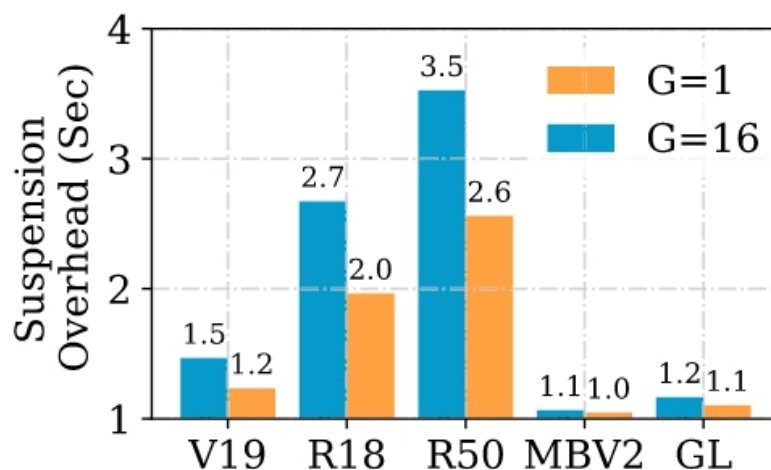
对于性能停止指标的作业。其基本思想是：用一组概率型学习曲线模型对观测到的性能指标进行建模。

$$T_i^{\text{iter}} = \begin{cases} T_i^{\text{comp}} & \text{if } N_i^{\text{gpu}} = 1, \\ T_i^{\text{comp}} + (N_i^{\text{gpu}} - 1) \cdot T_i^1 & \text{if } N_i^{\text{node}} = 1, N_i^{\text{gpu}} \geq 2, \\ T_i^{\text{comp}} + (N_i^{\text{gpu}} - 1) \cdot T_i^2 & \text{otherwise.} \end{cases} \quad (2)$$

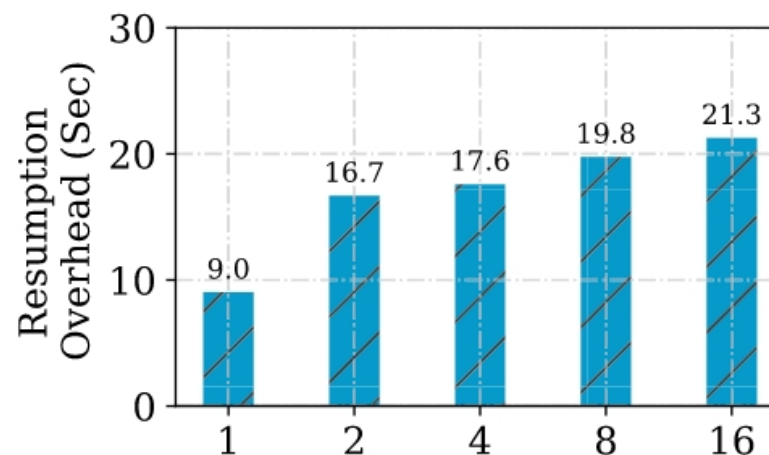
Estimator



只有当调度器为作业分配了一个租约周期时，该作业才允许运行。当租约到期时，作业需要进行续约，否则，会被挂起。



(a) Suspension Overhead



(b) Resumption Overhead

Estimator

$$T_i^{\text{dur}} = T_i^{\text{sr}} + N_i^{\text{iter}} \cdot T_i^{\text{iter}}$$

我们假设 DLT 作业发生挂起与恢复的事件服从均匀概率分布。

$$\frac{C_{n-1}^k C_{m-n+1}^{k+1}}{C_m^n}, \text{ where } k \in [0, \min(n-1, m-n)]$$

对于一个需要 n 个租期的尽力而为 (best-effort) 作业, 发生挂起与恢复的概率为 $\frac{1}{2}$

$$T_i^{\text{sr}} = \sum_{k=0}^{\min(n-1, m-n)} k \cdot t_i^{\text{sr}} \cdot \frac{C_{n-1}^k C_{m-n+1}^{k+1}}{C_m^n}. \quad T^{\text{sr}} \text{ is } \frac{n}{2} \cdot t_i^{\text{sr}}$$

Selector

$$\max_{\mathbf{S}} \sum_{i \in [N]} \sum_{f \in [F_i]} s_{f,i} V_{f,i}, \quad (4) \quad \text{最大化所有作业的总奖励值}$$

subject to:

$$x_{k,i}, s_{f,i} \in \{0, 1\}, \forall i \in [N], f \in [F_i], \quad (5) \quad \text{二元变量}$$

$$\sum_{f \in [F_i]} s_{f,i} \leq 1, \forall i \in [N], \quad (6) \quad \text{至多只能获得一个可行解}$$

$$\sum_{k \in [Q_{f,i}]} s_{f,i} x_{k,i} \leq s_{f,i} L_i, \forall i \in [N], f \in [F_i]. \quad (7) \quad \text{必须满足对于的截止时间要求}$$

Selector

$$y_{i,p} \in \{0, 1\}, \forall i \in [N], p \in [P_i], \quad (9) \quad \text{二元变量}$$

$$\sum_{g=0}^3 2^g \cdot N_g^{\text{cell}} \leq M, \quad (10) \quad \text{使用的不超过总的GPU数量}$$

$$N_g^{\text{con}} = \sum_{i \in [N]} \sum_{p \in [P_i]} y_{i,p} A_{i,p}(g), \forall g \in \{0, 1, 2, 3\}, \quad (11) \quad \text{保证资源分配方案的可行性}$$

$$\begin{aligned} & \sum_{k \in [Q_{f,i}]} y_{i,p} s_{f,i} x_{k,i} \bar{\phi}(j_i, A_{i,h}) \\ & \geq y_{i,p} s_{f,i} L_i, \forall i \in [N], f \in [F_i], p \in [P_i], \end{aligned} \quad (12) \quad \text{确保作业能完成}$$

$$\sum_{p \in [P_i]} y_{i,p} \leq 1, \forall i \in [N]. \quad (13) \quad \text{只被分配一个可行的方案}$$

EVALUATION

实验环境：（Physical Evaluation）搭建了一个由 16 个 GPU 节点组成的集群。

（Simulator Evaluation）我们评估了两个同构 GPU 集群 C120 和 C96 的性能，它们分别包含 120 个和 96 个节点，每个节点配备 8 个 GPU。

工作负载：使用了两个真实的 DLT 作业轨迹，自 SenseTime 的 Helios 轨迹 和来自 Microsoft 的 Philly 轨迹。

对于严格 SLO 作业，在作业持续时间的 1.1 到 2 倍范围内随机生成截止时间。

对于软 SLO 作业：首个截止时间 $D_{0,i}$ 与严格 SLO 作业相同方式生成，额外设置三个截止时间： $1.1 \times D_{0,i}$ （奖励 80） $1.2 \times D_{0,i}$ （奖励 50） $1.5 \times D_{0,i}$ （奖励 20）这些奖励值基于用户调查。

80% 作业采用迭代准则，20% 作业采用性能准则。

EVALUATION

Table 1: Summary of workloads in our experiments

Workload	Strict/Soft/BE (%)	Trace	Cluster	# of jobs
Short	0 / 0 / 100	Helios	–	5,735
H_SLO	100 / 0 / 0	Helios	C96	6,599
H_MIX1	70 / 0 / 30	Helios	C96	6,599
H_MIX2	30 / 60 / 10	Helios	C96	6,599
P_SLO	100 / 0 / 0	Philly	C120	30,940
P_MIX1	70 / 0 / 30	Philly	C120	30,940
P_MIX2	30 / 60 / 10	Philly	C120	30,940

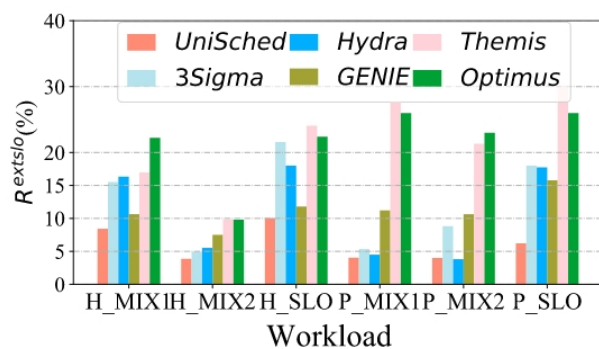
指标: Weighted Deadline Miss Rate.

$$R^{\text{slo}} = \frac{1}{|J^{\text{slo}}|} \sum_{j_i^{\text{slo}} \in J^{\text{slo}}} \frac{\mathcal{W}(j_i^{\text{slo}}) - \mathcal{W}_{\min}}{\mathcal{W}_{\max} - \mathcal{W}_{\min}}.$$

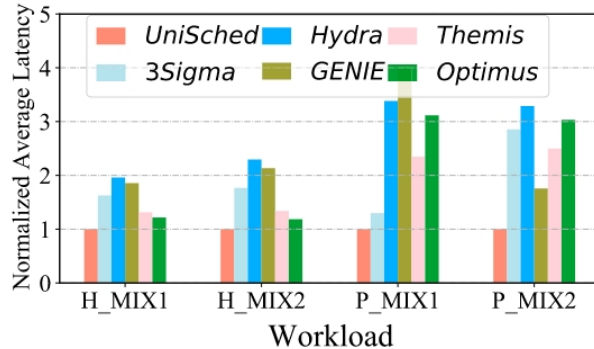
Job Completion Time (JCT).

EVALUATION

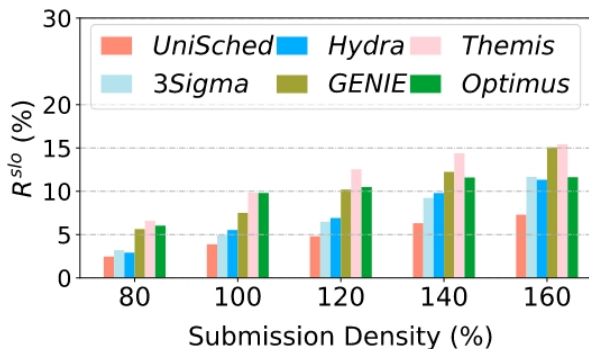
Job Load	$T[360]$		$T[720]$	
Metric	R^{slo} (%)	Avg JCT (min)	R^{slo} (%)	Avg JCT (min)
Simulator	4.97	266.39	13.52	253.98
Kubernetes	3.92	274.42	16.11	267.40
Relative Diff	1.08%	2.93%	2.57%	5.38%



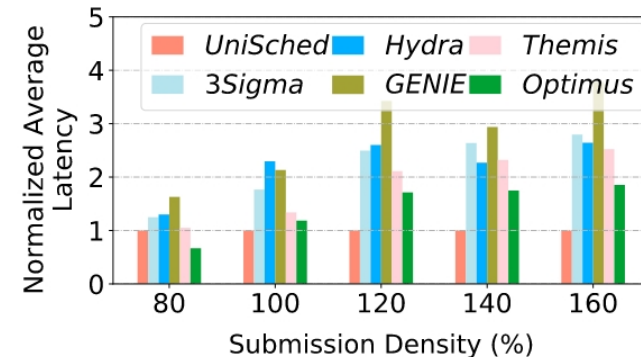
(a) R^{slo}



(b) Average JCT



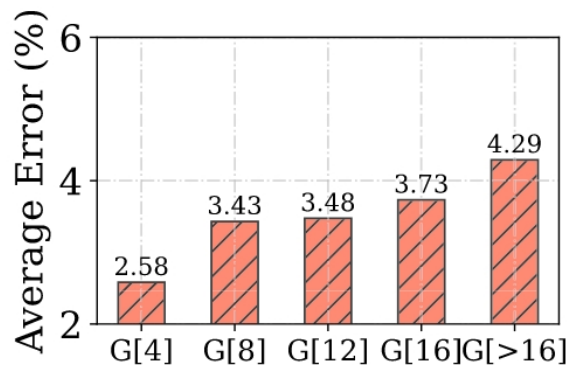
(c) R^{slo} versus job density



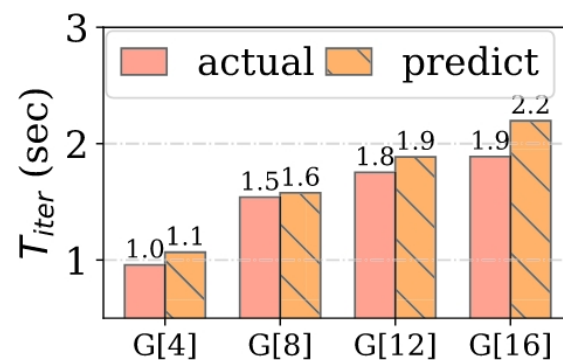
(d) Avg. JCT versus job density

EVALUATION

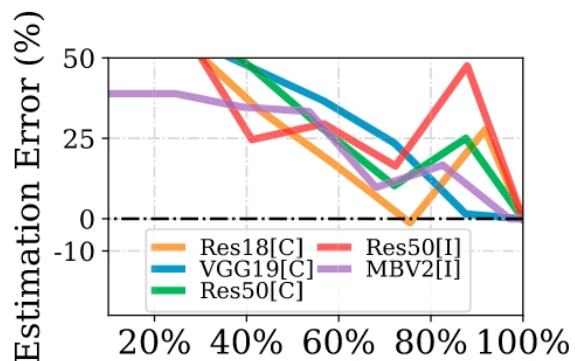
Training Iteration Estimator



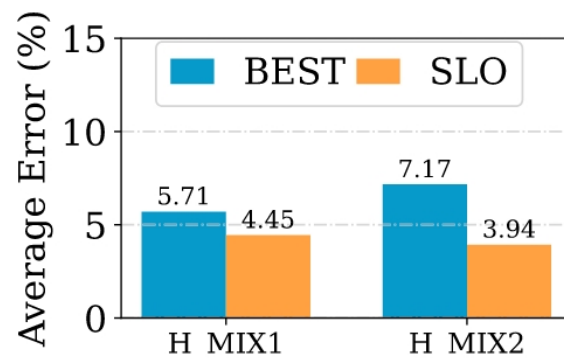
(a) Runtime Speed Estimator



(b) BERT's Runtime Speed

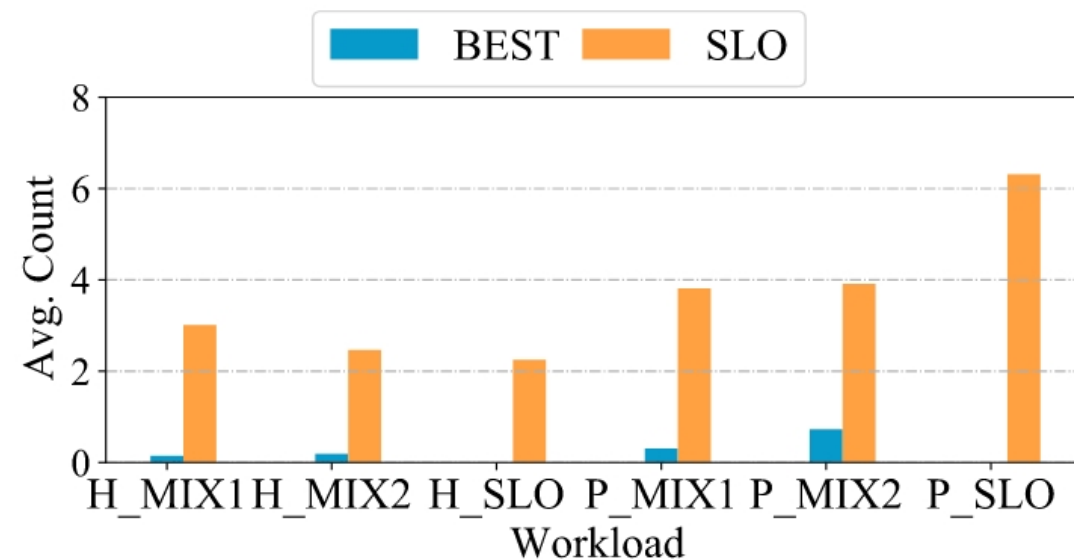


(c) Iteration Estimator



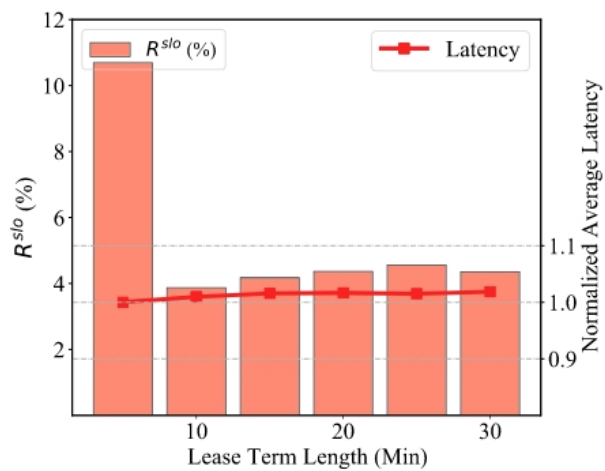
(d) Job Type

Analysis of suspension and resumption.

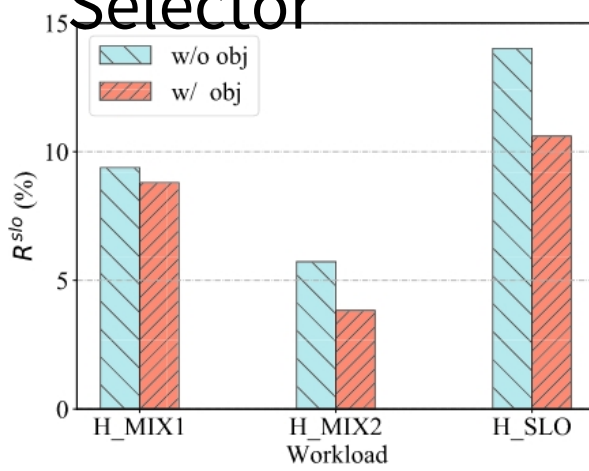


EVALUATION

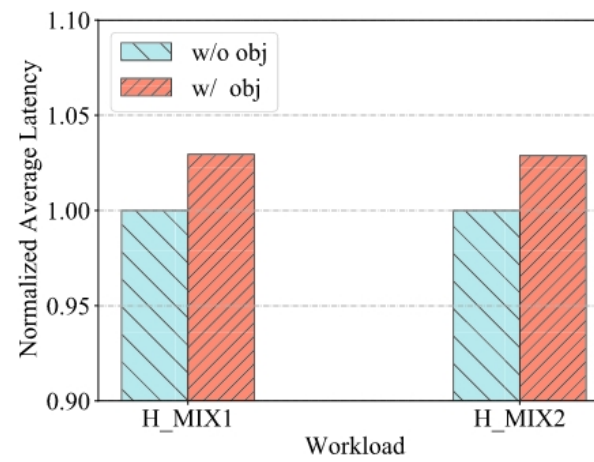
Performance analysis of the Selector



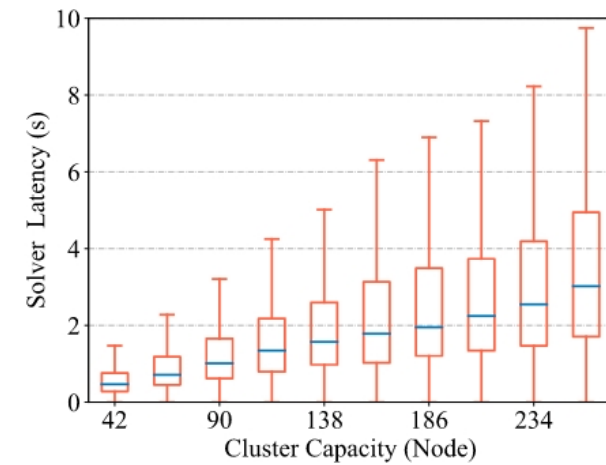
(a) Impact of the SLO lease length



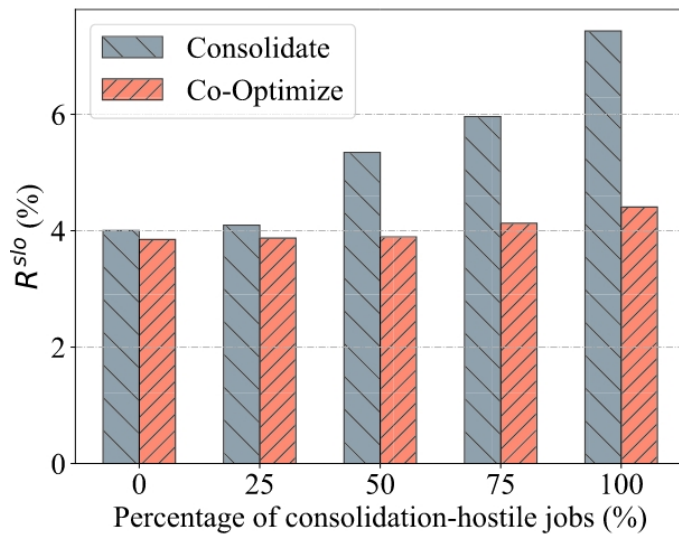
(b) SLO Impact of the objective



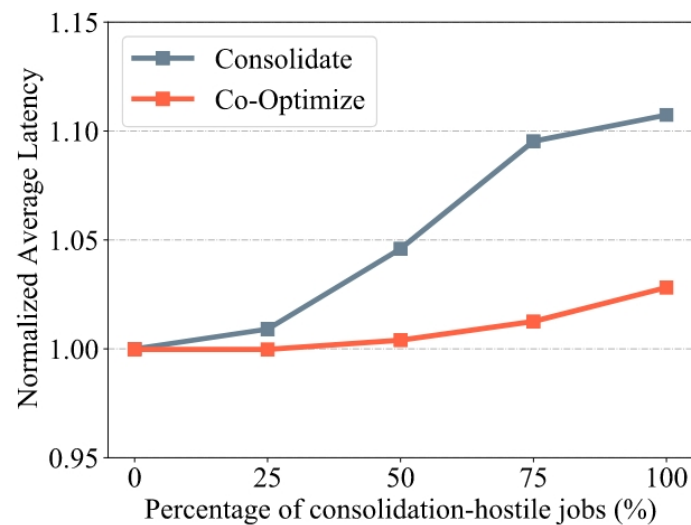
(c) Latency Impact of the objective



(d) Scalability analysis

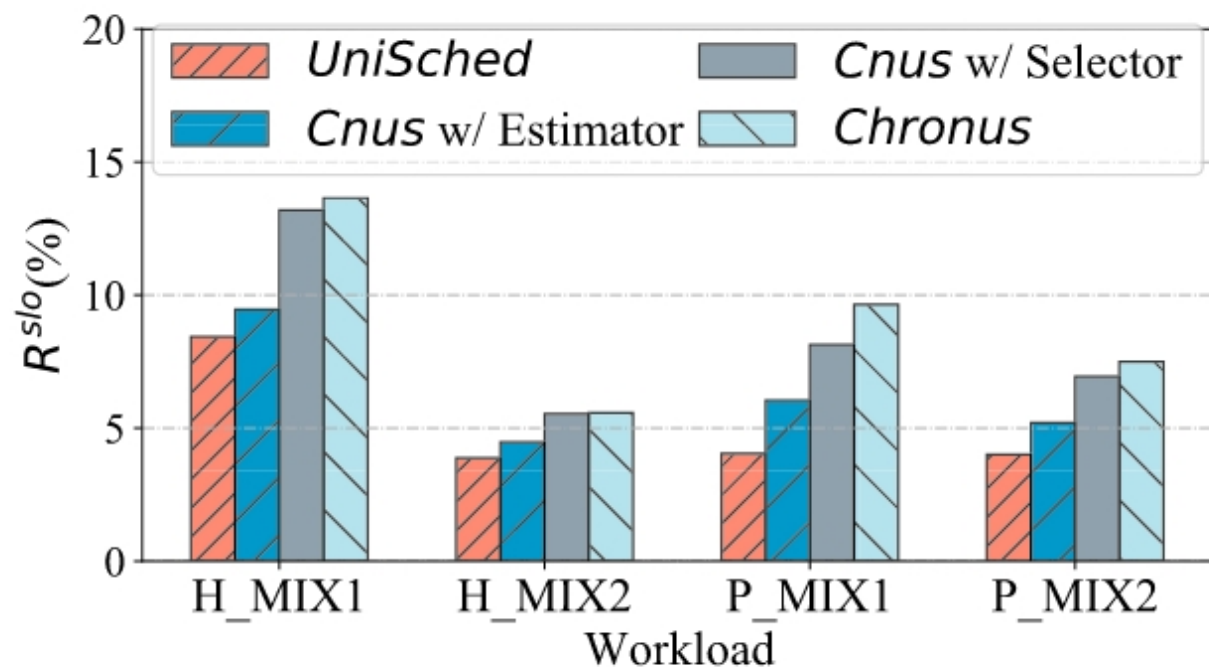


(a) R^{slo}

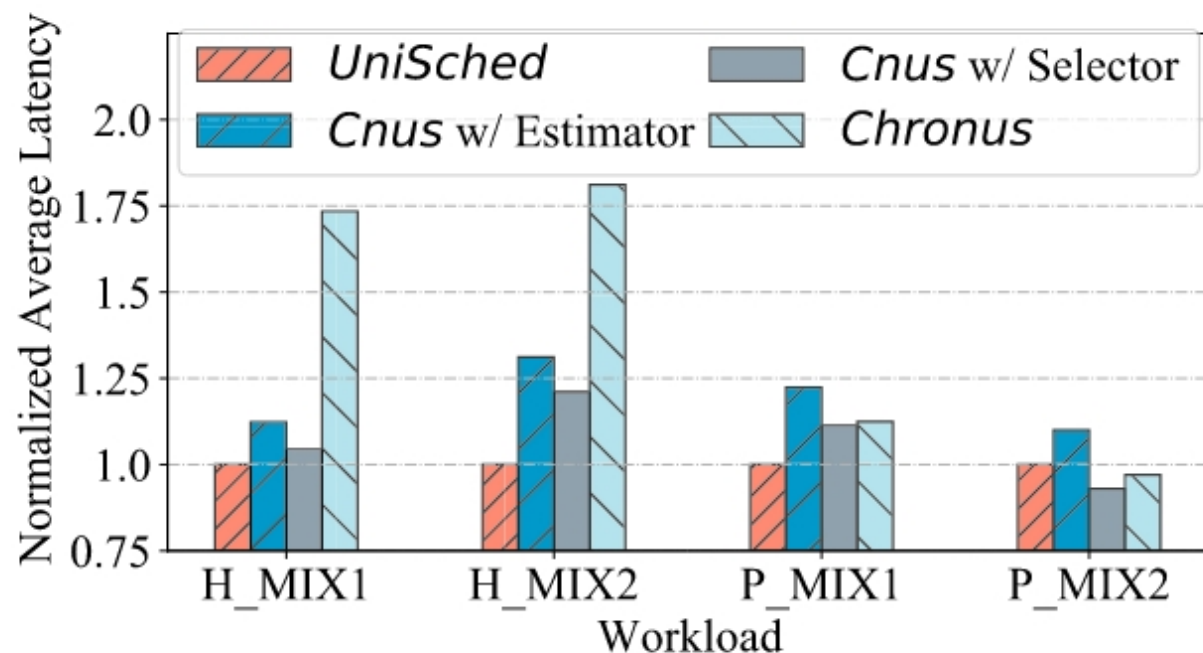


(b) Normalized Average Latency

EVALUATION



(a) R^{slo}



(b) Normalized Average Latency