CIFellows 2020-2021

Computing Innovation Fellows

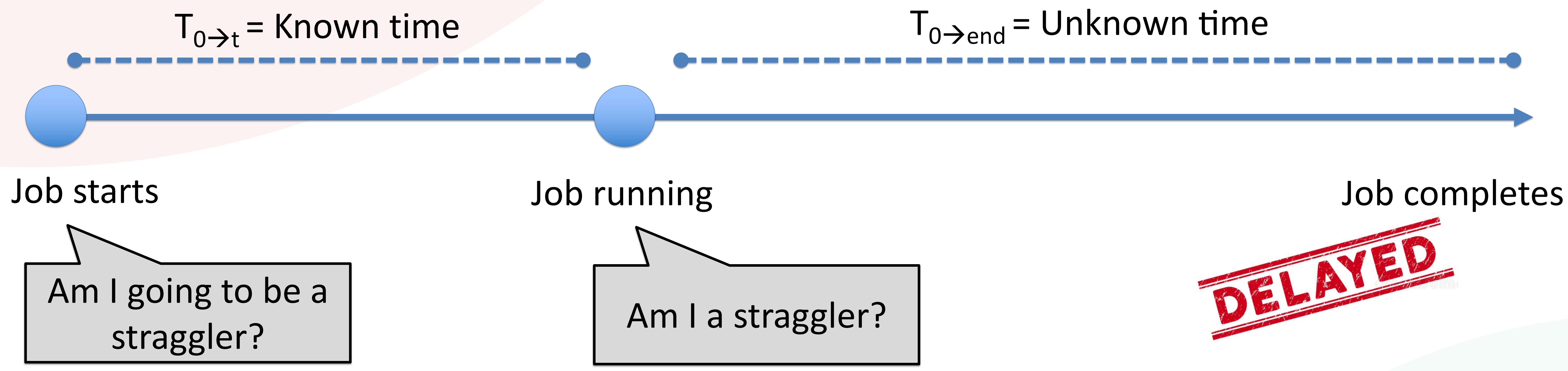
Yi Ding

Massachusetts Institute of Technology

NURD : Negative-Unlabeled Learning for Online Datacenter Straggler Prediction (MLSys'22)

Motivation and Problem Statement:

- A small amount (e.g., 1%) of *stragglers* (i.e., extremely long-latency tasks) account for a disproportionate amount of time (e.g., 10%) spent within a job in datacenters.
- Existing ML methods require complete training labels, or strong assumptions about the underlying latency distributions, which are hard to obtain for online running tasks.
- How to predict stragglers early and accurately within a running job?



$T_{0\to t}$ = Known time

$T_{0\to end}$ = Unknown time

Job starts

Job running

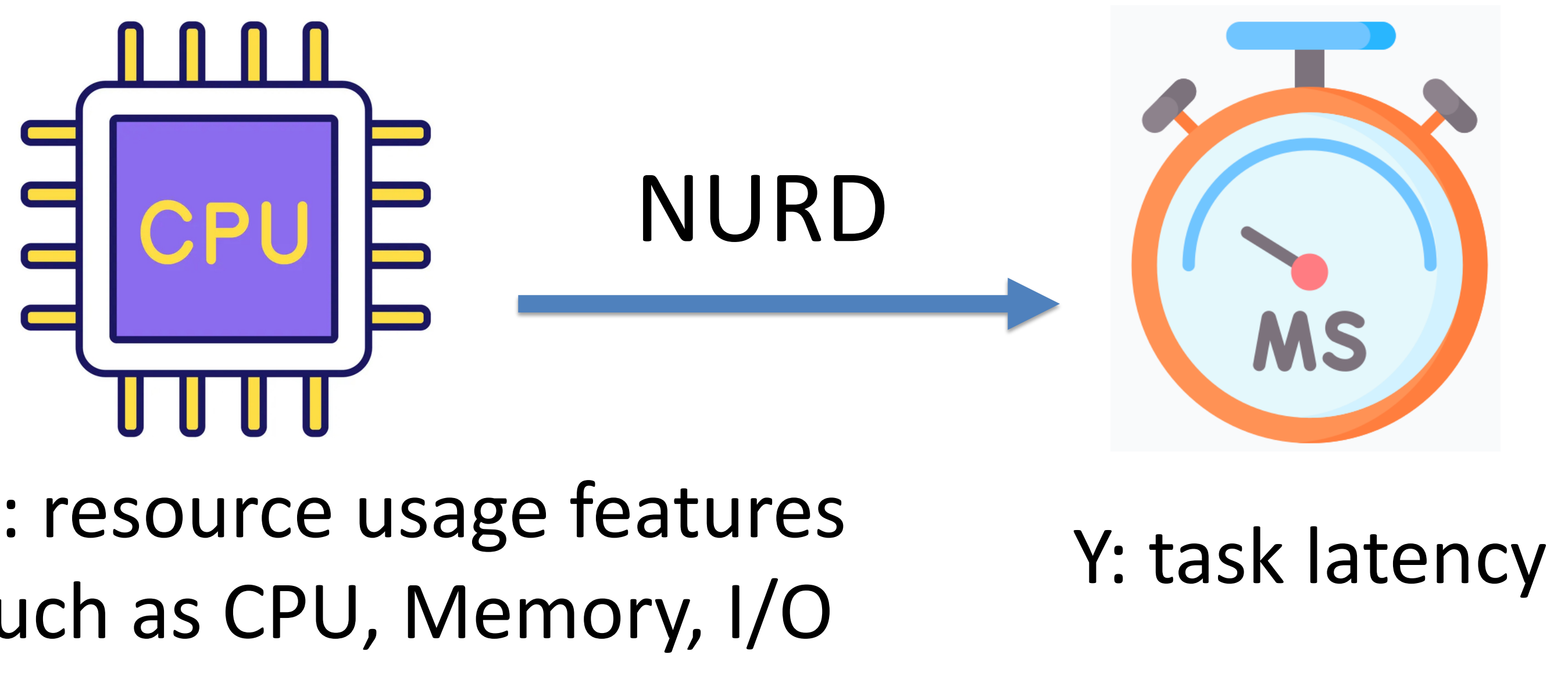
Job completes

Am I going to be a straggler?

Am I a straggler?

DELAYED

Proposed Approach: NURD



X: resource usage features such as CPU, Memory, I/O

Y: task latency

- Train with finished tasks.

$$\hat{y}_{ti} = h_t(x_{ti})$$

- Reweight based on feature space.

$$z_{ti} = \mathbb{P}(y_i \leq \tau_t^{\text{run}} | x_{ti}) \qquad w_{ti} = \max(\epsilon, \min(z_{ti} + \delta, 1))$$

$$\hat{y}_{ti}^{\text{adj}} = \frac{\hat{y}_{ti}}{w_{ti}}$$

- Update models online.

Experimental Methodology:

- Datacenter trace datasets from Google, Alibaba.
- Comparing to 1 supervised learning method, 14 outlier detection methods, 2 positive-labeled methods, 3 censored regression methods, and a system approach Wrangler (SOCC'14).

Experimental Results:

- Improved prediction accuracy: 2–11% ⬆ F1.
- Improved job completion time: 4.7–8.8% ⬇.

CRA Computing Research Association

CCC Computing Community Consortium Catalyst

NSF