Providing SLOs for Resource-Harvesting VMs in Cloud Platforms

Pradeep Ambati[*], Íñigo Goiri, Felipe Frujeri, Alper Gun[*], Ke Wang[*], Brian Dolan, Brian Corell, Sekhar Pasupuleti, Thomas Moscibroda, Sameh Elnikety, Marcus Fontoura, and Ricardo Bianchini

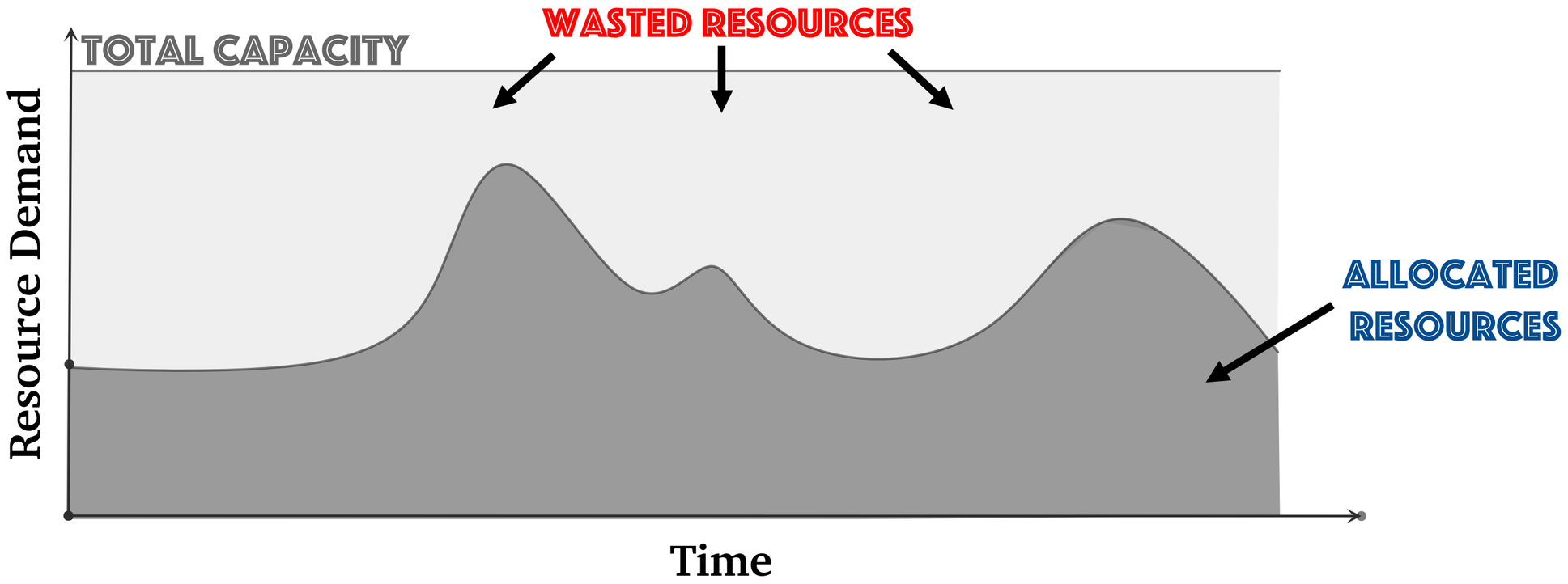
Microsoft Research

Microsoft Azure

*Ambati is with UMass Amherst, Gun and Wang are now with Google

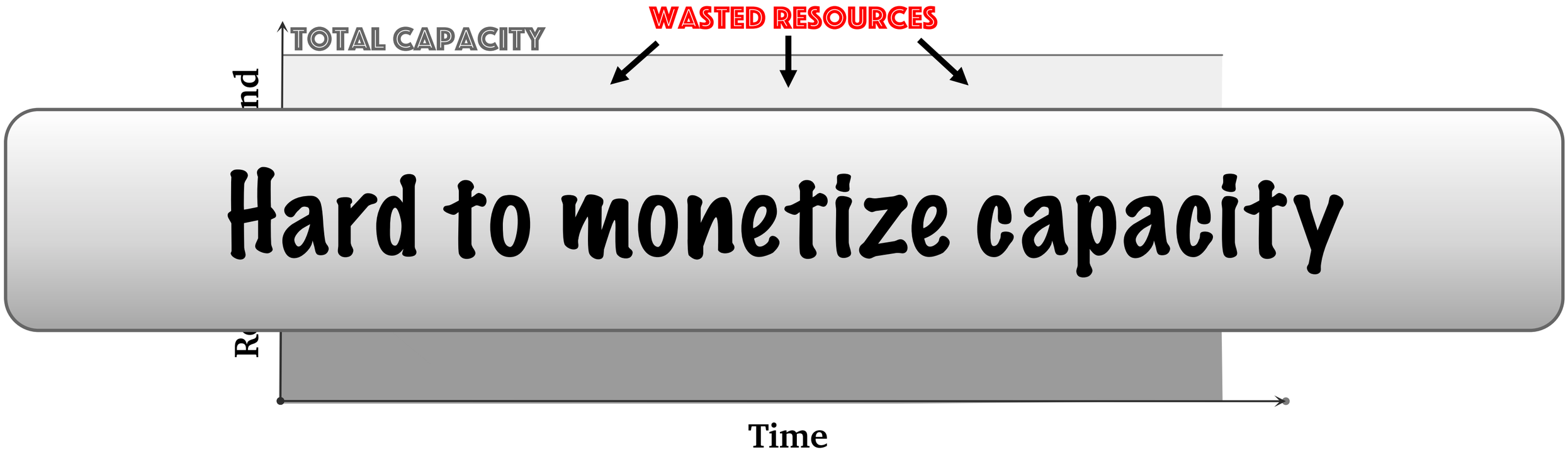
Public Cloud Platforms

- Cloud Platforms offer compute resources as **virtual machines (VM)**
  - Users can keep the VMs from seconds to years and request more VMs
  - Cloud platforms provide **illusion of infinite scalability**
    - To allow user growth, handle hardware failures etc.



Public Cloud Platforms

- Cloud Platforms offer compute resources as **virtual machines (VM)**
  - Users can keep the VMs from seconds to years and request more VMs
  - Cloud platforms provide **illusion of infinite scalability**
    - To allow user growth, handle hardware failures etc.

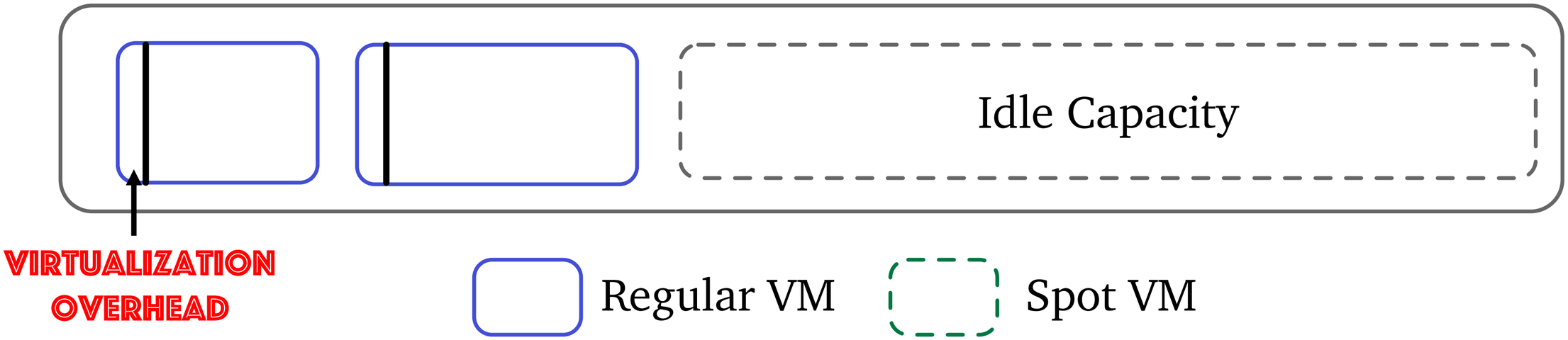


Hard to monetize capacity

Evictable (Spot) VMs

- **Unallocated resources** leveraged as spot VMs with **relaxed SLOs**
  - Spot VMs can be **revoked** anytime for regular-priority VMs
  - Cost ~**50-90% less** than regular-priority VMs

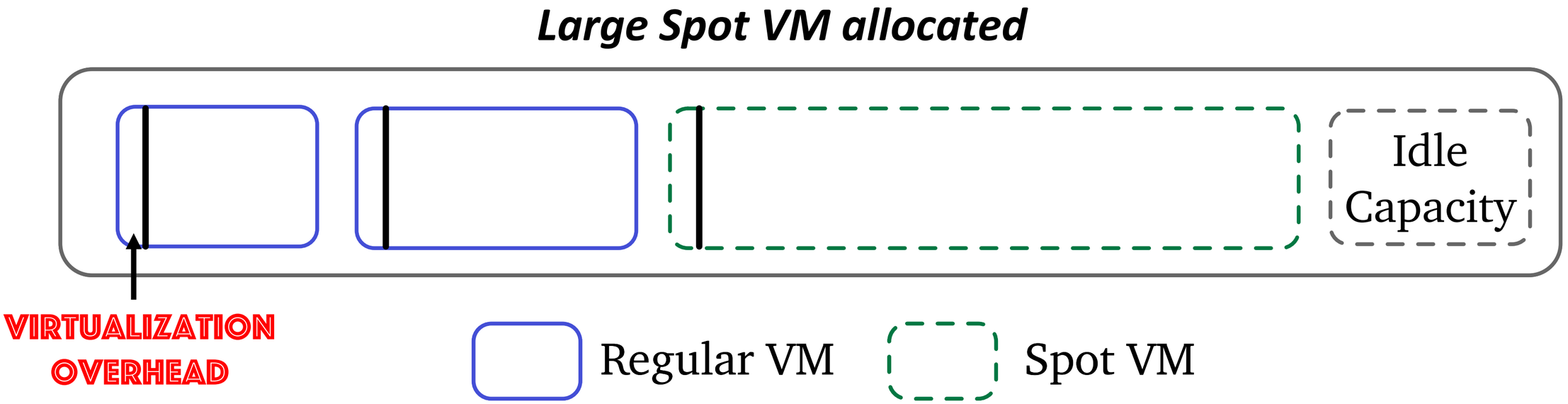
- But spot VMs are **fixed size** *VMs* and …



Evictable (Spot) VMs

- **Unallocated resources** leveraged as spot VMs with **relaxed SLOs**
  - Spot VMs can be **revoked** anytime for regular-priority VMs
  - Cost ~**50-90% less** than regular-priority VMs

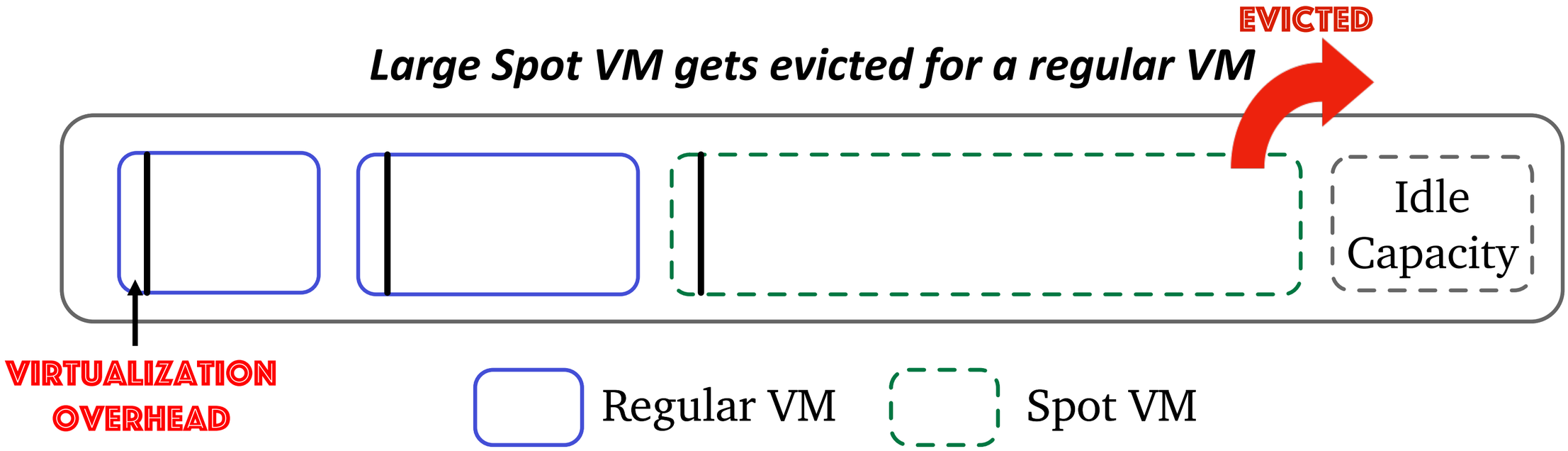
- But spot VMs are **fixed size** *VMs* and …



Evictable (Spot) VMs

- **Unallocated resources** leveraged as spot VMs with **relaxed SLOs**
  - Spot VMs can be **revoked** anytime for regular-priority VMs
  - Cost ~**50-90% less** than regular-priority VMs

- But spot VMs are **fixed size** *VMs* and …

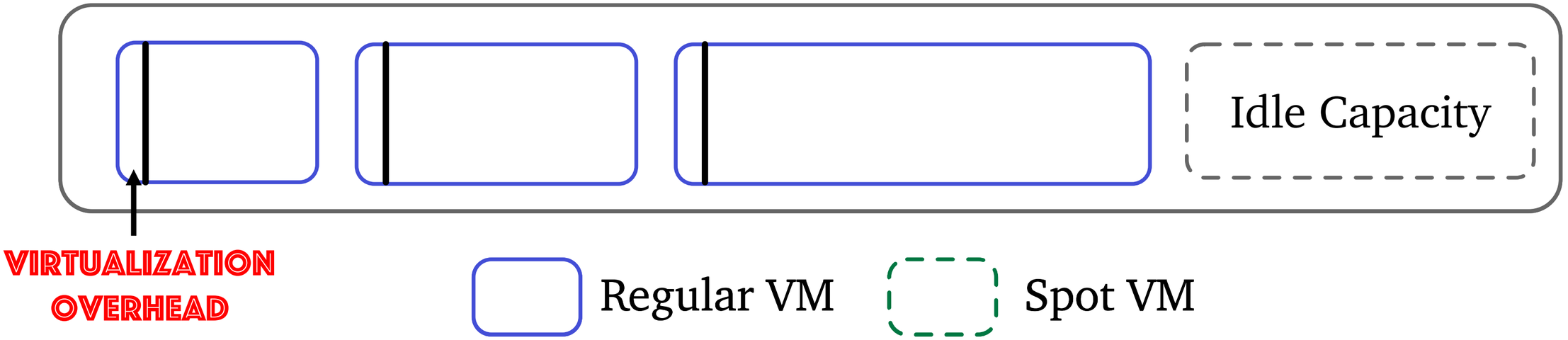


Evictable (Spot) VMs

- **Unallocated resources** leveraged as spot VMs with **relaxed SLOs**
  - Spot VMs can be **revoked** anytime for regular-priority VMs
  - Cost ~**50-90% less** than regular-priority VMs

- But spot VMs are **fixed size** *VMs* and …

Large Spot VM gets evicted for a regular VM

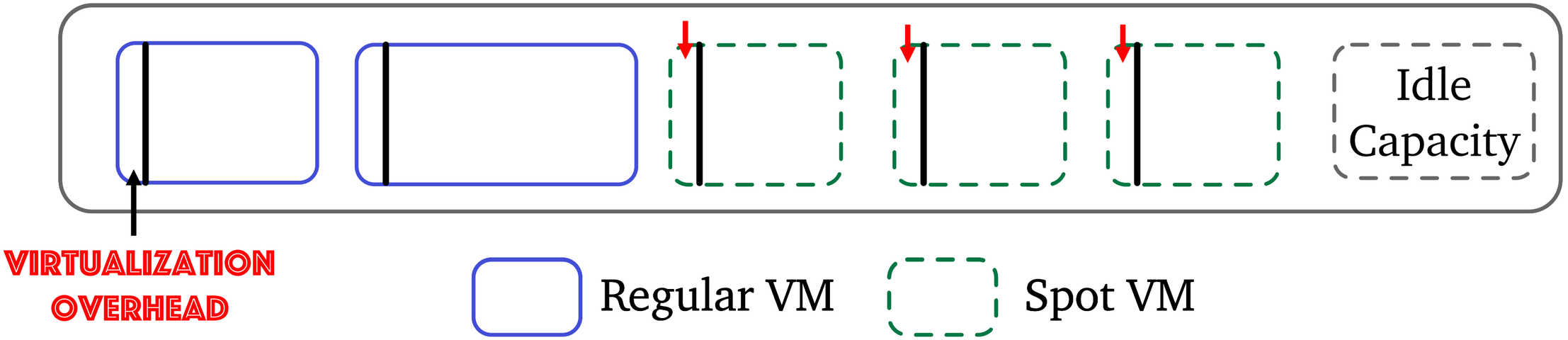


Evictable (Spot) VMs

- **Unallocated resources** leveraged as spot VMs with **relaxed SLOs**
  - Spot VMs can be **revoked** anytime for regular-priority VMs
  - Cost ~**50-90% less** than regular-priority VMs

- But spot VMs are **fixed size** *VMs* and …

Multiple Small Spot VMs entail high and eviction overhead



Our Proposal: Harvest VM and SLOs for Them

- Harvest VM - a new class of **evictable** VMs
  - Allocated with a minimum size (physical resources)
  - Dynamically **grows and shrinks** to harvest unallocated resources on the host
  - Only **evicted** if its minimum size is needed for a regular VM

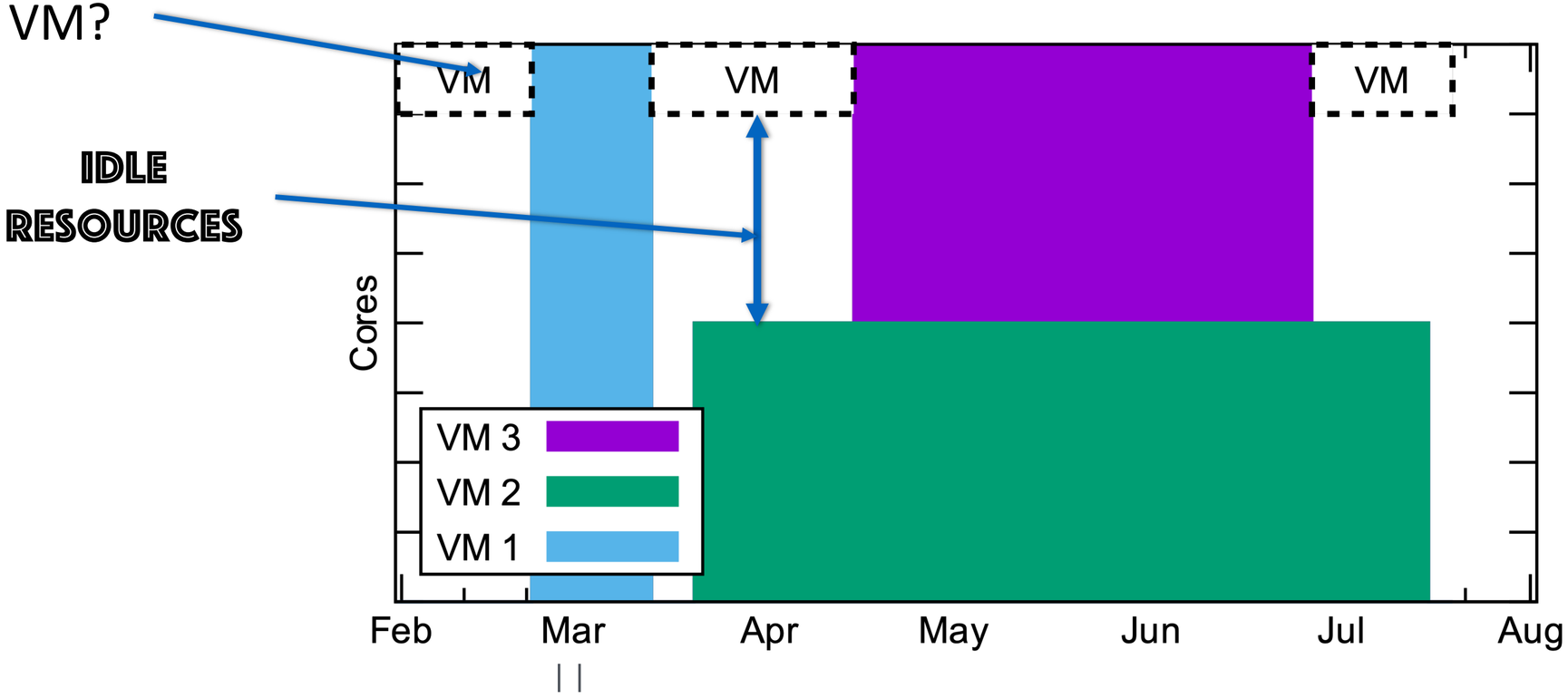
- Harvest VMs different than Burstable VMs
  - Burstable VMs **only burst for brief time** up to their max size after accumulating credits
  - Harvest VMs grow to consume **all unallocated resources at all times**

Road Map

- Characterize the unallocated resources of all Azure clusters
- Harvest VMs: new VM type that harvests unallocated resources
- SLO for Harvest VM: predict survival and amount of harvested resources
- Harvest Hadoop: platform to leverage harvested resources transparently
- Lessons and experiences from production

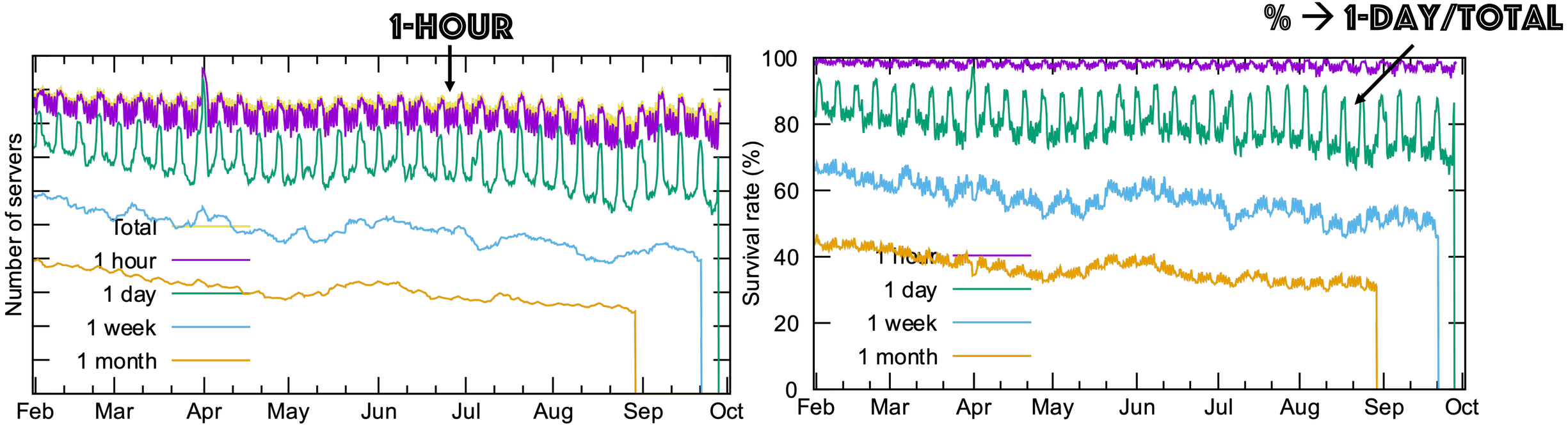
Characterizing unallocated resources

- Methodology
  - **6-month long traces** from February to October 2019
  - All **azure production** clusters for regular compute (e.g., no storage or GPUs)
  - Compute unallocated resources for each host server
  - Could we place a VM?



Characterization: Temporal patterns

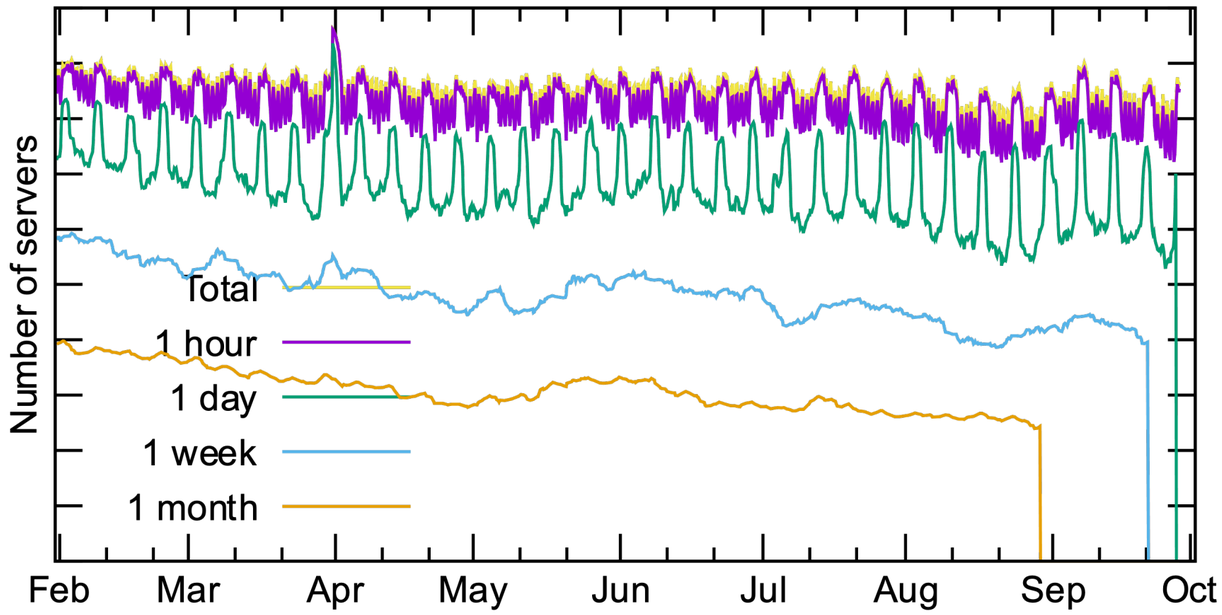
- Unallocated resources for a region
  - 1-hour shows *diurnal* pattern (nights have more)
  - 1-day shows *weekly* patterns (weekends have more)
  - Fewer servers have enough unallocated capacity over longer horizon



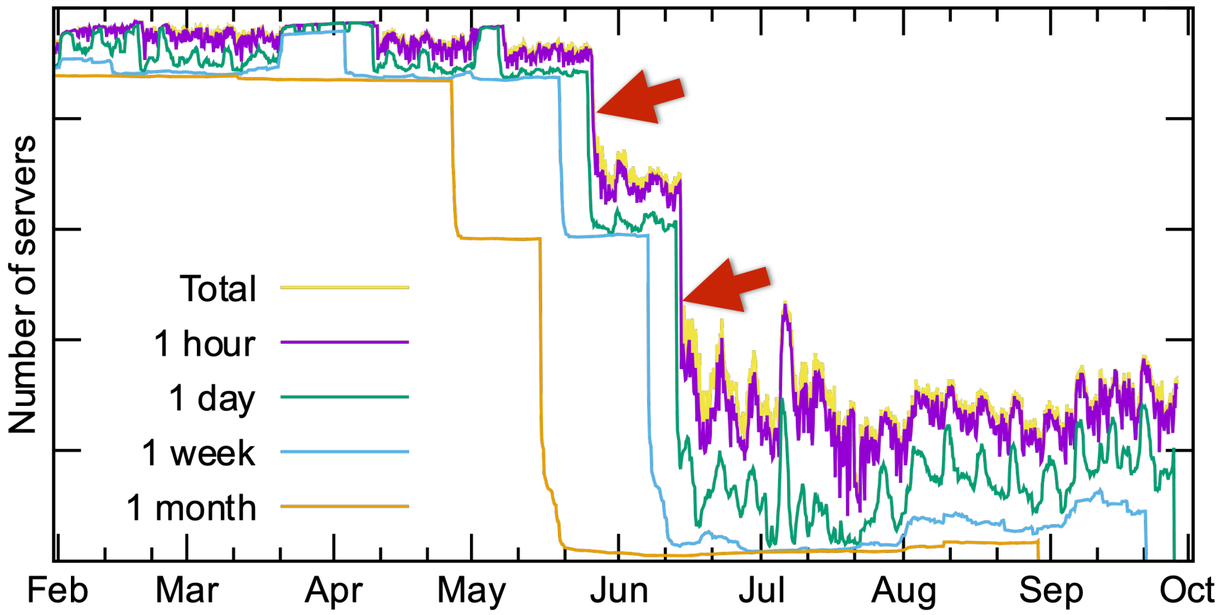
Characterization: Cluster behaviors

- Unallocated resources at region level are **stable**
- Unallocated resources at cluster level can change **abruptly**

Region Level



Cluster Level



Characterization: Key Takeaways

- Many unallocated resources available for harvesting
  - Dynamic temporal and spatial behaviors
- Unallocated resources **not evenly** distributed across clusters
  - Smaller amount of resources more widely available
  - Larger amounts of resources may last longer
- Many additional unallocated resources beyond spot VMs size
- Filling with spot VMs takes many more VMs (and many more evictions)

Harvest VMs: Overview

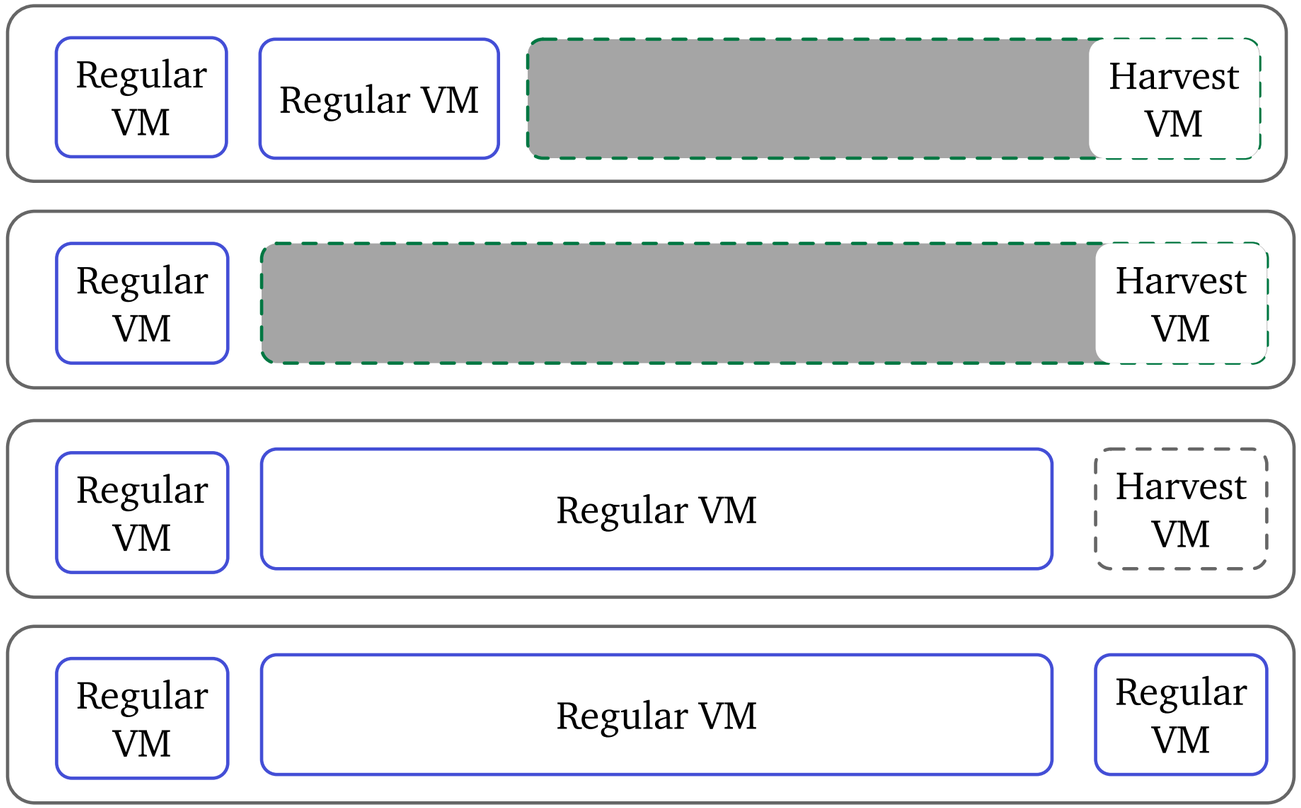
- **New VM class**
  - User picks *minimum/maximum* size
- **Harvest unallocated resources dynamically**

T_0: All unallocated first

T_1: *Grow* when VM leaves

T_2: *Shrink* when new VM lands in host

T_3: *Evicted* if providers needs minimum



Harvest VMs: Implementation

- Based on Azure VM EXv3
  - Example: 1, 2, and 4 physical cores ← **Harvested resource**
  - Fixed number (e.g., 40) of virtual cores
- Currently at most one Harvest VM per host server
- Changes in physical resources exposed to VMs
- Pricing
  - Same price as spot VM for minimum size
  - Further discount (e.g., 50%) on any additional cores beyond the minimum size
- Available in production for interval users

SLOs for Harvest VMs

- Hard to provision just enough VMs with variable resources
  - ***Key***: VMs survival rate?
  - ***Key***: How many resources will I get on average?

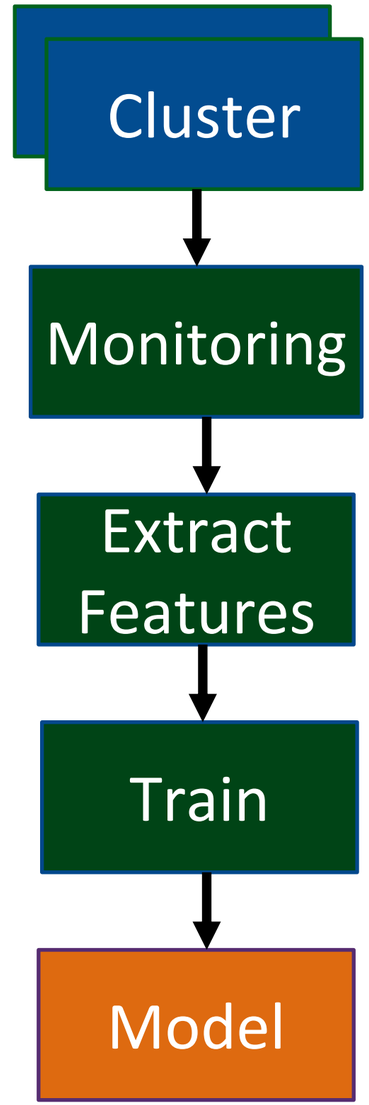
- Example SLO: User requests 100 Harvest VMs in East US
  - 85% of them survive →1-hour and 35% for → 1-month
  - An average of 8.5 cores
  - 95% confidence intervals (80-90% survive → 1-hour)

SLO Predictor Features

- Random Forest Regressor

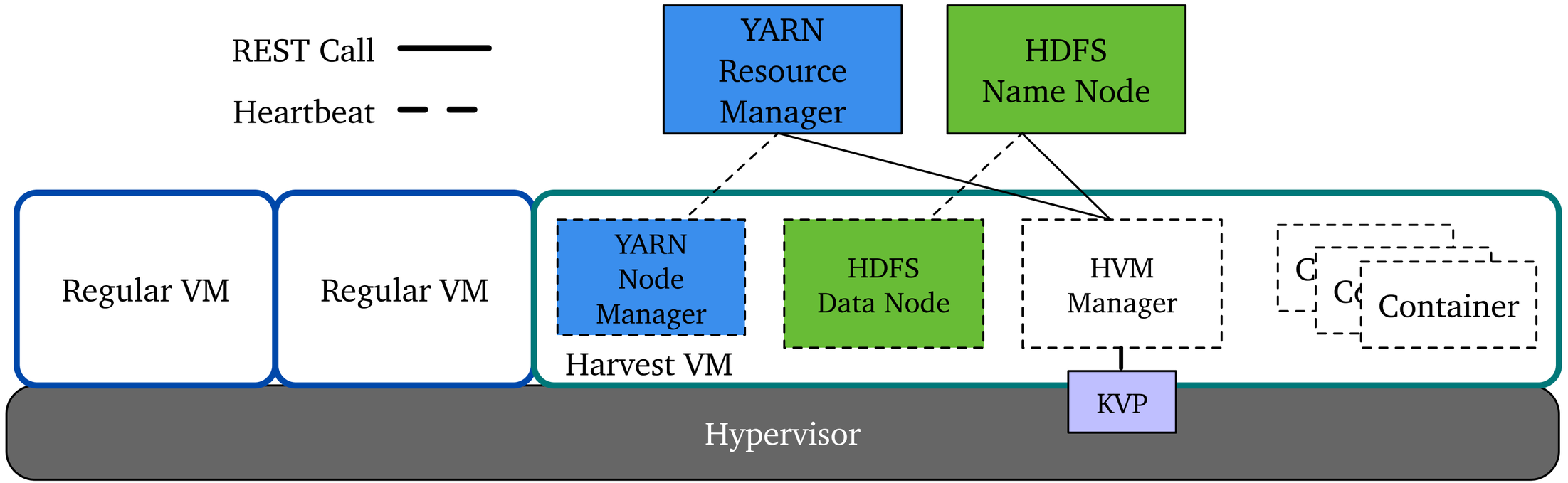
- Features
  - Total VMs in the cluster
  - Total cores/memory allocated and available
  - Cluster characteristics (generation, number of racks,…)
  - Auto-regressive (e.g., values 1 day ago)
  - Moving average (average values for the last week)

- Integrated into Resource Central (SOSP'17)



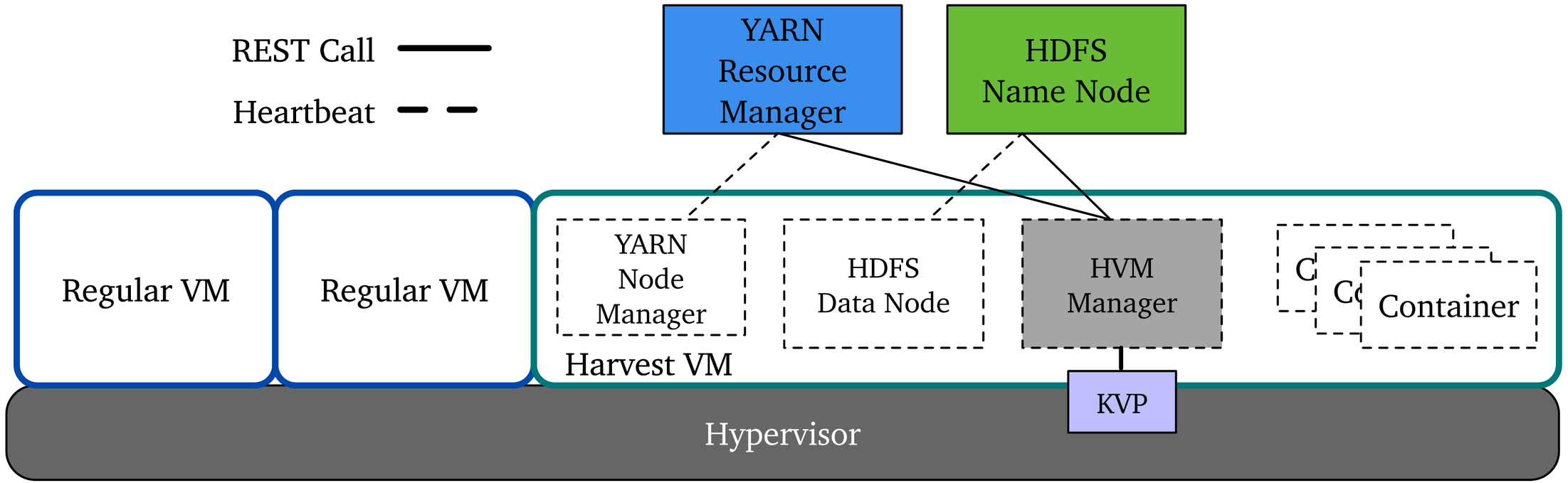
Building Application on Harvest VMs

- Applications can naively use Harvest VMs
  - Leverage fault tolerance for evictions → **Inefficient**
  - Run using minimum resources available → **May be slower**
- **Extend Hadoop** - run many applications (Spark, MapReduce,…)



Building Application on Harvest VMs

- Applications can naively use Harvest VMs
  - Leverage fault tolerance for evictions → **Inefficient**
  - Run using minimum resources available → **May be slower**
- **Extend Hadoop** - run many applications (Spark, MapReduce,…)



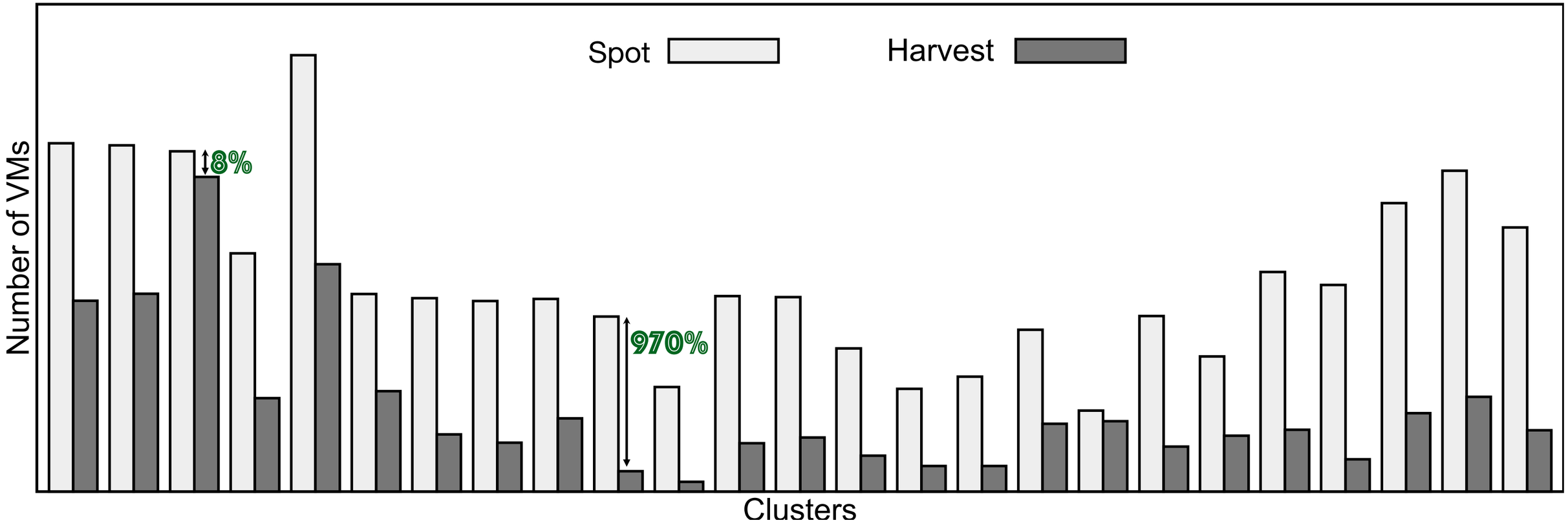
Evaluation

- Use production data for evaluation
  - **25 clusters** from **14 regions** across **2 server** generations
  - December 2019 to April 2020 (train/test split January 15th)

- **Extreme scenario**: every possible hole is filled
  - Simulate real traces and insert as many Harvest VMs as possible

- **Harvest Hadoop deployment**
  - *Private cluster* – **not many** VMs coming and going (**stable**)
  - *Canary cluster* – **many** VMs created and destroyed (**stress test**)

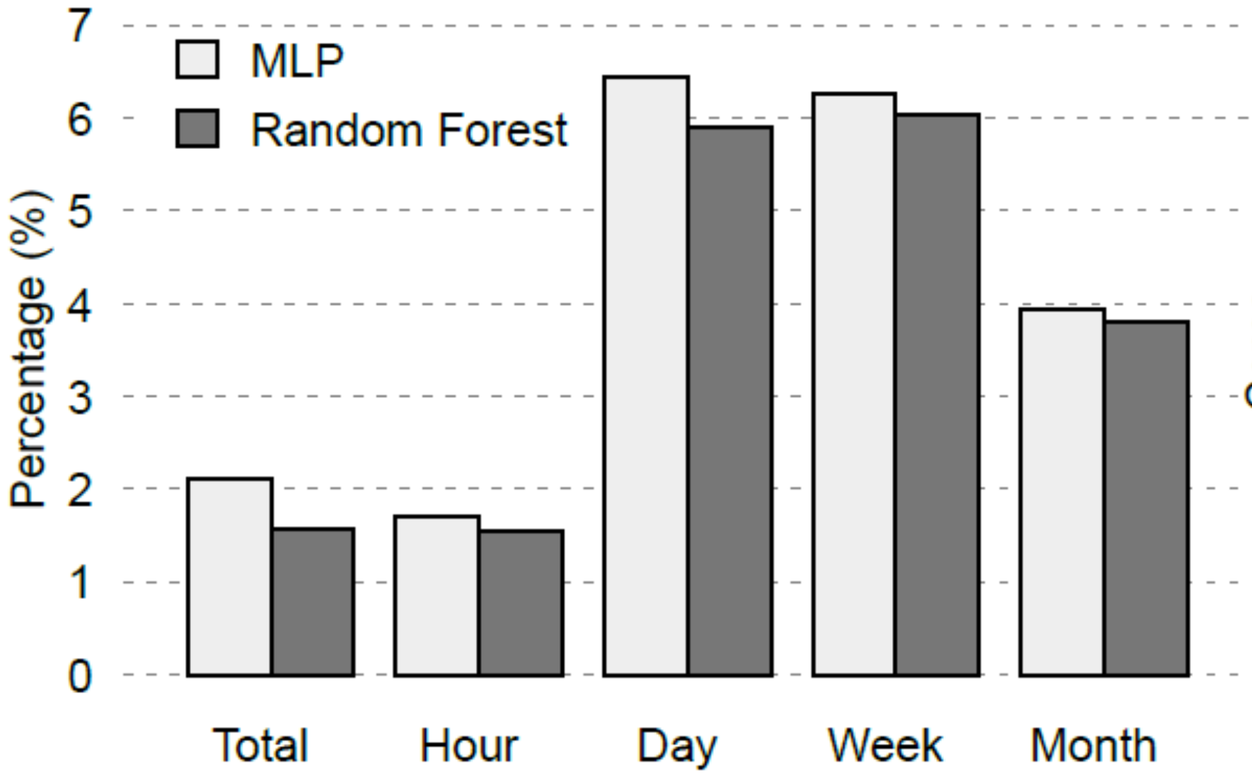
Evaluation: Spot VMs vs Harvest VMs



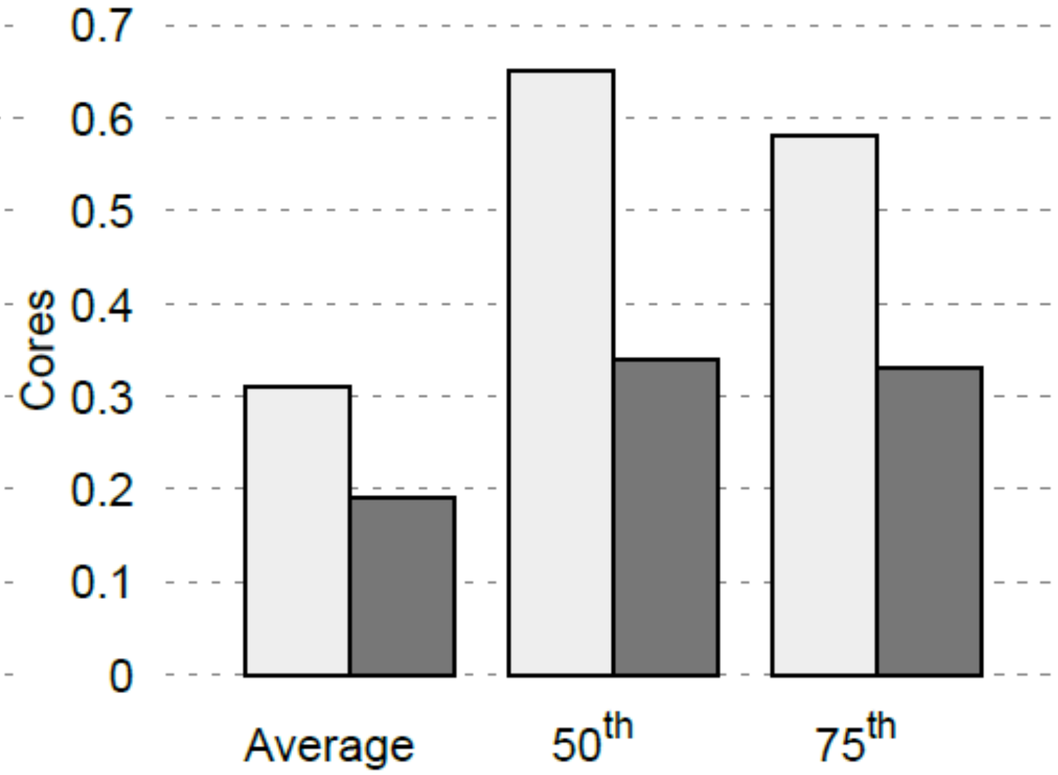
Requires around 3.7x more evictable VMs on average than Harvest VMs to fill unallocated capacity across all clusters

Evaluation: Random Forest vs MLP

Survival rate prediction

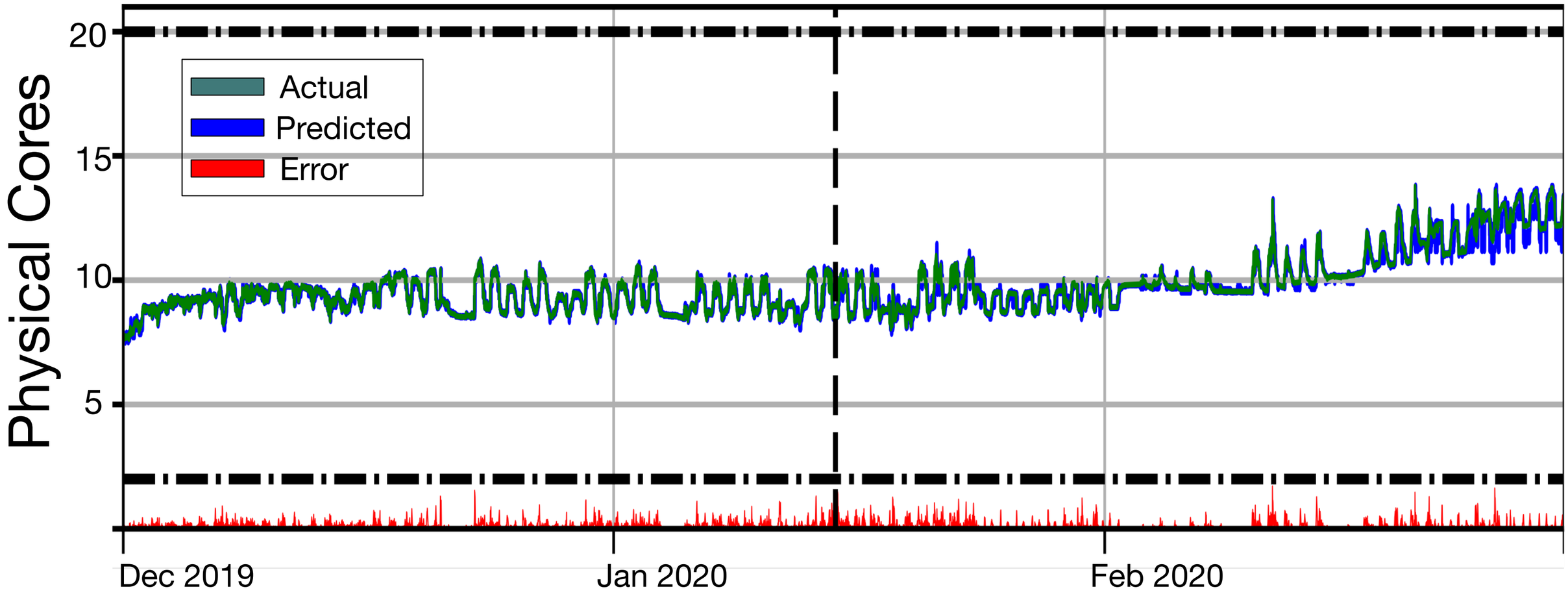


Average cores prediction



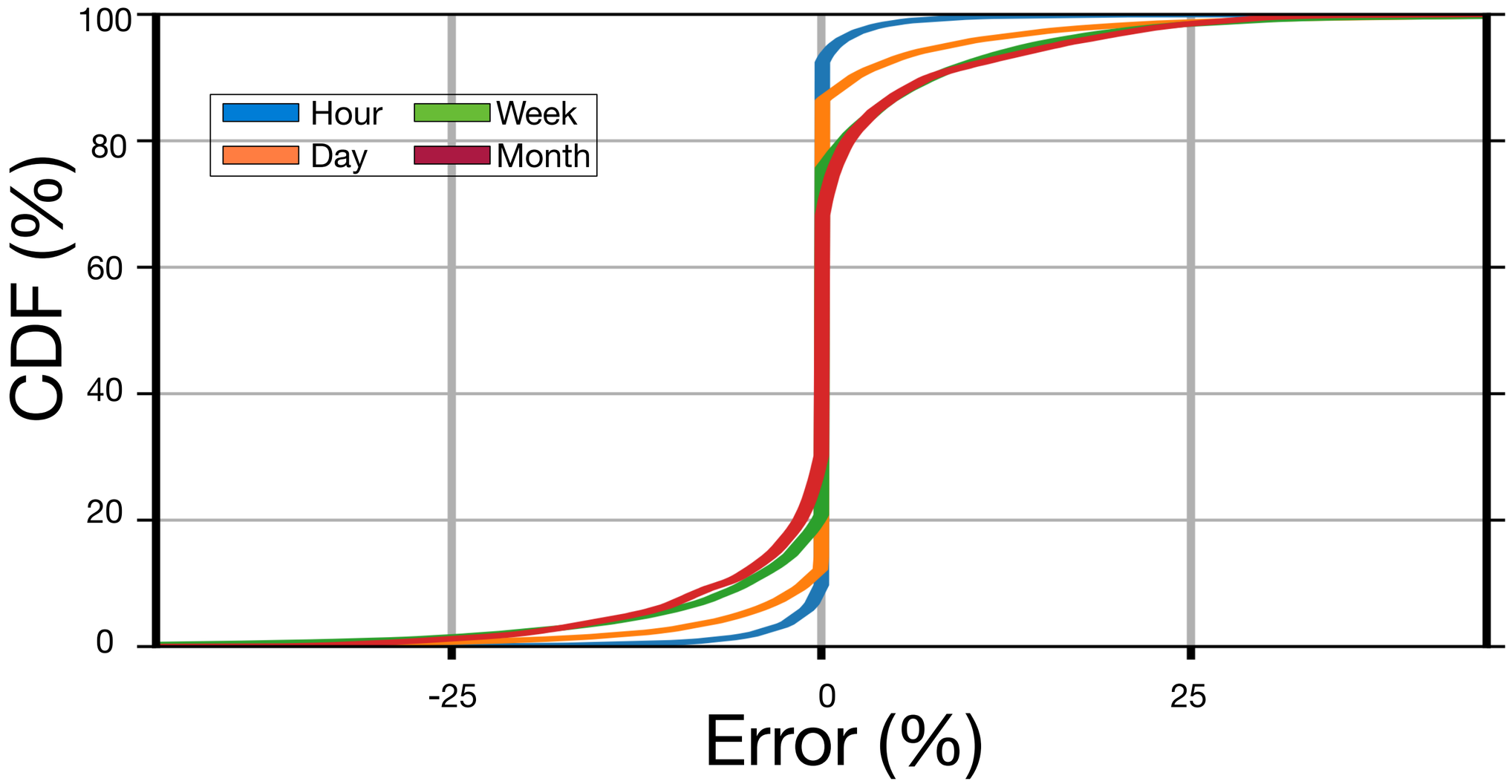
Random Forest yields an (overall) accuracy of ~98% with mean error of ~0.2 cores

Evaluation: SLOs for Harvested Cores



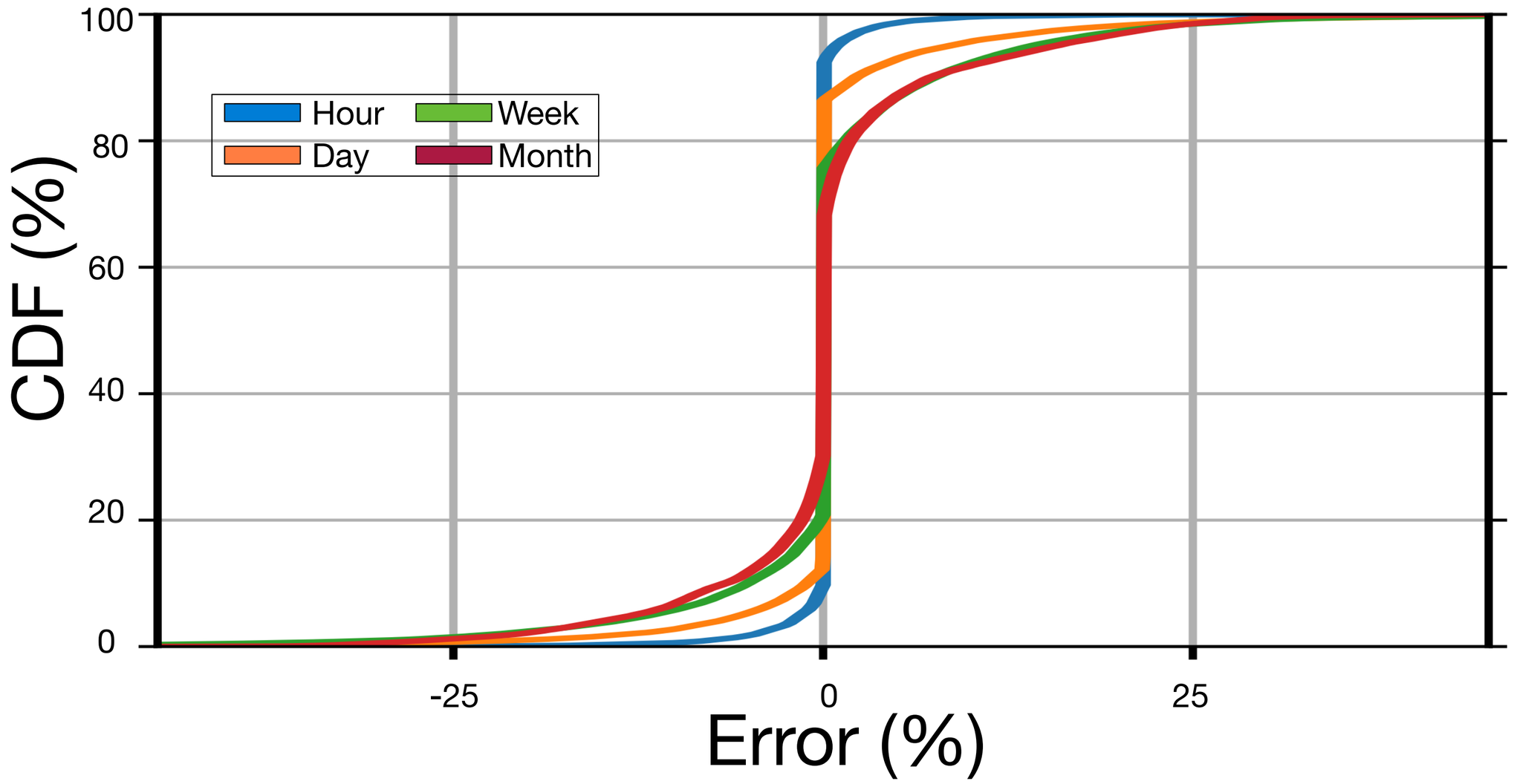
Prediction accuracy is very high i.e. average cores SLO would be accurate

Evaluation: SLOs for Survival Rate



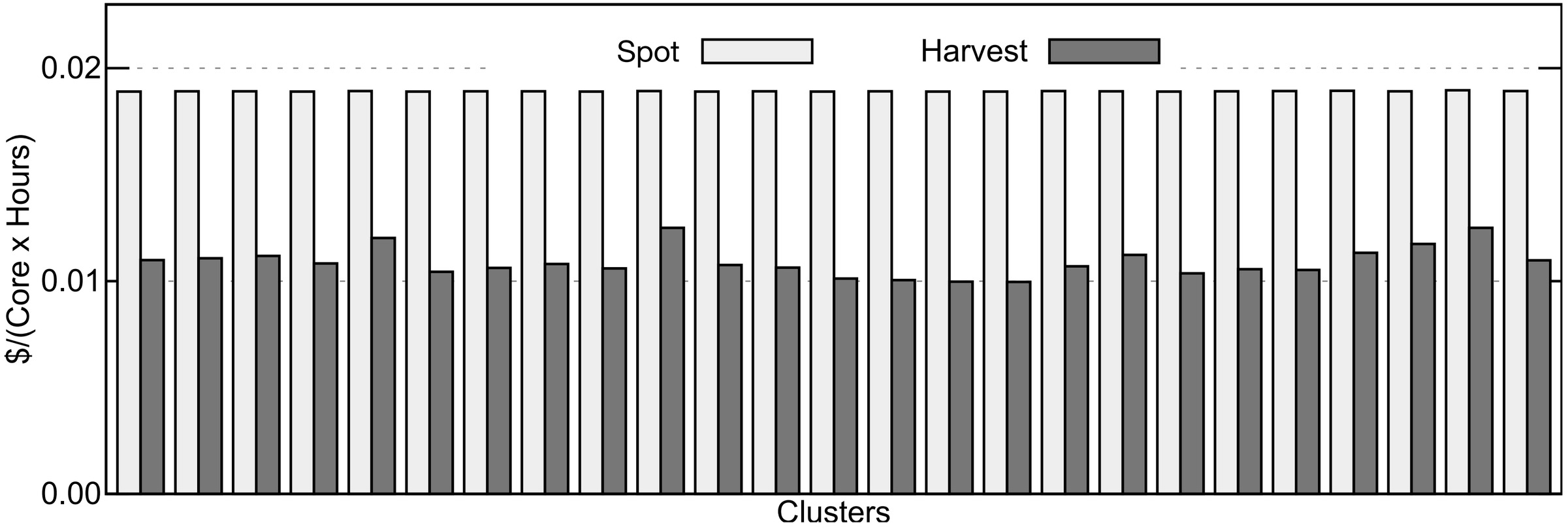
Short-term predictions have an avg error < 2% and < 6% for longer terms

Evaluation: SLOs for Survival Rate



Errors are balanced and there are as many overpredictions as underpredictions

Evaluation: Cost Comparison



Harvest VMs 91% cheaper than regular VMs and 45% cheaper than spot VMs

Lessons from Production

- Adapting applications is the main blocker

- When a Harvest VM gets 40 virtual cores, it becomes unbalanced
  - 2 cores/16GB of memory → 40 cores/16GB of memory

- Allowing multiple Harvest VMs per server
  - Add the maximum size of each Harvest VM

- Impact to regular VMs
  - Optimization to reduce impact in creation time

Conclusion

- Characterization shows many unallocated resources for harvesting
  - Dynamic temporal and spatial behaviors
- Harvest VMs successful at leveraging unallocated resources
- We provide SLOs for the availability of harvested resources
  - Our prediction models show high accuracy (~98%)
- Harvest Hadoop can adjust to changing harvested resources
- Harvest VMs and Harvest Hadoop running in production in Azure
  - 91% cheaper than regular VMs
  - 45% cheaper than spot VMs and with 73% fewer evictions

Thank You

Questions?

Email:
lambati@umass.edu
inigog@microsoft.com
ricardob@microsoft.com