Prediction-Based Power Oversubscription in Cloud Platforms

Alok Kumbhare, Reza Azimi, Ioannis Manousakis, Anand Bonde, Felipe Frujeri, Nithish Mahalingam, **Pulkit A. Misra**, Seyyed Ahmed Javadi, Bianca Schroeder, Marcus Fontoura, Ricardo Bianchini

Microsoft Research

Azure

Motivation

Motivation

- Soaring demand for datacenter capacity
  - $200B+ spent worldwide on datacenter systems [Gartner'21]

Motivation

- Soaring demand for datacenter capacity
  - $200B+ spent worldwide on datacenter systems [Gartner'21]

- Efficient resource utilization is key
  - Lower costs and fewer datacenters to build
  - Better sustainability

Motivation

- Soaring demand for datacenter capacity
  - $200B+ spent worldwide on datacenter systems [Gartner'21]
- Efficient resource utilization is key
  - Lower costs and fewer datacenters to build
  - Better sustainability
- Power is typically a bottleneck resource
  - Massive underutilization due to provisioning peak power for each server

Prior Work: Power Capping and Oversubscription

Prior Work: Power Capping and Oversubscription

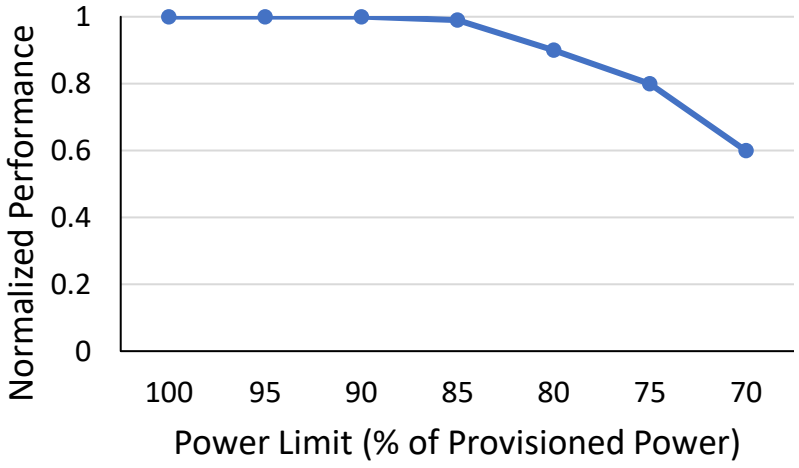
- Harvest unutilized power for adding more servers
  - Use power capping for safety

Prior Work: Power Capping and Oversubscription

- Harvest unutilized power for adding more servers
  - Use power capping for safety
- Hardware-based capping on servers
  - Throttle CPU (all cores) and memory to honor cap

Prior Work: Power Capping and Oversubscription

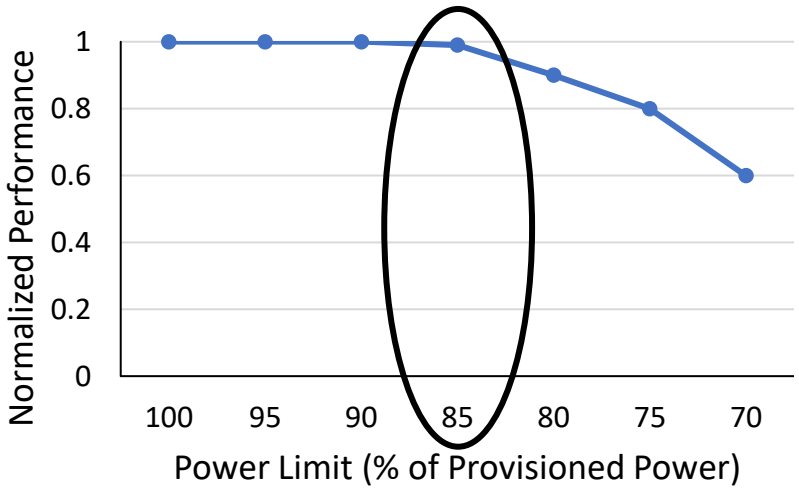
- Harvest unutilized power for adding more servers
  - Use power capping for safety
- Hardware-based capping on servers
  - Throttle CPU (all cores) and memory to honor cap
- Profile impact of capping on workloads
  - Oversubscribe power while protecting performance



Power capping impact on workload performance
(baseline: un-capped performance)

Prior Work: Power Capping and Oversubscription

- Harvest unutilized power for adding more servers
  - Use power capping for safety
- Hardware-based capping on servers
  - Throttle CPU (all cores) and memory to honor cap
- Profile impact of capping on workloads
  - Oversubscribe power while protecting performance



Power capping impact on workload performance
(baseline: un-capped performance)

Oversubscription Challenges for Cloud Providers

Oversubscription Challenges for Cloud Providers

1. Opaque workloads on Virtual Machines (VMs)
   - Which ones are critical (e.g., latency-sensitive or user-facing)?

Oversubscription Challenges for Cloud Providers

1. Opaque workloads on Virtual Machines (VMs)
   - Which ones are critical (e.g., latency-sensitive or user-facing)?

2. Dynamic system (VM arrivals/departures) prevents pre-defined grouping
   - Harvest power while protecting performance of critical workloads

Oversubscription Challenges for Cloud Providers

1. Opaque workloads on Virtual Machines (VMs)
   - Which ones are critical (e.g., latency-sensitive or user-facing)?

2. Dynamic system (VM arrivals/departures) prevents pre-defined grouping
   - Harvest power while protecting performance of critical workloads

3. Multiple VMs with differing performance requirements per server
   - Impact of full-server throttling on critical VMs?

Oversubscription Challenges for Cloud Providers

1. Opaque workloads on Virtual Machines (VMs)
   - Which ones are critical (e.g., latency-sensitive or user-facing)?

Oversubscription is currently limited by performance impact of capping

3. Multiple VMs with differing performance requirements per server
   - Impact of full-server throttling on critical VMs?

Fine-grained Power Capping for Oversubscription

Fine-grained Power Capping for Oversubscription

Insight #1: Not all VMs are performance-critical (e.g., non-production, batch)

- Predictions to identify performance criticality of opaque VMs

Fine-grained Power Capping for Oversubscription

Insight #1: Not all VMs are performance-critical (e.g., non-production, batch)

- Predictions to identify performance criticality of opaque VMs

Insight #2: Per-core dynamic voltage and frequency scaling (DVFS) for throttling VMs

Fine-grained Power Capping for Oversubscription

Insight #1: Not all VMs are performance-critical (e.g., non-production, batch)

- Predictions to identify performance criticality of opaque VMs

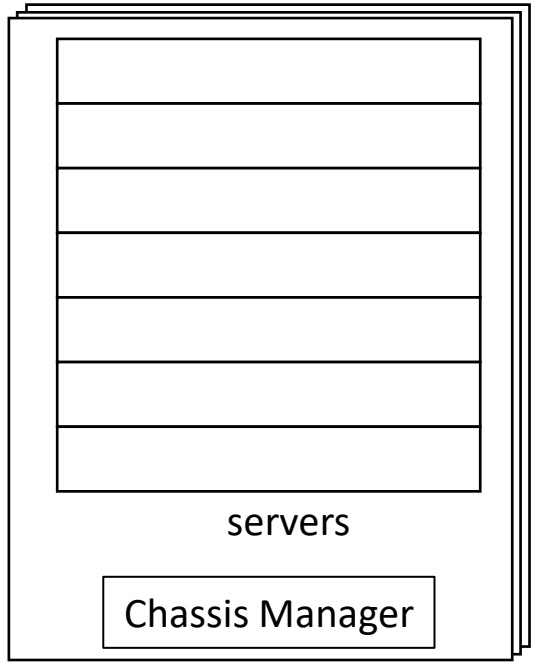
Insight #2: Per-core dynamic voltage and frequency scaling (DVFS) for throttling VMs

Solution: Criticality-aware per-VM power capping and oversubscription

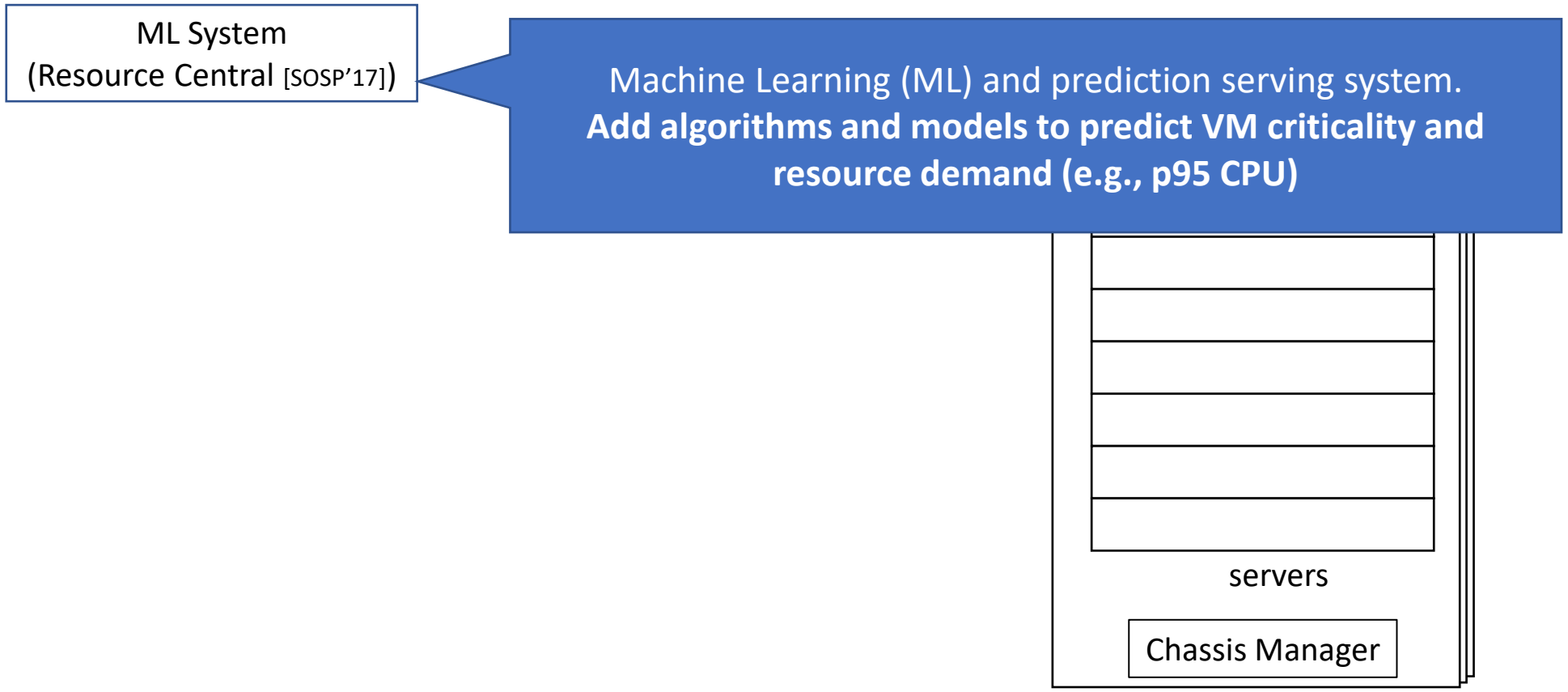
- Provide power safety while protecting performance of critical VMs
- Strategy for criticality-aware oversubscription

Per-VM Power Capping Overview

Per-VM Power Capping Overview



Per-VM Power Capping Overview

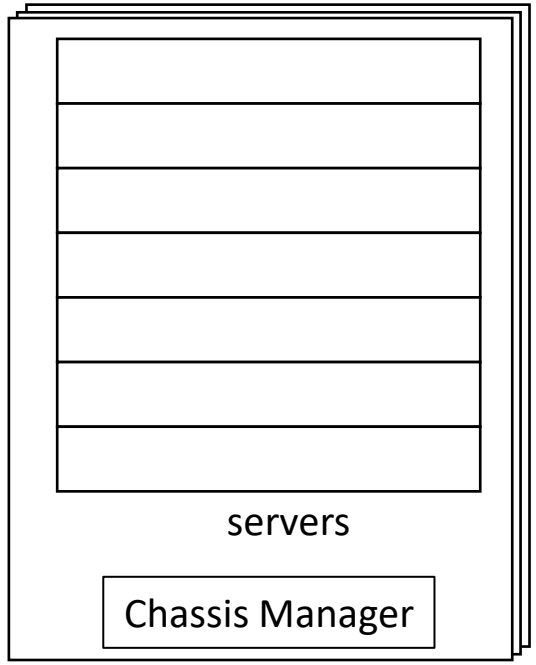


Per-VM Power Capping Overview

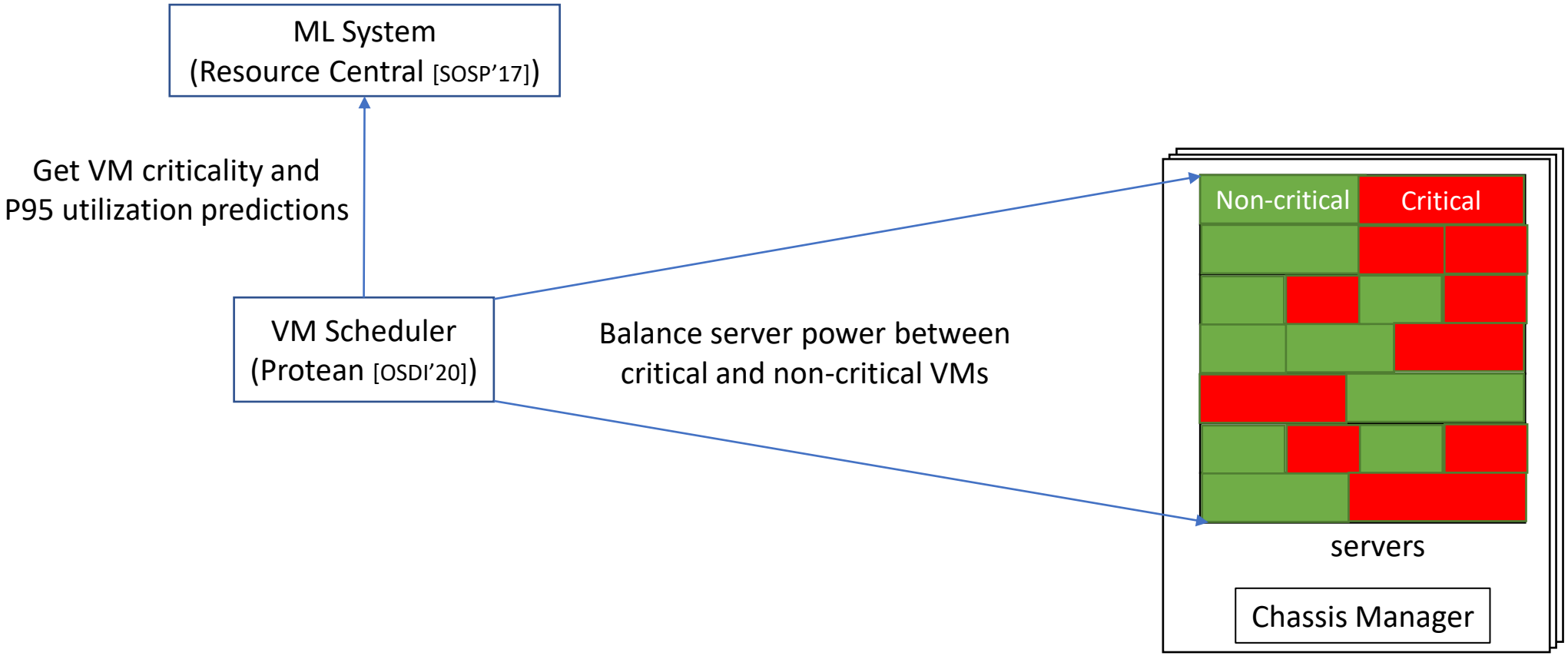
ML System
(Resource Central [SOSP'17])

VM Scheduler
(Protean [OSDI'20])

VM placement with rules to tightly pack VMs on servers.
Add rules for distributing power via criticality and utilization-aware VM placement.



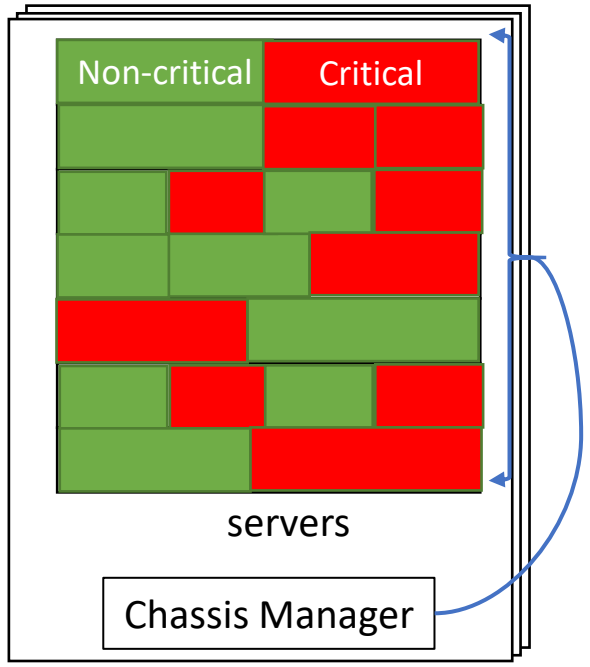
Per-VM Power Capping Overview



Per-VM Power Capping Overview

ML System
(Resource Central [SOSP'17])

VM Scheduler
(Protean [OSDI'20])



Chassis power draw > limit

- Start power capping on servers

Per-VM Power Capping Overview

ML System
(Resource Central [SOSP'17])

VM Scheduler
(Protean [OSDI'20])

Criticality-aware capping for safety while protecting perf of critical VMs

Non-critical
VM

Critical
VM

Per-VM capping controller

Hypervisor

C C C C C C

server

Non-critical
Critical

servers

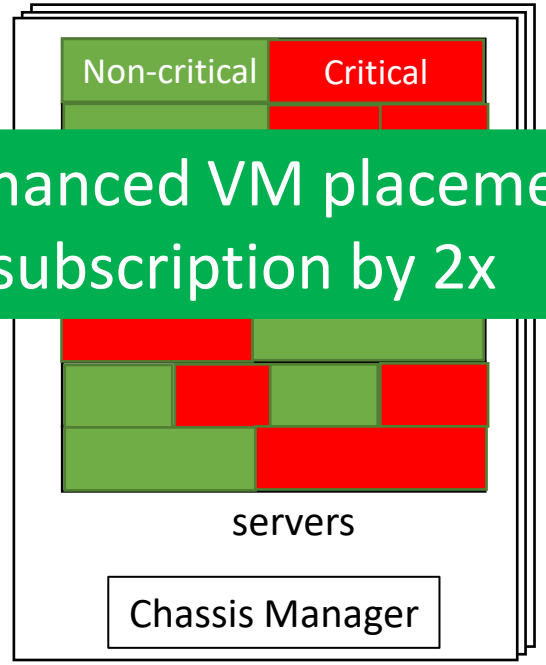
Chassis Manager

Chassis power draw > limit

- Start power capping on servers

Per-VM Power Capping Overview

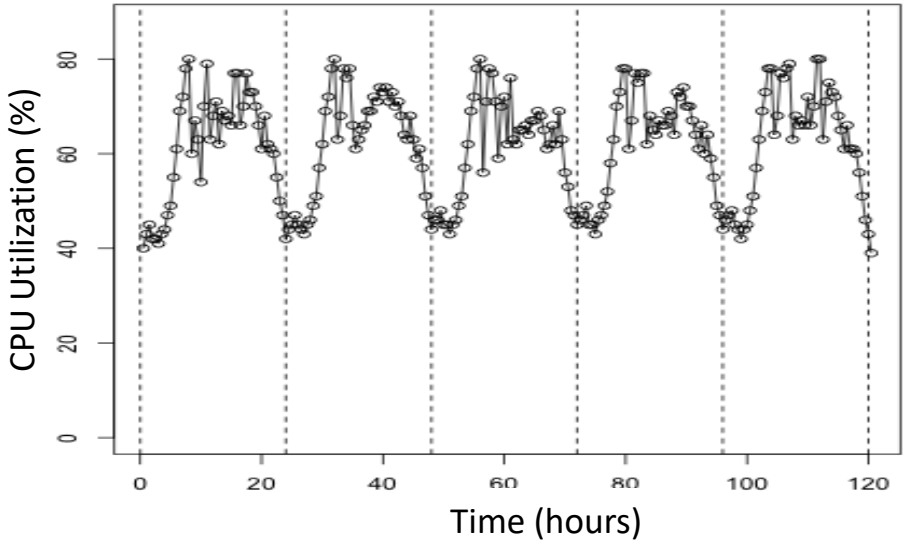
ML System
(Resource Central [SOSP'17])



Per-VM power management using ML models, enhanced VM placement and per-VM power capping can increase oversubscription by 2x

Inferring criticality of opaque VMs (challenge #1)

- Insight: User-facing workloads exhibit diurnal load pattern

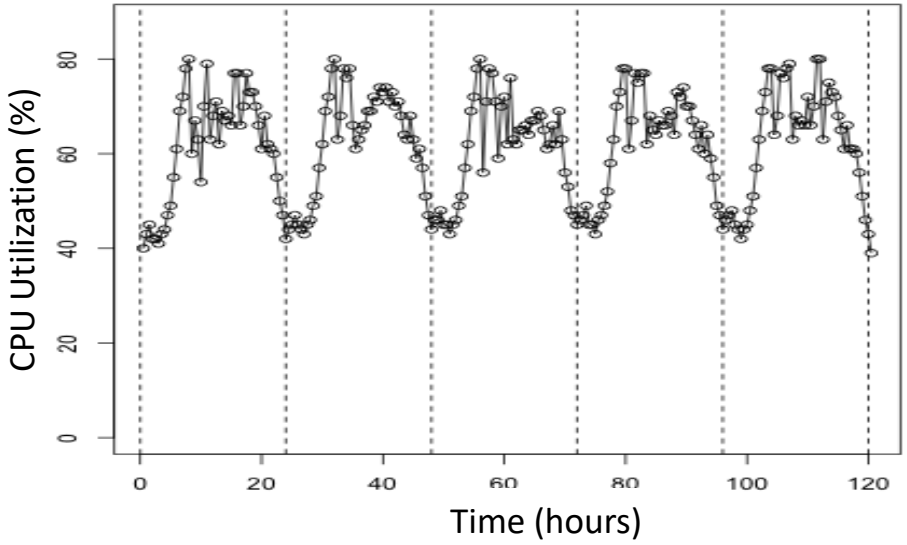


CPU utilization pattern of a workload

Inferring criticality of opaque VMs (challenge #1)

- Insight: User-facing workloads exhibit diurnal load pattern

- Algorithm to identify periodicity in CPU utilization



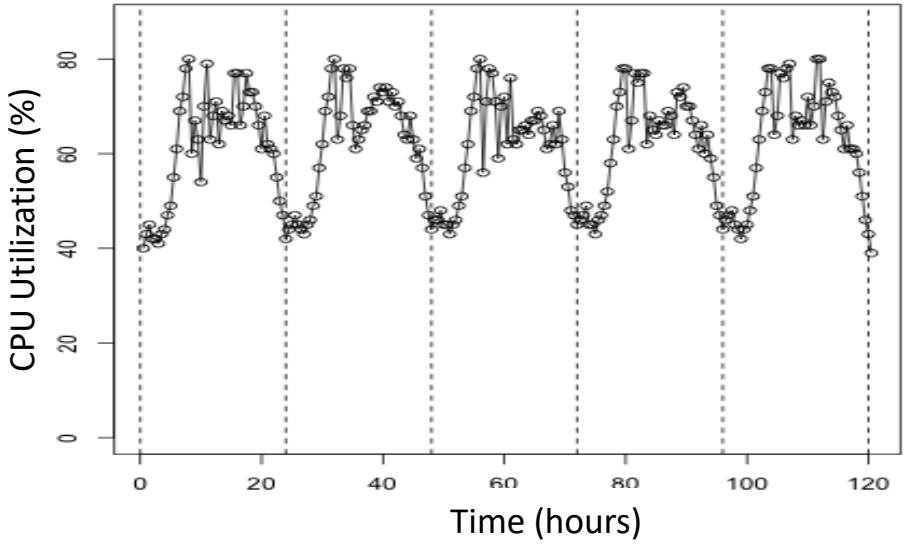
CPU utilization pattern of a workload

Inferring criticality of opaque VMs (challenge #1)

- Insight: User-facing workloads exhibit diurnal load pattern

- Algorithm to identify periodicity in CPU utilization

- ML model to predict VM criticality for placement
  - Algorithm provides training labels
  - 99% precision and recall for user-facing workloads



CPU utilization pattern of a workload

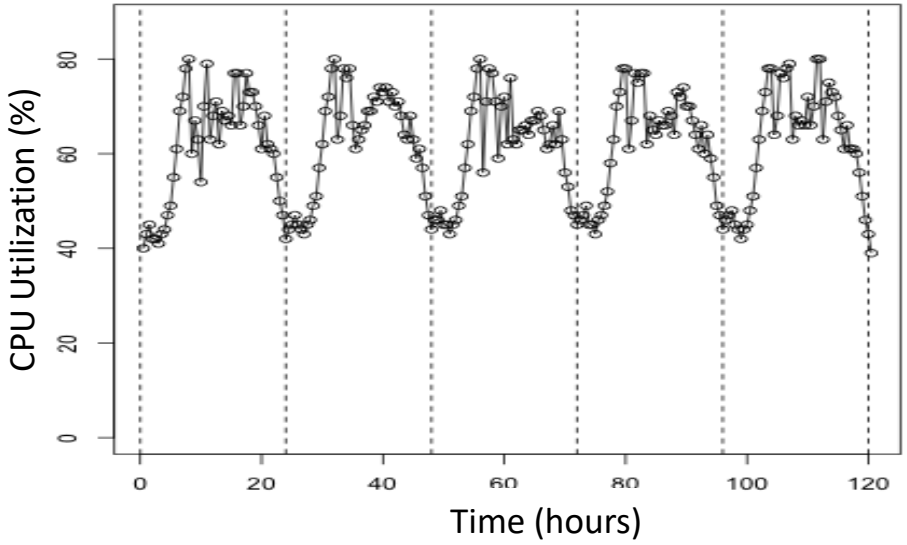
Inferring criticality of opaque VMs (challenge #1)

- Insight: User-facing workloads exhibit diurnal load pattern

- Algorithm to identify periodicity in CPU utilization

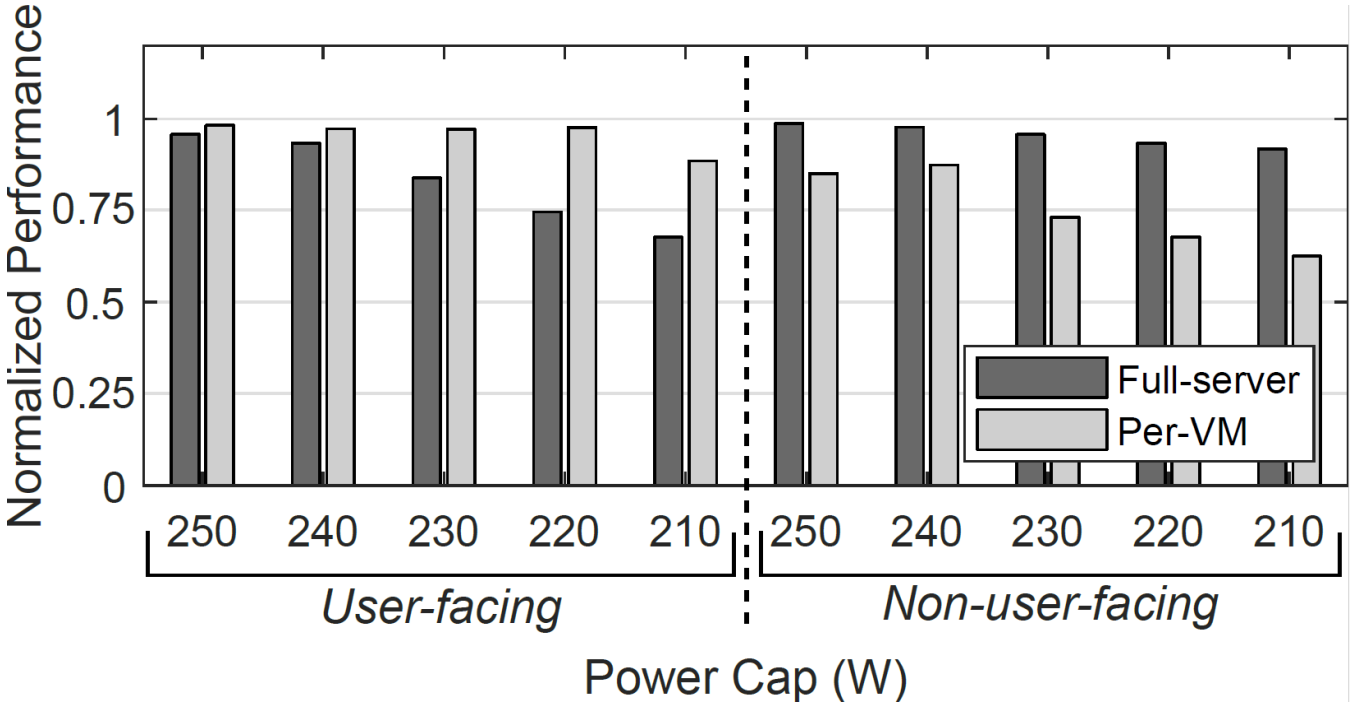
- ML model to predict VM criticality for placement
  - Algorithm provides training labels
  - 99% precision and recall for user-facing workloads

- Static overrides
  - “Always-throttle” list of VMs (e.g., internal non-production)
  - “Do-not-throttle” list of VMs (e.g., all third-party, gaming)

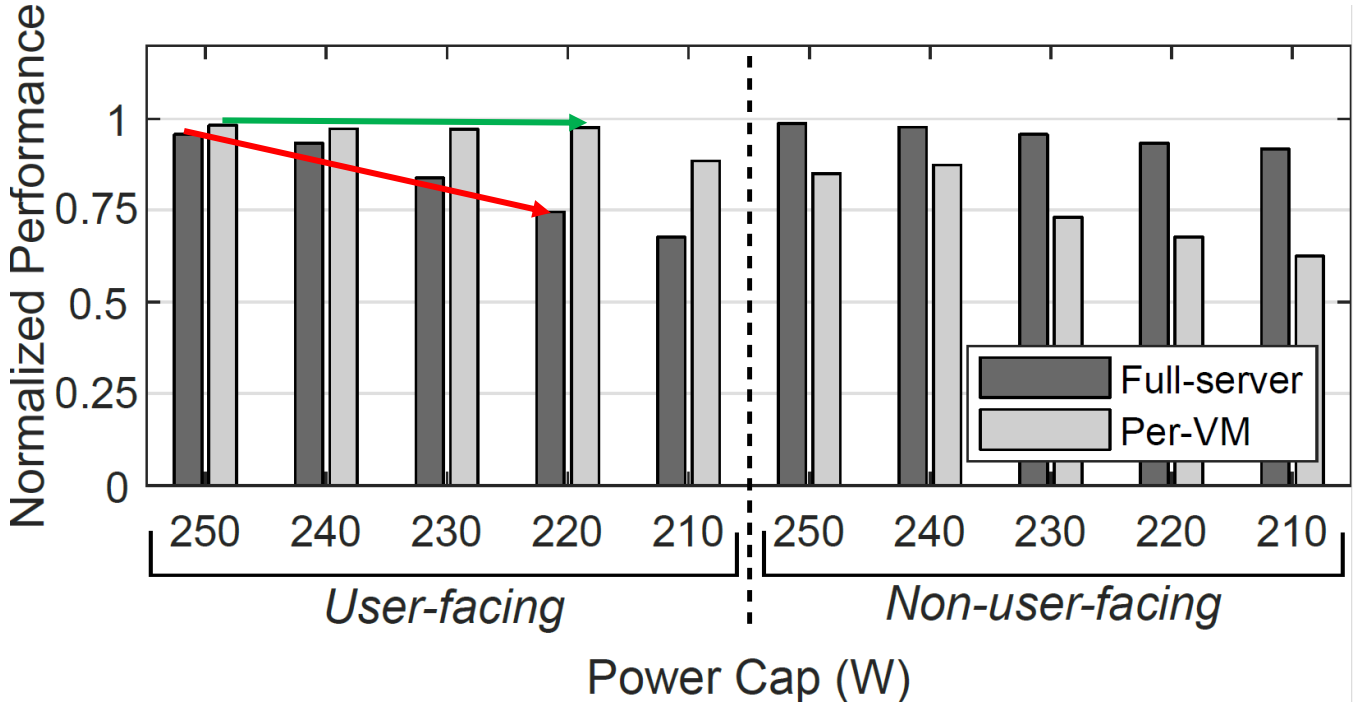


CPU utilization pattern of a workload

Full-server throttling (challenge #3)

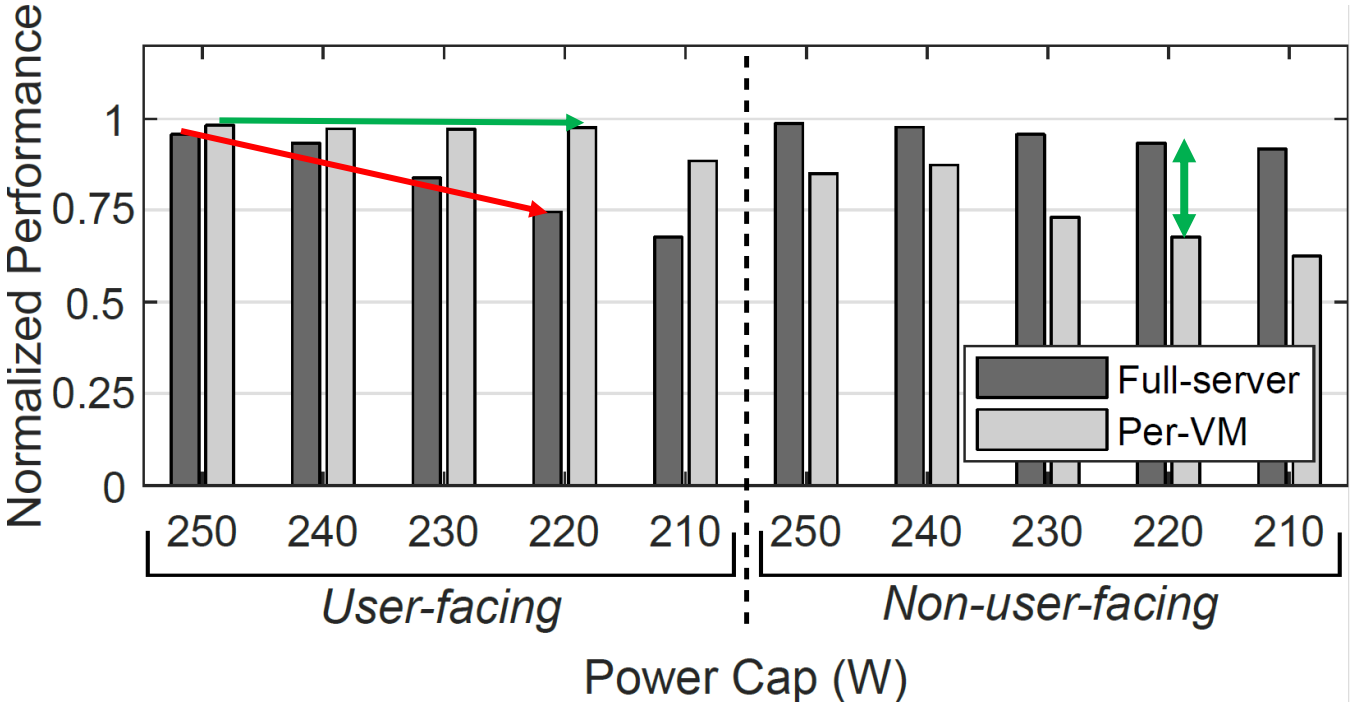


Full-server throttling (challenge #3)



- Per-VM enables additional harvesting while protecting perf of critical VMs

Full-server throttling (challenge #3)



- Per-VM enables additional harvesting while protecting perf of critical VMs
  - Trade-off: Increased perf degradation for non-critical VMs
  - Relaxed perf requirement of workloads on non-critical VMs (e.g., internal non-production)

Oversubscription Strategy with Per-VM Capping

- Insight: Differentiated (per-VM) capping for harvesting power from chassis
  - **Constraints**: # capping events and perf (frequency) reduction for critical and non-critical VMs

Oversubscription Strategy with Per-VM Capping

- Insight: Differentiated (per-VM) capping for harvesting power from chassis
  - **Constraints**: # capping events and perf (frequency) reduction for critical and non-critical VMs

- Use historical draws to calculate harvesting opportunity with per-VM capping

Oversubscription Strategy with Per-VM Capping

- Insight: Differentiated (per-VM) capping for harvesting power from chassis
  - **Constraints**: # capping events and perf (frequency) reduction for critical and non-critical VMs

- Use historical draws to calculate harvesting opportunity with per-VM capping

Approach	Harvested power (%)	Savings ($10/W)
Traditional (no oversubscription)	0	0
State-of-the-art (w/ full-server capping)	6.2%	$79.4M
Predictions for internal and non-premium external VMs	12.1%	$154.9M

Oversubscription potential with per-VM capping

Oversubscription Strategy with Per-VM Capping

- Insight: Differentiated (per-VM) capping for harvesting power from chassis
  - **Constraints**: # capping events and perf (frequency) reduction for critical and non-critical VMs

- Use historical draws to calculate harvesting opportunity with per-VM capping

Approach	Harvested power (%)	Savings ($10/W)
Traditional (no oversubscription)	0	0
State-of-the-art (w/ full-server capping)	6.2%	$79.4M
Predictions for internal and non-premium external VMs	12.1%	$154.9M

Oversubscription potential with per-VM capping

Oversubscription Strategy with Per-VM Capping

- Insight: Differentiated (per-VM) capping for harvesting power from chassis
  - **Constraints**: # capping events and perf (frequency) reduction for critical and non-critical VMs

- Use historical draws to calculate harvesting opportunity with per-VM capping

Approach	Harvested power (%)	Savings ($10/W)
Traditional (no oversubscription)	0	0
State-of-the-art (w/ full-server capping)	6.2%	$79.4M
Predictions for internal and non-premium external VMs	12.1%	$154.9M

Oversubscription potential with per-VM capping

Oversubscription Strategy with Per-VM Capping

- Insight: Differentiated (per-VM) capping for harvesting power from chassis
  - **Constraints**: # capping events and perf (frequency) reduction for critical and non-critical VMs

- Use historical draws to calculate harvesting opportunity with per-VM capping

Approach	Harvested power (%)	Savings ($10/W)
Traditional (no oversubscription)	0	0
State-of-the-art (w/ full-server capping)	6.2%	$79.4M
Predictions for internal and non-premium external VMs	12.1%	$154.9M

Oversubscription potential with per-VM capping

Oversubscription Strategy with Per-VM Capping

- Insight: Differentiated (per-VM) capping for harvesting power from chassis
  - **Constraints**: # capping events and perf (frequency) reduction for critical and non-critical VMs

Per-VM capping allow us to be selective
and increase the amount of **oversubscription by 2x!**

Approach	Harvested power (%)	Savings ($10/W)
Traditional (no oversubscription)	0	0
State-of-the-art (w/ full-server capping)	6.2%	$79.4M
Predictions for internal and non-premium external VMs	12.1%	$154.9M

Oversubscription potential with per-VM capping

Production Impact and Lessons

Production Impact and Lessons

- Per-VM capping system and ML models deployed in many Azure datacenters
  - Significantly reduce throttling of critical VMs (vs full-server throttling mechanisms)

Production Impact and Lessons

- Per-VM capping system and ML models deployed in many Azure datacenters
  - Significantly reduce throttling of critical VMs (vs full-server throttling mechanisms)
- Working on deploying VM placement policy to enable aggressive oversubscription

Production Impact and Lessons

- Per-VM capping system and ML models deployed in many Azure datacenters
  - Significantly reduce throttling of critical VMs (vs full-server throttling mechanisms)
- Working on deploying VM placement policy to enable aggressive oversubscription
- Lessons (more in the paper)

Production Impact and Lessons

- Per-VM capping system and ML models deployed in many Azure datacenters
  - Significantly reduce throttling of critical VMs (vs full-server throttling mechanisms)
- Working on deploying VM placement policy to enable aggressive oversubscription
- Lessons (more in the paper)
  1. Refresh VM criticality prediction on servers

Production Impact and Lessons

- Per-VM capping system and ML models deployed in many Azure datacenters
  - Significantly reduce throttling of critical VMs (vs full-server throttling mechanisms)
- Working on deploying VM placement policy to enable aggressive oversubscription
- Lessons (more in the paper)
  1. Refresh VM criticality prediction on servers
  2. Increasing rack density (# of servers) with per-VM capping

Production Impact and Lessons

- Per-VM capping system and ML models deployed in many Azure datacenters
  - Significantly reduce throttling of critical VMs (vs full-server throttling mechanisms)
- Working on deploying VM placement policy to enable aggressive oversubscription
- Lessons (more in the paper)
  1. Refresh VM criticality prediction on servers
  2. Increasing rack density (# of servers) with per-VM capping
  3. Server support for per-VM capping

Conclusions

- Limited power oversubscription on cloud platforms to restrict performance impact
- Solution: Prediction-based per-VM power capping
  - Algorithm and ML models for predicting performance criticality and VM utilization
  - Criticality- and utilization-aware VM placement
  - On-server criticality-aware power management system
  - Strategy for criticality-aware oversubscription
- Main result: Increase oversubscription by 2x while protecting critical workloads

Thank you!