UNIVERSITY OF CAMBRIDGE

Pronto: Federated Asynchronous Learning for Data Center Scheduling

Andreas Grammenos
The Alan Turing Institute, University of Cambridge

Evangelia Kalyvianaki
University of Cambridge

Peter Pietzuch
Imperial College London

Data Centers

Data centers are the Cloud backbone infrastructure

- large group of networked commodity hardware servers
- heterogeneous hardware, TPUs, GPUs, virtualization, containers
- cheaper at scale, electricity, hardware, network, and operations

Workloads are diverse, demanding, upredictable

- cover a range of applications
- batch processing, real-time, AI/ML, etc

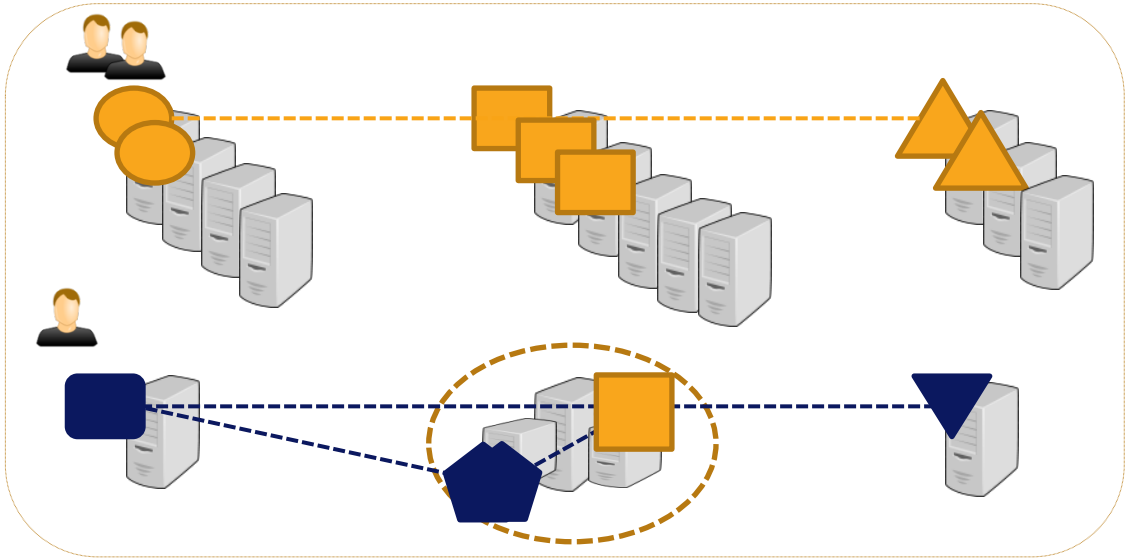
☛ How to **efficiently** use data center resources and satisfy QoS

Resource Management

data center
CPU, memory, network b/w, disk space

workloads
time-varying, unknown

over-provisioning
loss of revenue



accurate utilization prediction

colocation, time-sharing

virtual machines, containers

prediction+colocation hard in-practice

CPU Resource Management for VMs

100%

a VM's resource entitlement can be changed at run-time

used(40%) free(60%) used(60%) free(40%)

☛ How to accurately predict whether there is enough CPU for a VM to accept more incoming requests?

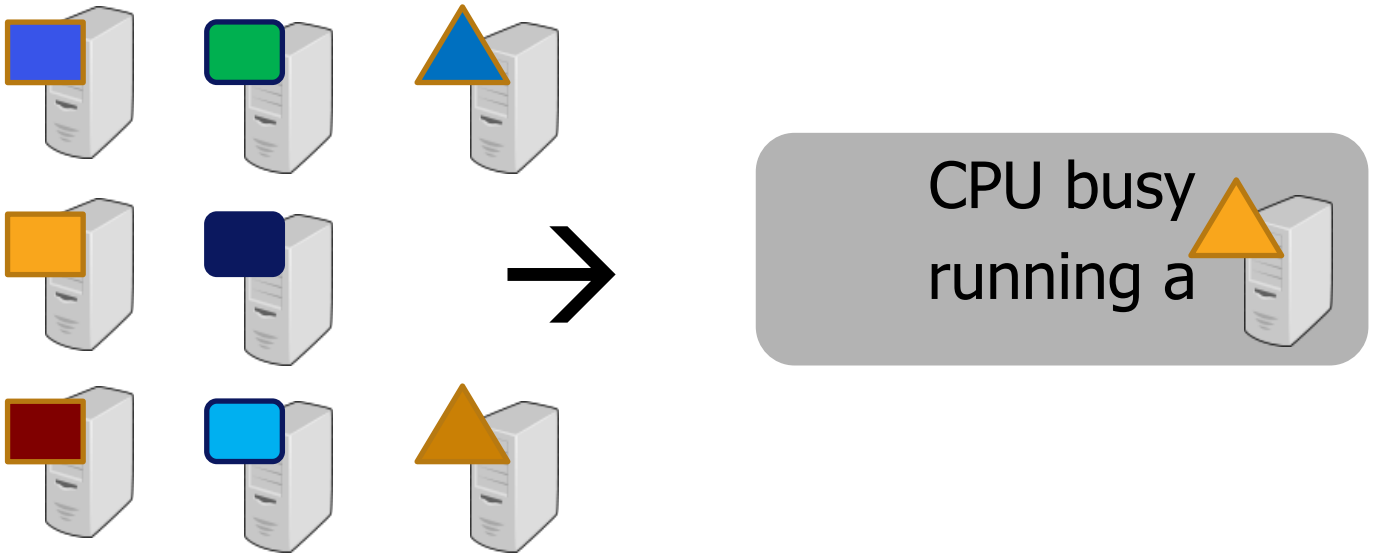
VM `CPU utilization` metric

free(CPU) = 100% - (*current*(CPU) → *predict*(CPU))

related work is rich but still an open problem

VM CPU Ready

% of time a VM is ready to run but is not scheduled in a CPU

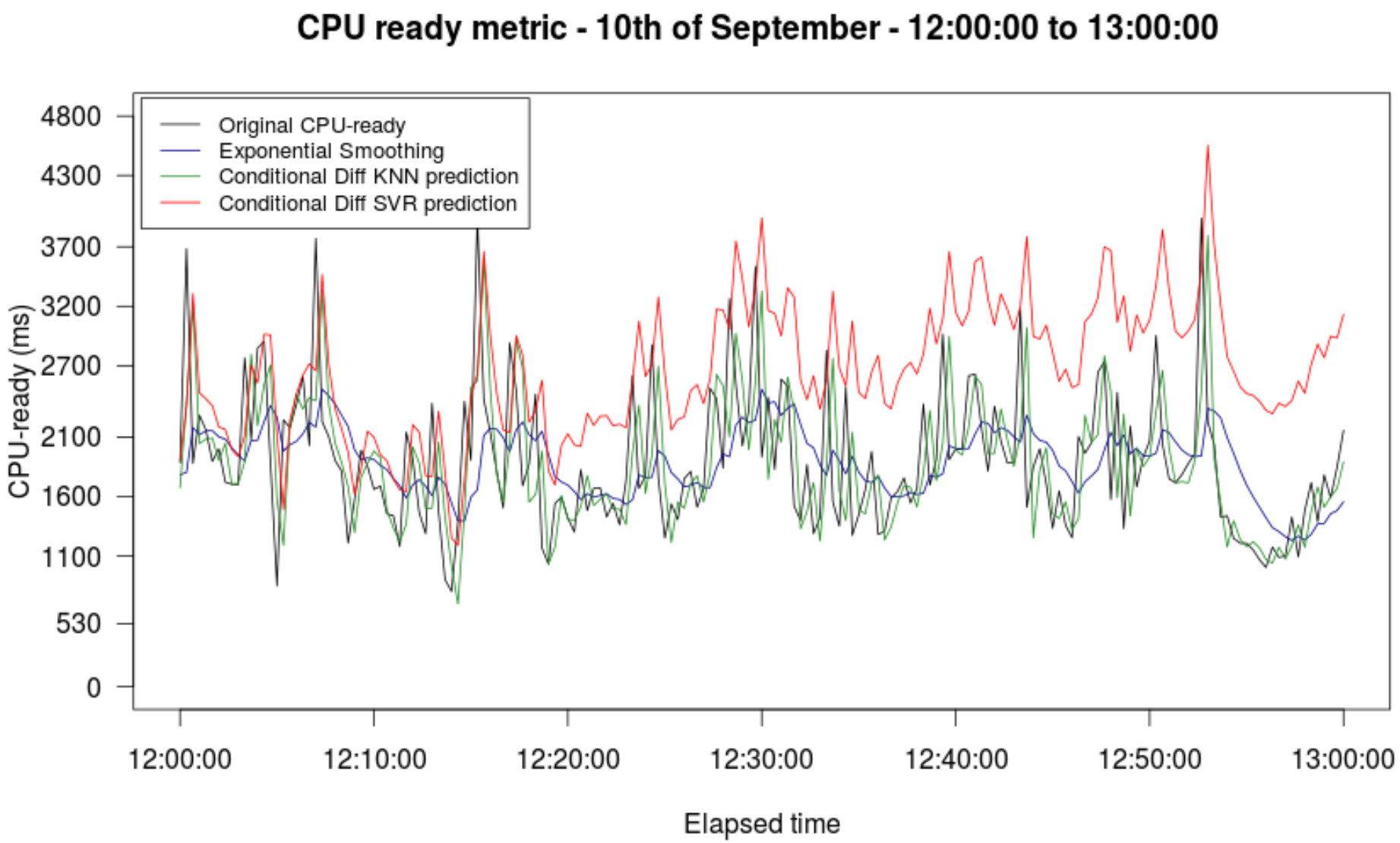


↑ CPU Ready → ↑ VM is waiting → its workload is not running

rule-of-thumb

CPU Ready < threshold

CPU Ready Prediction



`CPU Ready` Prediction using Past Values

Three approaches for prediction:

1. `CPU Ready` values → flag values > threshold
2. `CPU Ready` spikes → flag all spikes
3. **Pronto approach using Federated-PCA**

Dataset of a Company running VMware ESX:

- 100 clusters
- each cluster has ~14 ESX hosts and 250-350 virtual machines
- data related to CPU, memory, disc, and network utilization for hosts/VMs
- 134 metrics/host, 52 metrics/VM, new data every 20 secs
- 1TB of total dataset size

`CPU Ready` Values Prediction Results

Q1: *How many other VMs to look into?*

Options: *single VMs, same cluster VMs*

A1: Better results when using more VMs

Q2: *How much data from the past to look into?*

Options: *14 or 21 days*

A2: Better performance for longer past windows

Q3: *How far in advance can we predict?*

Options: *15', 30', 1, 3, 6, 12h, 1 day*

A3: Better results for long forecasting windows

Q4: *What is the best forecasting model to use?*

Options: *naïve, Exponential smoothing, ARIMA, SVM*

A4: SVM performs better than all other methods

none of the approaches can forecast `CPU Ready` values with high accuracy, RMSE

CPU Ready Spikes Prediction Results

a spike is a CPU Ready value > predefined threshold

CPU Ready values time-series → CPU Ready spikes time-series

Thresholds:

1. fixed numbers: 500, 800, and 1000
2. percentiles: 90th, 95th, and 99th
3. median
4. per-VM statistically based thresholds
5. statistical xbar threshold

Results:

1. The number of spikes depends on the method
2. Forecasting accuracy varies across methods: SVM gives best results
3. Highest accuracy for methods with few spikes

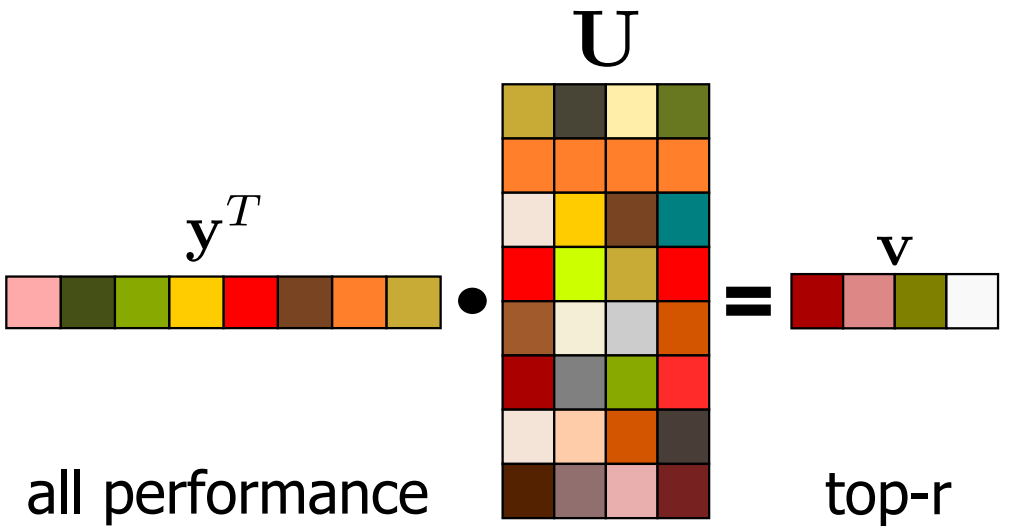
% of spikes detected is very low (<24%)

Thank you: Victoria Lopez Morales

Pronto: CPU Ready Prediction with Projections

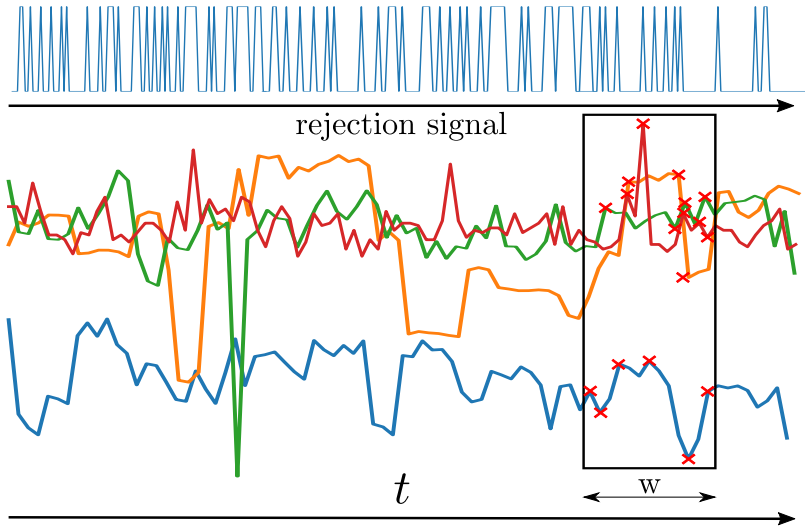
a spike in the top-r tracked projections
is indicative of incoming CPU Ready spikes

processing happens locally at each VM



all performance data from a VM

top-r projections



projections are tracked for abrupt changes over time

local estimate (updated globally) Federated-PCA[1,2]

Pronto Scheduler

Pronto accepts an incoming job at a single VM only if, by doing so, the performance of existing jobs(s) will not deteriorate

deterioration == one or more `CPU Ready` spikes

Reject-job algorithm @VM

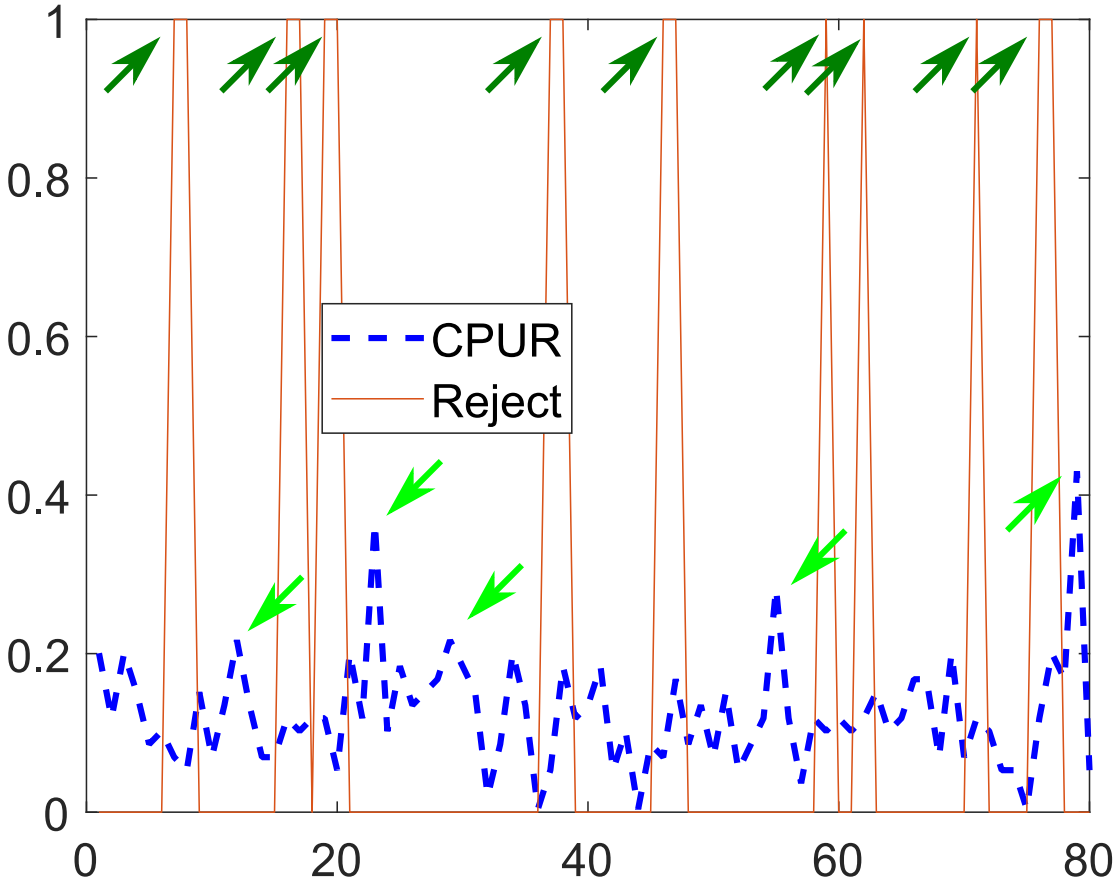
1. Compute projections
2. Find spikes for each projection
3. If one or more spikes are detected
   then: reject an incoming job
   else: accept an incoming job

--> Rejection Signal 1: reject 0: accept a job at *t*

Benefits:

- Use of unstructured data, unsupervised
- Based on existing work (Federated-PCA[1,2])
- Federated, scalable, online

Rejection and CPU Ready Signals



a raise in the Rejection Signal precedes a CPU Ready spike within *w*

Evaluation

High Level Goal:

To quantify the efficiency of Pronto to predict the `CPU Ready` spikes

Low Level Goal:

To predict at any *t* that a raise in the Rejection Signal happens shortly before or coincides with an observed `CPU Ready` spike in a window *w*

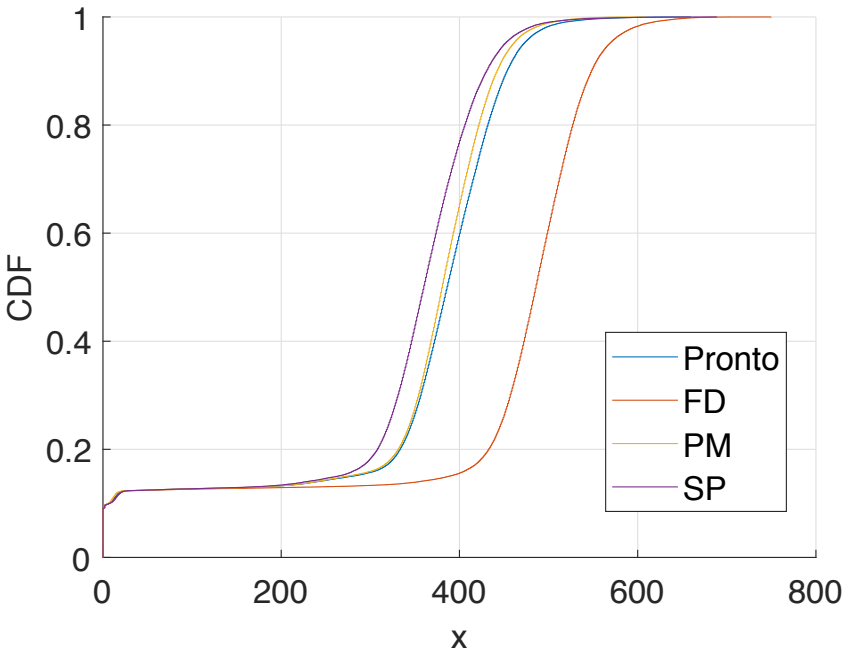
Logistics:

- Python simulation, Company's entire dataset, AMD TR1950, 64GB RAM
- window *w:* 10 timesteps
- rank *r* : 4
- baseline → the actual `CPU Ready` values
- comparison against state-of-the-art methods to generate **U**

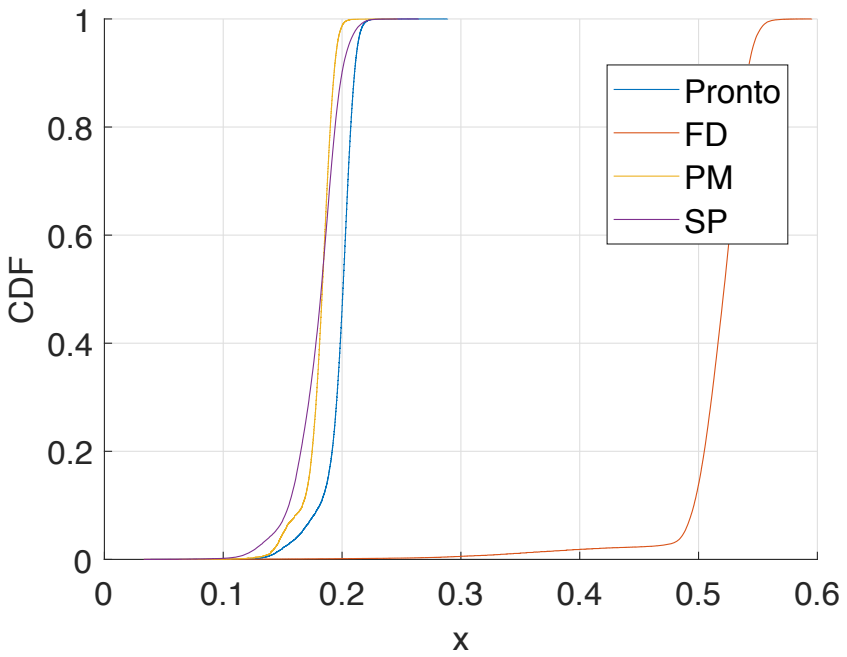
  SPIRIT (SP), Frequent Directions (FP), Power Method (PM)

Performance Overview

spikes detected



downtime



a peak is detected
→ a CPU spike is imminent

Pronto, PM, and SP provide more opportunities for accepting new jobs

Performance

	Execution time	Memory allocation
Pronto	15ms	148MB
PM	22ms	155MB
FD	25ms	151MB
SP	9ms	123MB

Pronto has one of the best performances and is federated

Related Work

1. **Microsoft's Hydra**

   a federated scheme, groups take independent scheduling decisions, groups are loosely coordinated for global policy management

2. **Microsoft's Resource Central**

   centrally gathers VM telemetry data for offline predictions for oversubscriptions

3. **VM Resource Control**

   VM resource management using feedback control (Kalman filters)

4. **Data center schedulers**

   Decima, Firmament, Omega, Tetrisched, Mesos, Kubernetes, Autopilot, Yarn, Sparrow

5. **Federated-PCA**

   [1] Grammenos, A., Mendoza-Smith, R., Mascolo, C., and Crowcroft, J. **Federated PCA with Adaptive Rank Estimation**. arXiv preprint arXiv:1907.08059v1, 2019.
   [2] Grammenos, A., Mendoza Smith, R., Crowcroft, J., and Mascolo, C. **Federated principal component analysis**. Advances in Neural Information Processing Systems, 33, 2020

Conclusions

"Pronto: Federated Task Scheduling" arXiv:2104.13429v1 **[cs.DC]**

- a federated scheme for near real-time identification of VM CPU saturation
- saturation is detected by `CPU Ready` spikes
- Pronto, an online, unsupervised accept/reject scheduler using unstructured data
- Pronto uses the local embedding @VM updated from other data
- simulation results show that Pronto correctly detects spikes and performs better than other methods

Future and Ongoing Work

- prototype evaluation on virtualized clusters
- combination of VM rejection signals for host planning
- pronto integration with existing capacity planning schedulers

Data center resource allocation using **Distributed Optimization**

1. Wei Jiang, Andreas Grammenos, Evangelia Kalyvianaki, and Themistoklis Charalambous. "**An Asynchronous Approximate Distributed Alternating Direction Method of Multipliers in Digraphs**". In Proc. of the IEEE Conference on Decision and Control (CDC), 2021.
2. Apostolos I. Rikos, Andreas Grammenos, Evangelia Kalyvianaki, Christoforos N. Hadjicostis, Themistoklis Charalambous, and Karl H. Johansson. "**Optimal CPU Scheduling in Data Centers via a Finite-Time Distributed Quantized Coordination Mechanism**". In Proc. of the IEEE Conference on Decision and Control (CDC), 2021.

Questions? Thank you!

ek264@cam.ac.uk https://www.cst.cam.ac.uk/people/ek264

andreas.grammenos@gmail.com