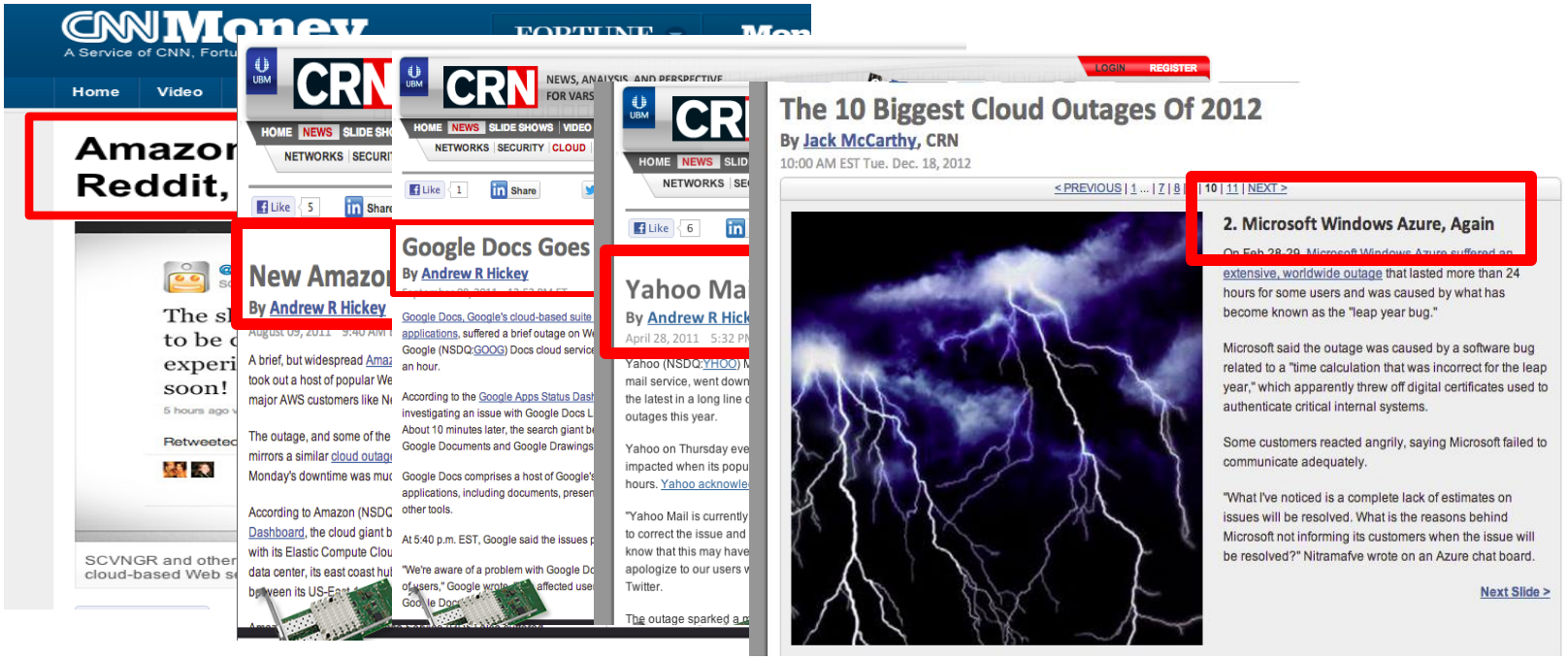
Automatic System Management using Unsupervised Machine Leaning

Xiaohui (Helen) Gu

North Carolina State University

Cloud Computing Challenges

- Robustness



- Resource and energy efficiency
  - Over-provision resources for performance
  - Average Resource utilization is 7-12%

Major Research Projects

- Online system anomaly management
  - [SRDS'16, TPDS'15, ICAC'15, IC2E'15, SOCC'14, HotCloud'14, USENIX ATC'14, ICDCS'13, ICDCS'12 (*best paper award*), ICAC'12, TPDS'12, SLAML'11, SRDS'11, PODC'10, MACOTS'10, ICAC'09, ICDE'09]
  - Sponsored by NSF, ARO, IBM, Google
- Resource and energy efficient cloud computing
  - [ICAC'13, SOCC'11, SOSP'11 (poster), CNSM'10 (*best paper award*), MASCOTS'10 (*best paper nominee*), HPDC'10, SOSP'09 (poster), IWQoS'09]
  - Sponsored by NSF CSR, two Google research awards, two IBM faculty awards
- Security monitoring for cloud and mobile devices
  - [ACM CSUR'16, CODESPY'14, TPDS'13, IWQoS'11, ASIACCS'10, CCS'10 (poster), ACSAC'09, STC'09]
  - Sponsored by NSA and ARO

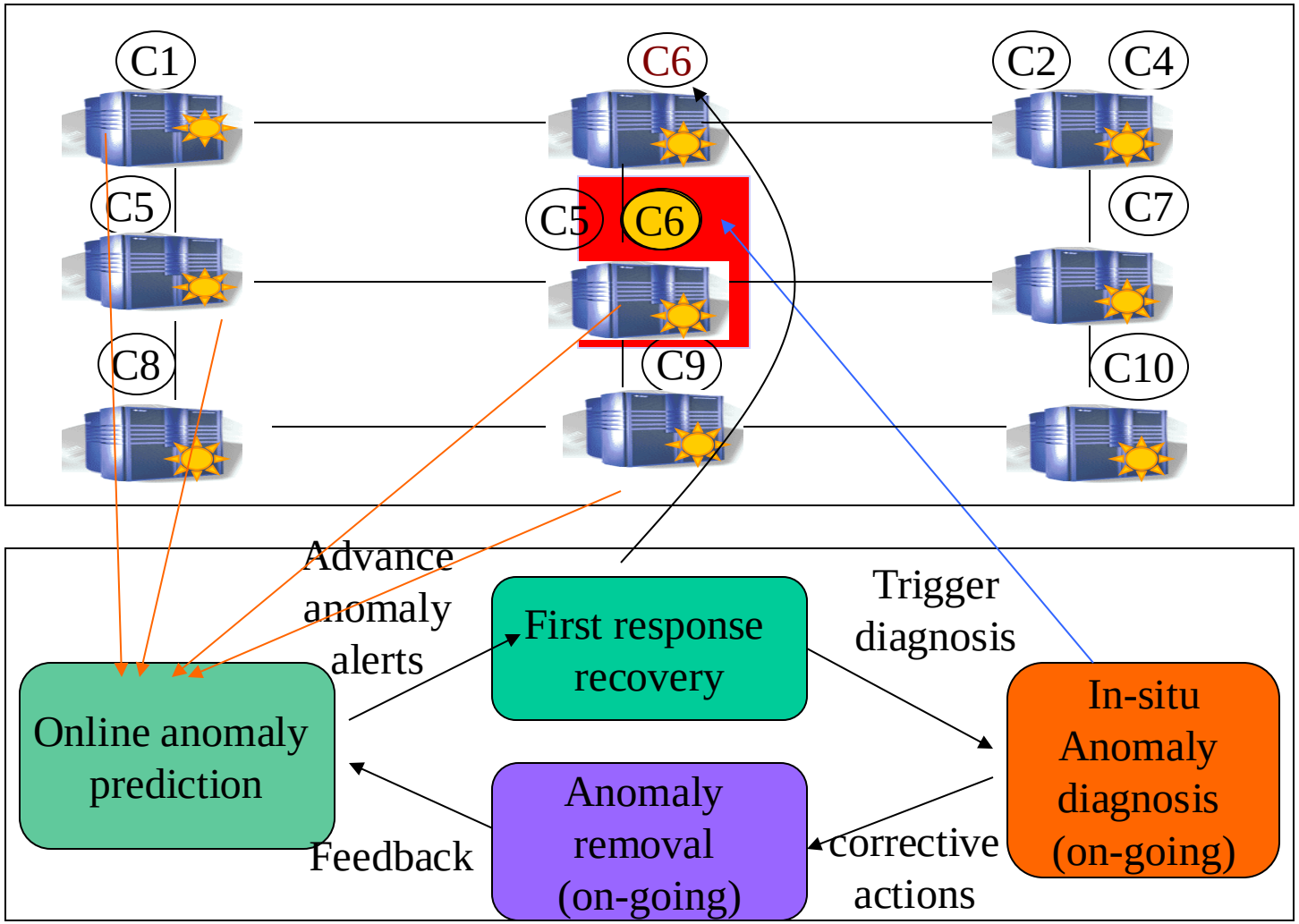
Cloud Computing Challenges

- Robustness
- Resource and energy efficiency
- Accountability

Existing Approaches

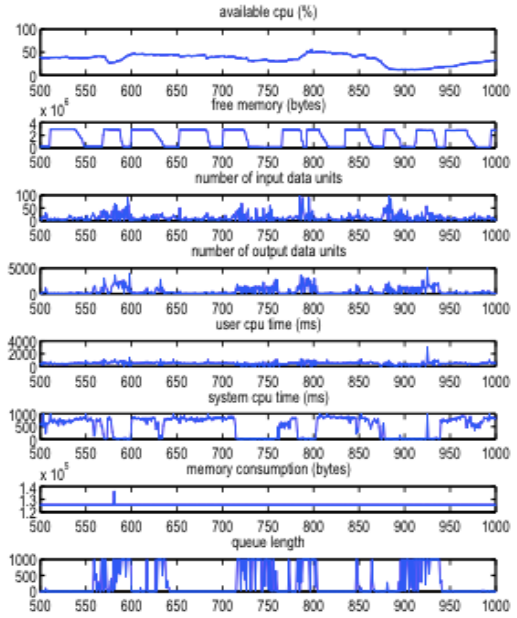
- *Reactive* approaches
  - Take corrective actions *after* an anomaly happens
  - No prevention cost but prolonged service downtime
  - Difficult to reproduce anomaly-inducing environments
- *Proactive* approaches
  - Take preventive actions on *all* system components
  - Provide better system reliability but incur large overhead
  - Do not know the underlying reasons

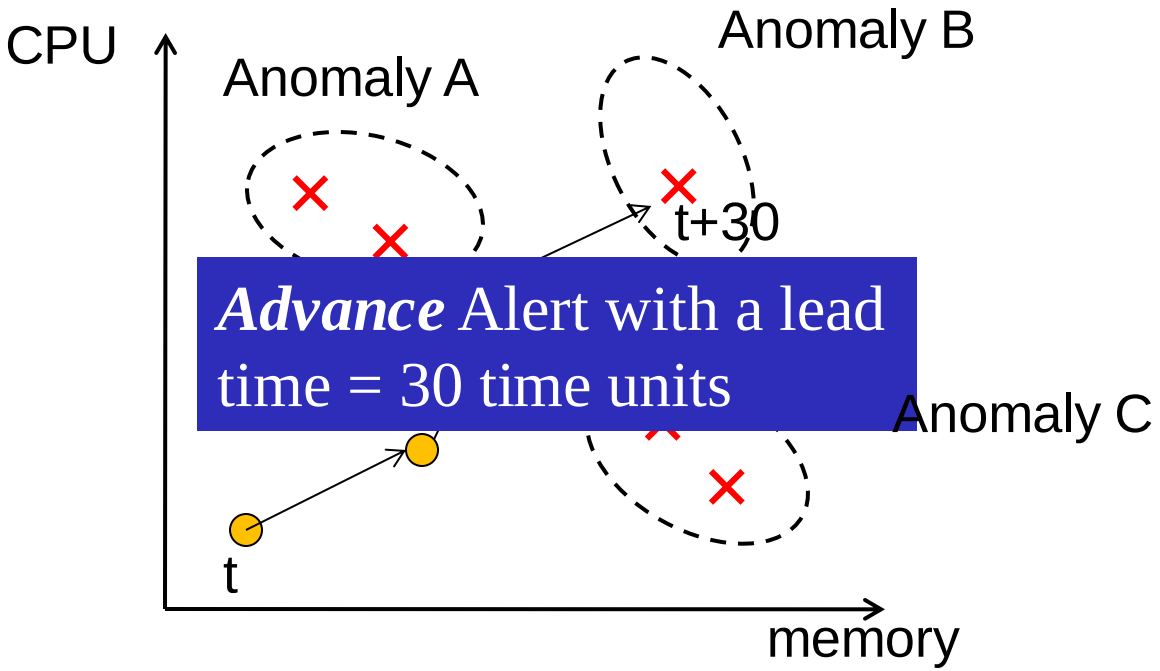
Predictive Online System Anomaly Management



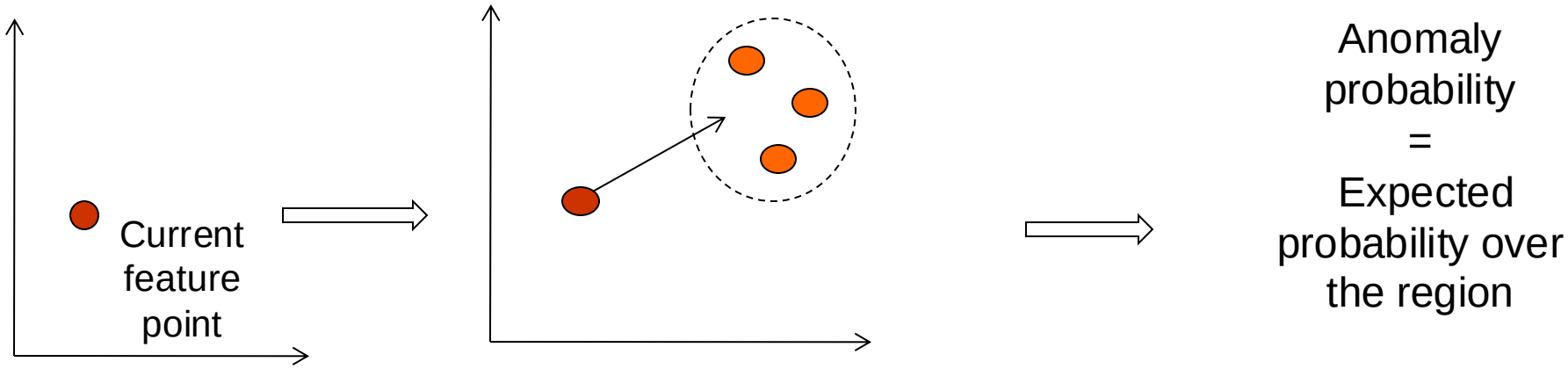
Online System Anomaly Prediction

- Performance Anomalies
  - SLO violations: response time > 50 ms
- Multivariate predictions
  - System level metrics: CPU, memory, network, disk I/O





Classification Over Future Data

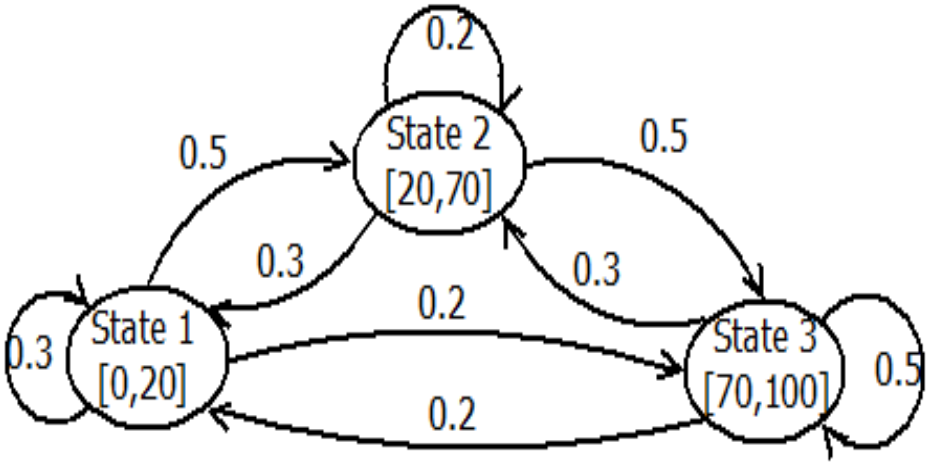


Feature value prediction
- Discrete-time Markov Chain model
- 2-dependent Markov model
- Wavelet predictions

Expected posterior Probability
- Naïve Bayesian
- Neuron Network

Feature Value Prediction

- Predict the metric value distribution at a future time
- Discrete-time Markov chain (DTMC)



CPU usage (%) ranging from 0 to 100 with three discrete states

$$\begin{bmatrix} p_{1,1}^{t} & p_{1,2}^{t} & p_{1,3}^{t} \\ p_{2,1}^{t} & p_{2,2}^{t} & p_{2,3}^{t} \\ p_{3,1}^{t} & p_{3,2}^{t} & p_{3,3}^{t} \end{bmatrix}$$

Multi-step state transition matrix e.g. for t-step (i.e. lead time t):

Wavelet-based Medium-Term Prediction

16 iterations

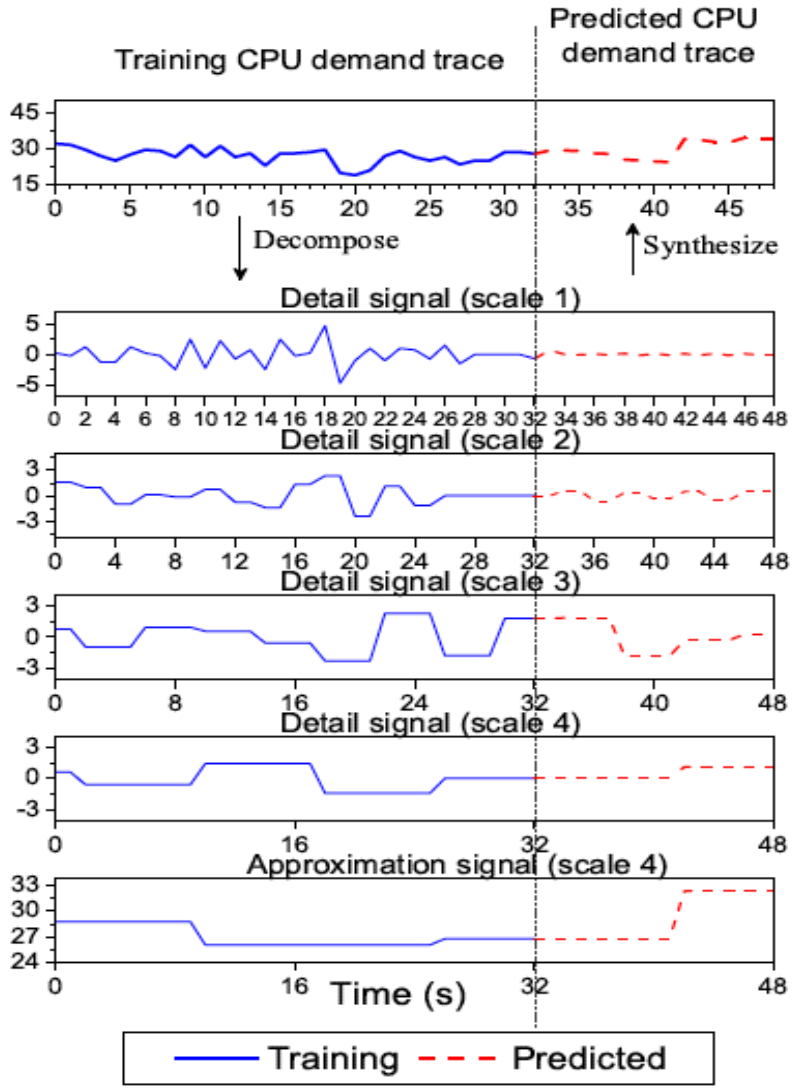
8 iterations

4 iterations

2 iterations

1 iterations

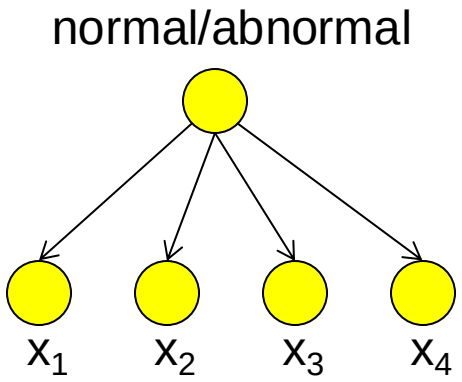
1 iterations



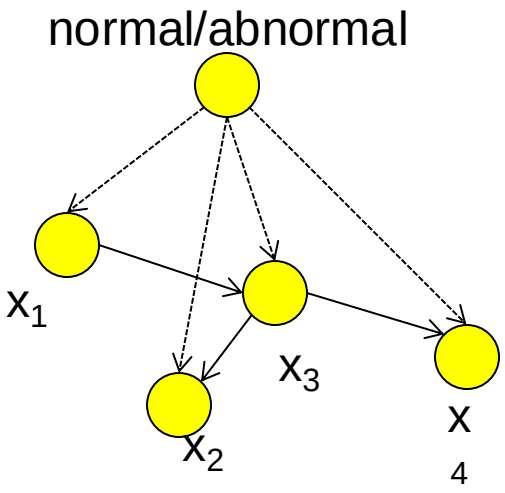
Statistical Anomaly Classification

- Naïve Bayesian classifier
  - Posterior probabilities can be reduced to conditional probabilities
  - Each metric is independent

$\mathbf{x} = [x_1, x_2, x_3, x_4]$



- Tree augmented Bayesian (TAN) network
  - Consider inter-metric dependencies
  - At most one parent other than the class label

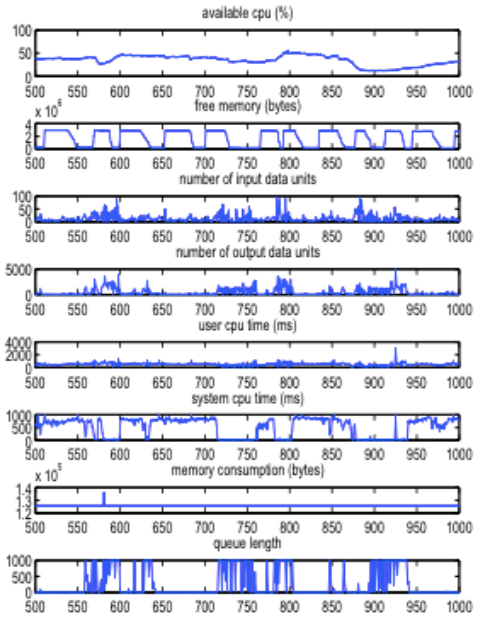


- Classification rule

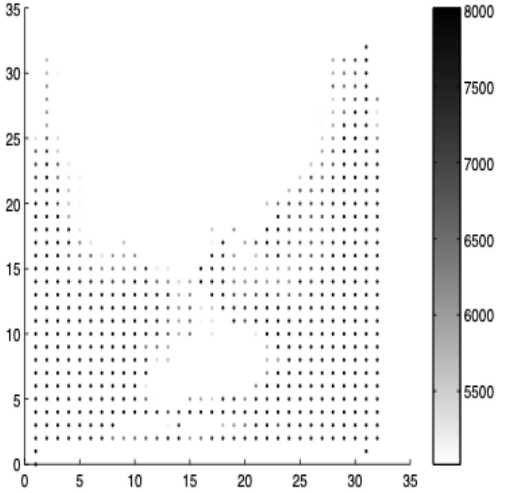
$$\log P(\text{“abnormal”}|\mathbf{x}) - \log P(\text{“normal”}|\mathbf{x}) > \delta$$

Unsupervised Behavior Learning (UBL)

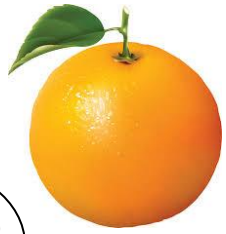
- Support online learning and prediction
  - linear time learning and constant time prediction
  - Incremental training
- Provide hints on root causes



Apache web server



UBL Learning



Neuron 1

Neuron 2

Color = yellow, Shape = long

Color = orange, Shape = round

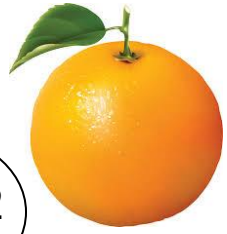
Neuron 3

Neuron 4

Color = green, Shape = round

Color = red, Shape = round

Anomaly Detection and Fault Inference



Neuron 1

Neuron 2

Color = yellow, Shape = long

Color = orange, Shape = round

Neuron 3

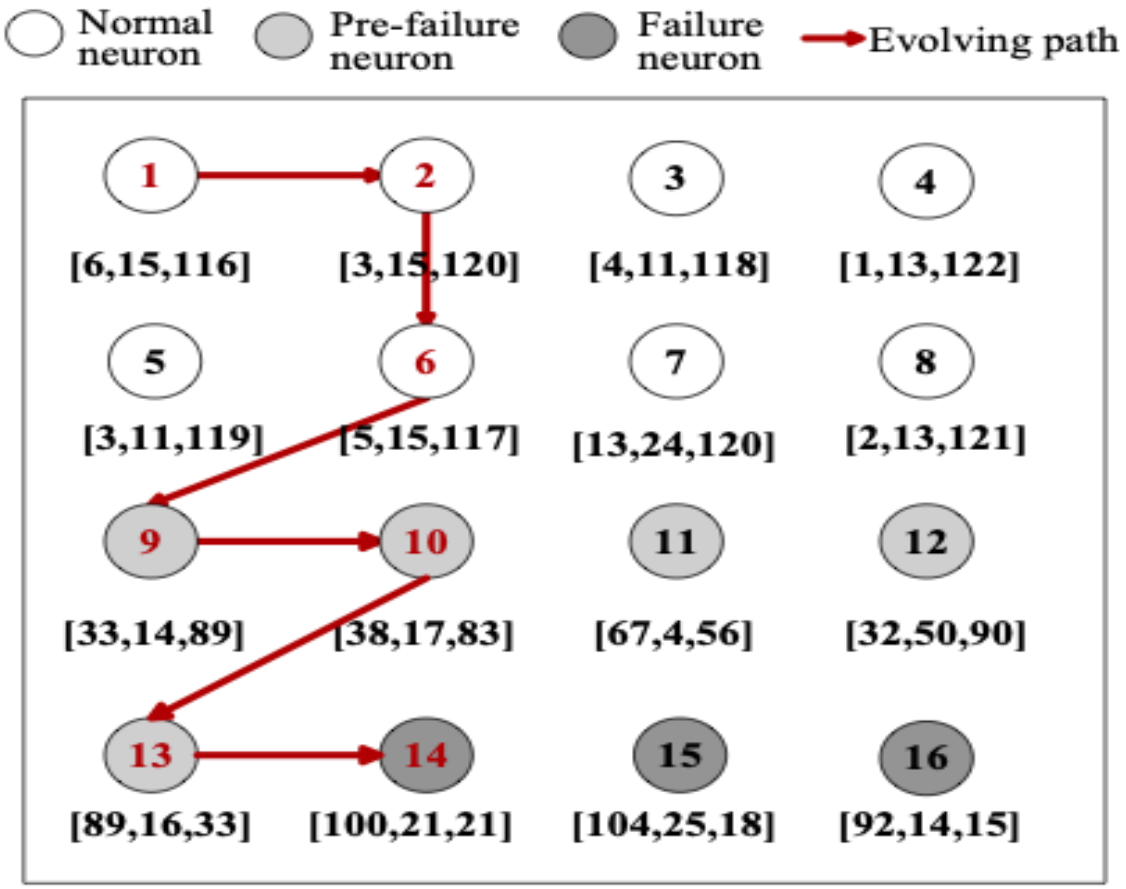
Neuron 4

Color = green, Shape = round

Color = red, Shape = round

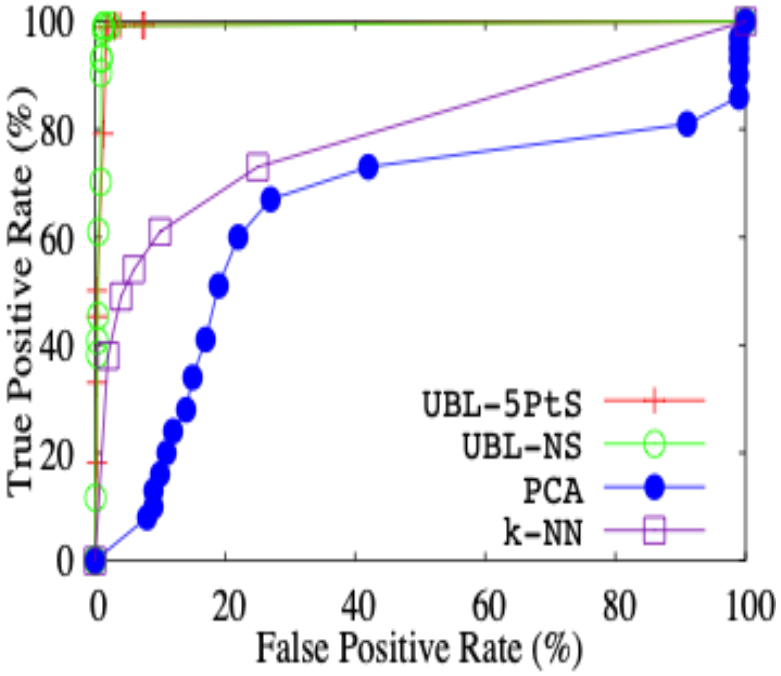
Real Server Example

CPU, memory, network		
6	15	116
3	15	119
5	16	117
33	14	90
38	20	80
90	15	30
100	20	20

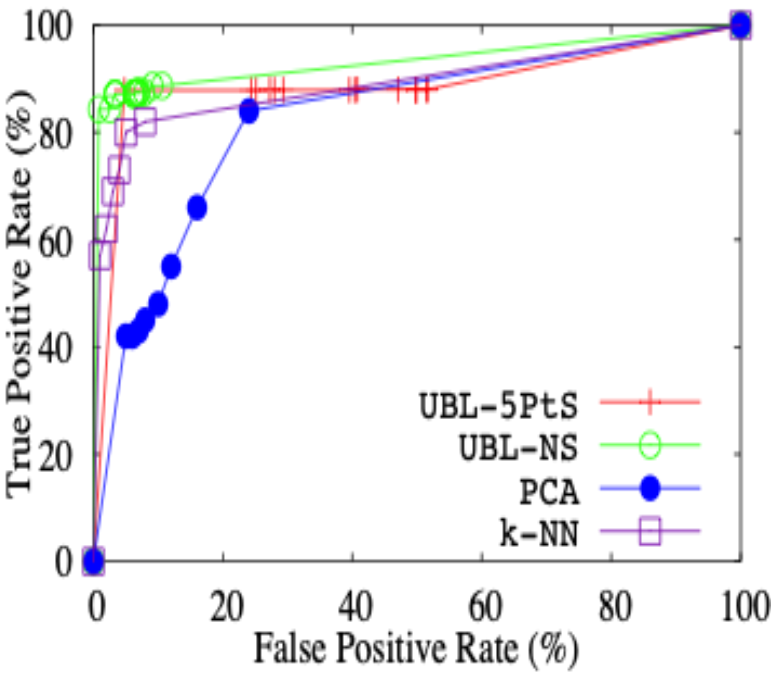


Network congestion in the Apache web server

UBL Prediction Results

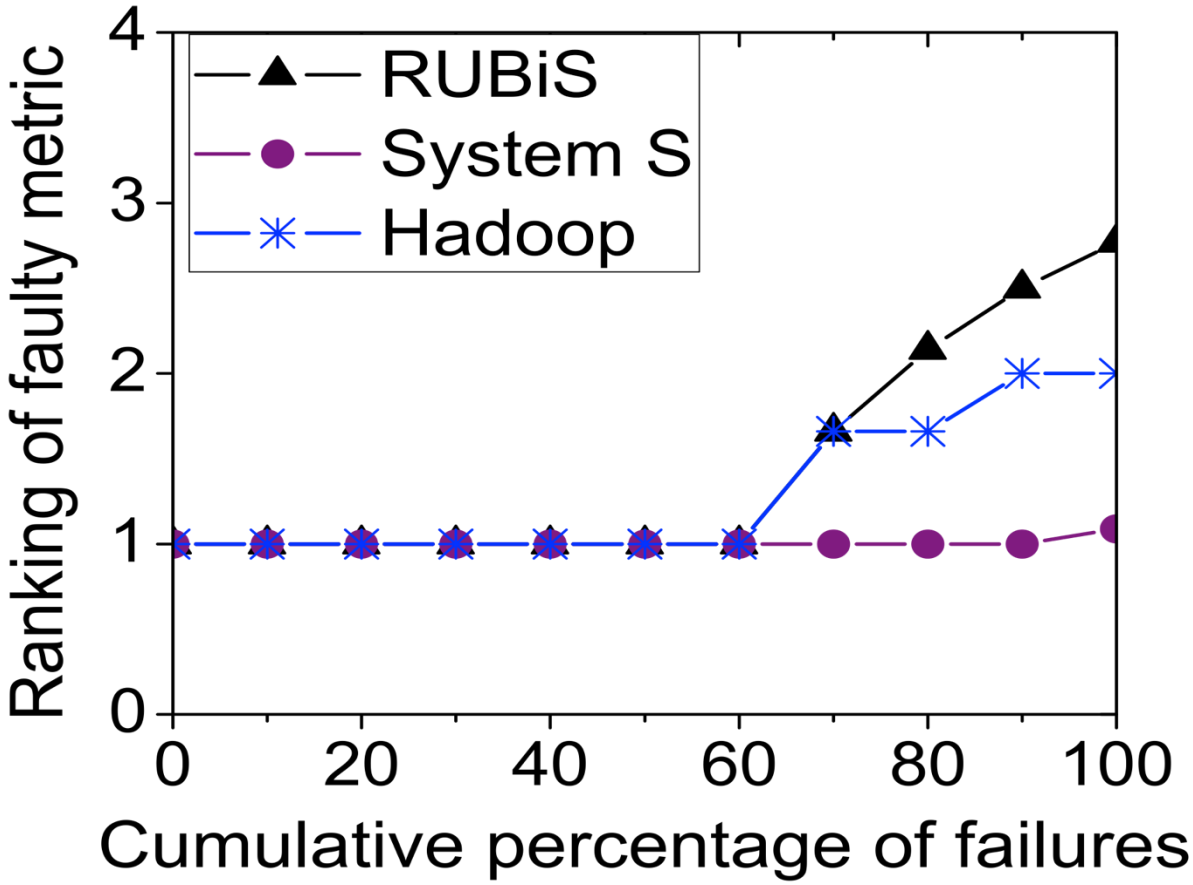


Memory leak in System S

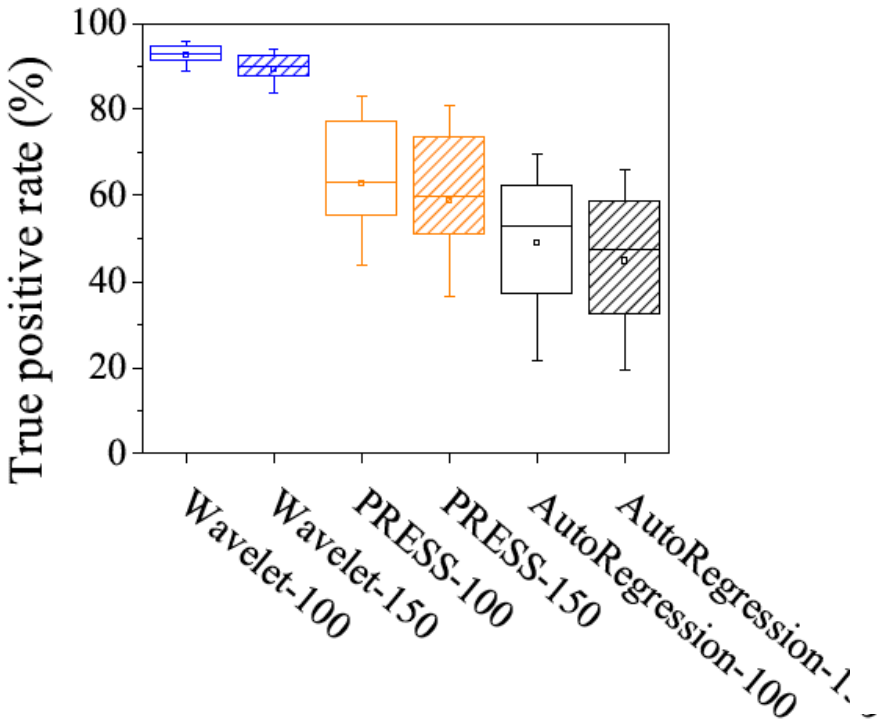


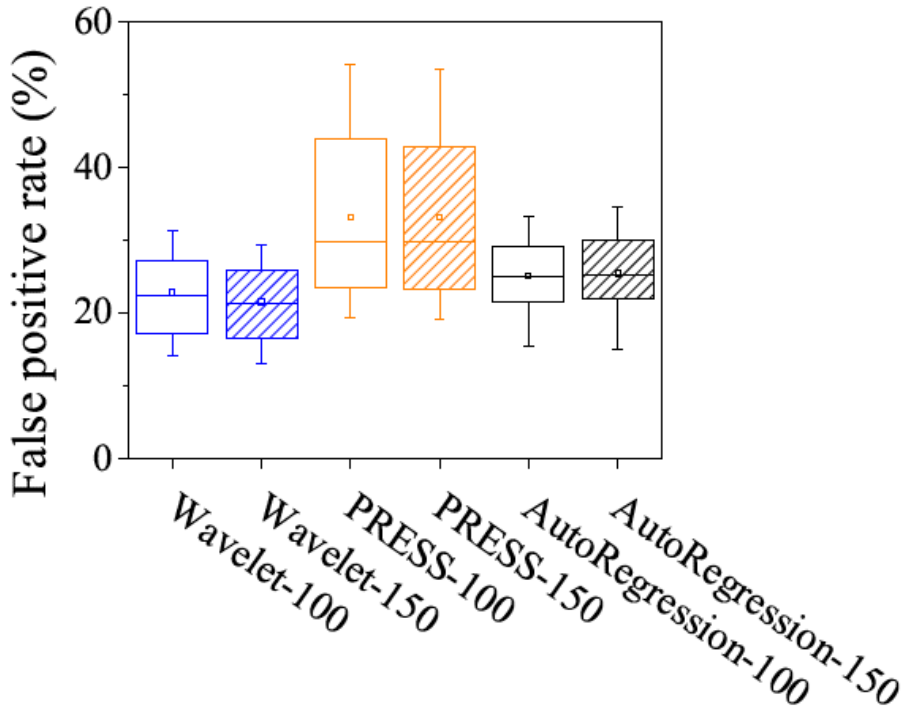
NetHog in RUBiS

Anomaly Cause Inference Results



Medium-Term Overload prediction accuracy

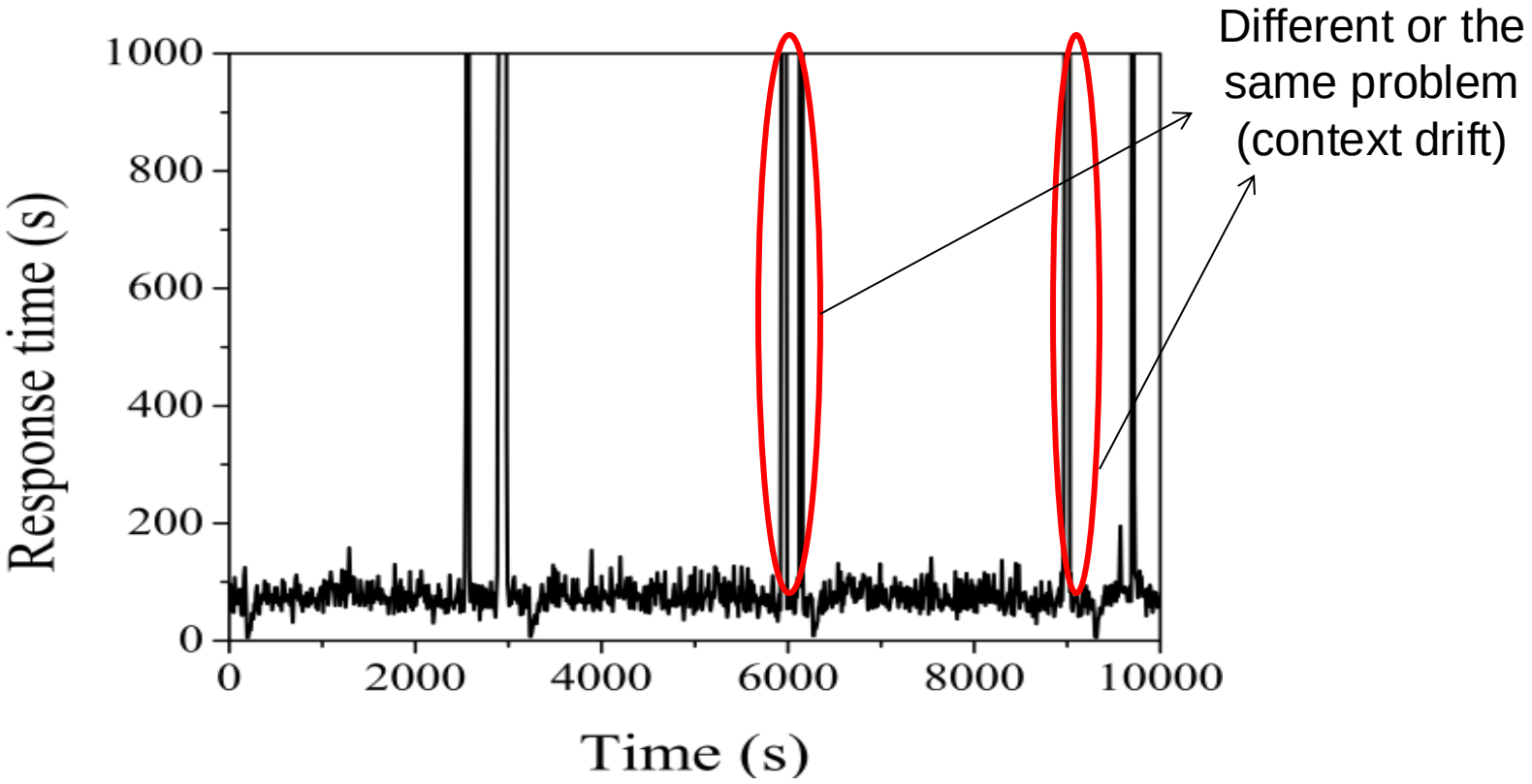




Google CPU traces

Prediction for Dynamic Systems

- Dynamic workloads
- Interference from co-located applications

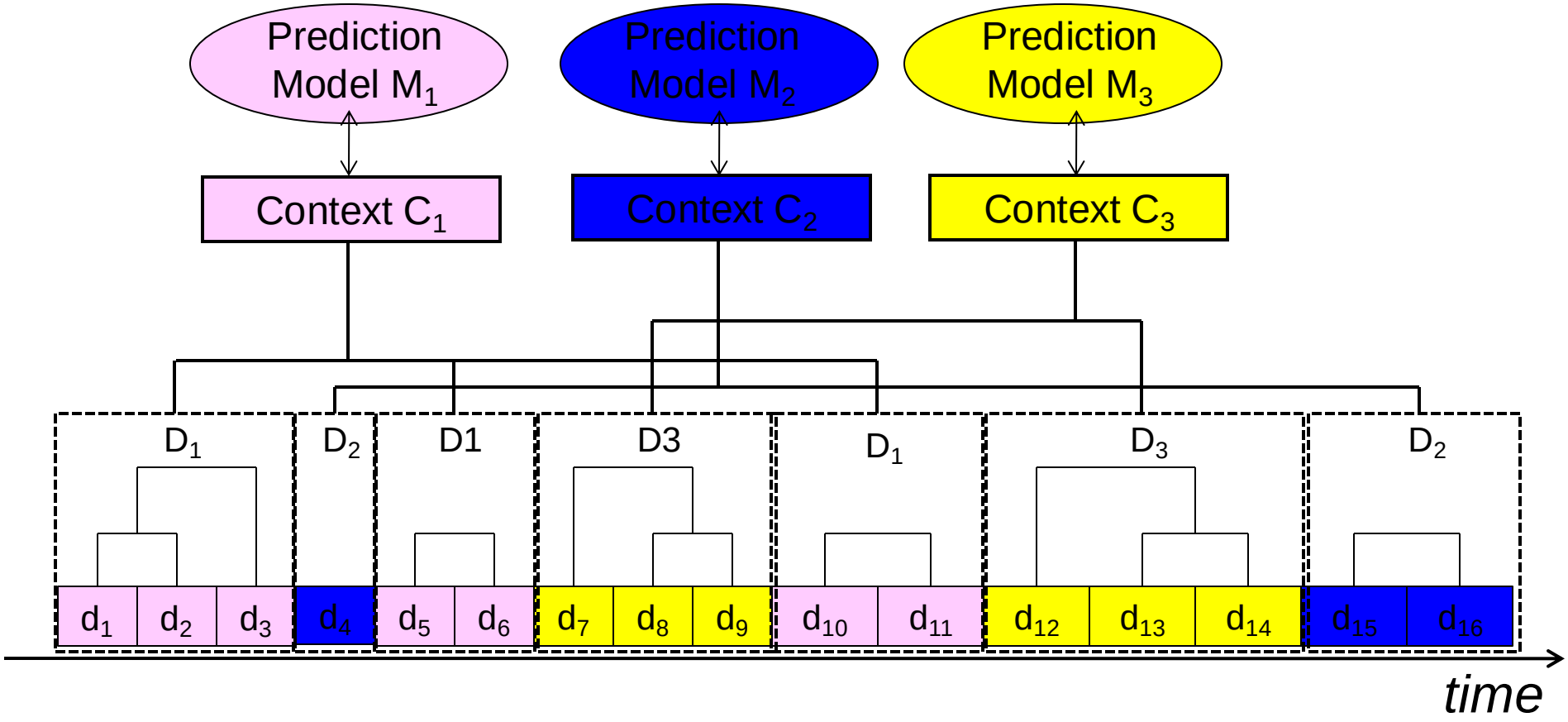


Context-Aware Anomaly Prediction for Dynamic Systems

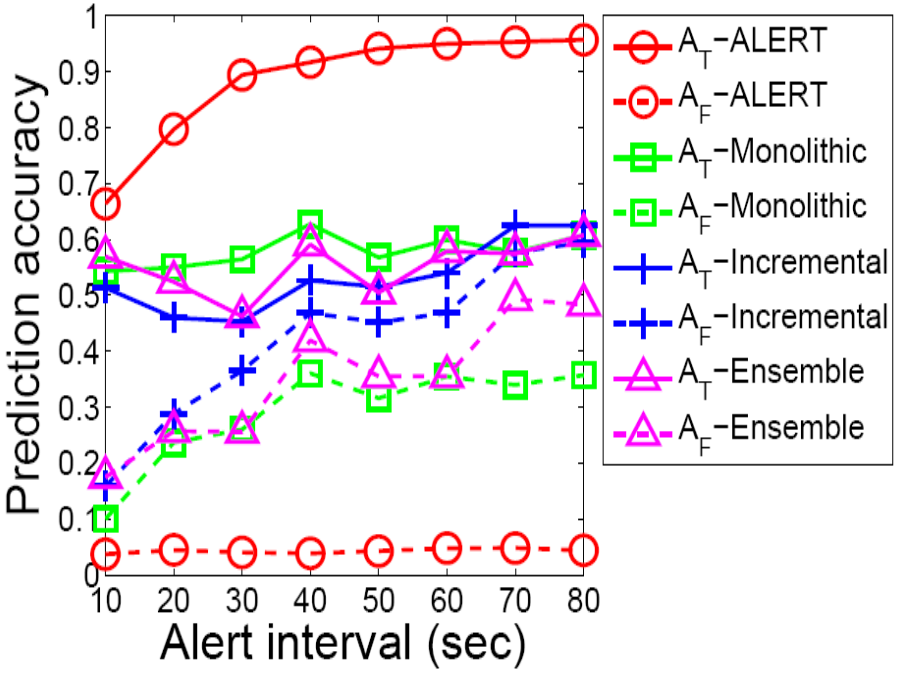
- More accurate model training
  - Induce an ensemble of prediction models from conflict-free training data
- Dynamic context discovery
  - Does not assume the context information is known
- Dynamic prediction model switching
  - Predict current context based on context evolving pattern
  - Dynamically switch models when context changes

Context Discovery

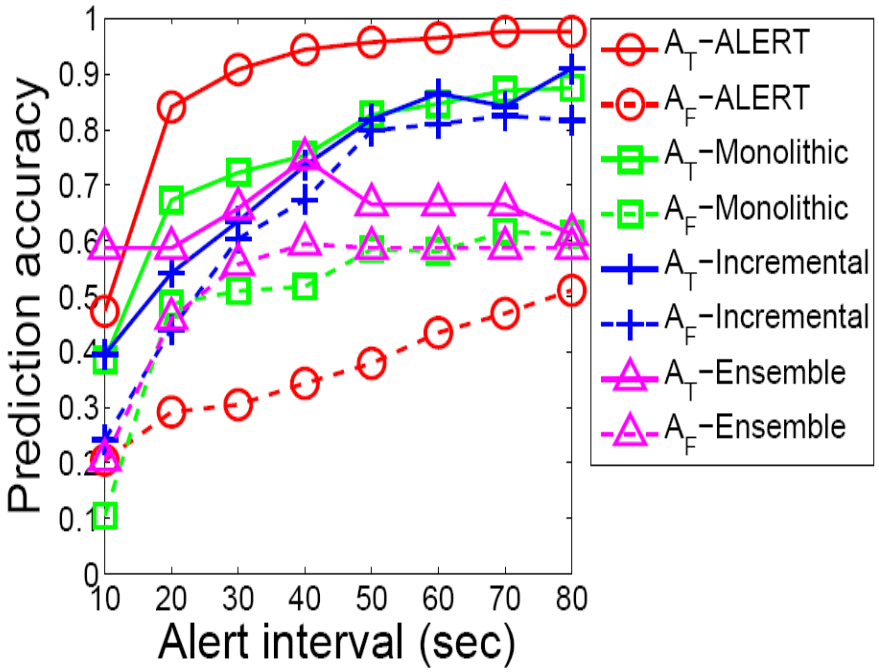
- Hierarchical clustering using the cross-validation error



IBM System S Prediction under Dynamic Workloads

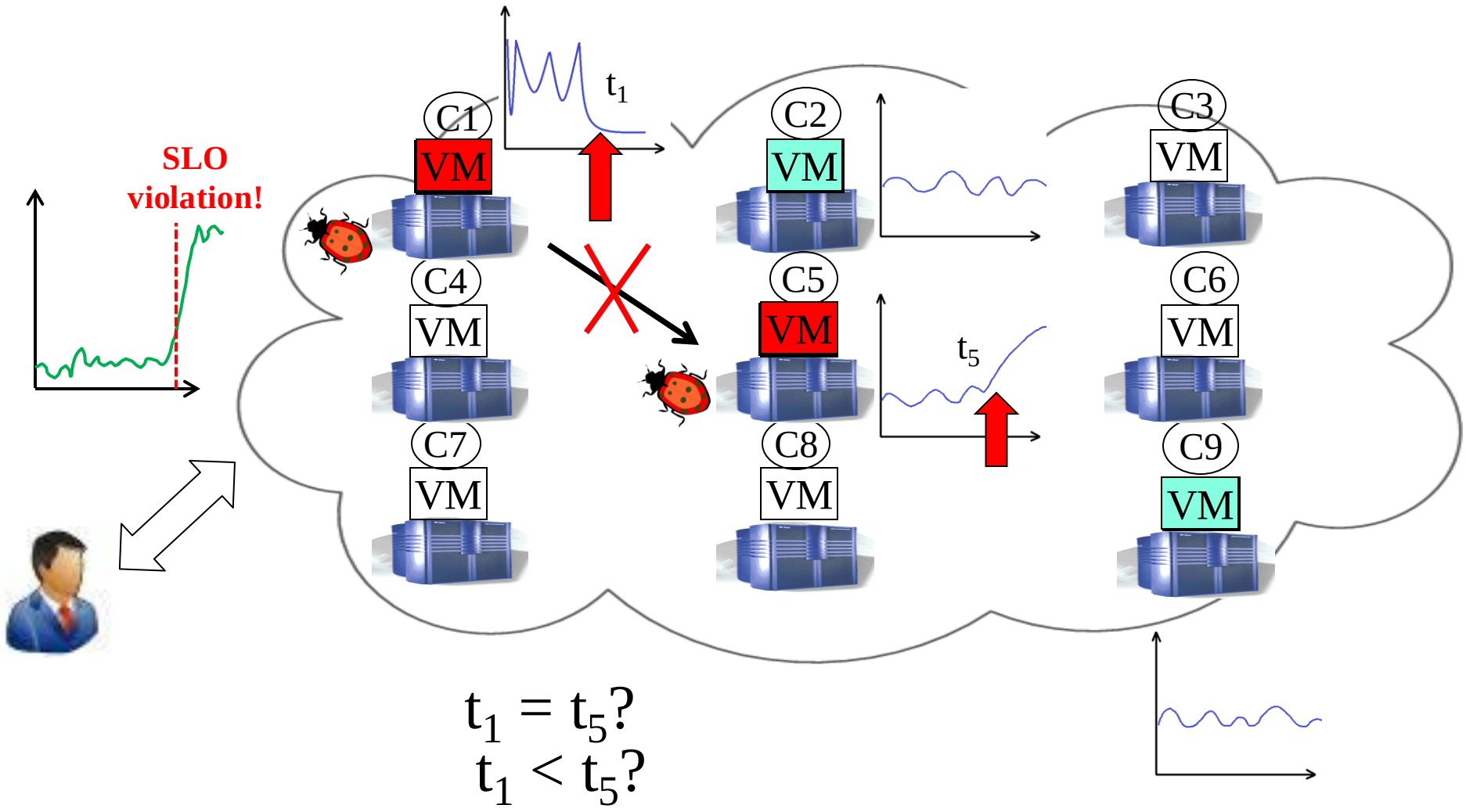


Memory Leak



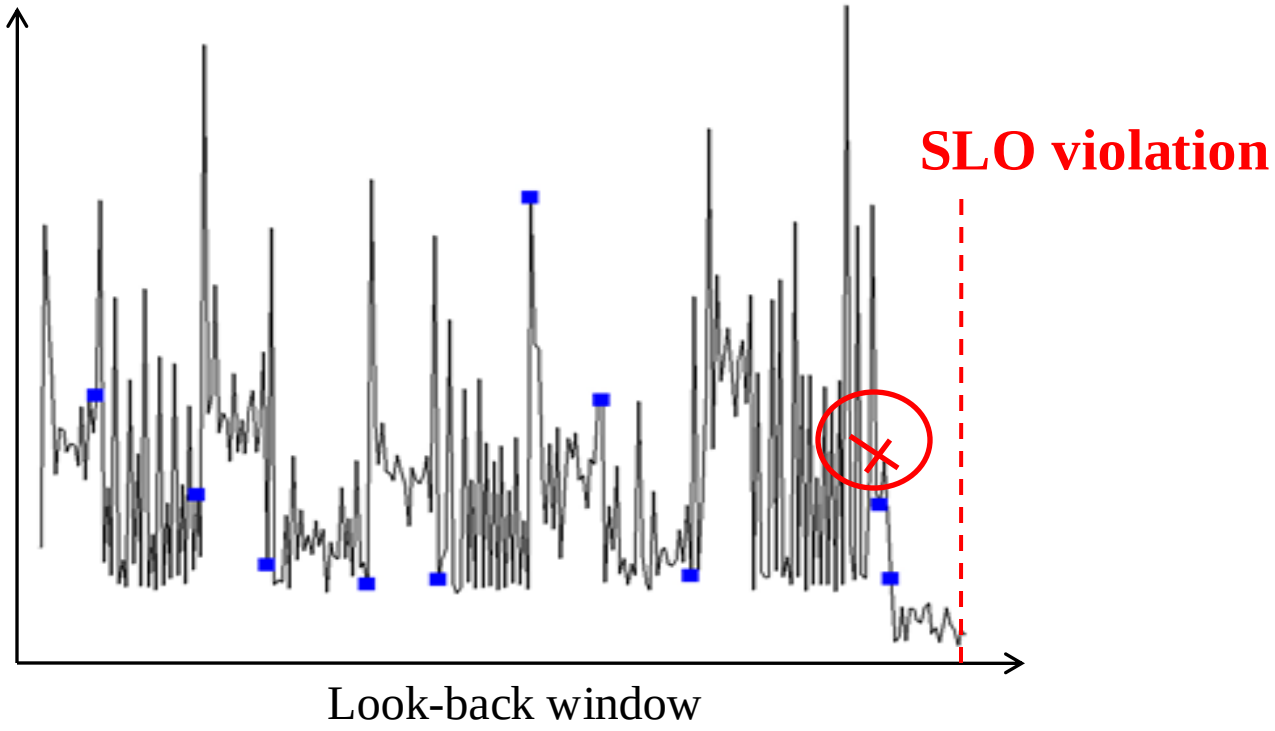
CPU Hog

FChain: Propagation-based Fault Localization



Abnormal change point selection

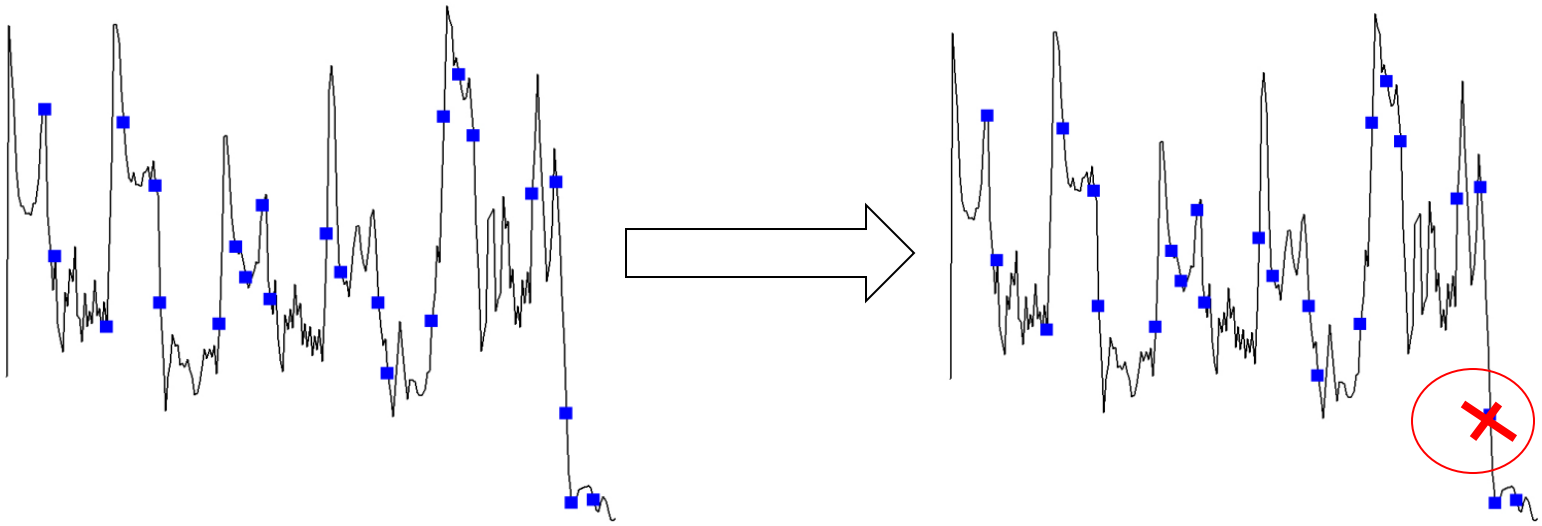
- **Mark the onset time of the fault manifestation**



Outlier Change Point Detection

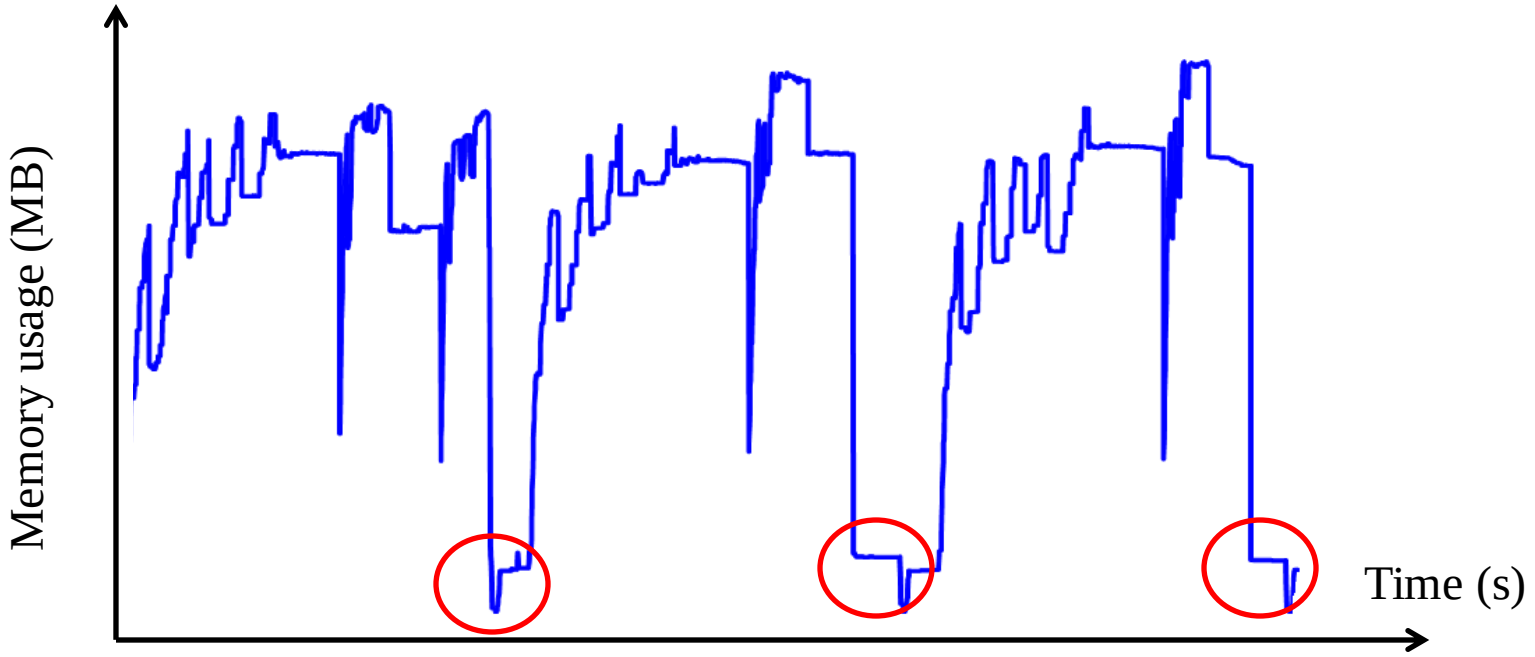
Smoothing
CUSUM + Bootstrap

Outlier detection



Outlier Change Point Filtering

- Change points caused by repeating workload fluctuations are predictable



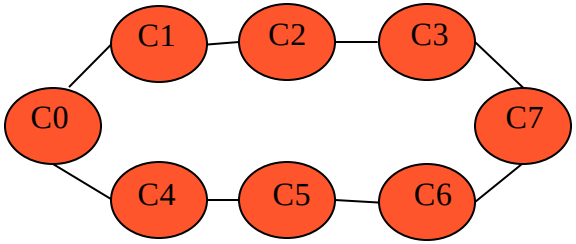
Memory usage in a Map node

Outlier Change Point Filtering

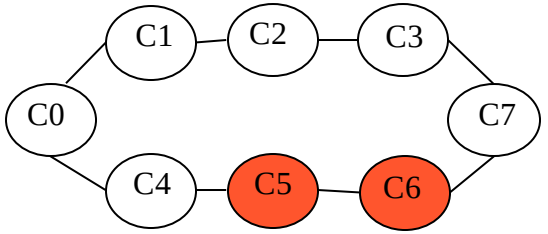
- Employ online learning models to capture normal fluctuations in system-level metrics
  - Markov chain model [Gong et al., CNSM'10]
  - Predict value at each change point
- Examine the prediction error ($\alpha = |y - y|$) of each outlier change point
  - Low (e.g., $\alpha \leq \varepsilon$): normal change point
  - High (e.g., $\alpha > \varepsilon$): abnormal change point

Propagation-based Fault Localization

- Workload change
  - Full coverage
  - Change points have the same upward trend

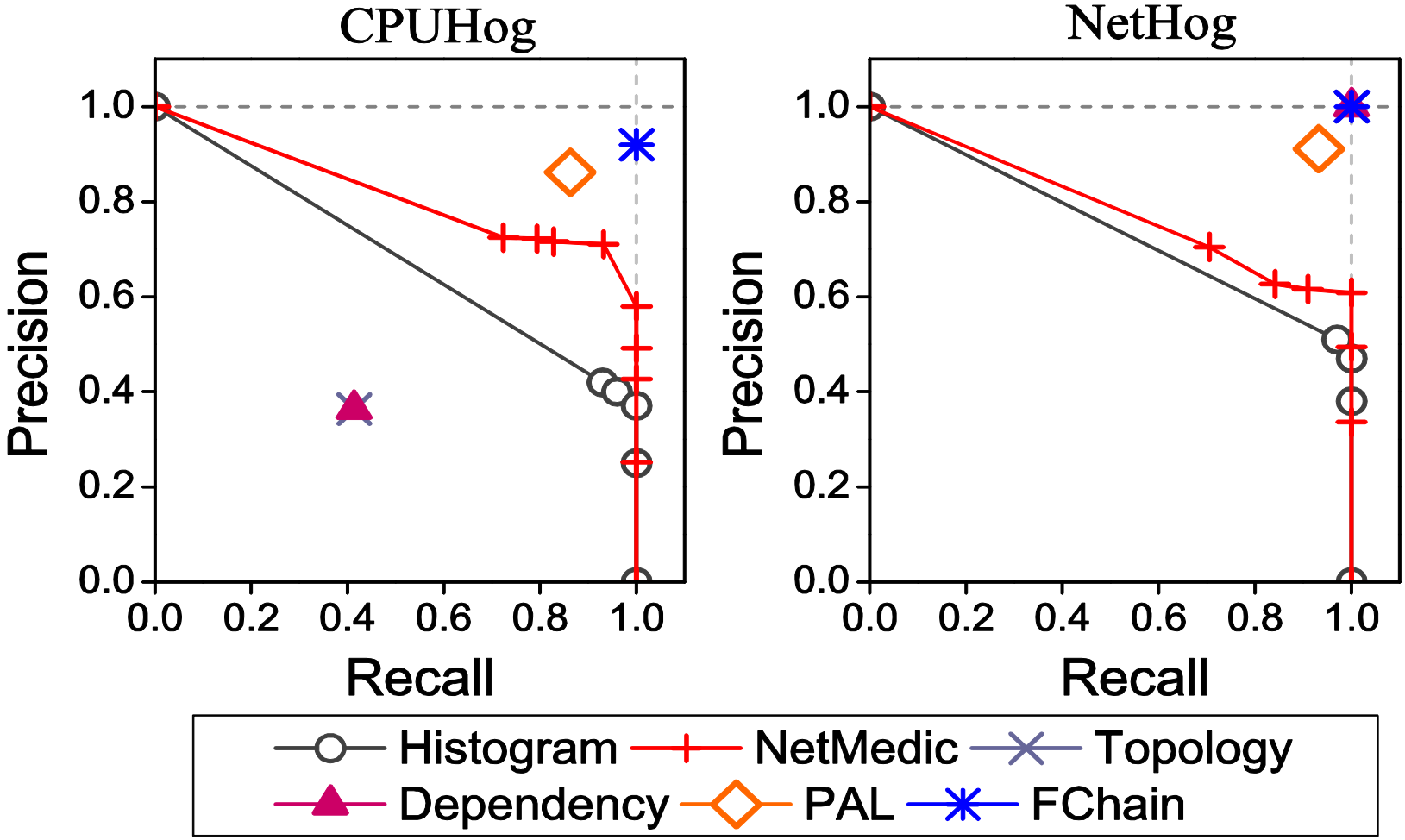


Full coverage

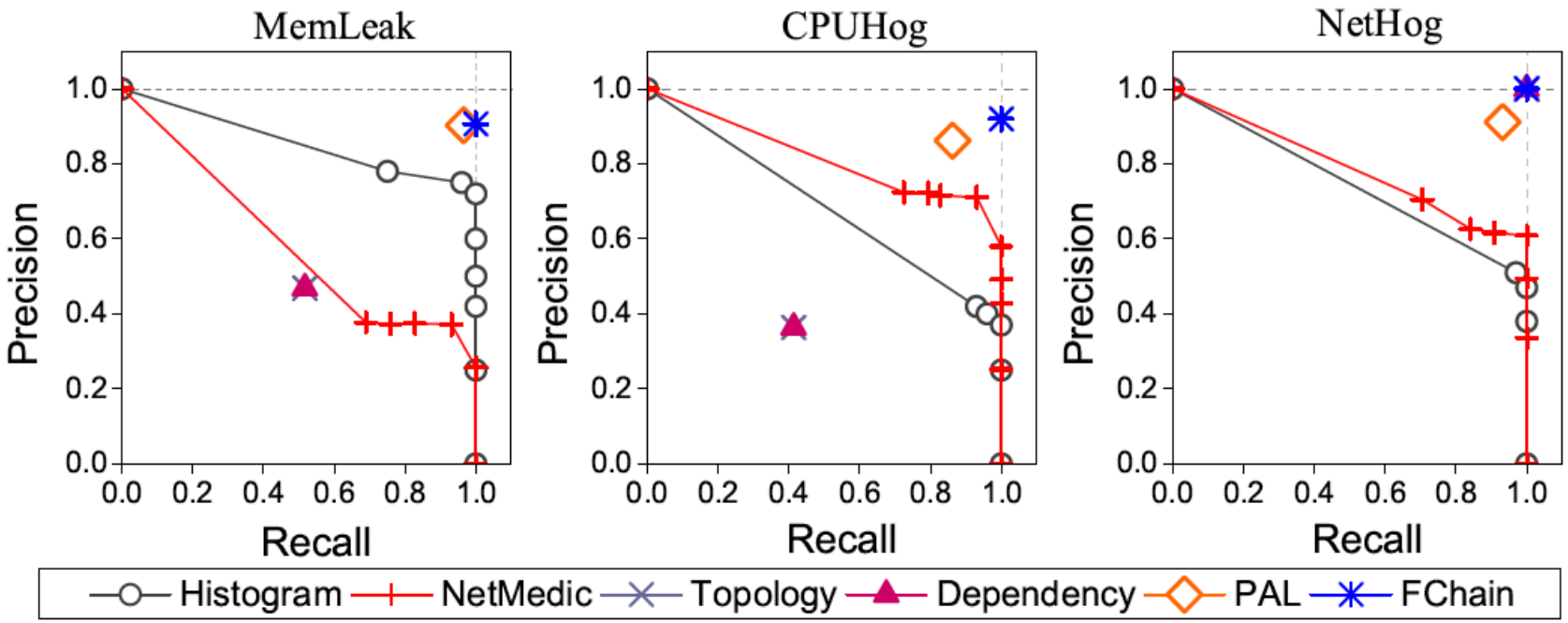


Partial coverage

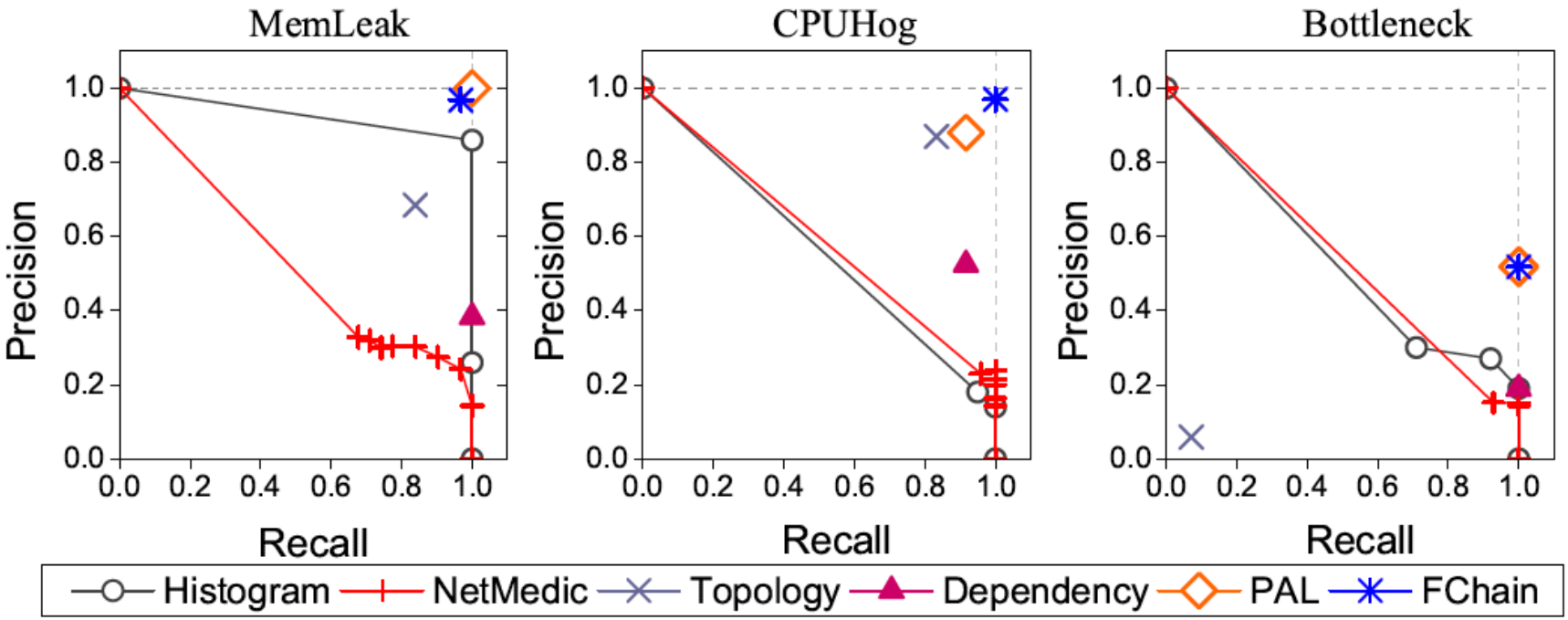
Faulty Component Pinpointing Results



RUBiS Pinpointing Results



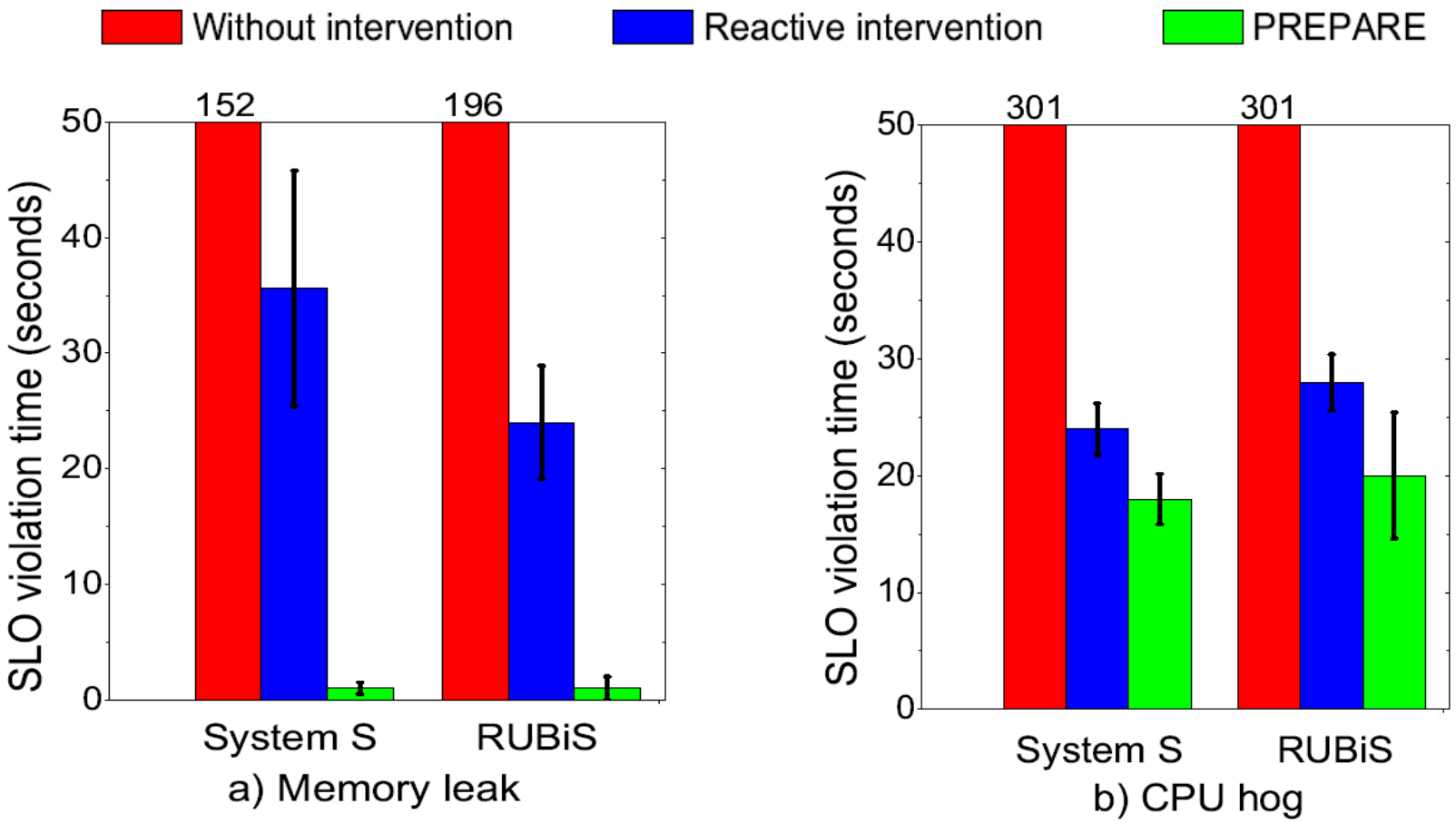
System S Pinpointing Results



Predictive Anomaly Prevention

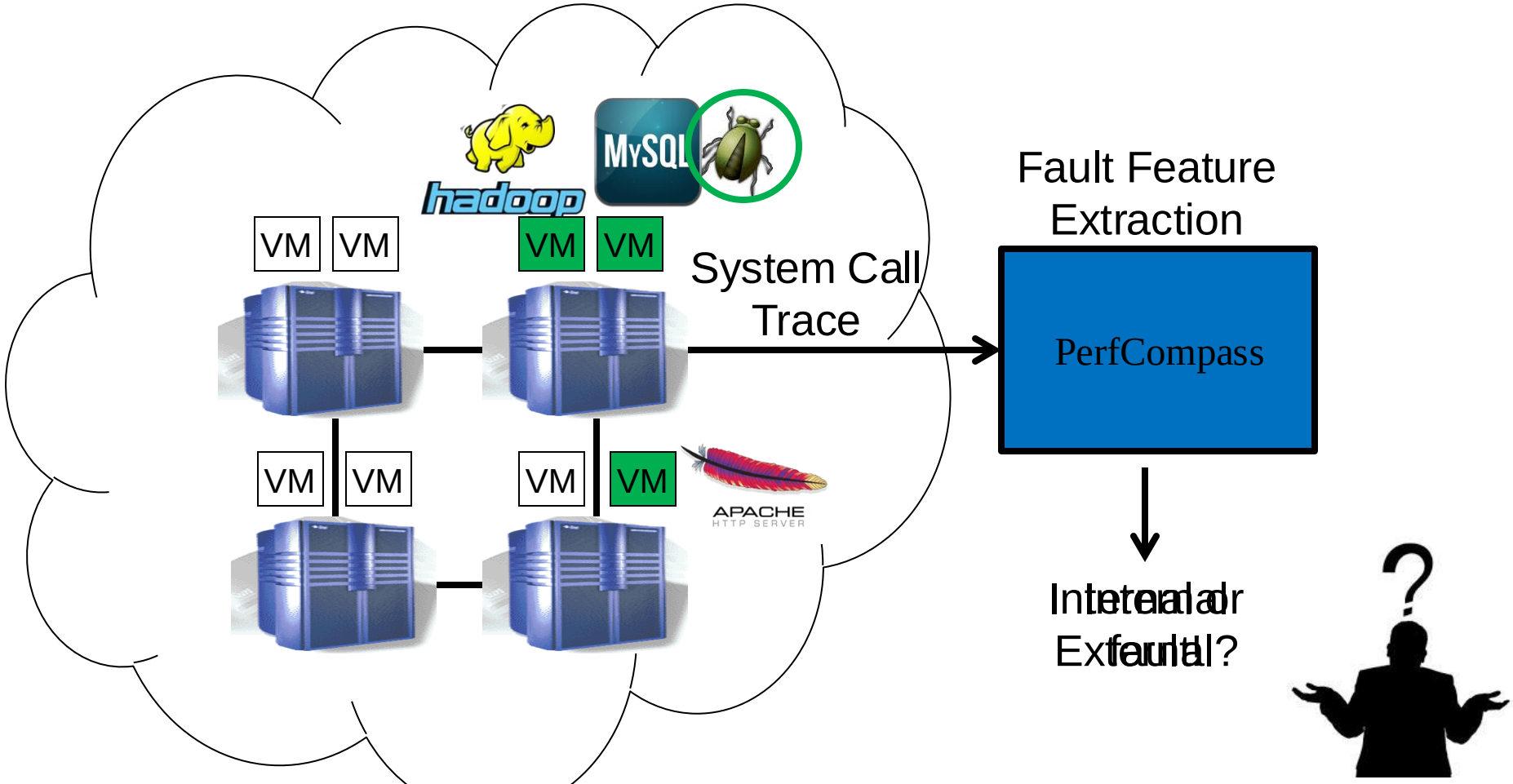
- Pinpoint faulty components
  - Anomaly propagation based pinpointing
- Coarse-grained anomaly cause inference
  - Metric attribution
- Safe, reversible intervention
  - Dynamic resource scaling
  - Live VM migration
- Online validation
  - Check whether anomaly alerts are gone
  - Compare resource usage before and after prevention

Predictive Performance Anomaly Prevention

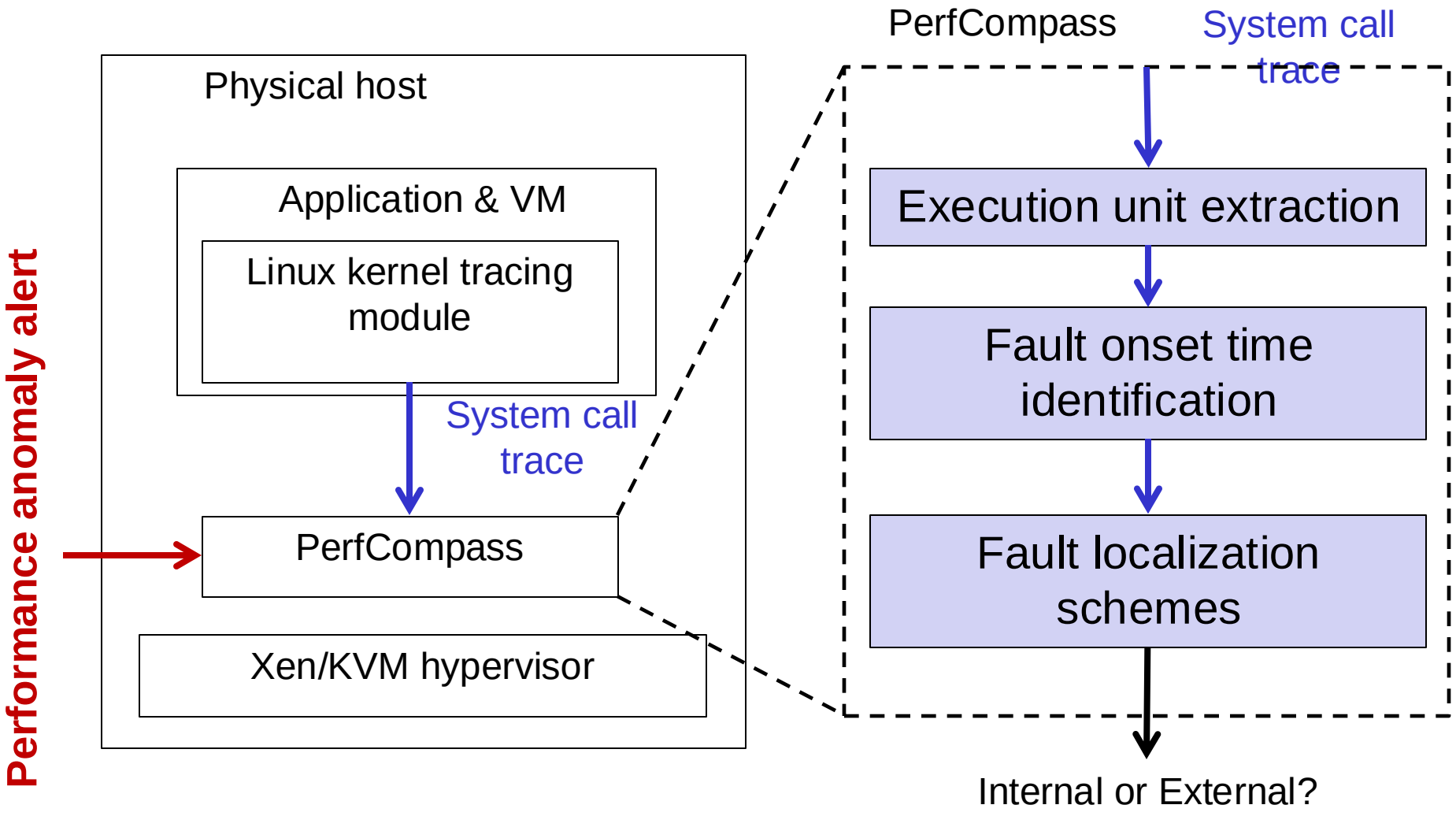


SLO violation time comparison using
the live VM migration as the prevention action.

PerfCompass: Runtime Performance Anomaly Fault Localization



System Architecture



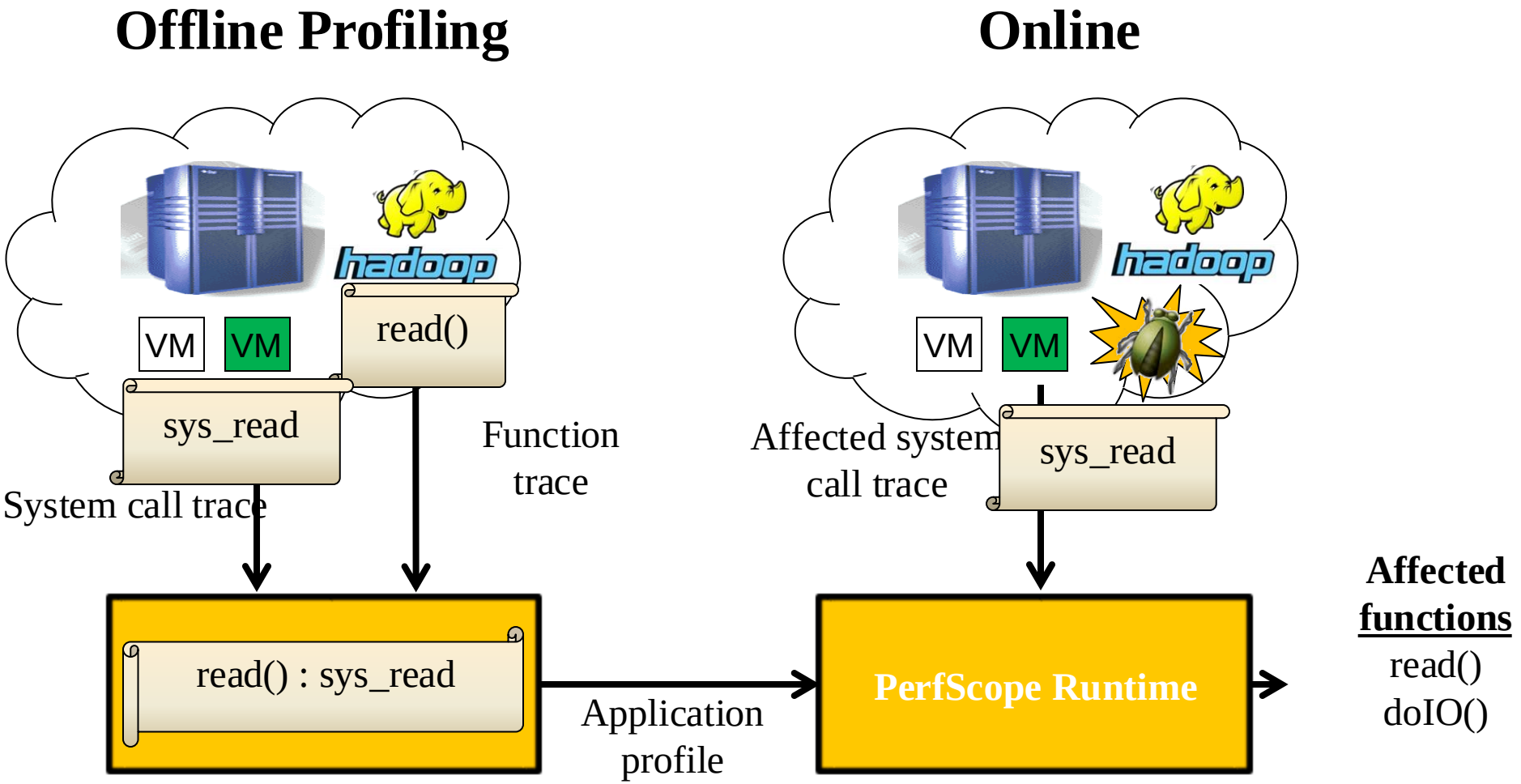
Experimental Evaluation – External Faults

System Name	Fault Description	Fault Impact Factor	Fault Onset Time Dispersion
Apache	CPU cap	100 +/- 0 %	7 +/- 1ms
Apache	Packet loss	100 +/- 0 %	4 +/- 1ms
MySQL	I/O interface	100 +/- 0 %	15 +/- 7ms
MySQL	CPU cap	94 +/- 2%	17.77 +/- 4ms
Squid	Packet loss	100 +/- 0 %	0.01 +/- 0.001ms
Cassandra	CPU cap	99 +/- 1.4%	28 +/- 4ms
Cassandra	I/O interference	100 +/- 0%	9 +/- 1ms
Hadoop	CPU cap	98 +/- 1%	39 +/- 5ms
Hadoop	I/O interference	98 +/- 0%	16 +/- 3ms

Experimental Evaluation – Internal Faults

System Name	Fault Description	Fault Impact Factor	Fault Onset Time Dispersion
Apache	Flag setting bug	50 +/- 0.5 %	374 +/- 63ms
MySQL	Deadlock bug	40 +/- 0 %	38 +/- 3ms
MySQL	Data flushing bug	62 +/- 3 %	721 +/- 4ms
Squid	File access bug	83 +/- 1 %	0.35 +/- 0.09ms
Cassandra	Endless loop	51 +/- 5.7 %	25 +/- 0.98ms
Hadoop	Endless read	81 +/- 0 %	23 +/- 6ms
Hadoop	Thread shutdown	85 +/- 0.5 %	110 +/- 20ms

PerfScope: Online Performance Bug Inference



Online Bug Inference Results

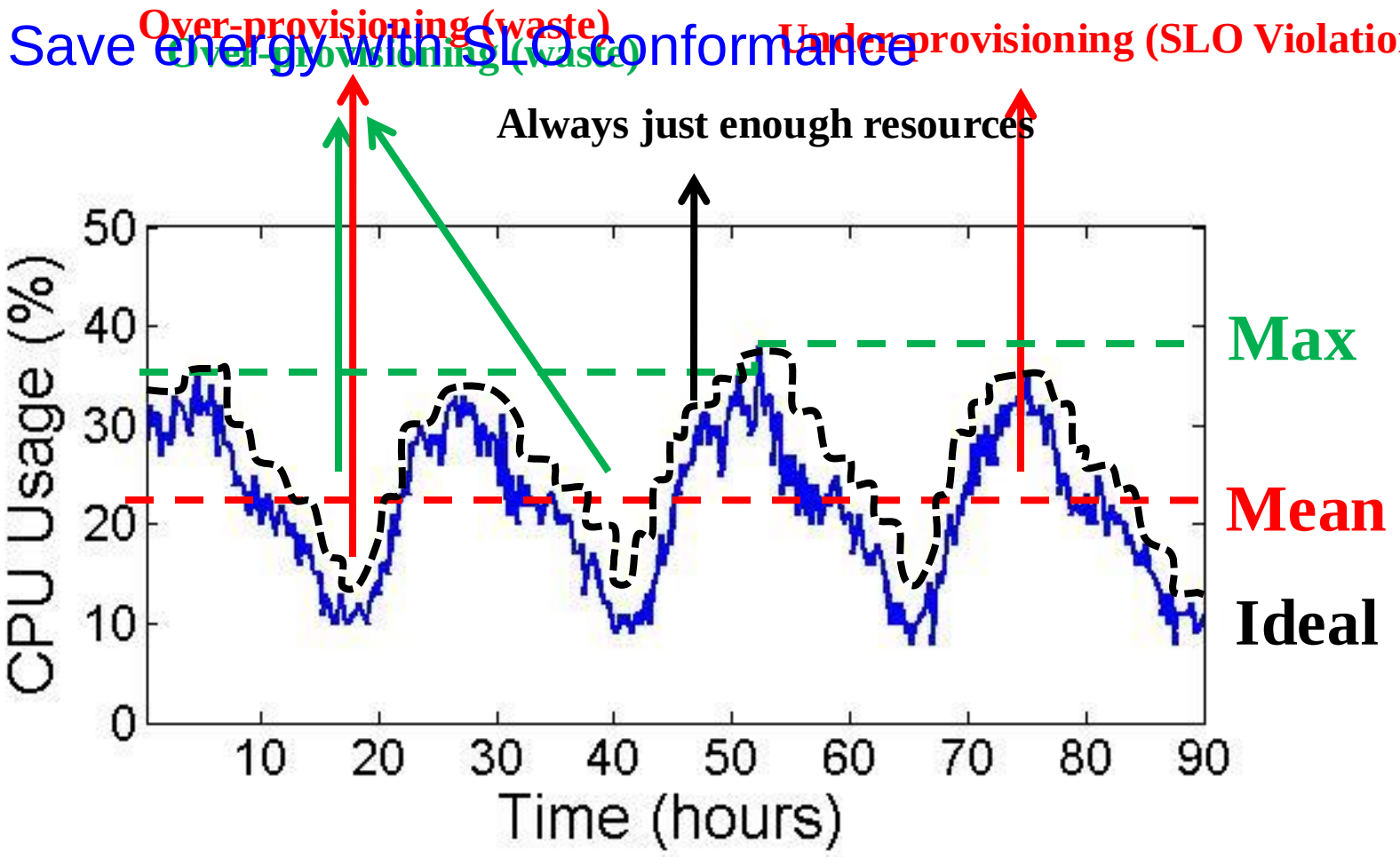
System Name	Percent of Identified Functions	Example Bug Related Function
Hadoop	0.52%	getState(4)
HDFS	1.5%	Reader.performIO(3)
Cassandra	0.6%	maybeSwitchMemtable(4)
Tomcat-1	1.5%	addLifeCycleListener(2)
Tomcat-2	0.8%	LimitLatch.countDown(11)
Tomcat-3	0.4%	Poller.run(3)
Apache-1	0.7%	apr_allocater_mutex_get(14)
Apache-2	2%	ssl_hook_pre_connection(9)
Lighttpd-1	0.1%	fdevent_poll(1)
Lighttpd-2	0.08%	connection_handle_read_state(3)
MySQL-1	0.6%	Ha_lock_engine(14)
MySQL-2	0.03%	buf_flush_list(4)

Cloud Computing Challenges

- Robustness
- Resource and energy efficiency
- Accountability

Elastic Resource Scaling

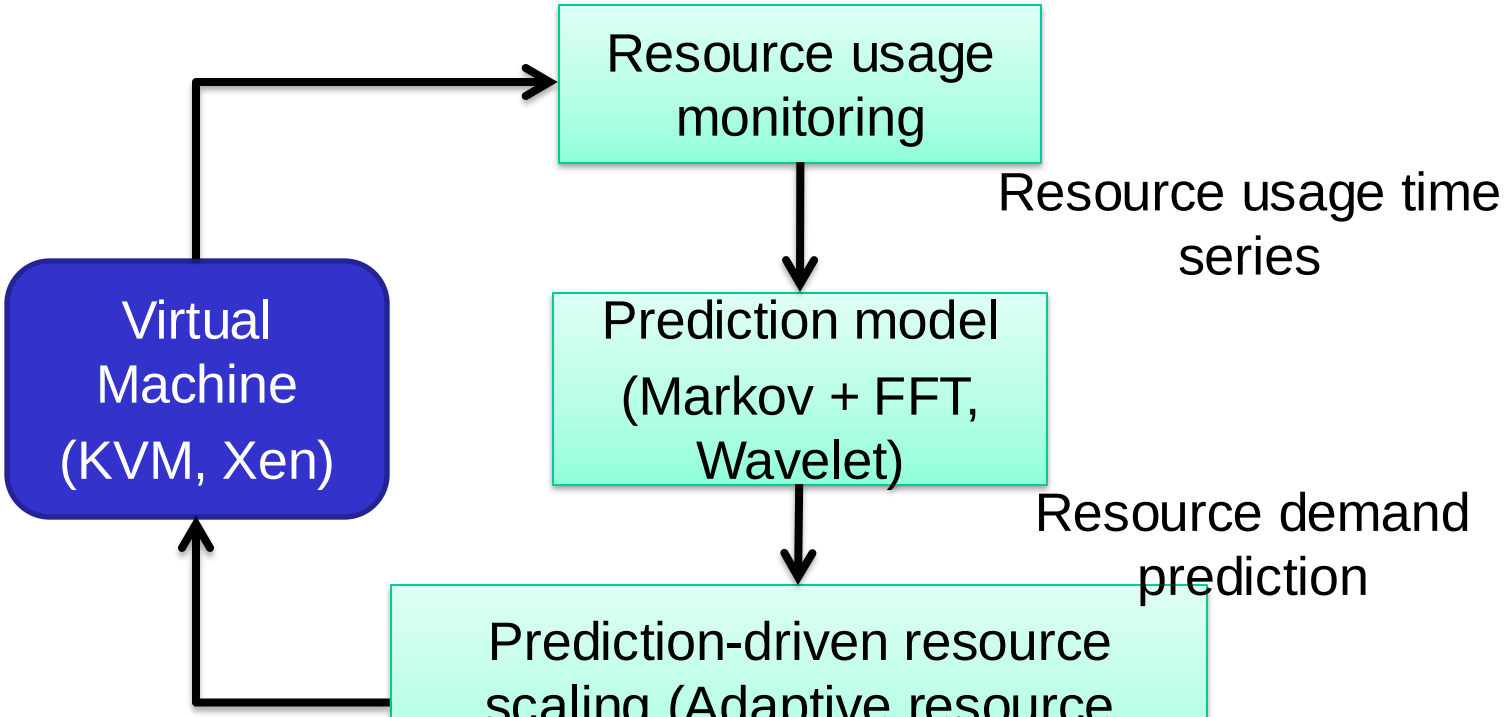
- Cannot assume prior application knowledge
- Save energy with SLO conformance



Apache web server under real server workload

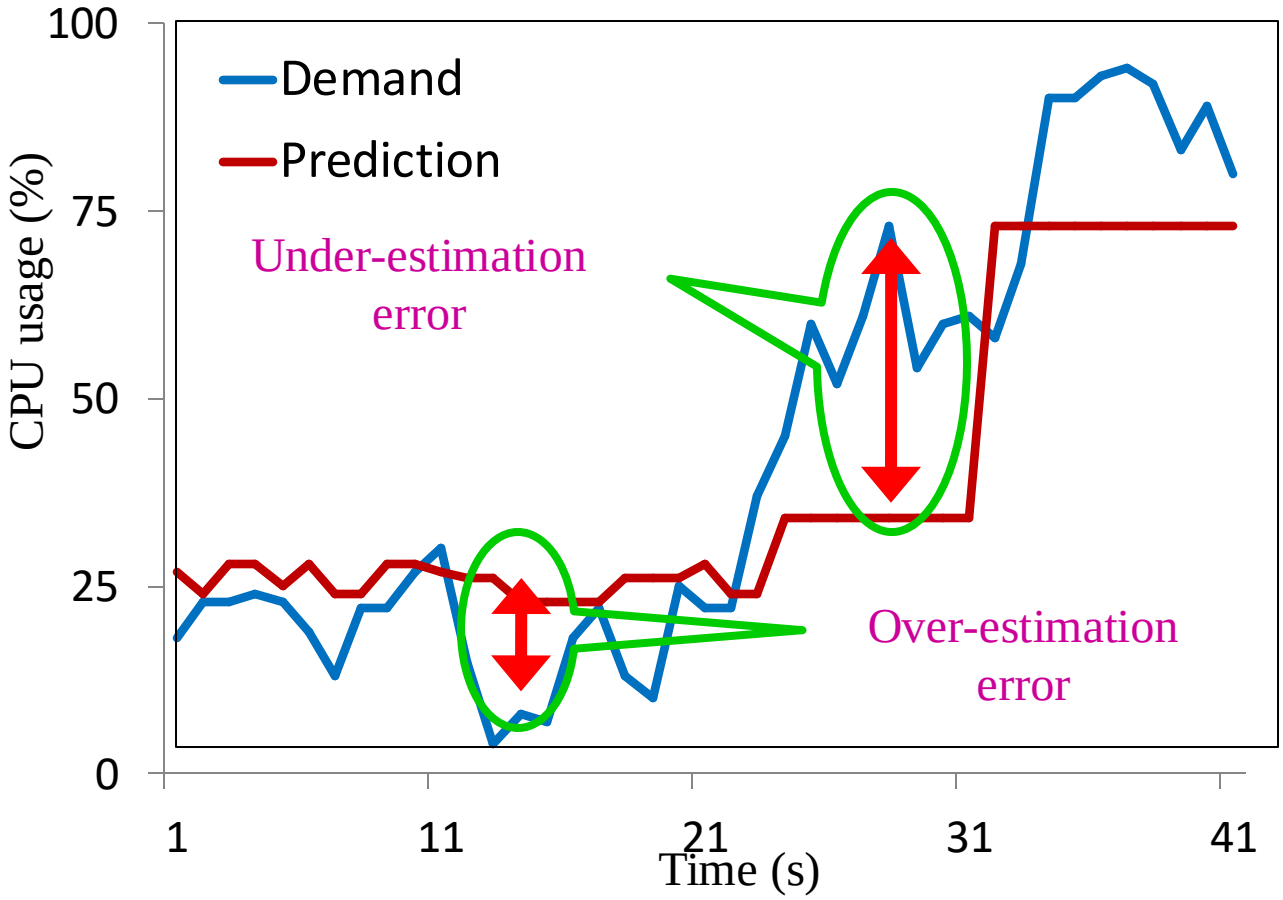
Predictive Elastic Resource Scaling

- Do not require advance application profiling or modeling

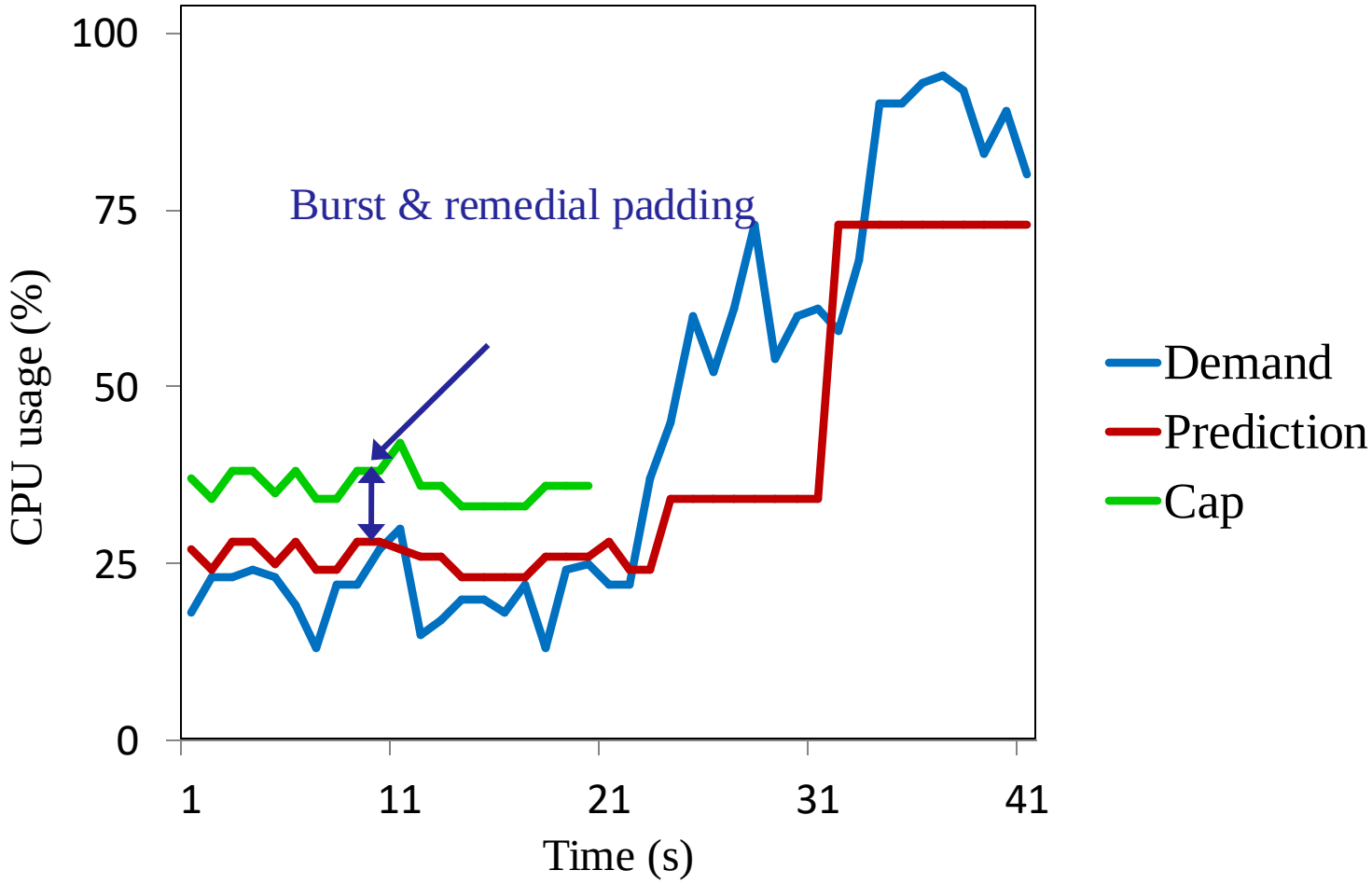


- Adjust resource cap
- Live VM migration
- Live VM cloning (pre-copy)
- Dynamic frequency & voltage scaling

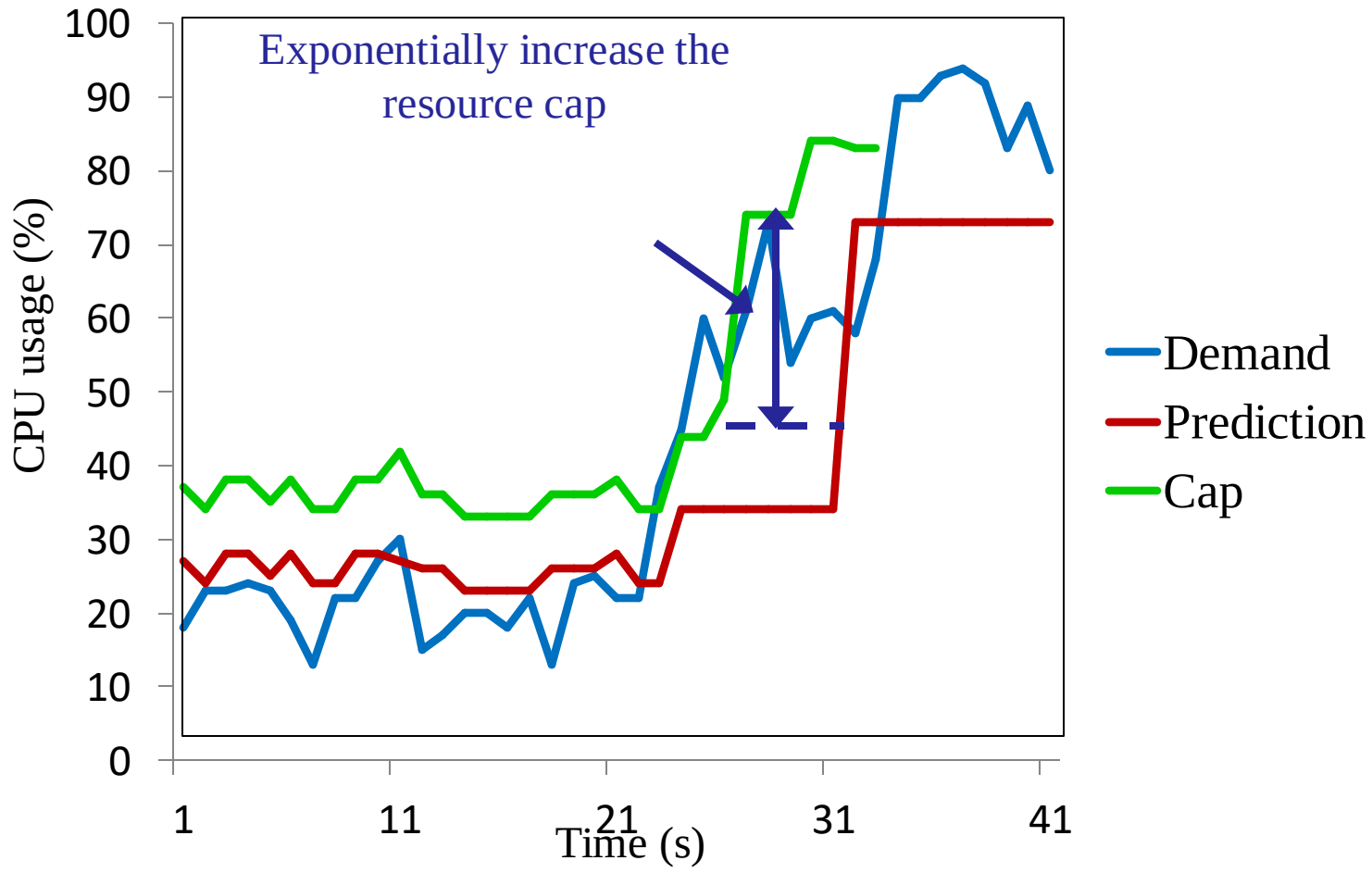
Prediction-driven Scaling Problems



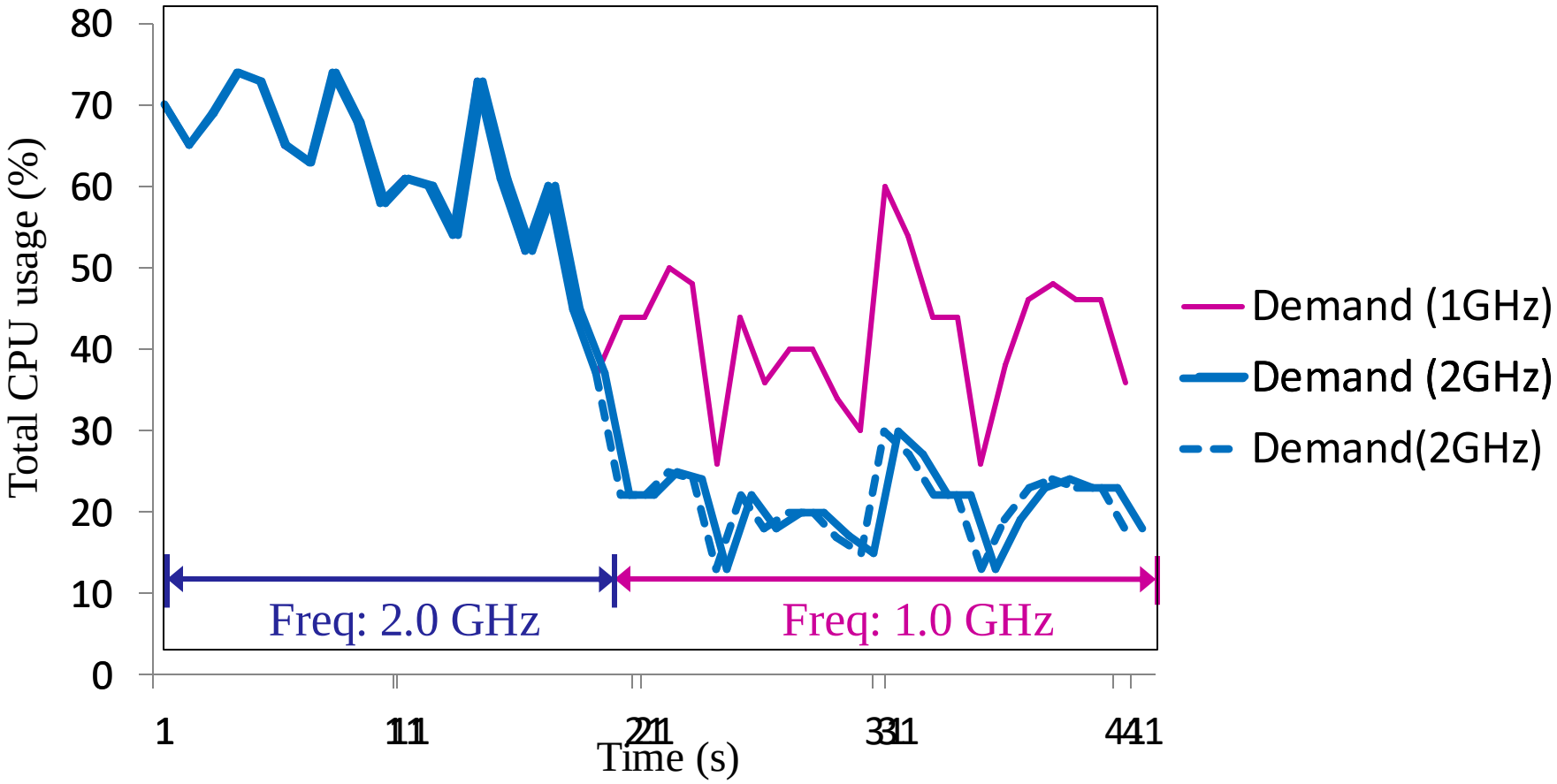
Under-Estimation Prevention



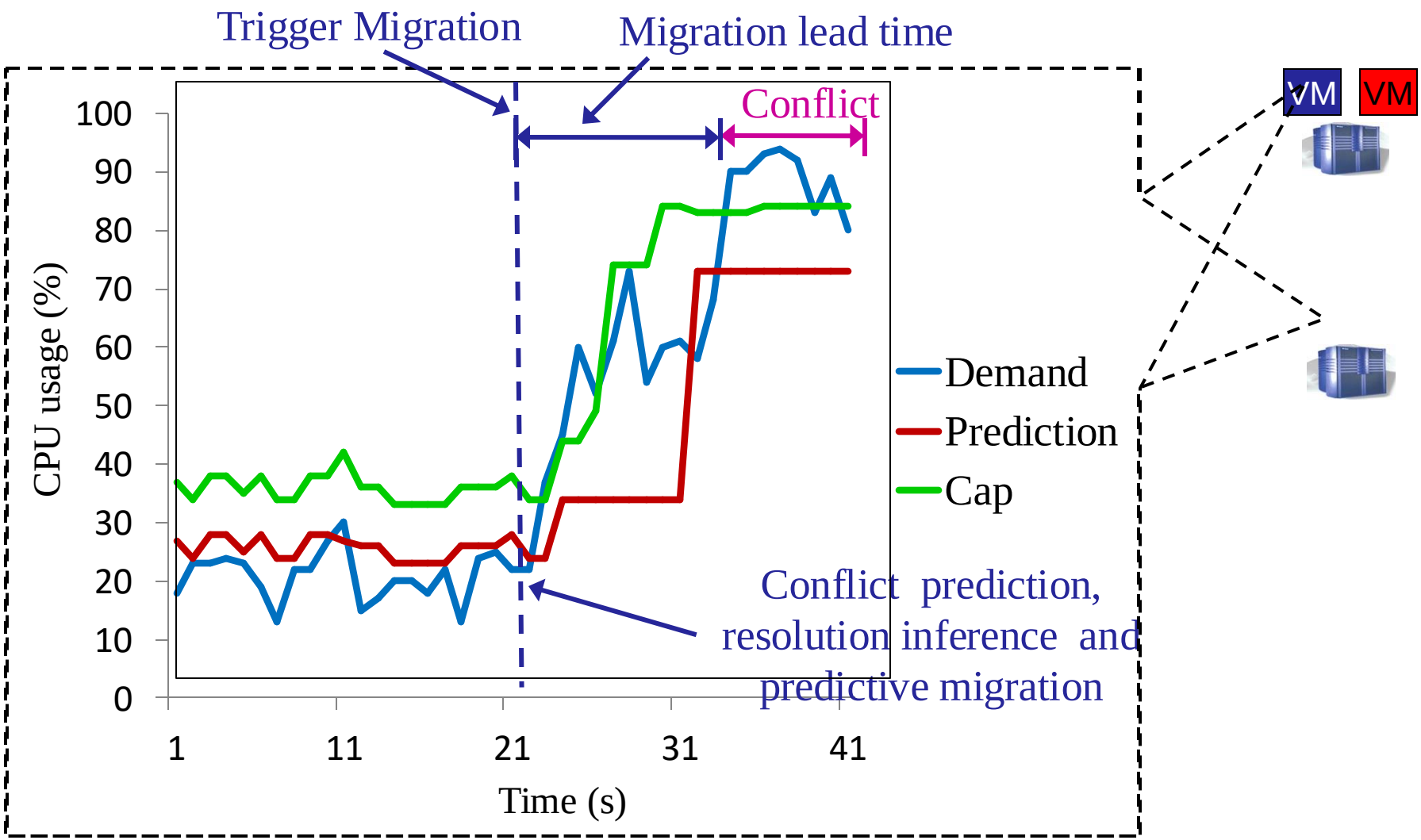
Fast Under-Estimation Correction



Predictive Frequency and Voltage Scaling for Energy Saving

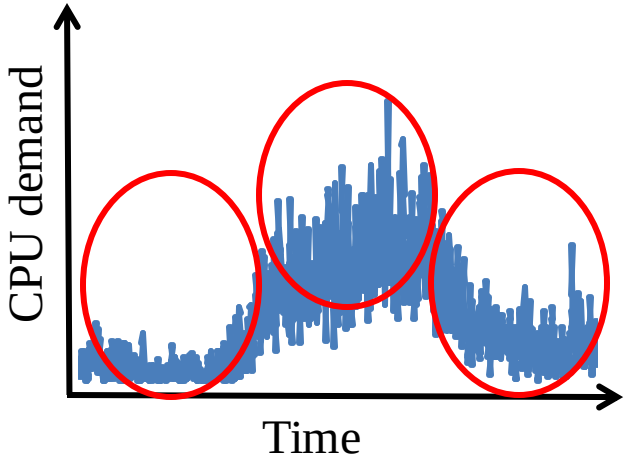


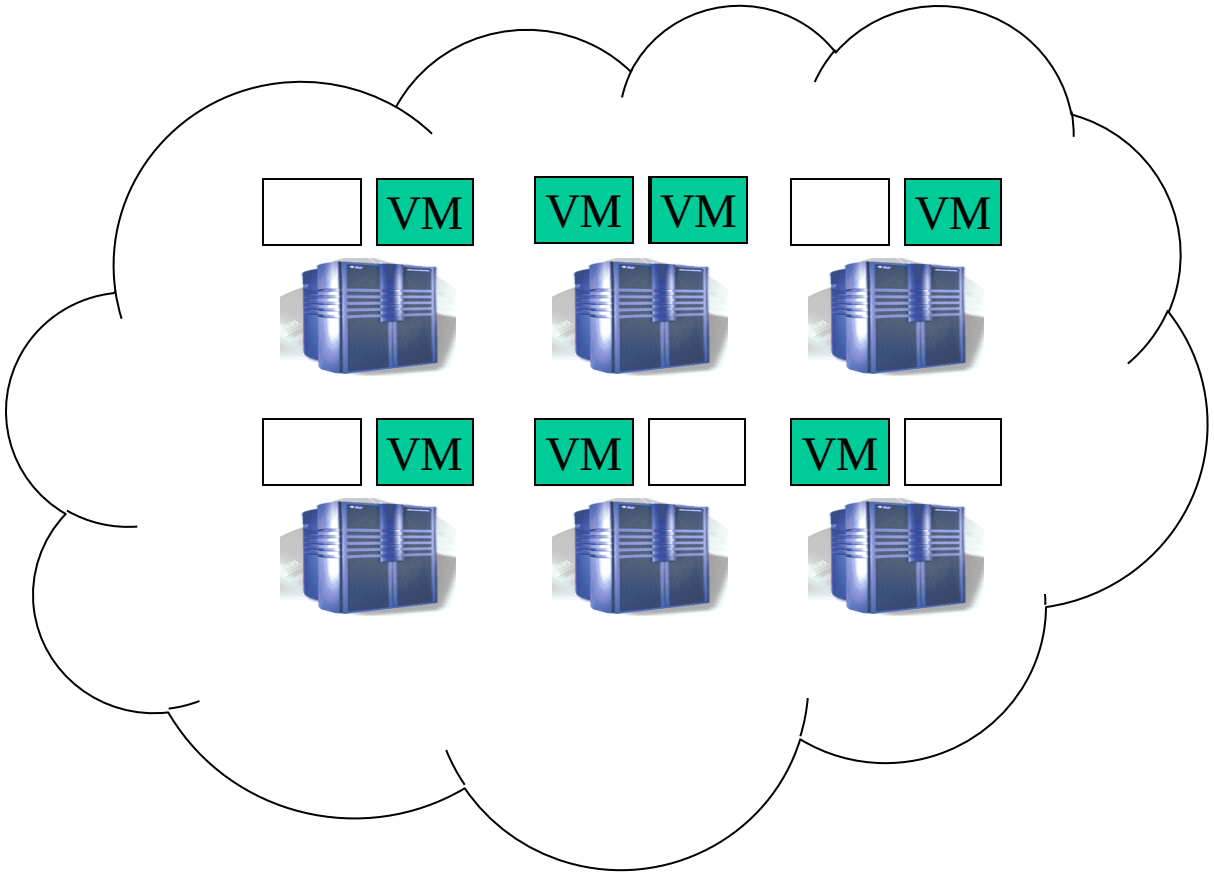
Predictive Live VM Migration



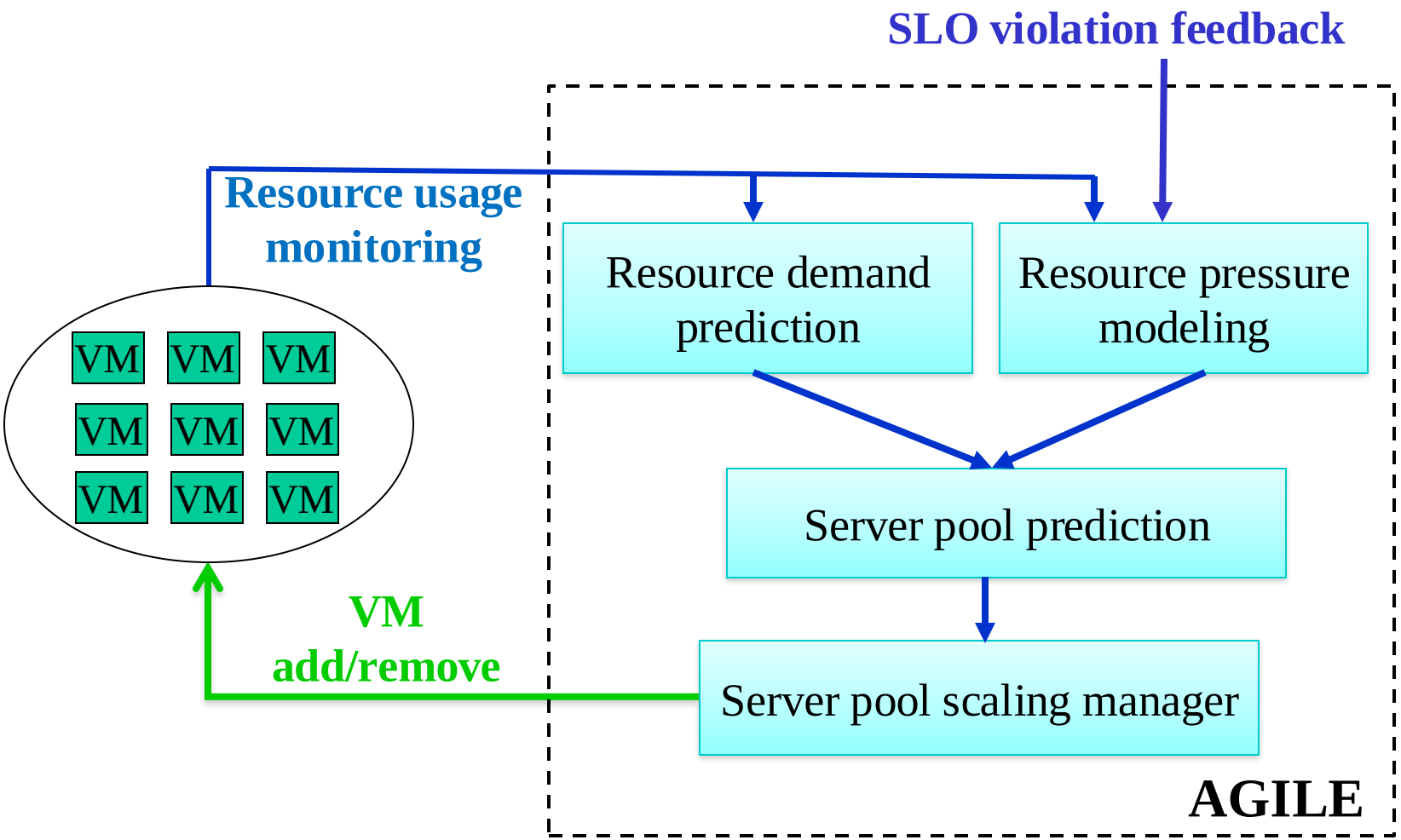
Elastic Auto-Scaling

- Elasticity: grow/shrink resource as required





AGILE System Overview



Experimental Evaluation

- Implemented on top of Xen and KVM
- Benchmark systems
  - Real resource usage traces from 100 machines in a Google cluster
  - RUBiS driven by real server workload traces
    - WorldCup' 98, EPA, NASA, ClarkNet
  - Hadoop
    - Word Count, Grep
  - IBM System S
    - Tax calculation application

Summary of Experimental Results

- **Prediction accuracy**
  - <10% prediction error
- **SLO conformance**
  - SLO violation rate: 50% → 2-8%
  - SLO violation time during conflicts: 350 → 60 seconds
- **Energy saving**
  - 8-10% total energy saving and 39-71% workload energy saving with little impact on the application performance

Overload handling

- **Web server: during scaling**