LEVELLER: Fair Communication Scheduling via Progress-Rate Awareness in Multi-Tenant Training Clusters

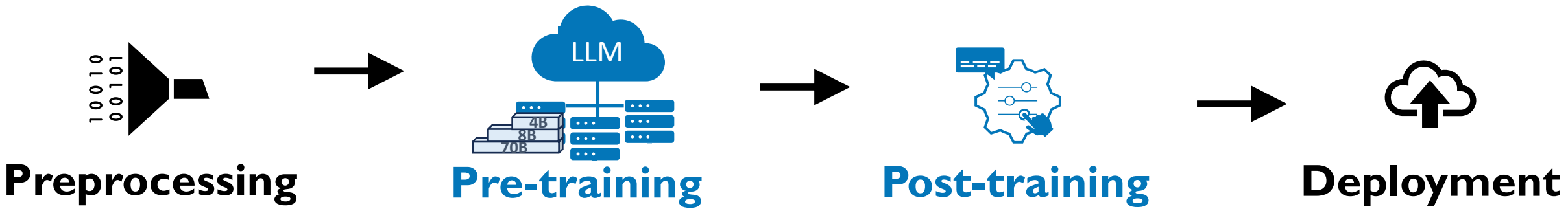
Geng Li, Yang Li, Mingyuan Zang, Jie Wu

China Telecom

Cloud Computing Research Institute

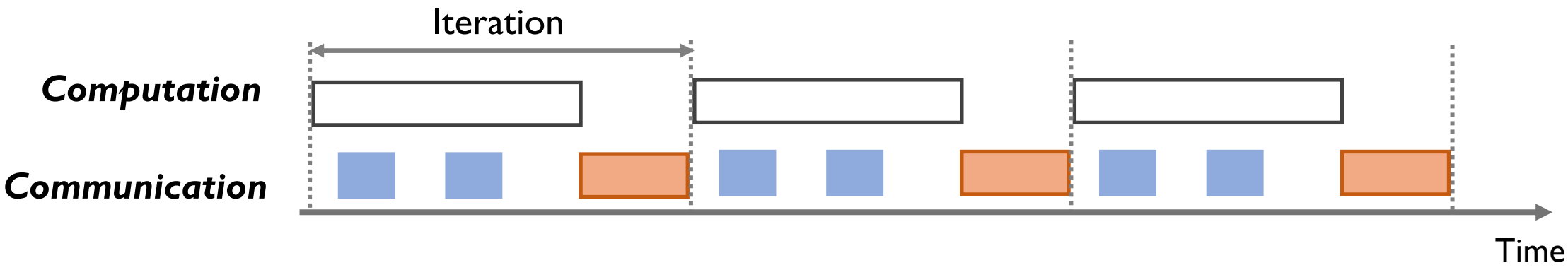
Deep Learning Training (DLT)

- AI explosion
- DLT is dominant in infrastructure complexity / capability frontier



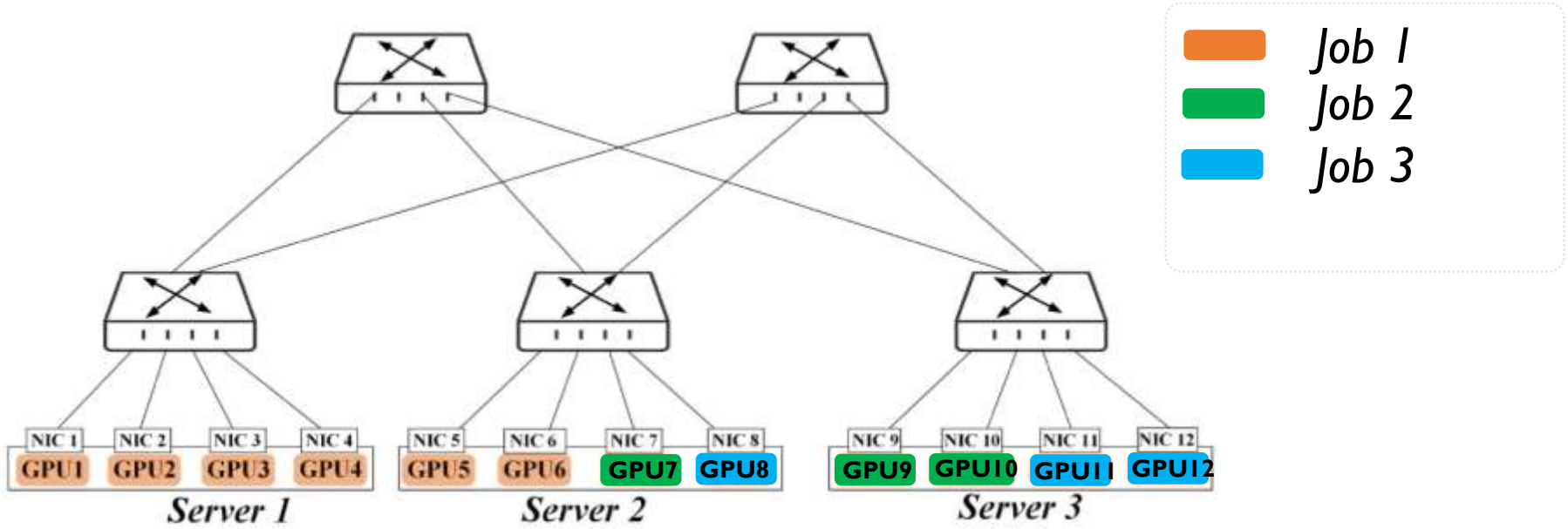
DLT Job

- Distributed across accelerators (data, pipeline, tensor parallelism)
- Iterations of ***computation*** and ***communication*** phases
  - Repeat periodically (excluding MoE)



Multi-Tenant Training Cluster

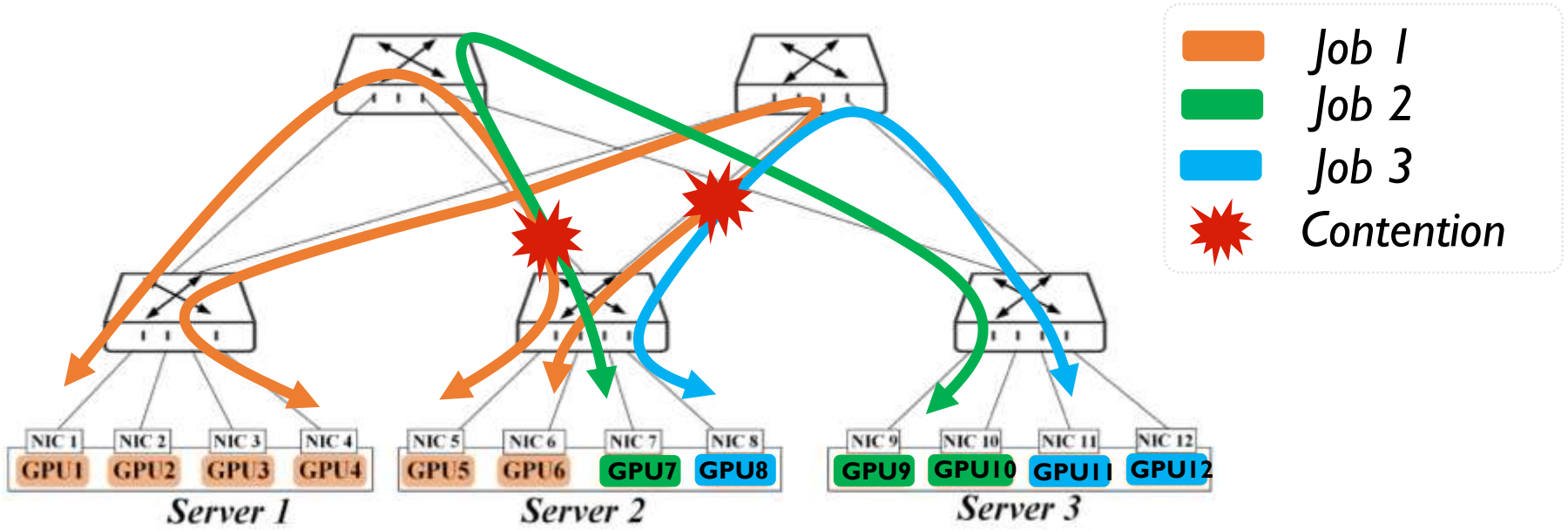
- Training multiple jobs concurrently
  - 30+ jobs in a 2,000-GPU cluster [1]



[1] http://github.com/alibaba/alibaba-lingjun-dataset-2023

Multi-Tenant Training Cluster

- Network shared among jobs
  - Inter-job network contention, e.g., BW oversubscription, imperfect LB
  - 36% of DLT jobs suffer from inter-job contention [1]



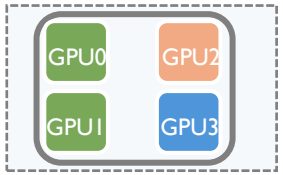
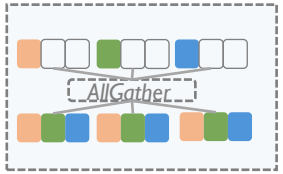
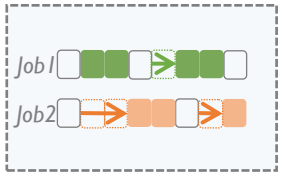
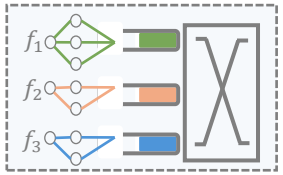
[1] Crux: GPU-efficient communication scheduling for deep learning training, SIGCOMM 24

Trade-off: Efficiency vs. Fairness

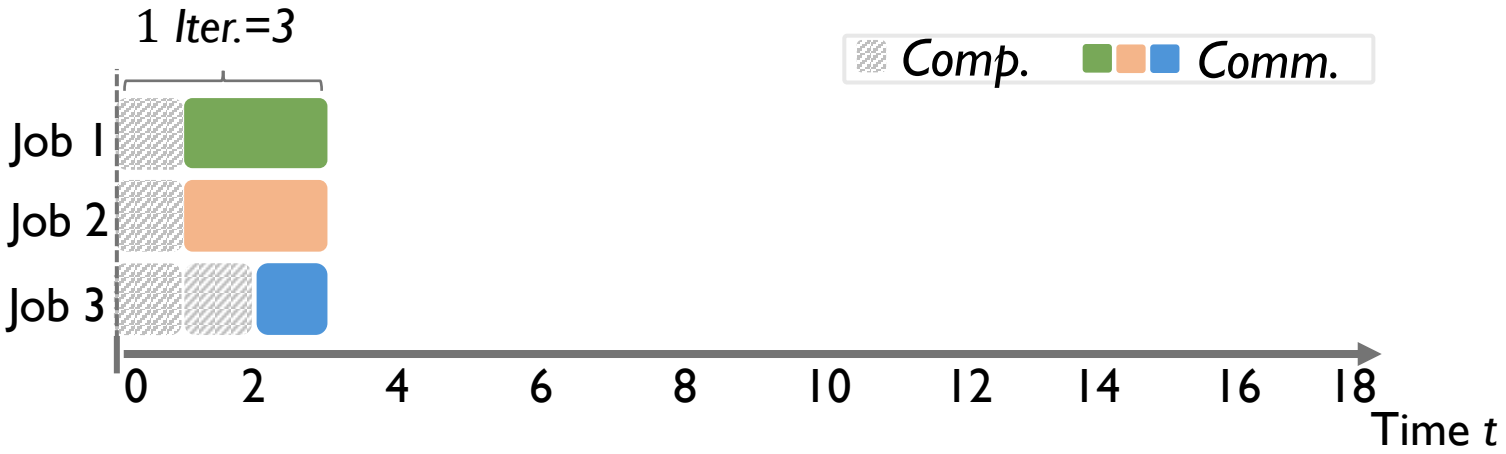
State of the Art

Maximize Utilization

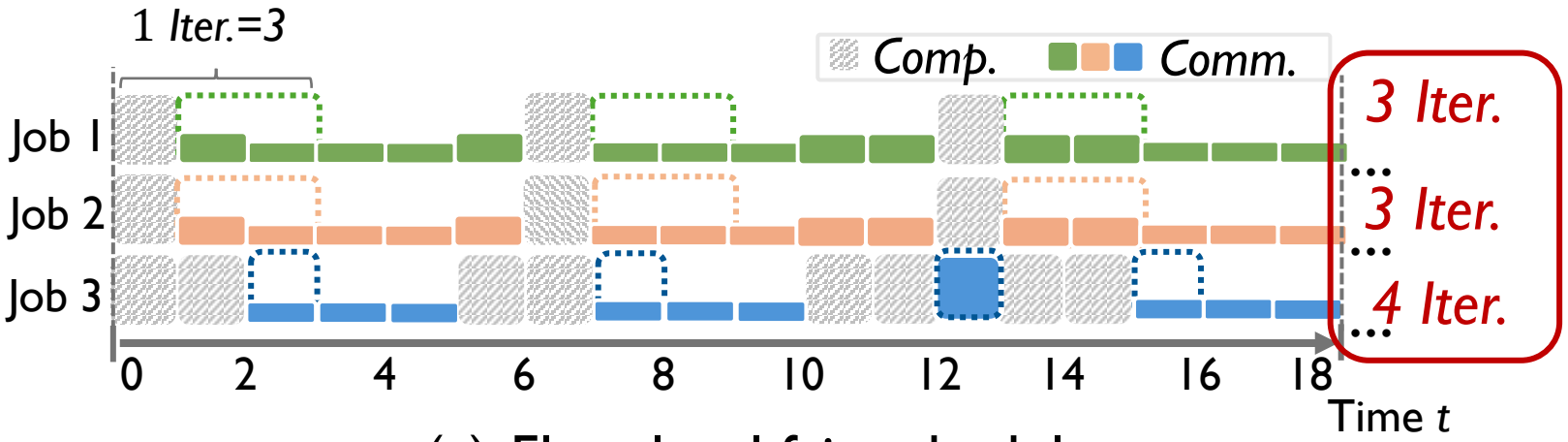
Cloud Integrity

	Efficiency e.g., measured by the average performance	Fairness e.g., assessed by the minimum of all	
 **Job placement & scheduling**	**HiveD** [OSDI 20], **Lucid** [SIGCOMM 23]	**THEMIS** [NSDI 20], **Shockwave** [NSDI 23]	Only computational fairness
 **Intra-job comm scheduling**	**TACCL** [NSDI 23], **TECCL** [SIGCOMM 24], **SyCCL** [SIGCOMM 25]	*NA*	Single job
 **Inter-job comm scheduling**	**CASSINI** [NSDI 24], **CRUX** [SIGCOMM 24], **MCCS** [SIGCOMM 24], **Symphony** [ICNP 25]	**Underexplored!** → **Ours: LEVELLER**	
 **Flow/coflow scheduling**	**pFabric** [SIGCOMM 13], **PDQ** [SIGCOMM 12], **Sincronia** [SIGCOMM 18]	**DCQCN** [SIGCOMM 15], **Swift** [SIGCOMM 20], **Coflex** [INFOCOM 17]	Fundamentally different!

Gap: Flow-level Fairness v.s. DLT Job-level Fairness

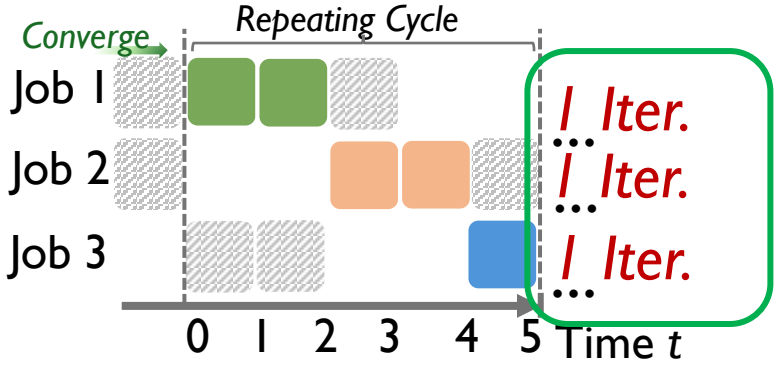


Gap: Flow-level Fairness v.s. DLT Job-level Fairness



(a) Flow-level fair schedule

Unfair!



(b) Job-level fair schedule

Fair!

Normalized Progress Rate as Fairness Metric

Normalized Progress Rate

$$R_j(t) = \frac{\text{ActualProgress}\ (t)}{\text{IdealProgress}\ (t)} = \frac{N_j(t)}{\left(t - t_j^0\right)/T_j^{\text{ideal}}}$$

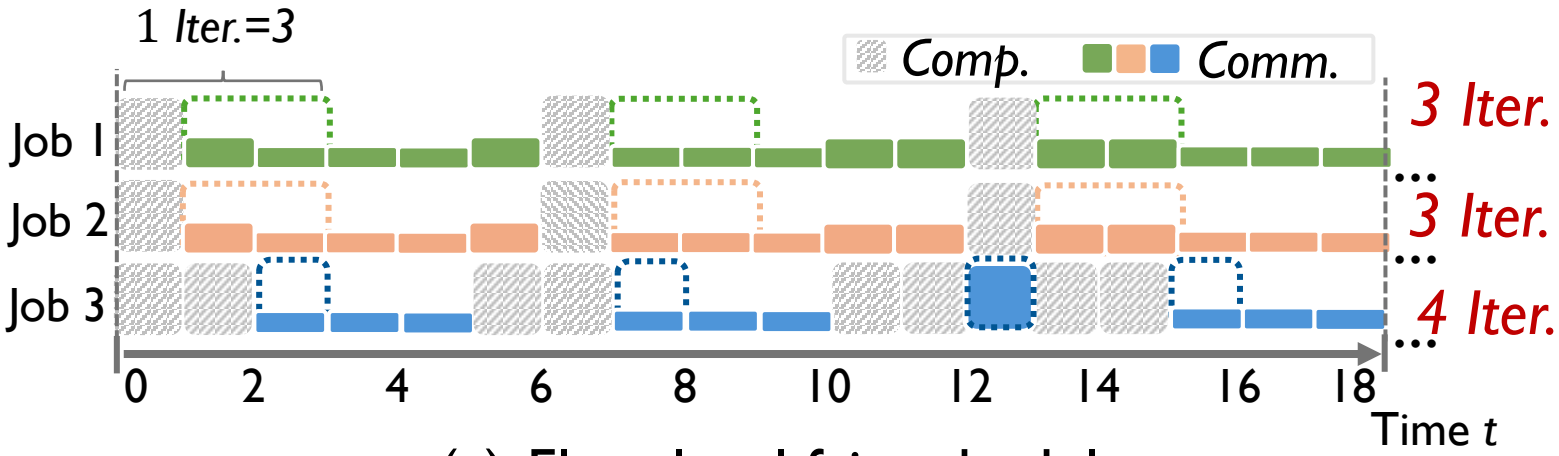
Job j's Progress at time t

Ideal iteration time of Job j (from profiling)

Job j starts at time t^0

- How closely a job approaches its maximum potential rate
  - Range [0, 1]
  - Approximately 1/iteration-time-slowdown
- A unified, job agnostic metric reflecting training experience, e.g., 2s delay
  - 1s-iteration job (rate=1/3)
  - 100s-iteration job (rate=100/102)

Gap: Flow-level Fairness v.s. DLT Job-level Fairness

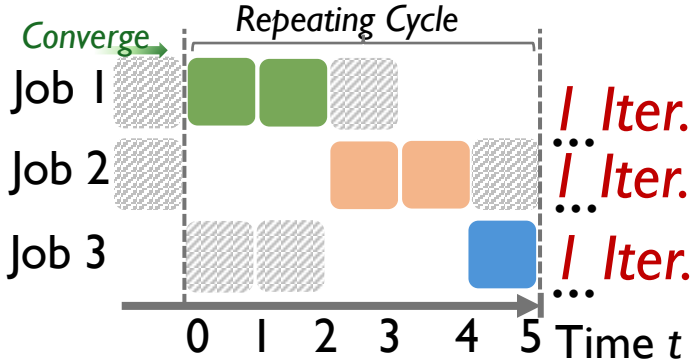


(a) Flow-level fair schedule

$$R_1(\infty) = \frac{3}{18/3} = 0.5$$

$$R_2(\infty) = \frac{3}{18/3} = 0.5$$

$$R_3(\infty) = \frac{4}{18/3} = 0.67$$



(b) Job-level fair schedule

$$R_1(\infty) = R_2 = R_3 = \frac{1}{5/3} = 0.6$$

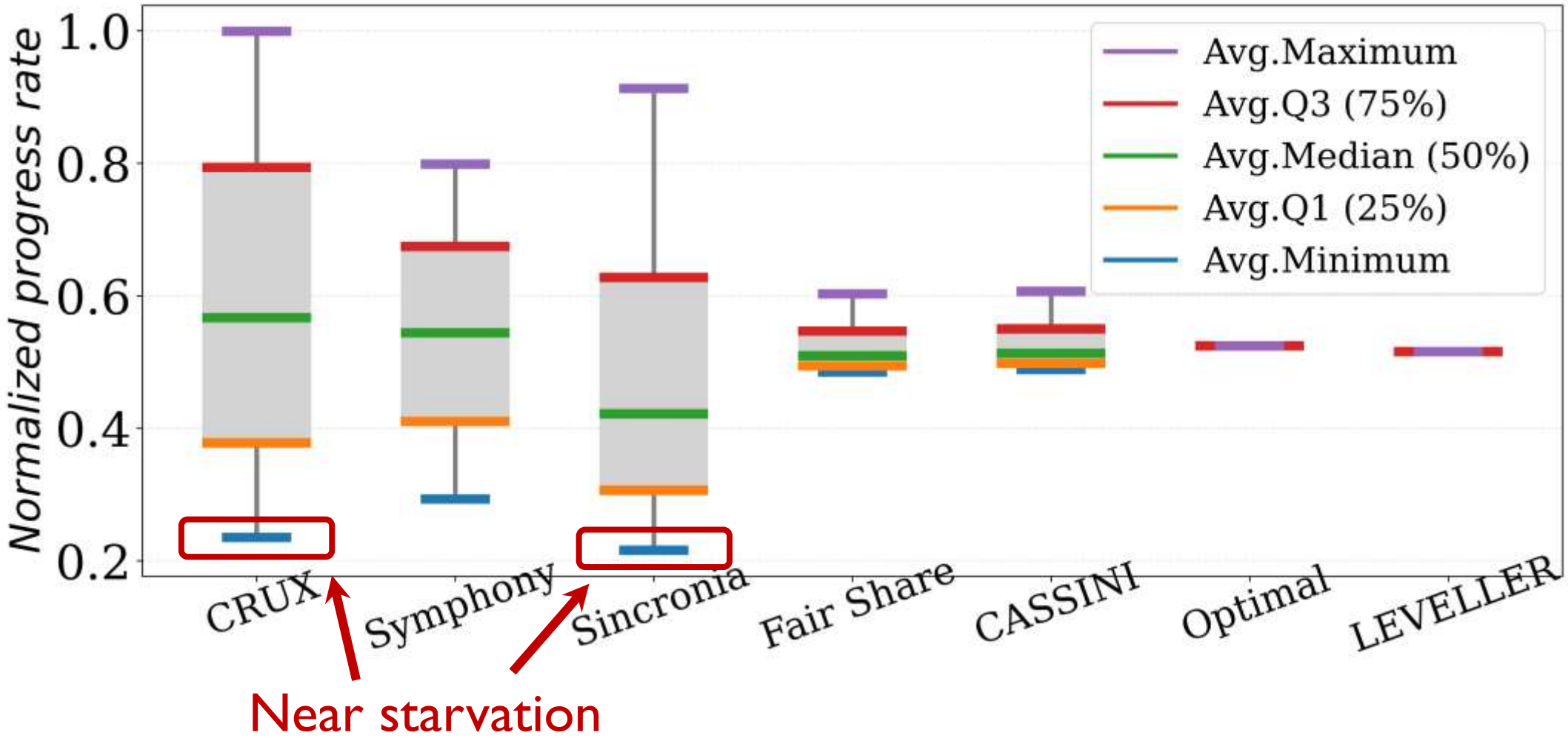
Job j's Progress at time t

$$R_j(t) = \frac{\text{ActualProgress}\ (t)}{\text{IdealProgress}\ (t)} = \frac{N_j(t)}{\left(t - t_j^0\right)/T_j^{\text{ideal}}}$$

Ideal iteration time of Job j (from profiling)

Job j starts at time t^0

Unfairness in Existing Schedulers



Our Goal: Max-Min Fairness

- Flow-scheduling policy S in a long execution window $[T_{\text{begin}}, T_{\text{end}}]$
- Lexicographic maximization problem

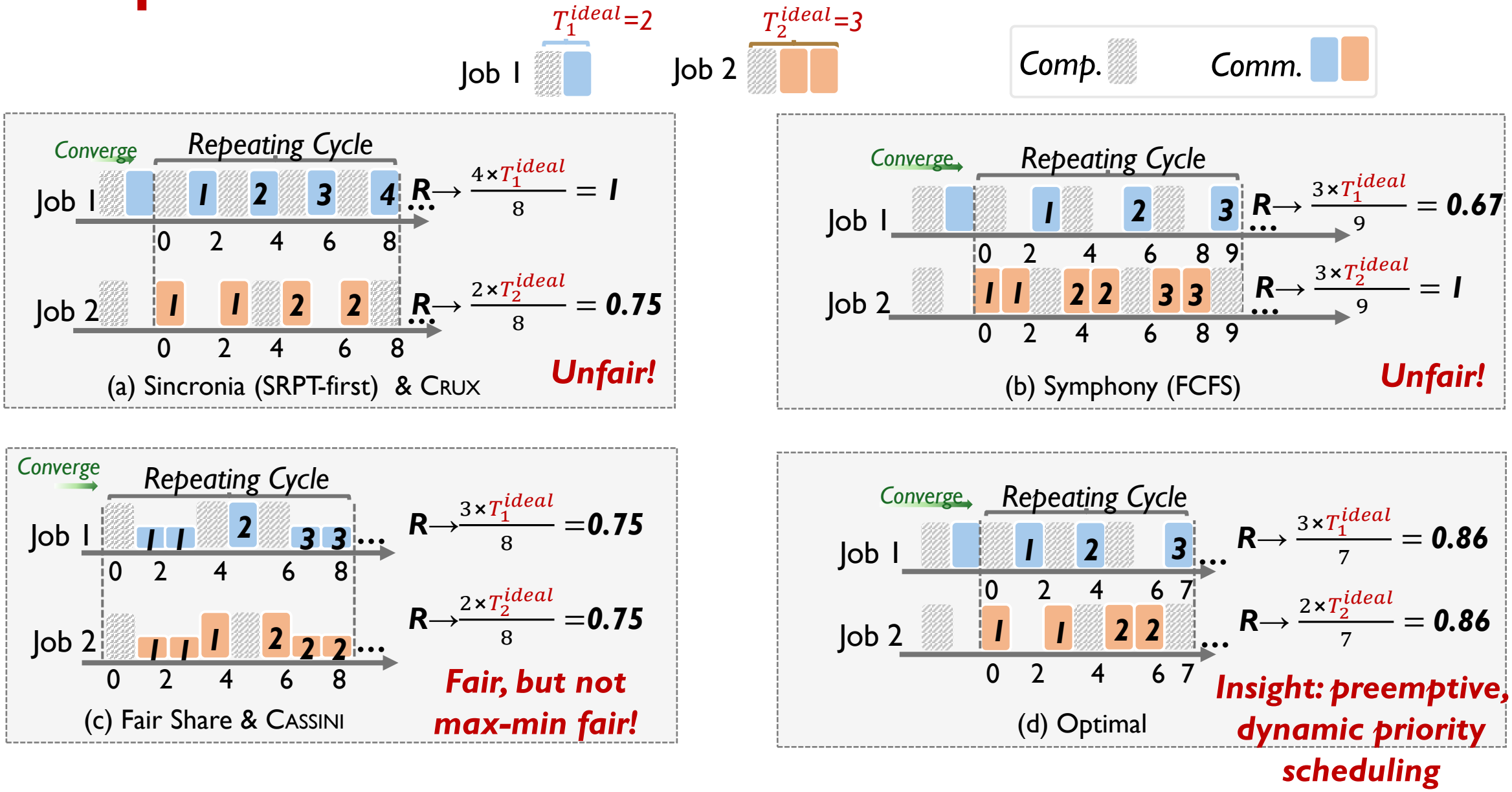
Non-decreasing order

$$\max_{S} \text{sort}_{\uparrow}\{R_j(T_{\text{end}})\} \mid j \in J$$

$$S = \{s(t) \mid t \in [T_{\text{begin}}, T_{\text{end}}]\}$$

- Intuition: universally share the slowdown
- Capture both efficiency and fairness

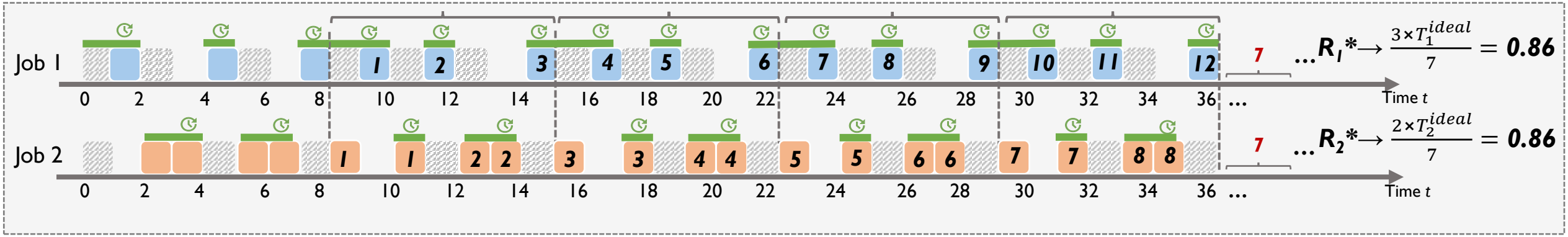
Example


T_1^{ideal}=2
T_2^{ideal}=3
Job 1
Job 2
Comp.
Comm.
Converge
Repeating Cycle
Job 1
Job 2
0 2 4 6 8
$R\rightarrow \frac{4\times T_1^{ideal}}{8} = 1$
$R\rightarrow \frac{2\times T_2^{ideal}}{8} = 0.75$
(a) Sincronia (SRPT-first) & CRUX
Unfair!
0 2 4 6 8 9
$R\rightarrow \frac{3\times T_1^{ideal}}{9} = 0.67$
$R\rightarrow \frac{3\times T_2^{ideal}}{9} = 1$
(b) Symphony (FCFS)
Unfair!
0 2 4 6 8
$R\rightarrow \frac{3\times T_1^{ideal}}{8} = 0.75$
$R\rightarrow \frac{2\times T_2^{ideal}}{8} = 0.75$
(c) Fair Share & CASSINI
Fair, but not max-min fair!
0 2 4 6 7
$R\rightarrow \frac{3\times T_1^{ideal}}{7} = 0.86$
$R\rightarrow \frac{2\times T_2^{ideal}}{7} = 0.86$
(d) Optimal
Insight: preemptive, dynamic priority scheduling

Theoretic Framework (1) of LEVELLER

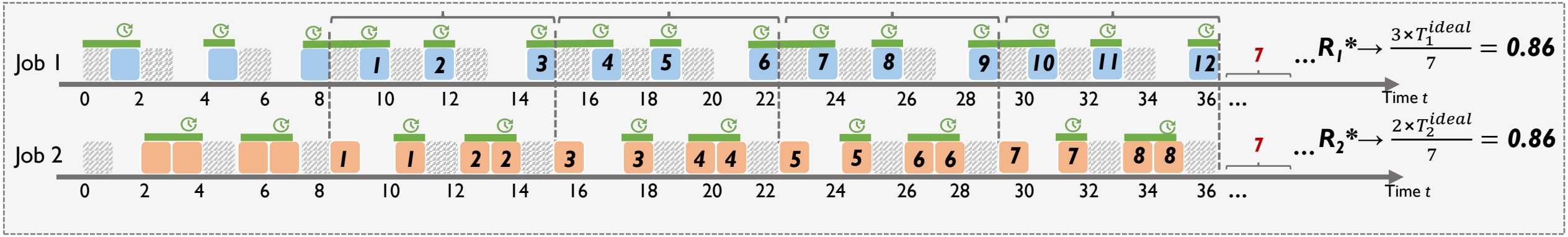
- Near-optimal schedule on a single link
  - Work-conserving, preemptive
  - Rate-aware greedy: non-decreasing $R_j(t)$

$$R_j(t) = \frac{N_j(t)}{\left(t - t_j^0\right)/T_j^{\text{ideal}}}$$



Theorem 1: The simple greedy scheduling achieves a 1/2-approximation of the max-min fairness.

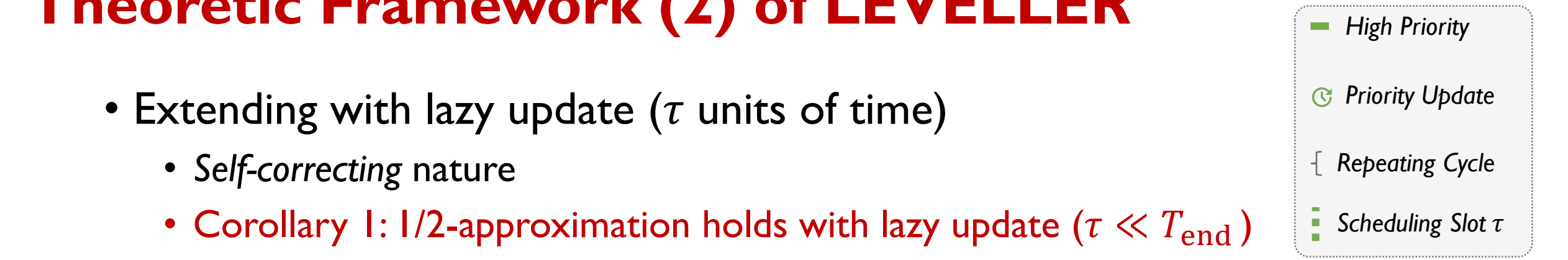
Theoretic Framework (1) of LEVELLER



Problem: frequent, real-time priority update, substantial overhead!

Theoretic Framework (2) of LEVELLER

- Extending with lazy update (τ units of time)
  - *Self-correcting* nature
  - Corollary 1: 1/2-approximation holds with lazy update ($\tau \ll T_{\text{end}}$)

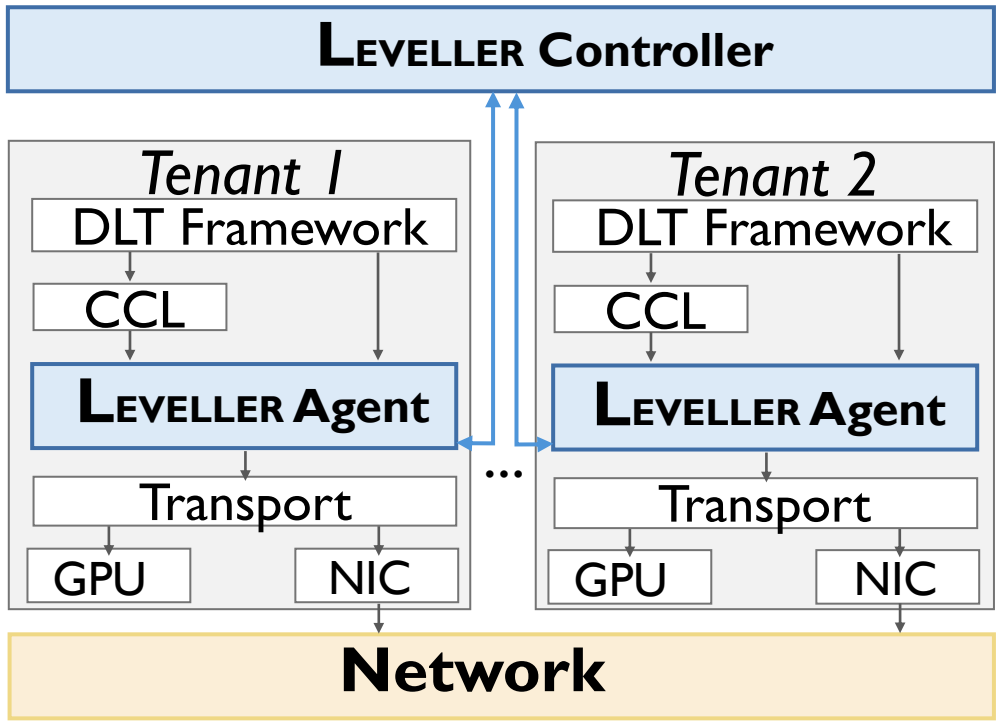


Theoretic Framework (3) of LEVELLER

- Extending to general networks and collectives (coflow with dependencies)
  - Corollary 2: 1/2-approximation holds in general networks
  - Global priority assignment based jobs' *normalized progress rates*

A loose bound, but guarantees
No Starvation!

LEVELLER System



- Priority compression over limited priority levels to minimize fairness loss
- Scheduling optimization by leveraging communication-computation overlap
- Dynamic RDMA priority (Traffic Class) adjustment via QP Resurrection

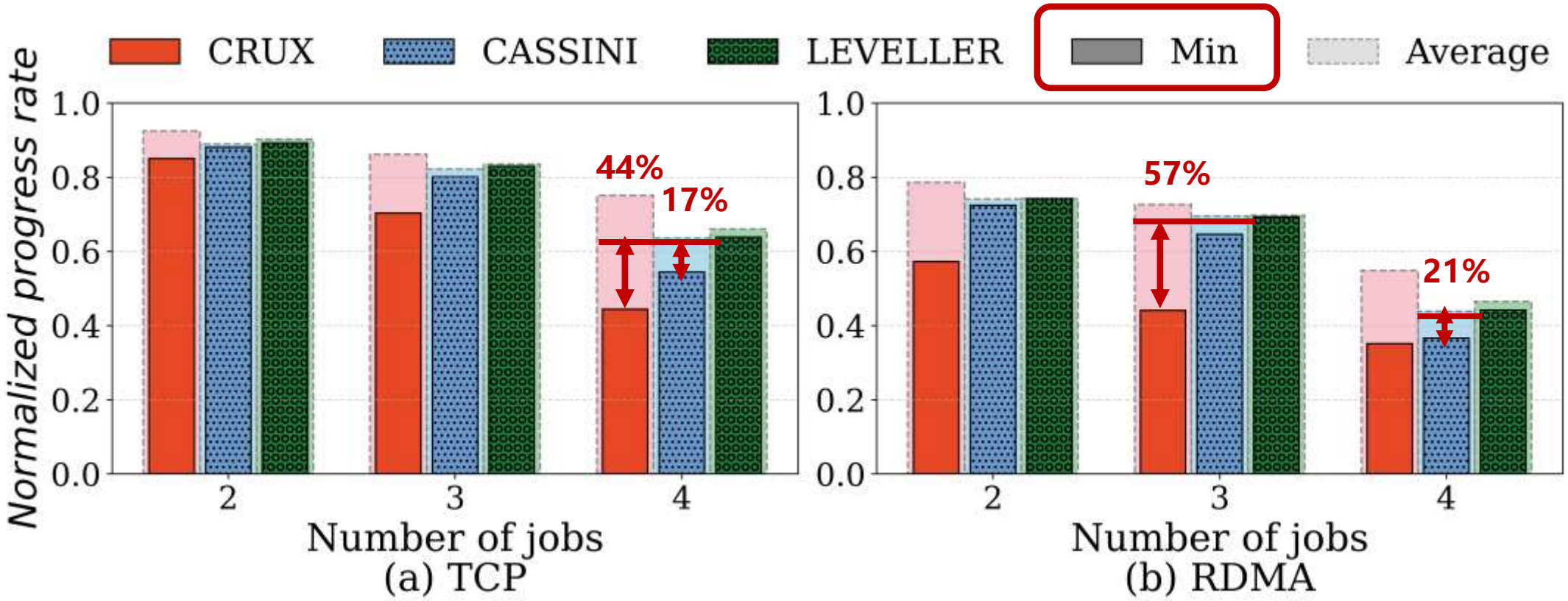
Evaluation Setup

- **Testbed**
  - 3 hosts, each with 4 NVIDIA A100 GPUs and 4 CX-6 NICs.
  - 3 leafs and 2 spines
- Llama-3.1, ChatGLM-3, Gemma, and DeepSeek-Coder
- ECMP and DCQCN by default
- CASSINI and CRUX (w/o routing)

- **Trace-based simulation**
  - 1,024 GPUs across 128 racks
  - (1) Oversubscribed (8:1): 3-tier Clos with 128 ToR, 16 leafs, and 8 spines
  - (2) Non-blocking (1:1): 2-tier multi-rail, 2 planes with 16 leafs
- Trace:
  - Empirical workloads [1]
  - 10 LLMs using STG Chakra and ASTRA-sim
- CASSINI, CRUX, Symphony, Sincronia, and Fair Share (DCQCN)

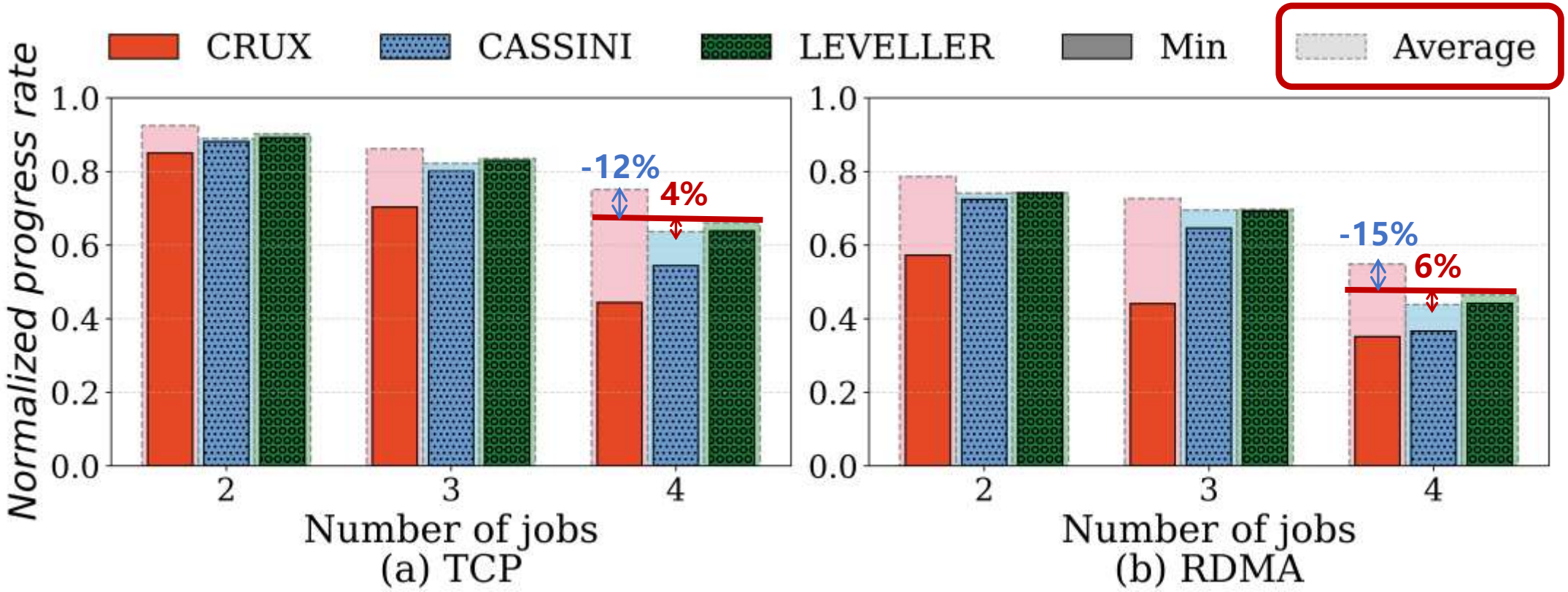
[1] http://github.com/alibaba/alibaba-lingjun-dataset-2023

Testbed Results



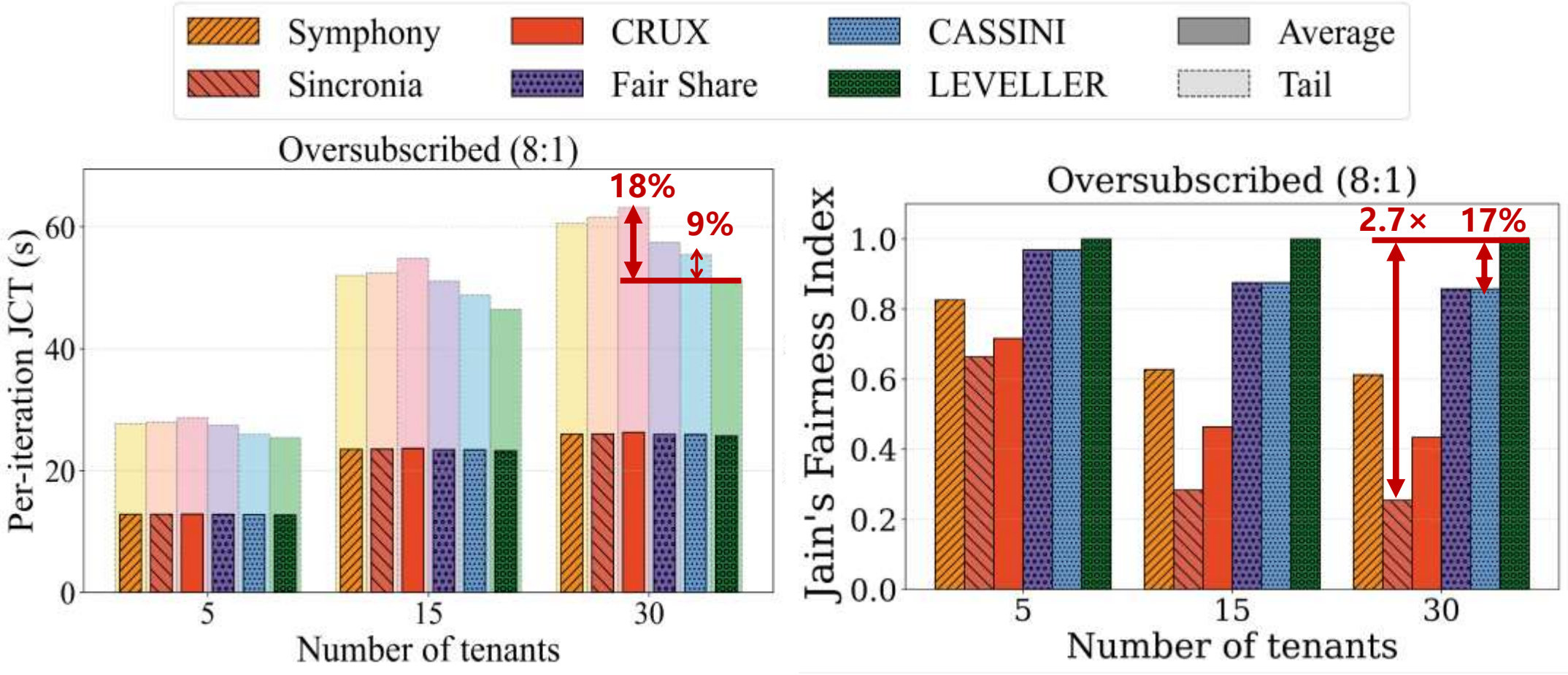
Improve the **minimum progress rate** by up to **57%**

Testbed Results



Efficiency vs. Fairness Trade-off: minimal efficiency loss, or even improve both

Simulation Results: JCT and Jain Index



Reduce **tail JCT** by **9% to 18%** and improve **fairness** by **17% to 2.7×**

Conclusion

- **Communication scheduling fairness in multi-tenant DLT clusters**
  - Gap between flow-level fairness and job-level fairness
- ***Normalized progress rate***
  - Ratio of a job's actual progress to its contention-free ideal
  - A unified, job agnostic metric reflecting training experience
  - Metric for future research (e.g., CC and TE) and potential standard for training service
- **LEVELLER, a practical system for max-min fairness**
  - Preemptive, dynamic-priority flow scheduling
  - 1/2-approximated theoretical guarantee
  - Improves the minimum progress rate by 57% and fairness by 17%, minimal efficiency loss

Thank you!

Geng Li
ligeng66@aliyun.com