# Applying Control Theory to Application Performance Management in the Cloud

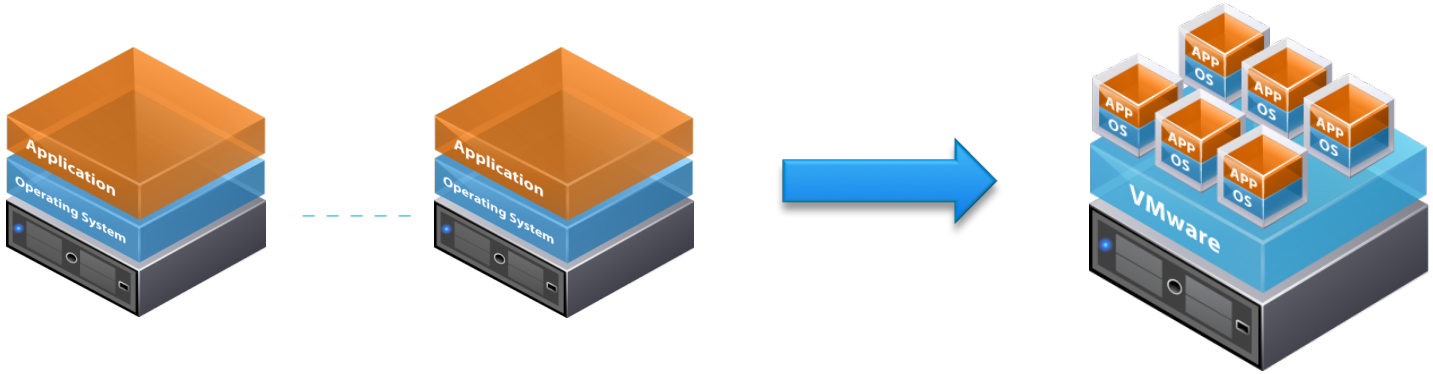
Xiaoyun Zhu

CDS@20

August 7, 2014

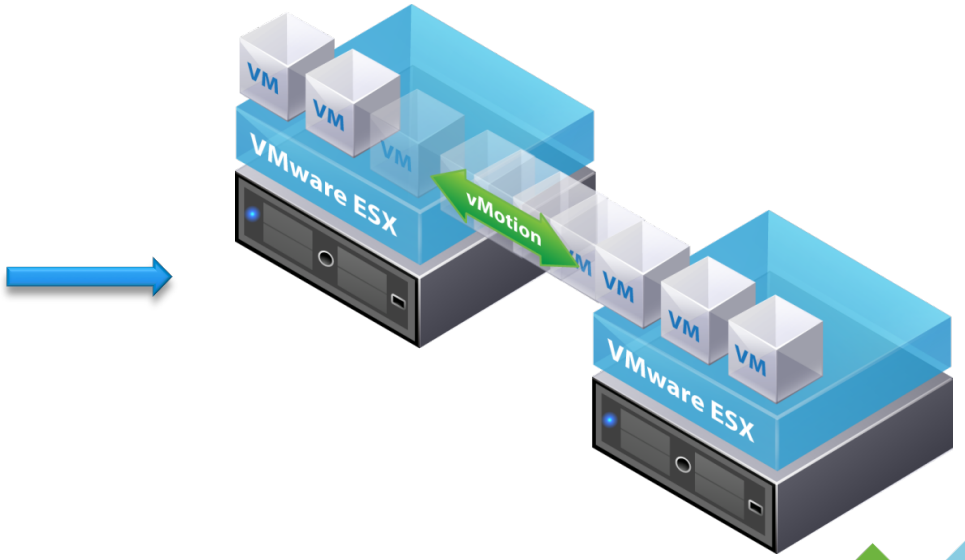
vmware

# Virtualization – a new computing paradigm

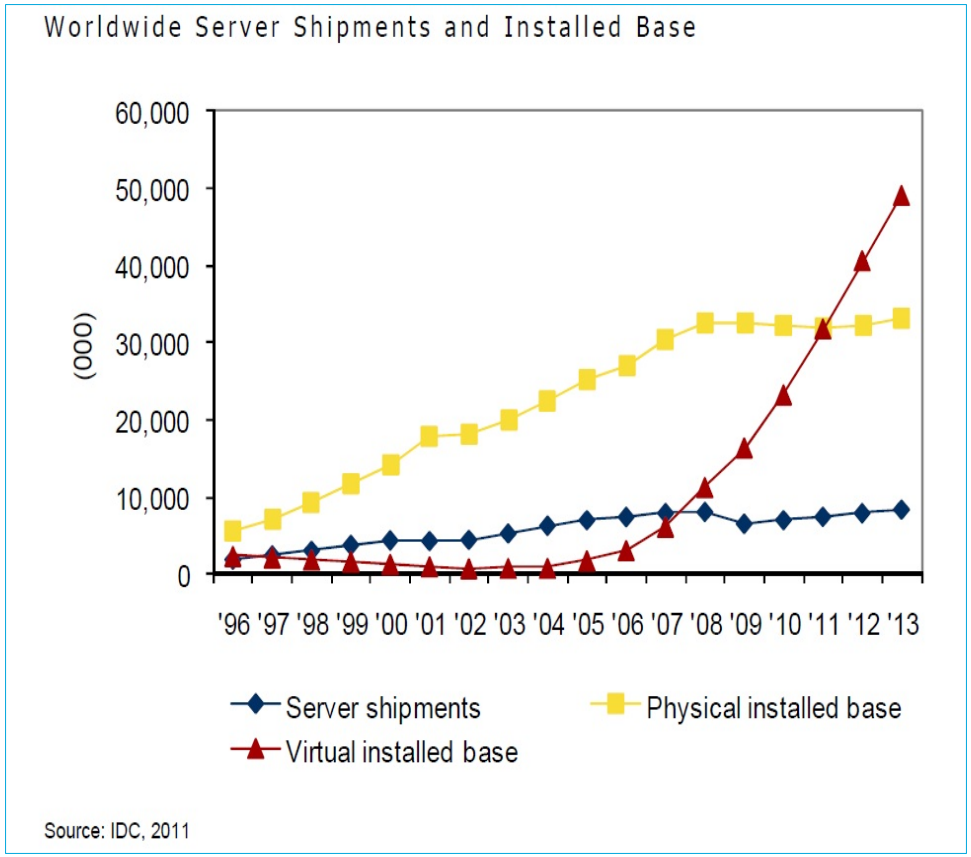


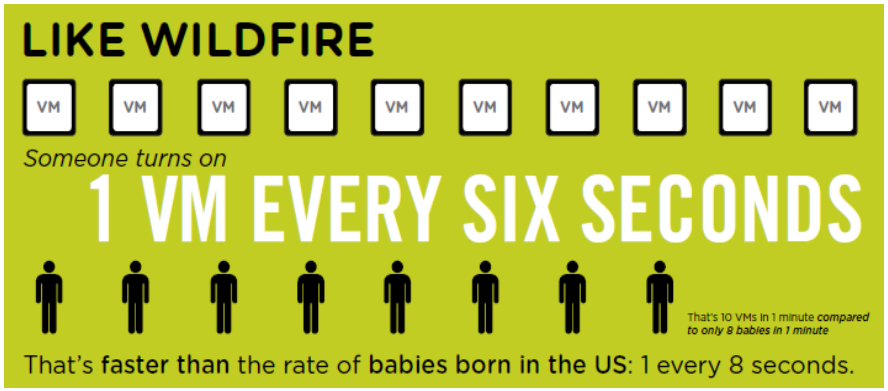

## Key benefits

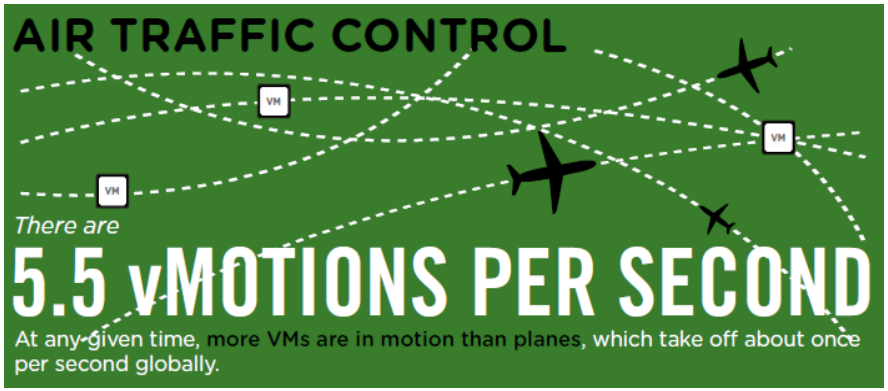
- Higher hardware utilization
- Easier deployment
- Elastic capacity
- Better agility via live migration
- Higher availability
- Fault tolerance
- Lower energy cost



# Virtual machines become mainstream in IT



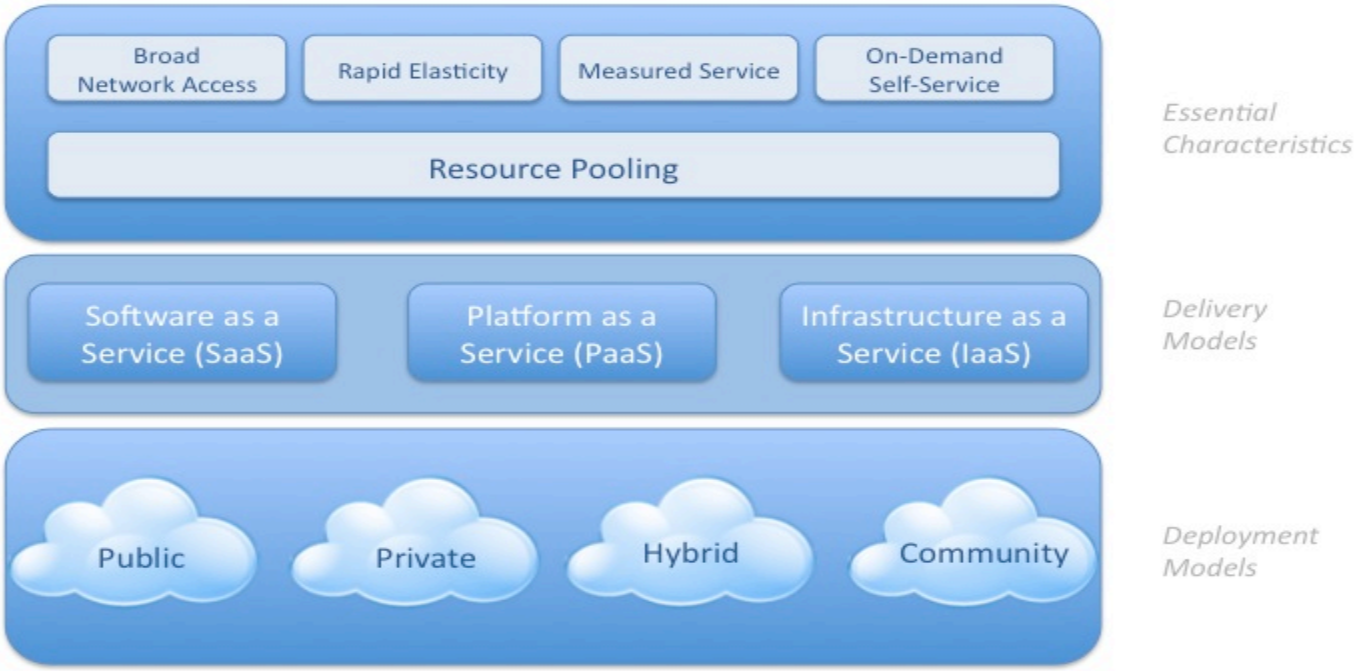




- (also from Gartner): 5 out of every 6 x86 server workloads are deployed in VMs by 2015.

- vSphere-infographic, VMworld 2011.

# What is cloud computing?

- Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models.



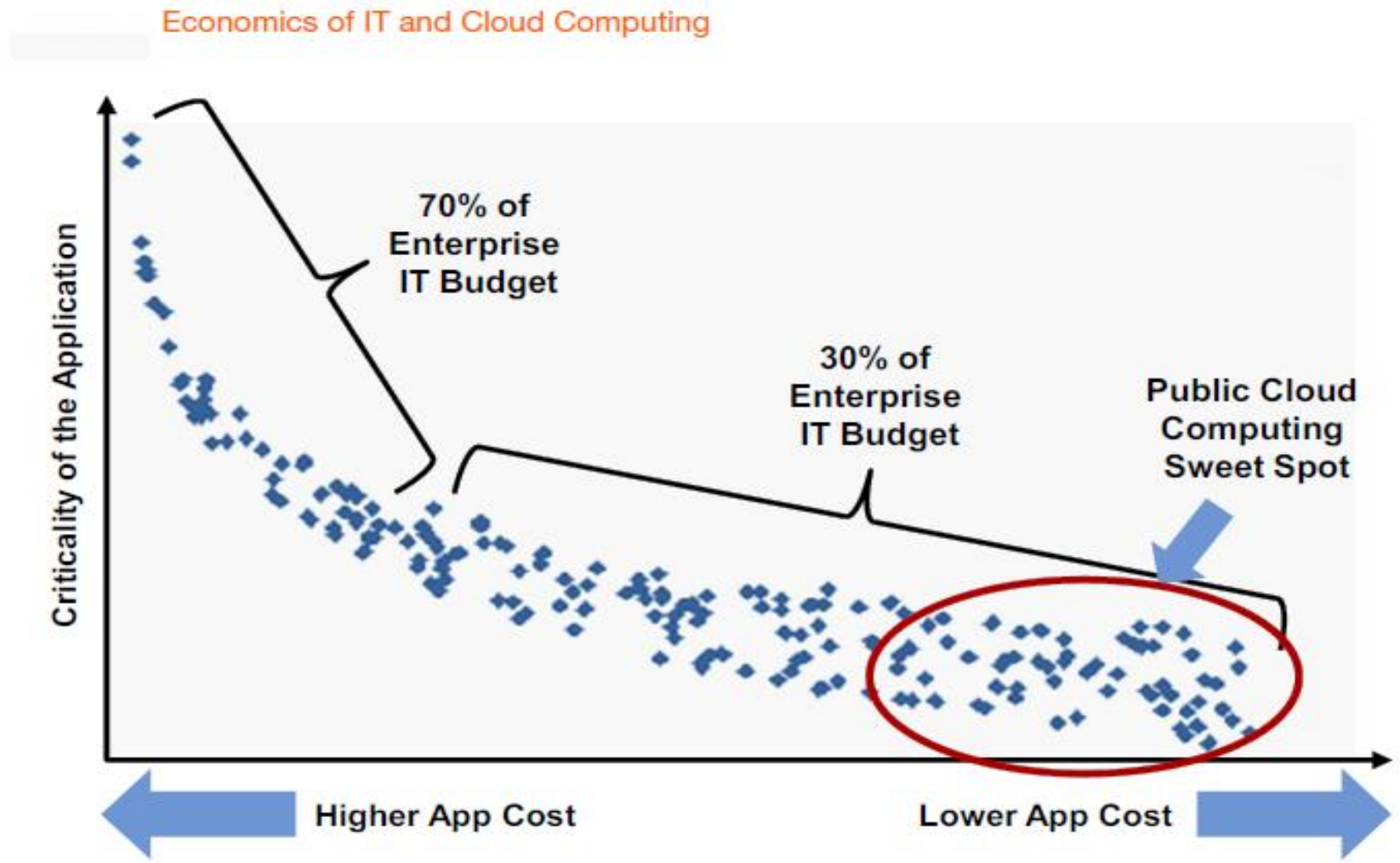

- *Source: NIST definition of cloud computing. NIST special publication 800-145, Sep. 2011.*

# Rapidly growing public cloud market


Source: Gartner (February 2013)

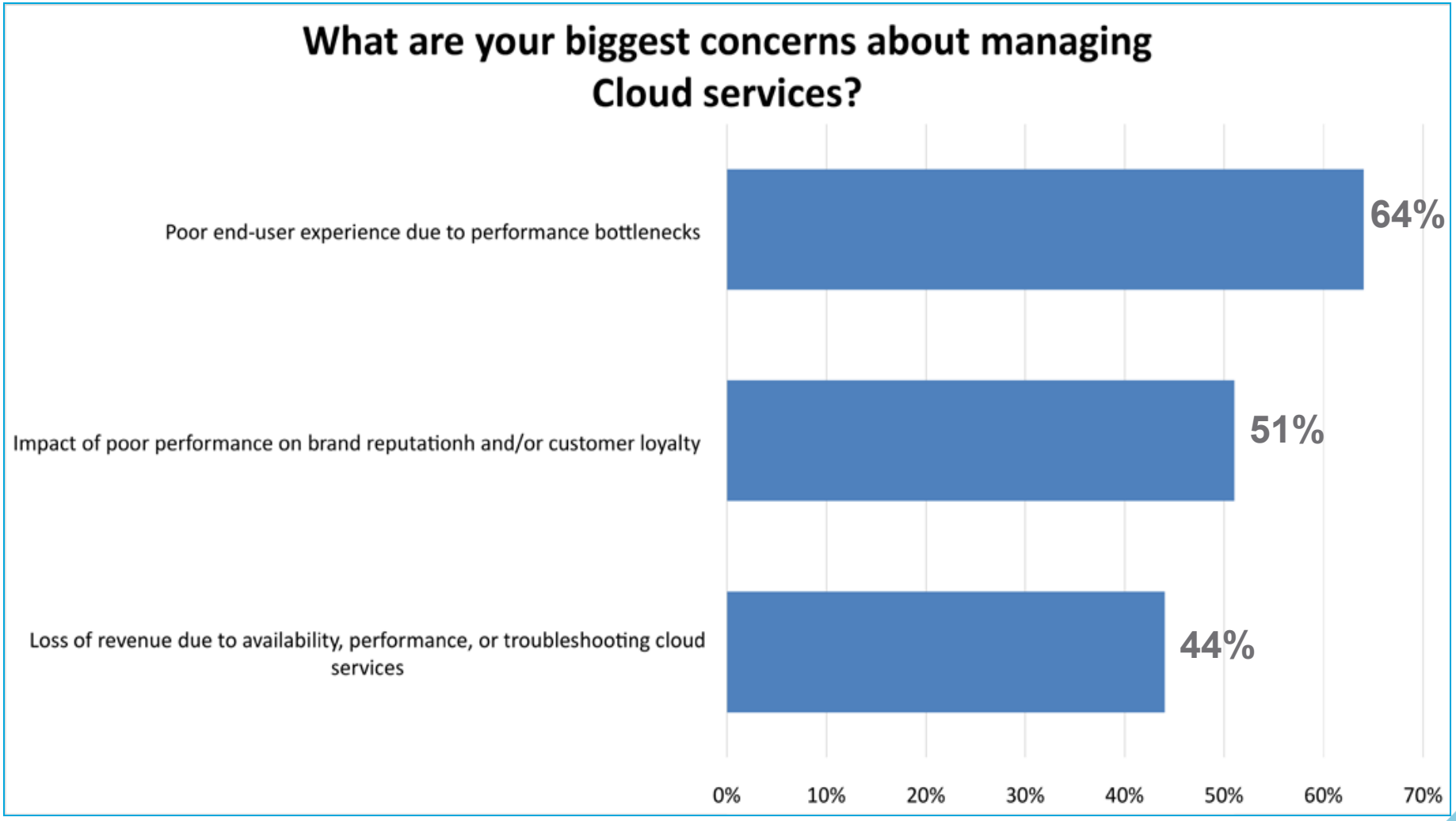
vmware

# How about hosting critical applications?




Source: Gartner (October 2012)

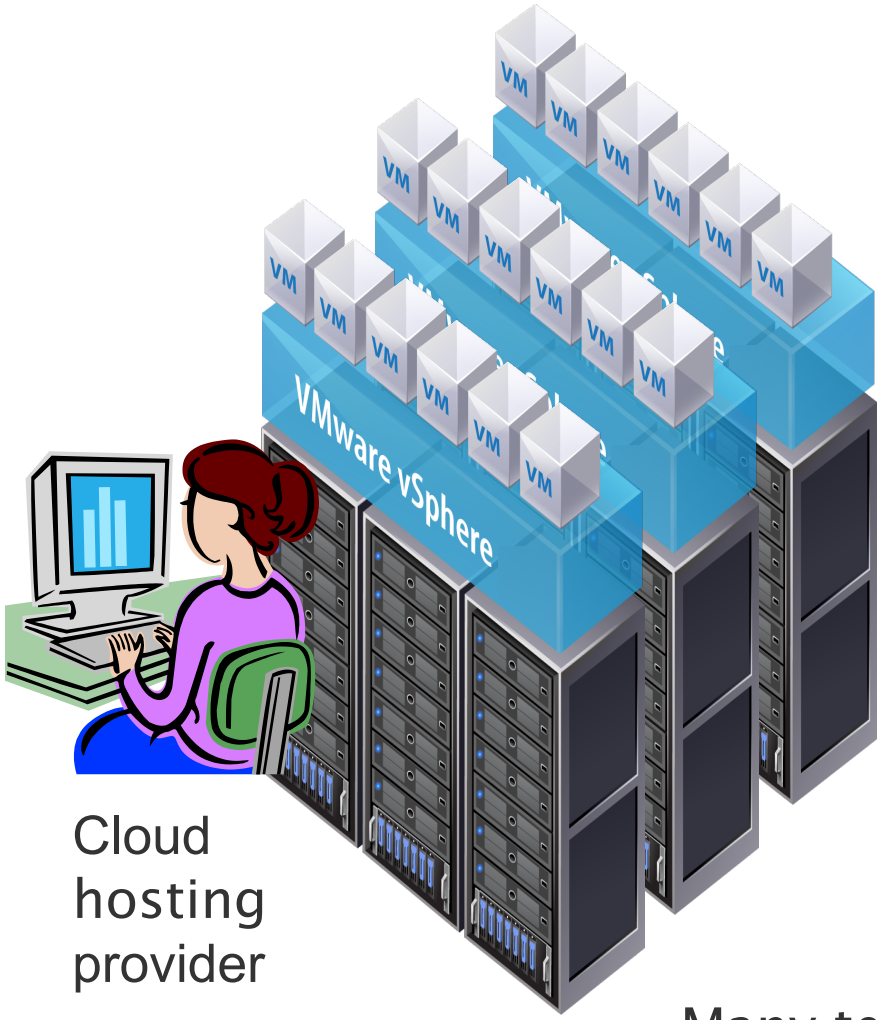
# Application performance – a real concern




*Source: "The hidden costs of managing applications in the cloud," Compuware/Research In Action White Paper, Dec. 2012, based on survey results from 468 CIOs in Americas, Europe, and Asia.*

vmware

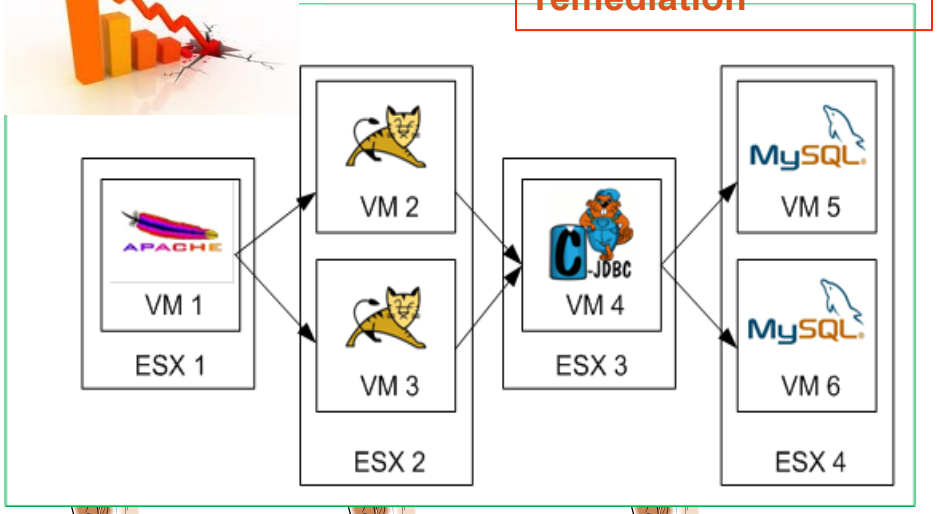
# Application performance management is hard

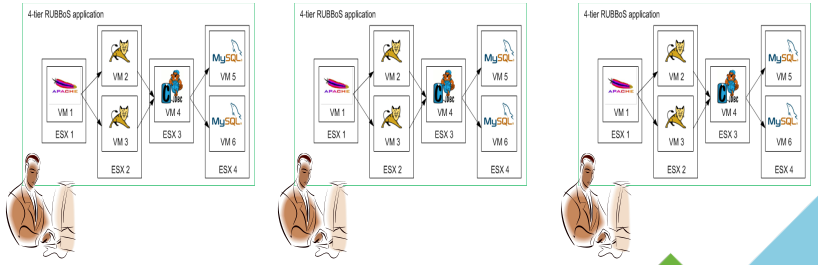



Cloud hosting provider

**Service Level Objective**: 95% of all transactions should be completed within 500ms

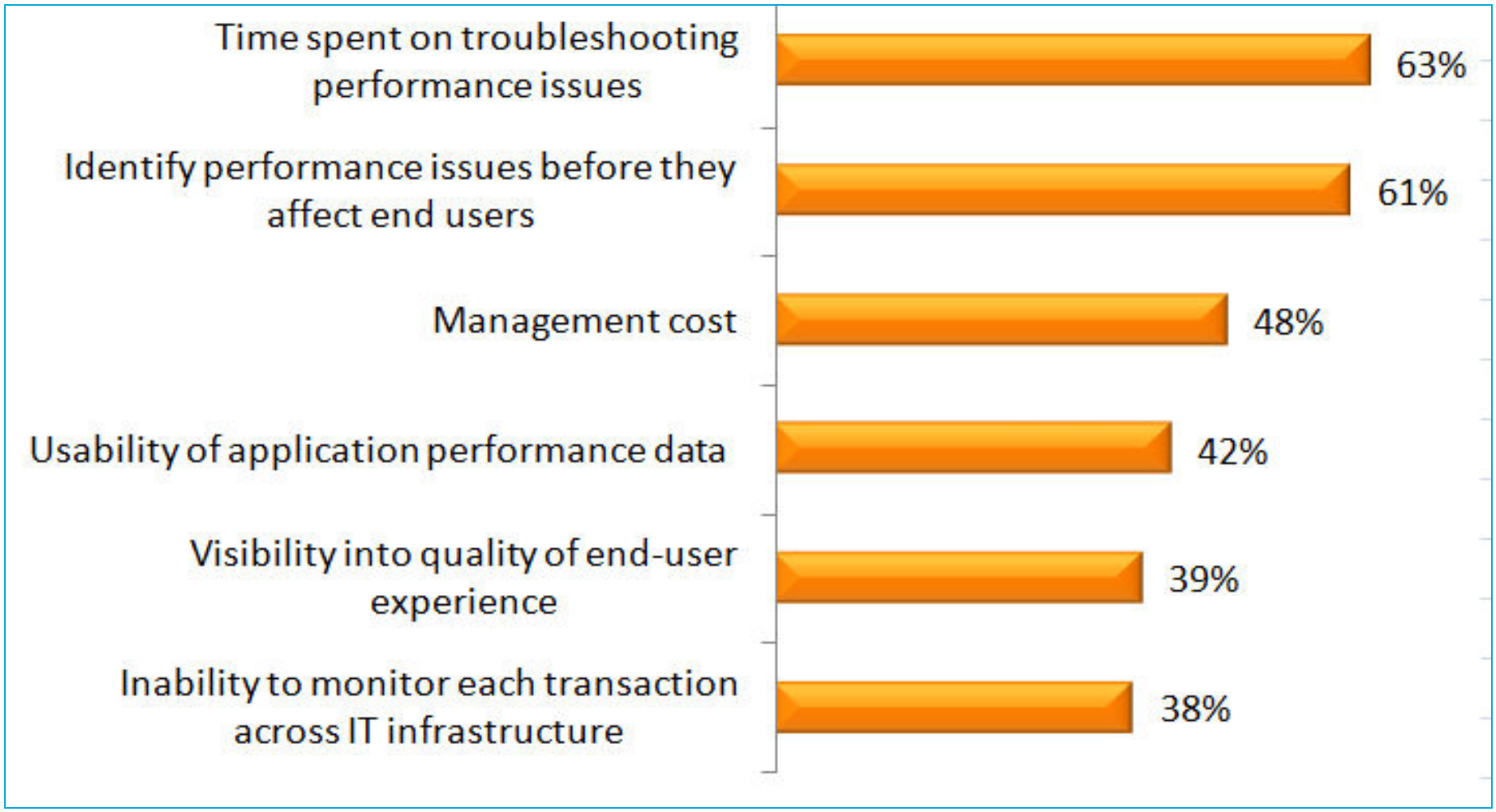
**SLO violation!** → **Performance troubleshooting & remediation**



Many tenant applications

# Challenges in managing application performance

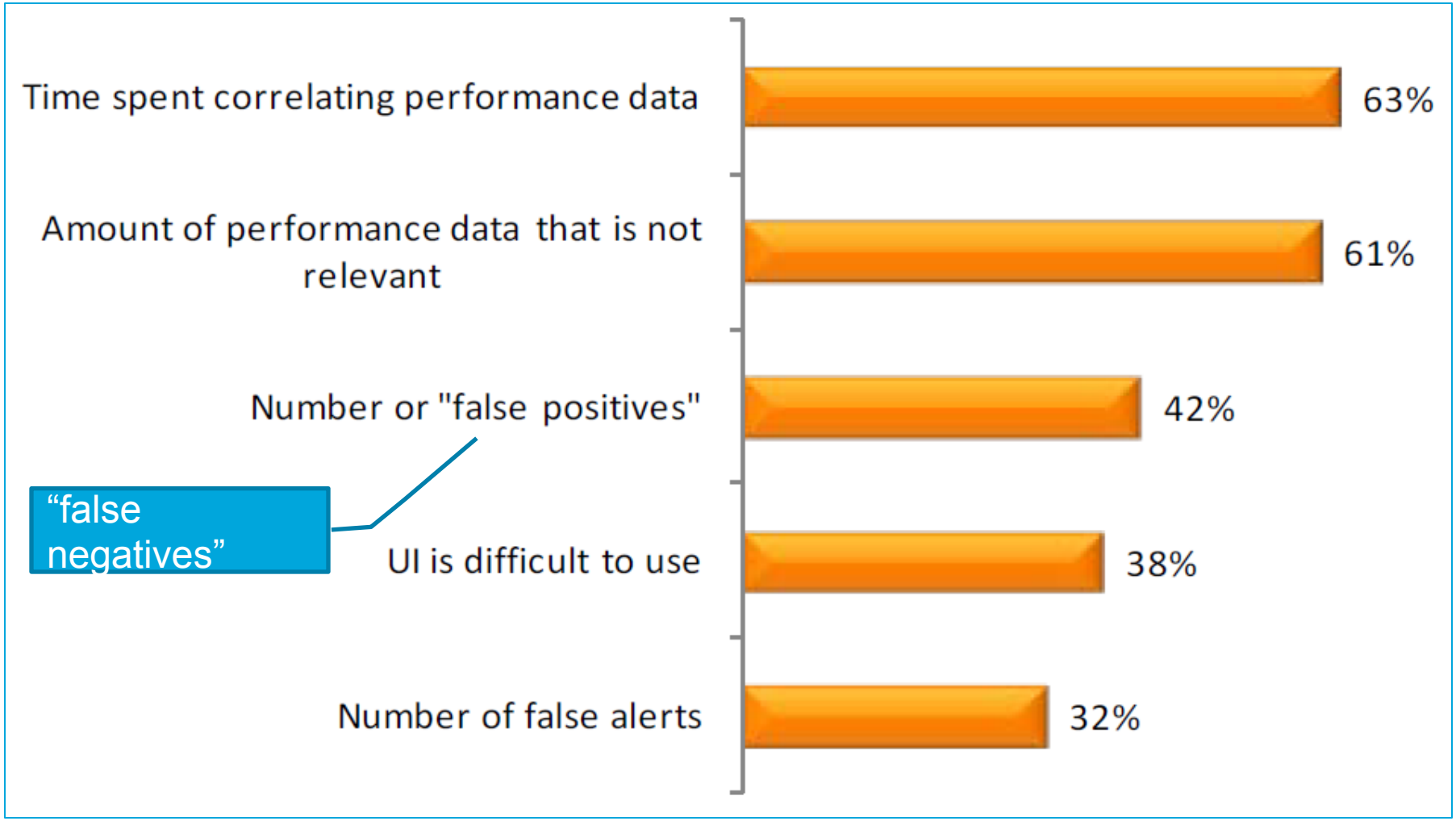



- On average, **46.2 hours** spend in "**war-room**" scenarios each month

*Source: Improving the usability of APM data: Essential capabilities and benefits. TRAC Research, June 2012, based on survey data from 400 IT organizations worldwide*

vmware

# Challenges in usability of performance data




*Source: Improving the usability of APM data: Essential capabilities and benefits. TRAC Research, June 2012, based on survey data from 400 IT organizations worldwide*

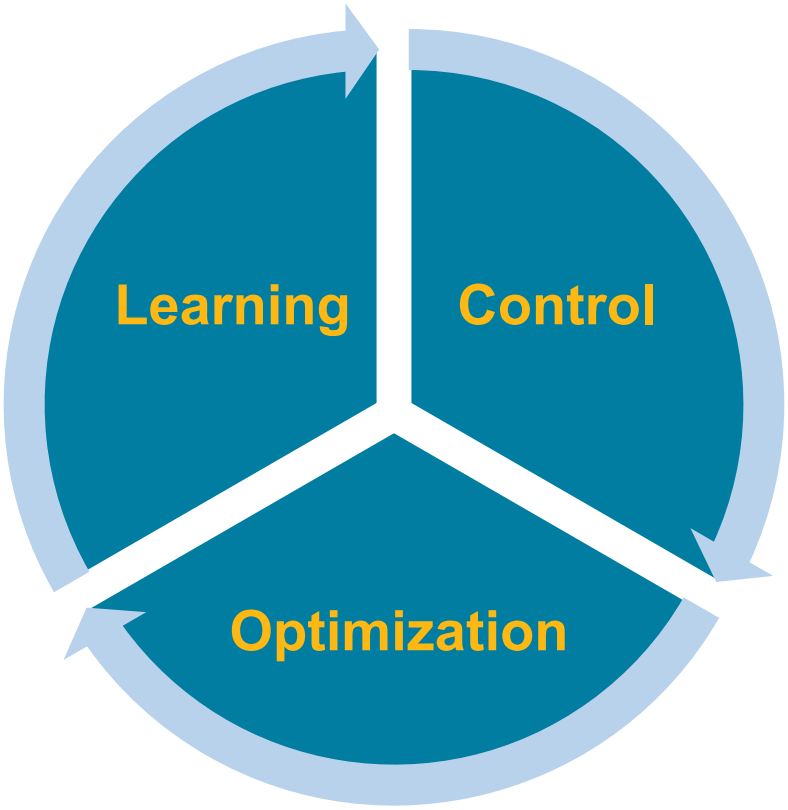
vmware

# APM goal: achieve service-level-objective (SLO)
## Technical challenges

- Enterprise applications are distributed or multi-tiered
- App-level performance depends on access to many resources
  - HW: CPU, memory, cache, network, storage
  - SW: threads, connection pool, locks
- Time-varying application behavior
- Time-varying hosting condition
- Dynamic and bursty workload demands
- Performance interference among co-hosted applications

# Better IT analytics for APM automation

## Three-pronged approach



# Why learning?

- Deals with APM-generated big data problem
- Fills the semantic gap with learned models
- Answers key modeling questions

# APM-generated big data

- "APM tools were part of the huge explosion in metric collection, generating thousands of KPIs per application."
- "83% of respondents agreed that metric data collection has grown >300% in the last 4 years alone."
- "88% of companies are only able to analyze less than half of the metric data they collect… 45% analyze less than a quarter of the data."
- "77% of respondents cannot effectively correlate business, customer experience, and IT metrics."

*Source: "APM-generated big data boom." Netuitive & APMDigest, July 2012, based on survey of US & UK IT professionals.*

# Real-time performance monitoring
## Infrastructure-level

### Physical host metrics

- System-level stats collected by the hypervisor
  - *e.g.*, `esxtop` – CPU, memory, disk, network, interrupt
- CPU stats
  - *%USED, %RUN, %RDY, %SYS, %OVRLP, %CSTP, %WAIT, %IDLE, %SWPWT*
- ~100s-1000s metrics per host!

### VM metrics

- Resource usage stats collected by the guest OS
  - *e.g.*, `dstat, iostat`
- ~10s metrics per VM

- Widely available on most platforms
- Available at a time scale of seconds to minutes

vmware®

# Real-time performance monitoring
## Application-level

### Metrics reflecting end user experience

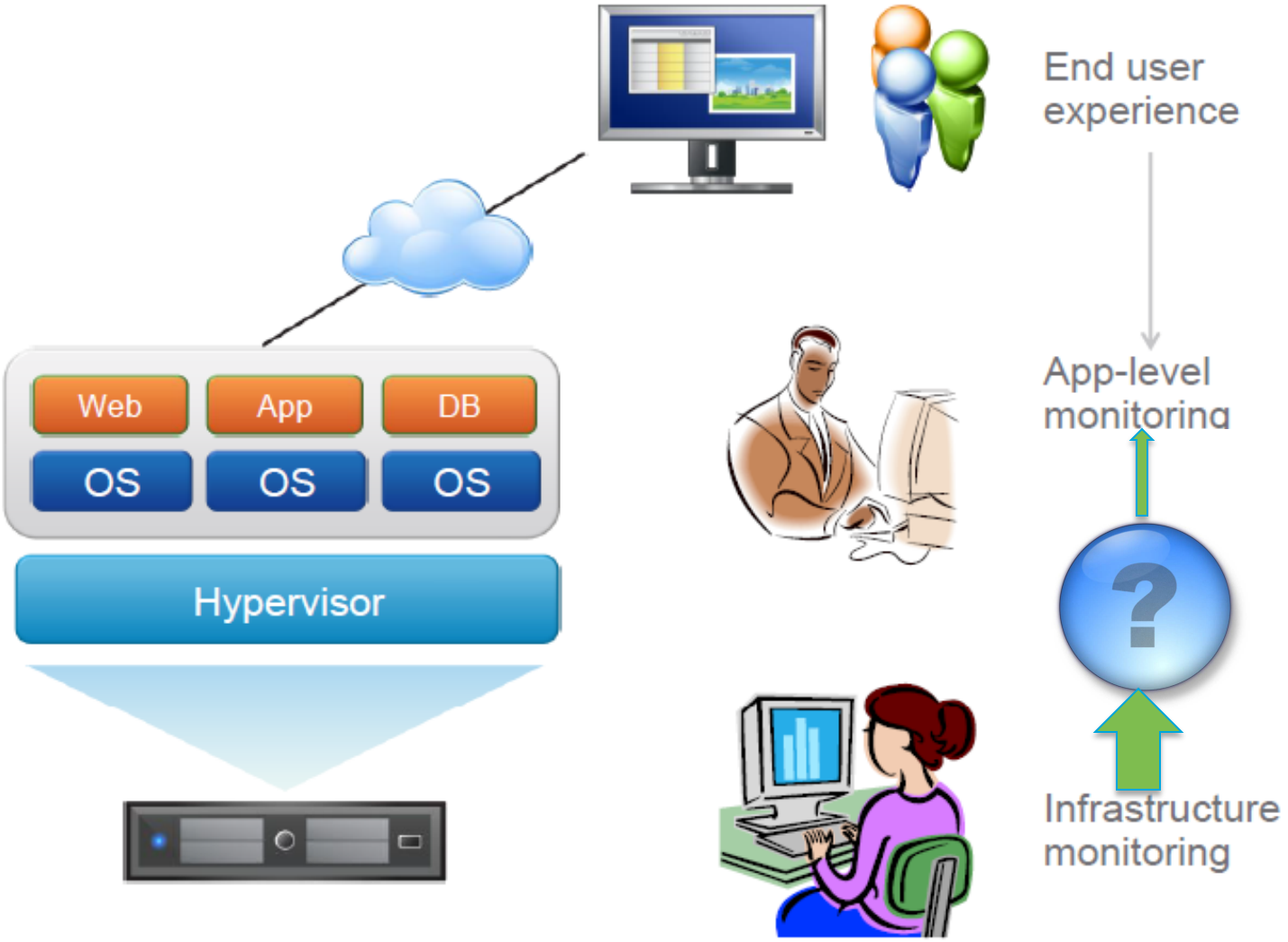
- Response times
- Throughput (or errors such as timed out requests)

### VMware Hyperic monitoring tool

- Agents deployed in VMs
- Auto-discovers types of applications running
- Plugins to extract application-related performance stats
- Stats available at a time scale of minutes
- Stats aggregated in Hyperic server
- Supports over 80 different application components
- Extensible framework to allow customized plugins

# The Semantic Gap challenge

## Correlating performance data from different sources



# Semantic gap filled by performance models

Traditional models harder to apply

- **First-principle models**: Only exist for special cases (e.g., flow models)
- **Queuing models**: More suitable for aggregate/average behavior
- **Architectural models**: Require domain knowledge, harder to automate

Empirical models via statistical learning

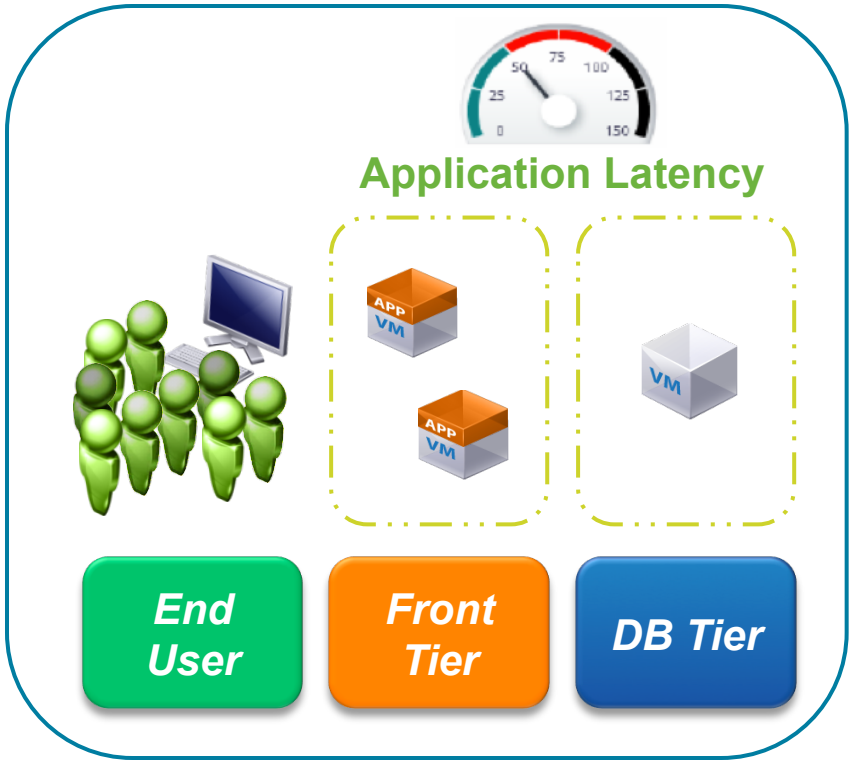
- Data driven, easier to automate and scale
- **Offline modeling** usually insufficient
  - Time-varying workloads
  - Changing system/software configurations
- **Online modeling**
  - Need to be low overhead

# Learning helps answer key modeling questions

- **Q1:** Which variables go into the model?
  - Which system resources or parameters affect application performance the most?
- **Q2:** What kind of model should we use?
  - Nonlinear models - better accuracy in general
  - Linear regression models - cheaper to compute and easier to interpret
- **Q3:** How to ensure our model captures recent behavior?
  - Continuous online adaptation
  - Online change-point detection

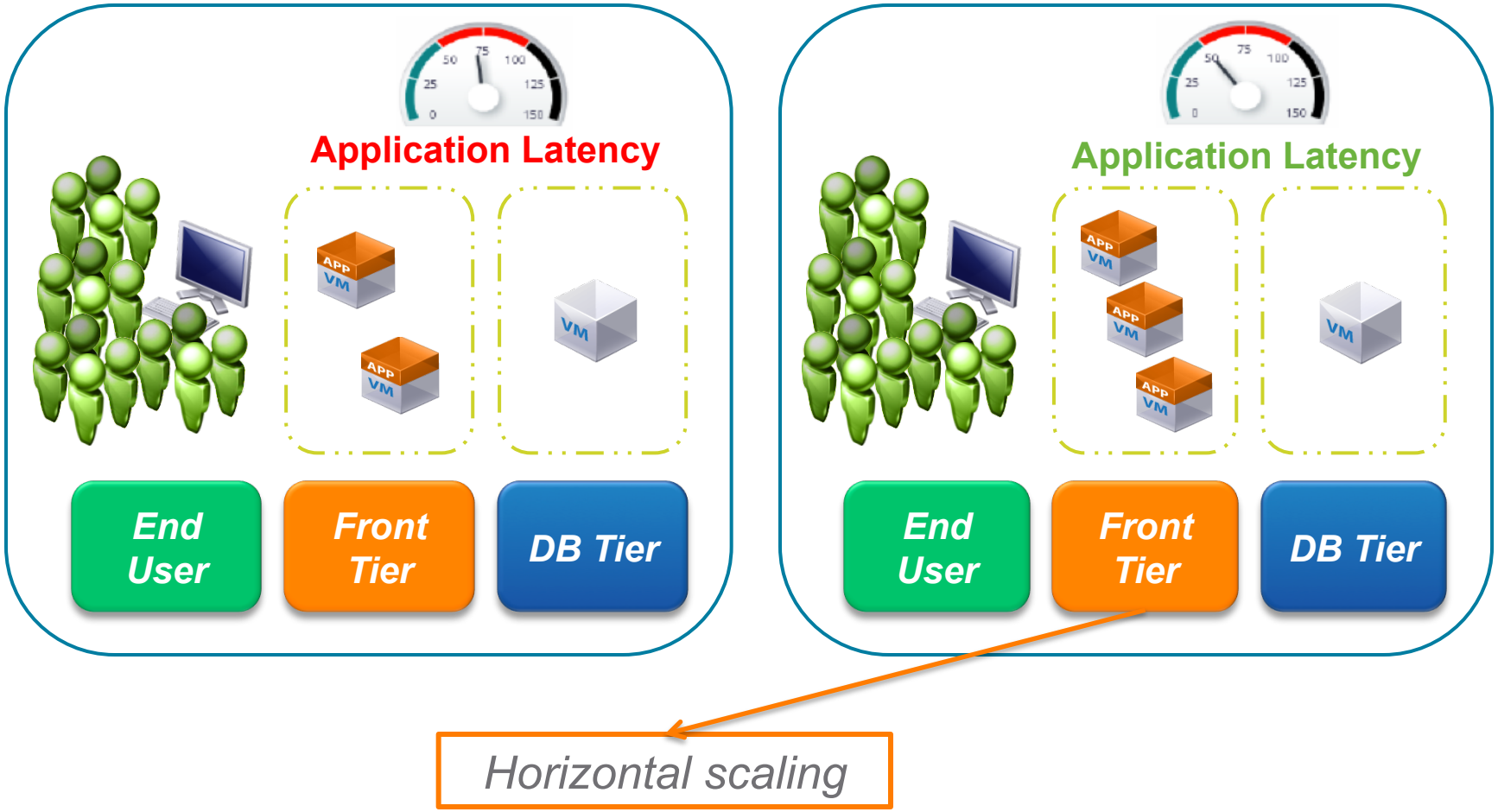
# Auto-Scaling to maintain application SLO

## A feedback-control approach



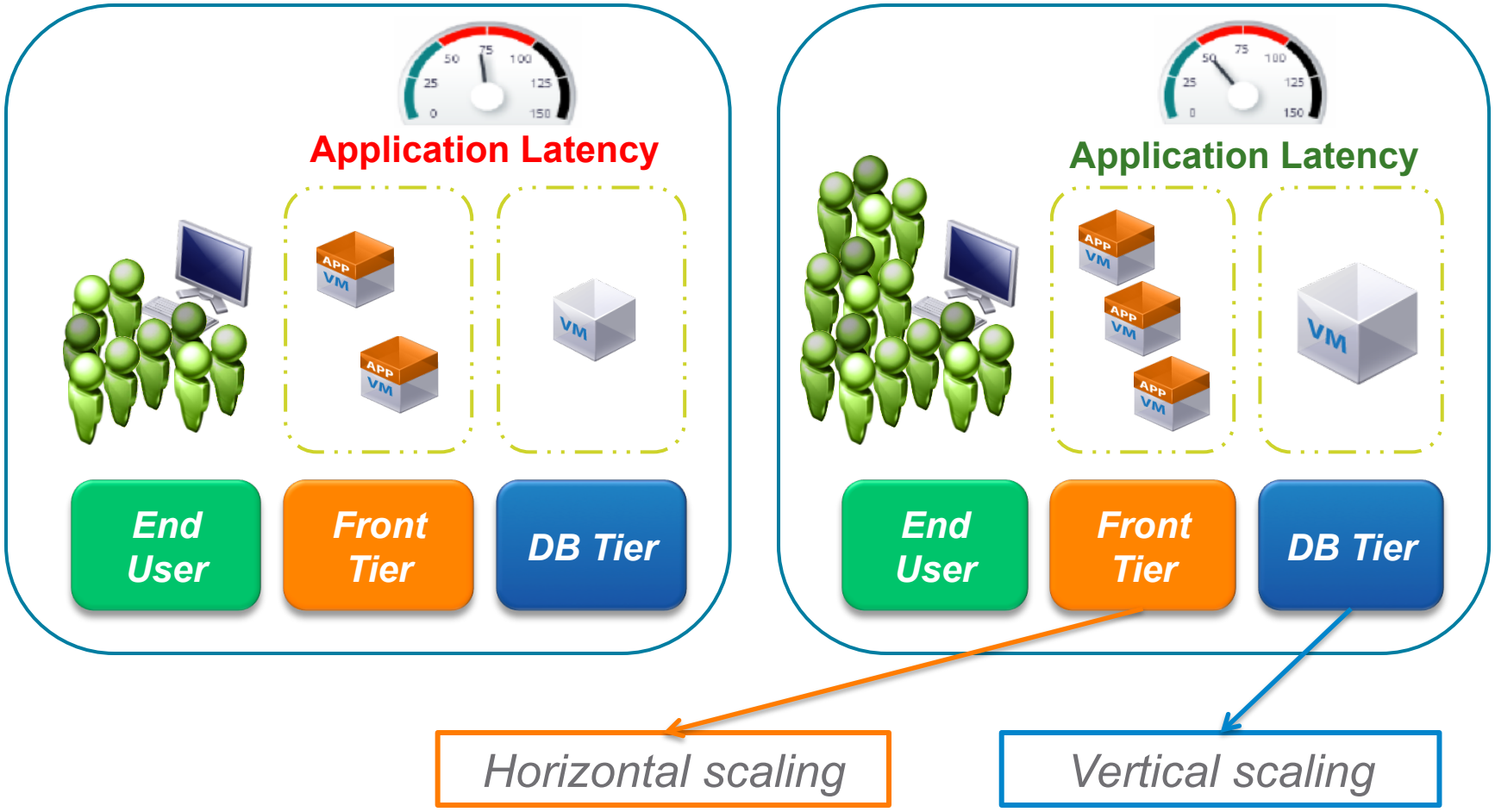
# Auto-Scaling to maintain application SLO

## A feedback-control approach



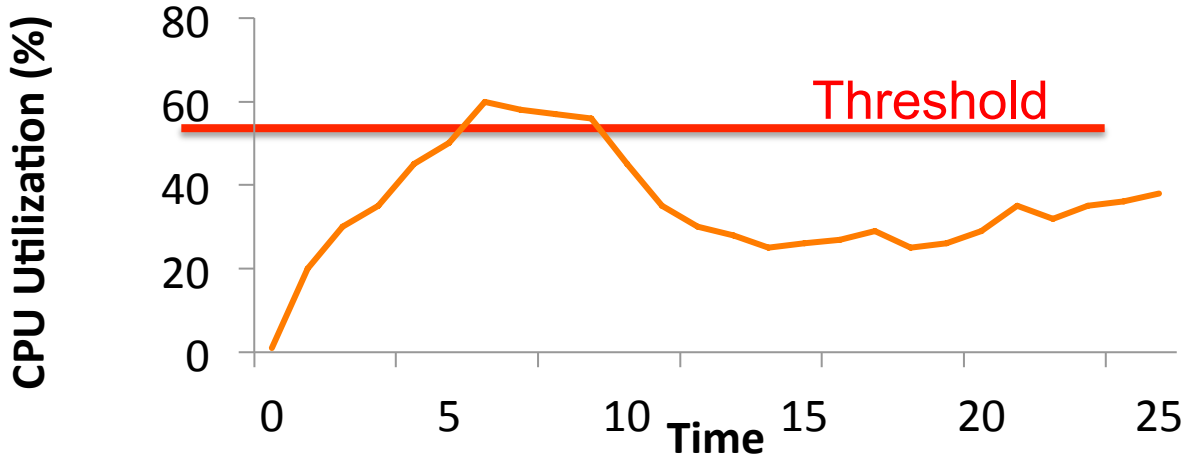
# Auto-Scaling to maintain application SLO

## A feedback-control approach



# Existing solutions to horizontal scaling
## Threshold-based approach

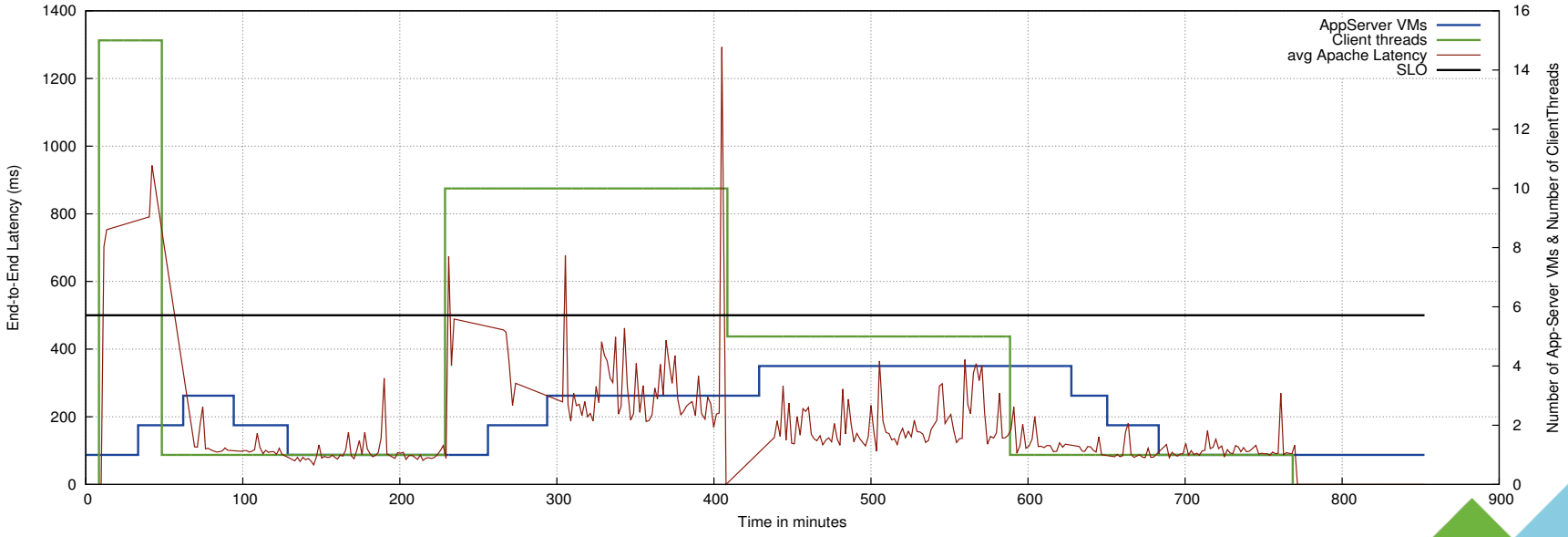
- User-defined threshold on a specific metric
  - Spin up new instances when threshold is violated
  - e.g. AWS Auto Scaling: http://aws.amazon.com/autoscaling/




- **Challenges**
  - How to determine the threshold value?
  - How to handle multiple application tiers?
  - How to handle multiple resources?

# Our Solution: Learning-based auto scaling

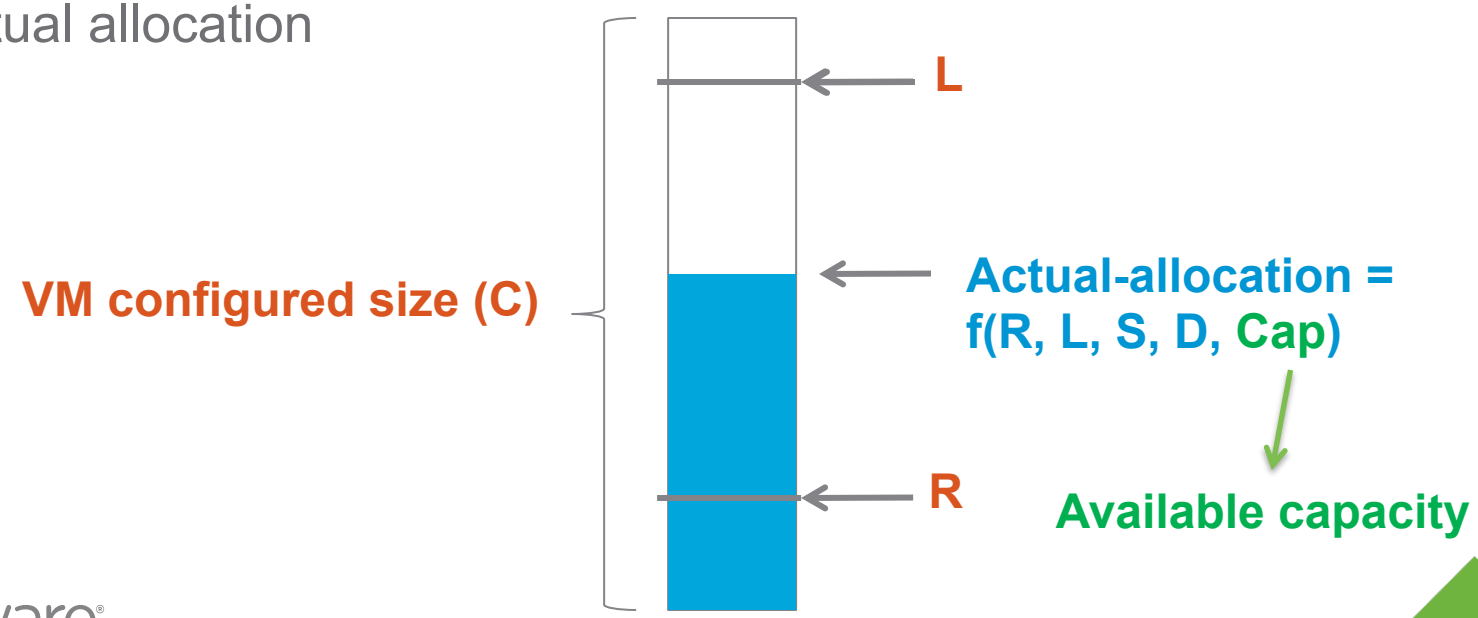
- Uses reinforcement learning to capture application's scaling behavior and inform future actions
- Uses heuristics to seed the learning process
- Handles multiple resources and tiers
- Fully automated without human intervention




vmware

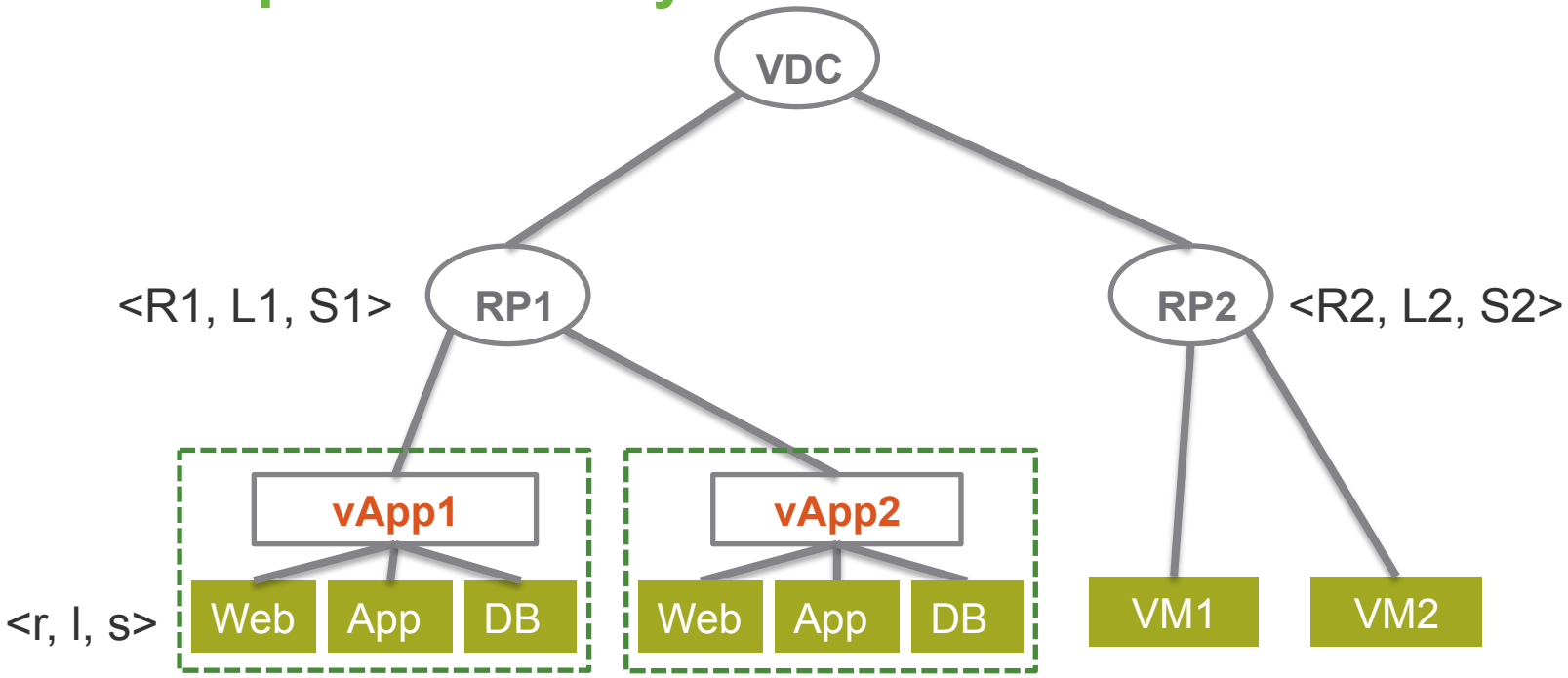
# Vertical scaling of resource containers
## Automatic tuning of resource control settings

- Available on various virtualization platforms
- For shared CPU, memory, disk I/O*, network I/O*:
  - ***Reservation (R)**** – minimum guaranteed amount of resources
  - ***Limit (L)*** – upper bound on resource consumption (non-work-conserving)
  - ***Shares (S)*** – relative priority during resource contention
- VM's CPU/memory *demand (D)*: estimated by hypervisor, critical to actual allocation



# DRS (Distributed Resource Scheduler)

## Resource pool hierarchy

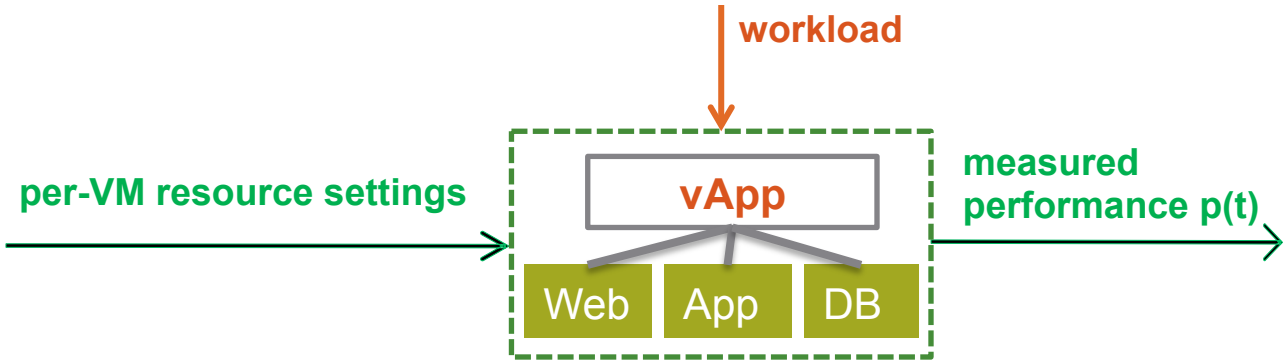



- Capacity of an RP divvied hierarchically based on resource settings
- Sibling RPs share capacity of the VDC
- Sibling VMs share capacity of the parent RP

** VMware distributed resource management: Design, implementation, and lessons learned, VMware Technical Journal, April 2012.*

# Powerful knobs, hard to use

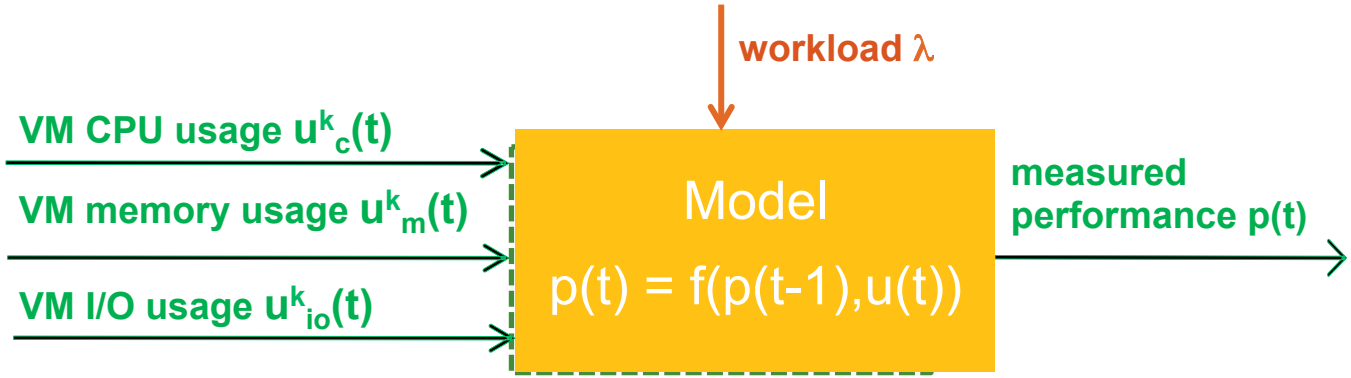
- How do VM-level settings impact application performance?
- How to set RP-level settings to protect high priority applications within the RP?
- Fully reserved (R=L=C) for critical applications
  - Leads to lower consolidation ratio due to admission control
- Others left at default (R=0, L=C) until performance problem arises
  - Increases reservation for the bottleneck resource (which one? by how much?)



# Performance model learned for each vApp

Maps VM-level resource allocations to app-level performance

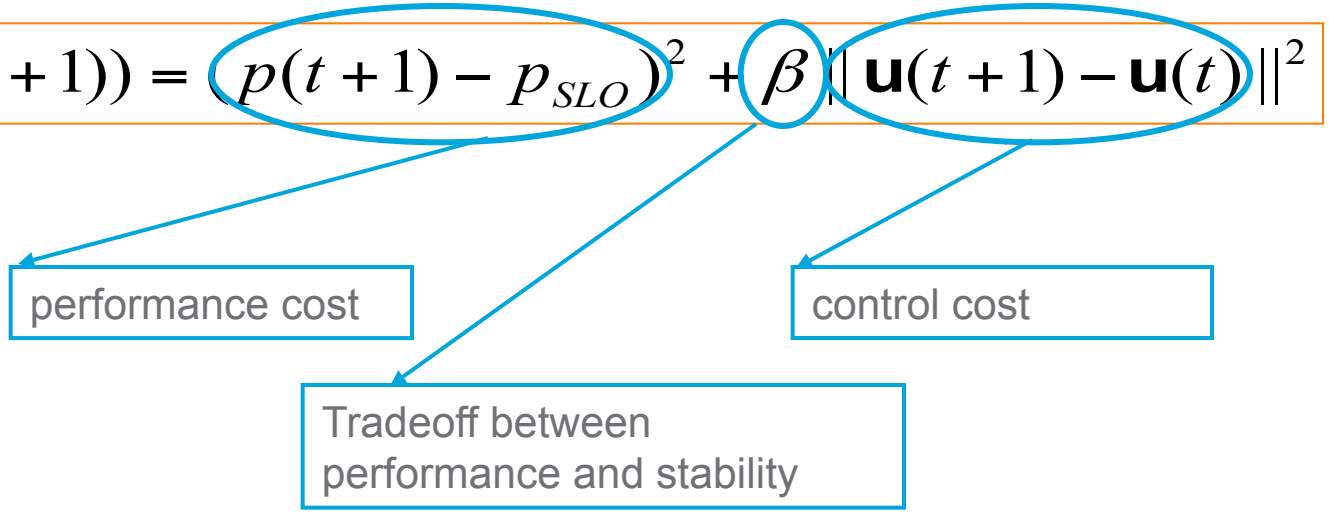
- Captures multiple tiers and multiple resource types
- Choose a linear low-order model (easy to compute)
- Workload indirectly captured in model parameters
- Model parameters updated online in each interval (tracks nonlinearity)



# Simplified optimal control law

- An example cost function

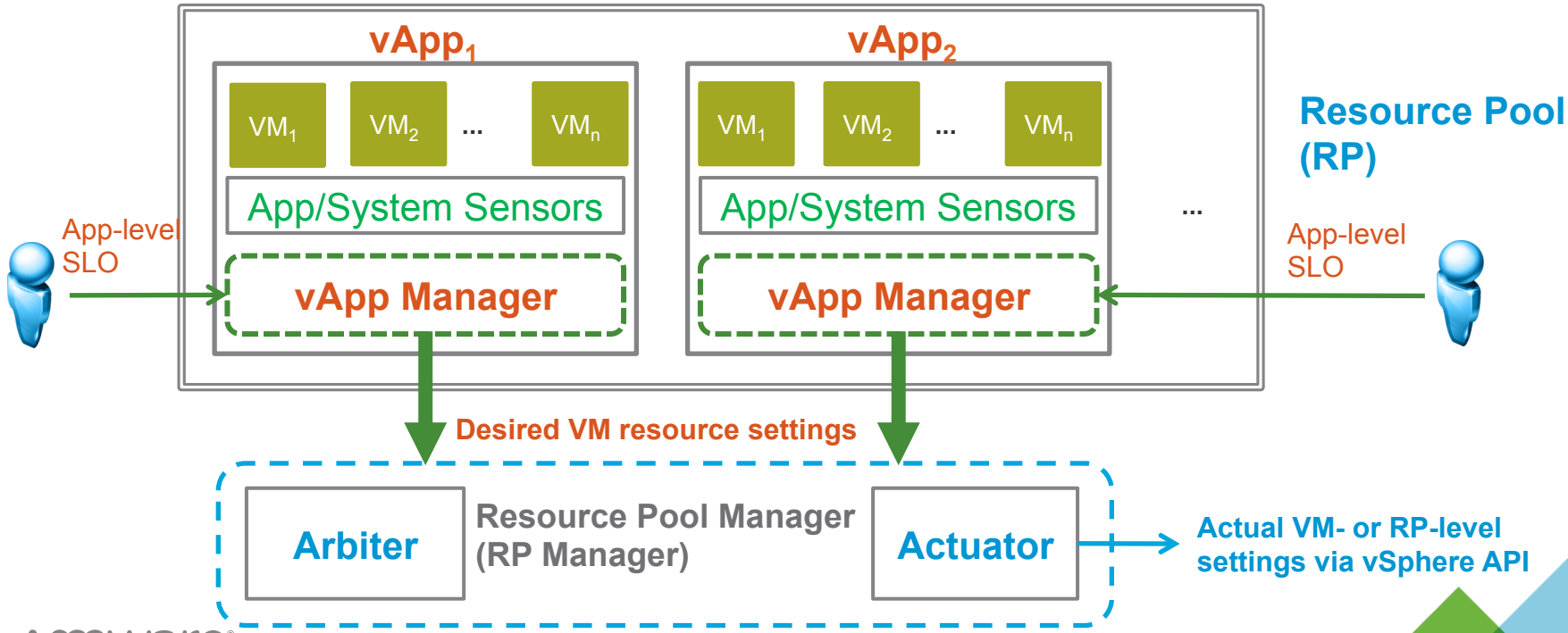
$$J(\mathbf{u}(t+1)) = (p(t+1) - p_{SLO})^2 + \beta \| \mathbf{u}(t+1) - \mathbf{u}(t) \|^2$$




- Compute optimal resource allocations online

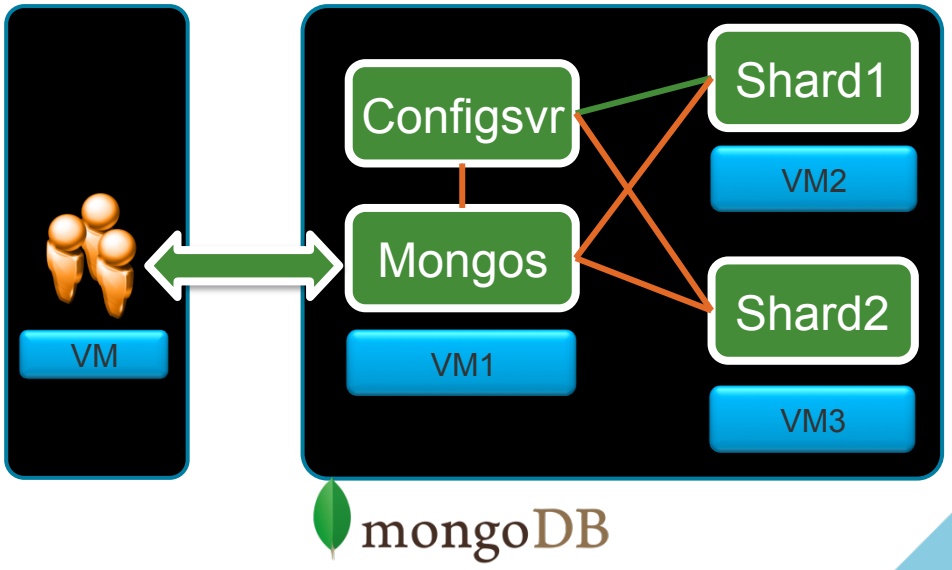
# Resource pool sharing among multiple vApps

- Auto-tunes VM-level and RP-level resource control settings to meet application SLOs
  - For each application, **vApp Manager** translates its SLO into ***desired*** resource control settings at individual VM level
  - For each resource pool, **RP Manager** computes the ***actual*** VM- and RP-level resource settings to satisfy all critical applications



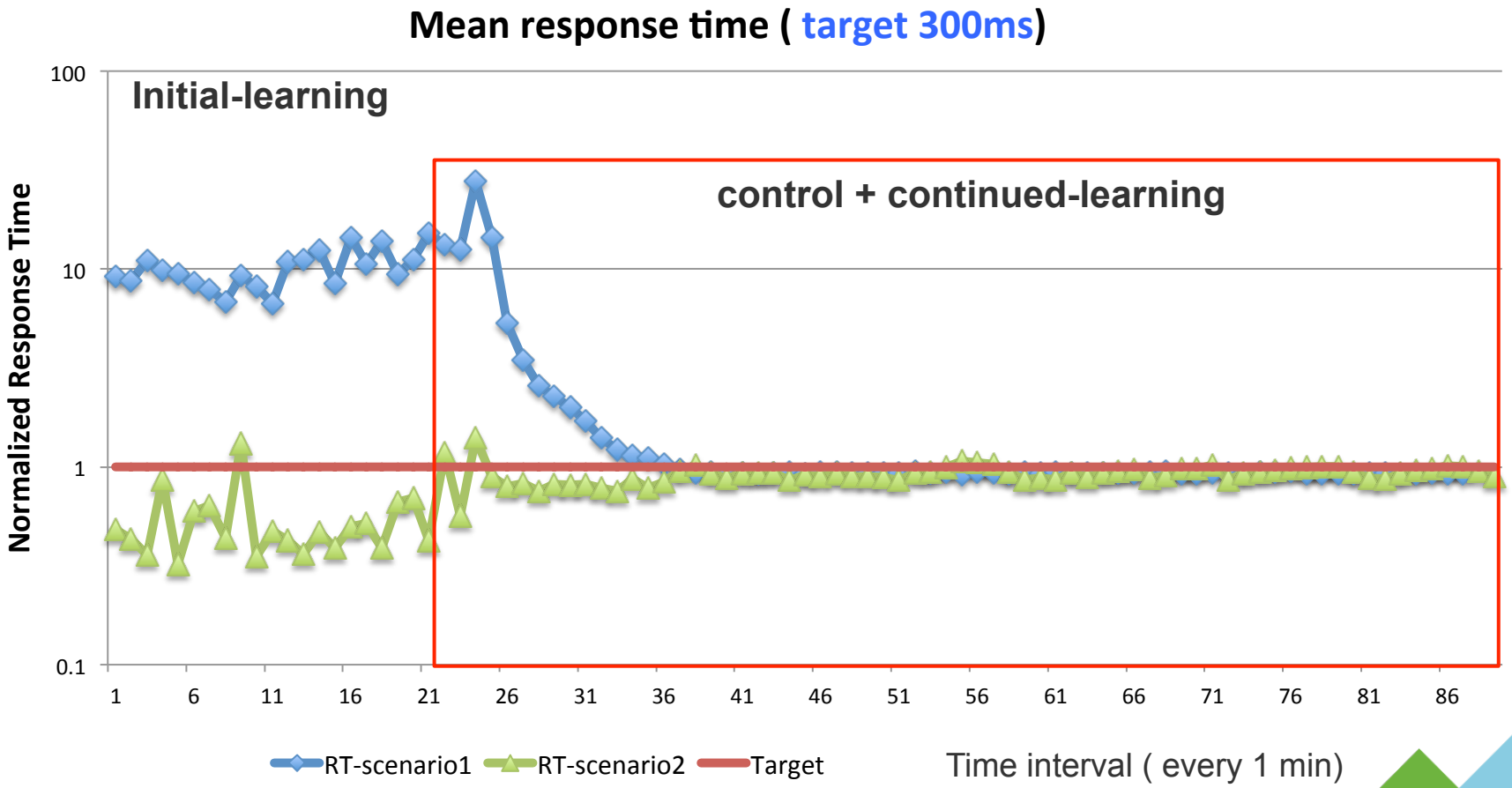
# Performance evaluation

- Application
  - **MongoDB** – distributed data processing application with sharding
  - **Rain** – workload generation tool to generate dynamic workload
- Workload
  - Number of clients
  - Read/write mix
- Evaluation questions
  - Can the vApp Manager meet individual application SLO?
  - Can the RP Manager meet SLOs of multiple vApps?



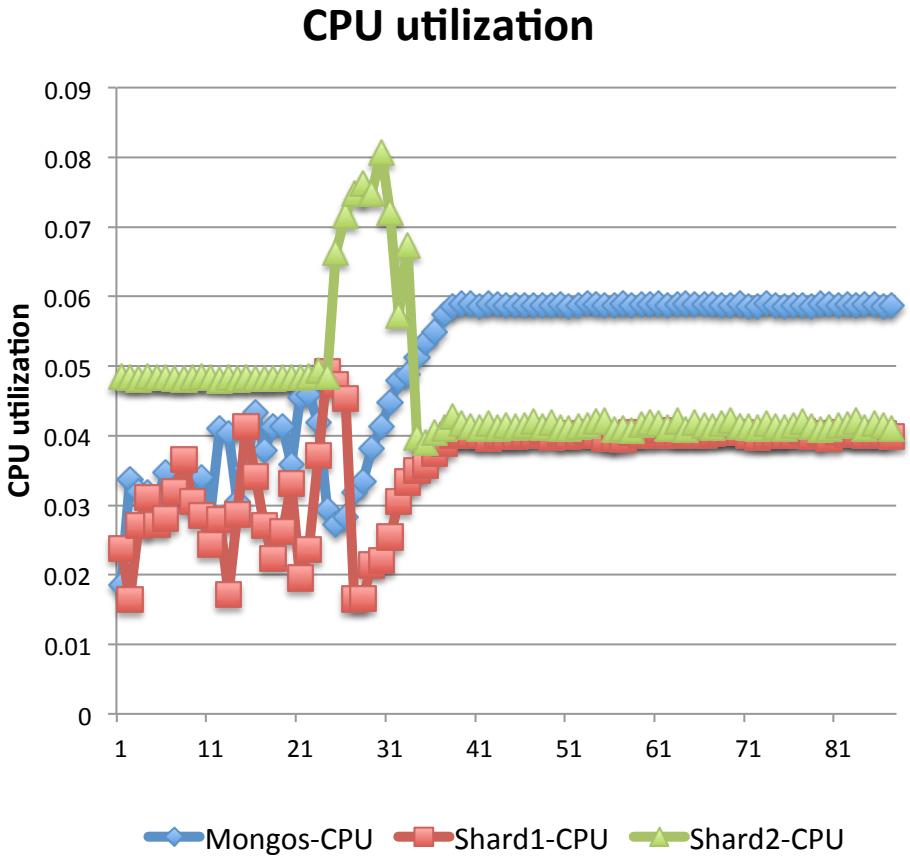
# Result: Meeting mean response time target

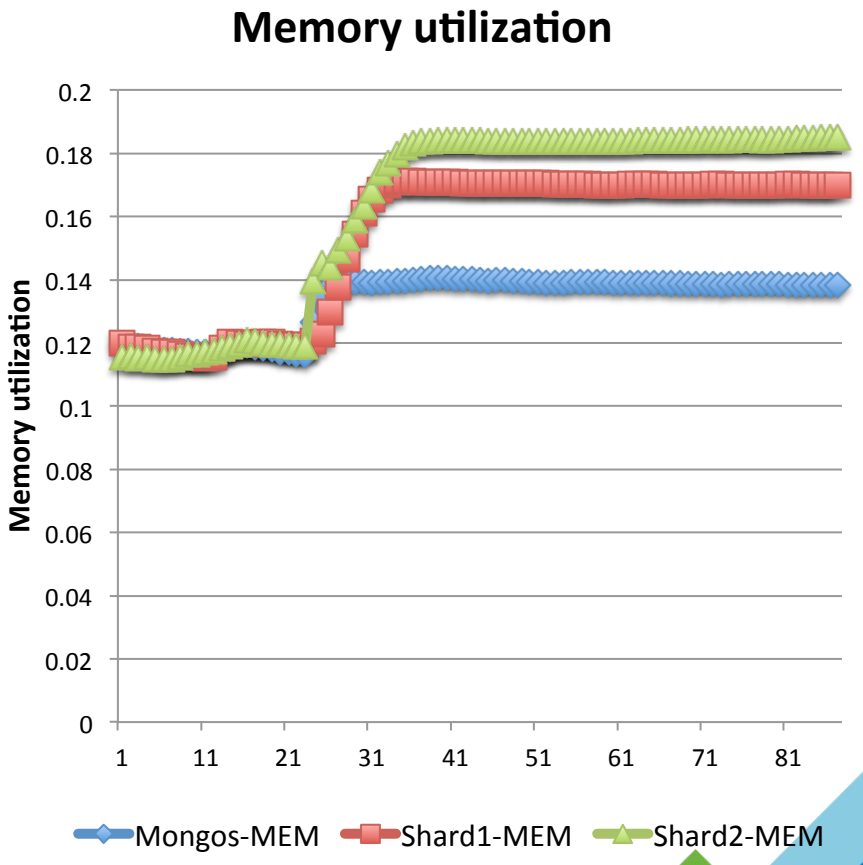
- Under-provisioned initial settings: R = 0, Limit = 512 (MHz, MB)
- Over-provisioned initial settings: R = 0, L = unlimited (cpu, mem)



# Resource utilization (under-provisioned case)

- Target response time = 300 ms
- Initial setting R = 0, L = 512 MHz/MB (under-provisioned)




vmware

# Grand challenge

**The Vision of Autonomic Computing, IEEE Computer, Jan. 2003.**

*"Systems manage themselves according to an administrator's goals. New components integrate as effortlessly as a new cell establishes itself in the human body. These ideas are not science fiction, but elements of the grand challenge to create self-managing computing systems."*

Enablers

- Widely deployed sensors and lots of (noisy) data
- New control knobs, resource fungibility and elasticity
- Increasing compute, storage, and network capacity
- Matured learning, control, and optimization techniques

Challenges

- Software complexity, nonlinearity, dependency, scalability
- Automated root-cause analysis, integrated diagnosis & control
- Need more collaborations between control and systems people
- How to teach control theory to CS students?

# Thanks to collaborators


## VMware

• Lei Lu, Rean Griffith, Mustafa Uysal, Anne Holler, Pradeep Padala, Aashish Parikh, Parth Shah

## HP Labs

• Zhikui Wang, Sharad Singhal, Arif Merchant (now Google)

## KIT

• Simon Spinner, Samuel Kounev

## College of William & Mary

• Evgenia Smirni

## Georgia Tech

• Pengcheng Xiong (now NEC Lab), Calton Pu

## University of Michigan

• Kang Shin, Karen Hou


vmware®

# Related venues

- International Conference on Autonomic Computing

  https://www.usenix.org/conference/icac14

- Feedback Computing Workshop (formerly known as FeBID)

  http://feedbackcomputing.org/

  http://www.controlofsystems.org/

- Lund University Cloud Control Workshop (LCCC)

  http://www.lccc.lth.se/index.php?page=Workshop201405Program

# References


- *X. Zhu, et al. "What does control theory bring to systems research?" ACM SIGOPS Operating Systems Review, 43(1), January 2009.*
- P. Padala *et al.* "Automated control of multiple virtualized resources." *Eurosys 2009.*
- A. Gulati *et al.* "VMware distributed resource management: Design, implementation, and lessons learned." *VMware Technical Journal*, Vol. 1(1), April 2012.
- P. Xiong *et al*. "vPerfGuard: An automated model-driven framework for application performance diagnosis in consolidated cloud environments." *ICPE 2013*.
- A. Gulati , "Towards proactive resource management in virtualized datacenters," *RESoLVE 2013.*
- L. Lu, *et al.,* "Application-Driven dynamic vertical scaling of virtual machines in resource pools." *NOMS 2014*.



vmware®