Online Load Balancing via Learned Weights

Silvio Lattanzi, **Thomas Lavastida**,
Benjamin Moseley, Sergei Vassilvitskii

October 22, 2019

Carnegie Mellon University
Tepper School of Business

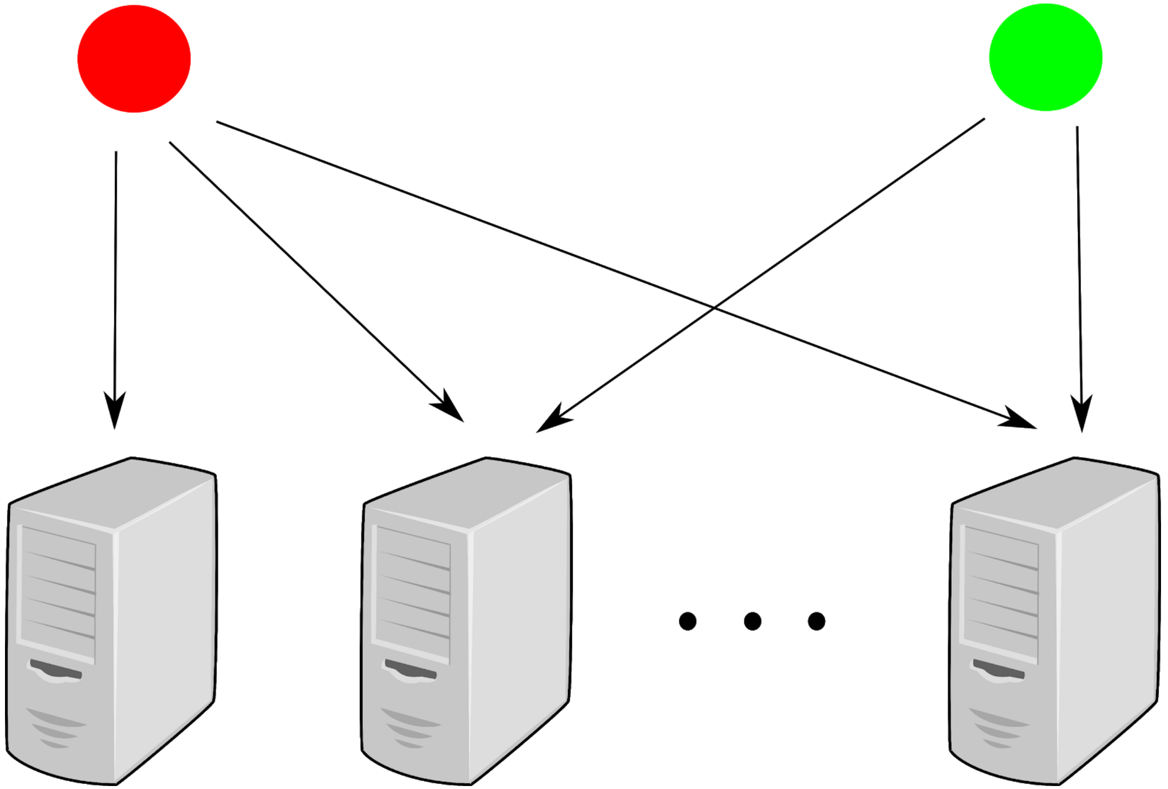
Google

Data Center Scheduling

- Jobs arrive online
- Heterogeneous machines and jobs w/ constraints
- Minimize maximum load
- Example: allocating VM's to physical machines in AWS

Online LB w/ Restricted Assignments

- m machines
- n jobs arrive in online list
  - $N(j) =$ subset of feasible machines for job j
  - $p_j =$ size of job j
- Machine load: total size of jobs assigned to a machine
- Goal: minimize makespan



Worst Case Analysis

- Online algorithm c-competitive if for all inputs

$$ALG \leq c \cdot OPT$$

- Every algorithm $\Omega(\log m)$-competitive
- Greedy algorithm $O(\log m)$-competitive
  - [Azar, Naor, Rom 1995]
- Worst case examples pathological

Learning Augmented Algorithms

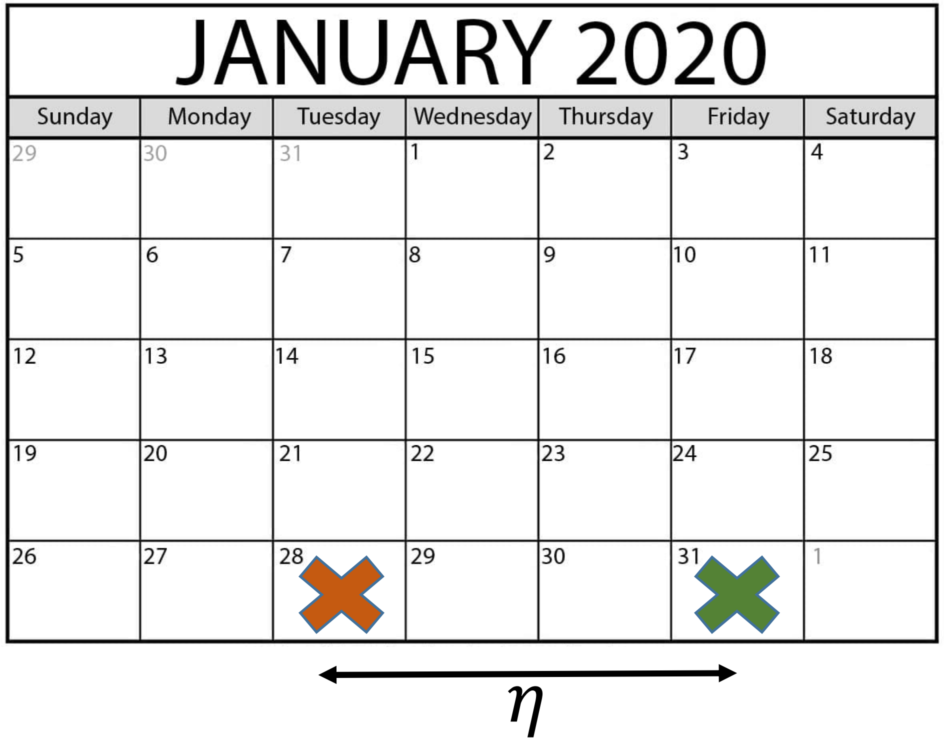
- Access to many traces of past jobs
- Learnable patterns may occur in practice
- Can ML be used to augment the design of online algorithms?
- Prediction about online instance
  - What to predict?
  - Handle errors?

Learning Augmented Algorithms

- Caching Problem [Lykouris and Vassilvitskii 2018]
- Ski Rental [Purohit et al. 2018]
- Non-Preemptive Scheduling [Purohit et al. 2018]
- Heavy Hitters Sketches [Hsu et al 2019]
- Improved Bloom Filters [Mitzenmacher 2018]
- Learned Index Structures [Kraska et al 2018]

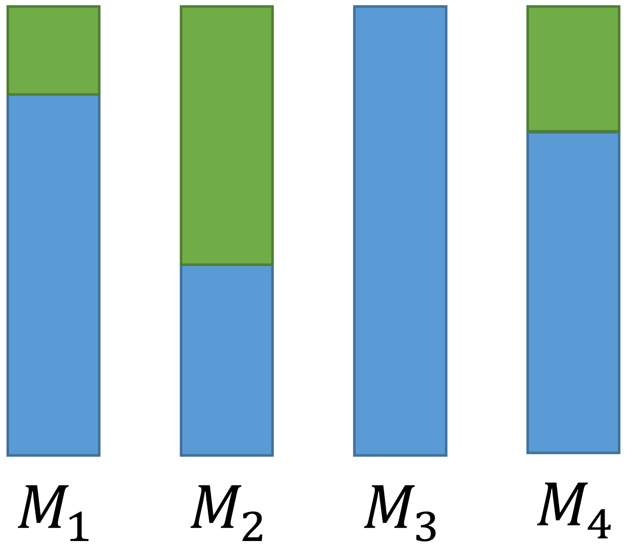
Online Algorithms + Predictions

- Ski Rental problem
  - Predict length of trip
- η := prediction error in hindsight
- Competitive ratio $= f(\eta)$
- Beat worst case for small η?
- Retain worst case for large η



What to Predict?

- Load of machines in OPT?
  - Pad the instance
- Dual variables?
  - Too sensitive to small errors
- Distribution over job subsets?
  - Potentially too many!
- Our approach:
  - compactly represent fractional solutions
  - Online rounding algorithm to get assignment



Results on Predictions

Theorem 1 – Machine Weights

Let $T =$ optimal makespan. For any $\epsilon > 0$ and any restricted assignment instance there exists weights $w \in R_+^m$ and a fractional assignment rule with fractional makespan at most $(1+\epsilon)T$

Given predictions w' of weights, there exists online algorithm yielding fractional assignment with fractional makespan at most $O(\log(\eta)\, T)$, $\eta := \max_i \frac{w_i'}{w_i}$ is relative error

Machine Weights

- Associate weight w_i to each machine
- Fractional Assignment:

$$x_{ij}(w) = \frac{w_i}{\sum_{i' \in N(j)} w_{i'}}$$

- Weights should satisfy

$$\sum_j p_j x_{ij}(w) \leq (1+\epsilon)T$$

- Idea builds off of [Agrawal et al. 2018]

Online Rounding Problem

- Receive j's size, neighborhood, fractional assignment online
$$\{x_{ij}\}_{i \in N(j)}\ s.t. \sum_{i \in N(j)} x_{ij} = 1$$
- Use x_{ij}'s to compute integral assignment online
- Rounding algorithm c-competitive if
$$ALG \leq c \cdot T'$$
- $T' \coloneqq \max\{\max_i \sum_j p_j x_{ij}, \max_j p_j\}$

Results on Rounding

Theorem 2 – Online Rounding

There exists a $O((\log\log m)^3)$-competitive randomized online rounding algorithm for restricted assignment and succeeds with high probability.

Theorem 3 – Lower Bounds

Every deterministic online rounding algorithms is $\Omega(\frac{\log m}{\log\log m})$-competitive and every randomized online rounding algorithm is $\Omega(\frac{\log\log m}{\log\log\log m})$-competitve

Conclusions

- Theorems 1 and 2 imply $O((\log \log m)^3 \log \eta)$-competitive algorithm with predictions
- Moderately accurate predictions go beyond worst case
- Connect prediction error to competitiveness

Questions?