Automated Resource Management in Virtualized Data Centers

Sharad Singhal — **Hewlett Packard Laboratories**

Xiaoyun Zhu — **VMware Inc.**

IM 2009 Tutorial: Recent Advances in the Application of Control Theory to Network and Service Management

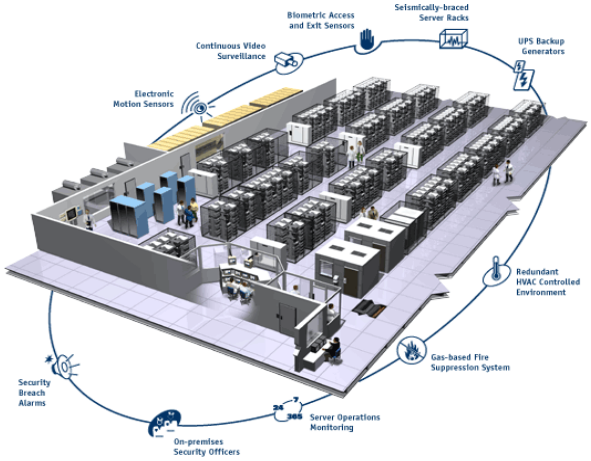
Outline

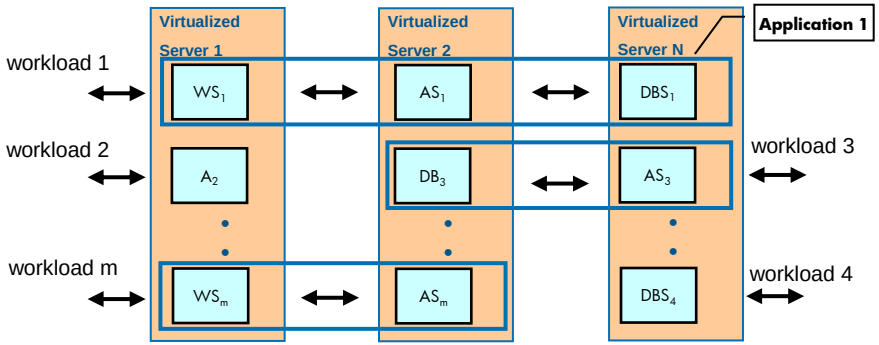
- Background
  - Virtualized data center
    - Server consolidation and virtualization
    - Challenges in workload management
    - Control basics
- Case studies
  - CPU utilization control [DSOM'05]
  - Application response time control [ACC'06]
  - Predictive control
    - Temporal prediction [NOMS'06]
    - Performance models [FeBID'07]
  - Multi-input multi-output control
    - Memory + CPU control [IM'09 MiniConference]
    - CPU + disk control [Eurosys'09]
- Summary

Virtualized data center

A large-scale shared hosting platform

- Virtualization enables resource sharing
  - Improved utilization
  - Higher compute density
  - More efficient
- Issues
  - Reduced performance isolation
  - Service levels more susceptible to workload variations

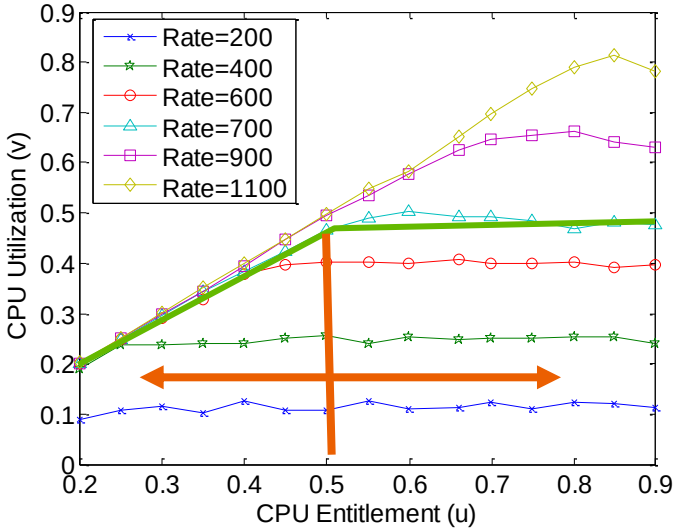


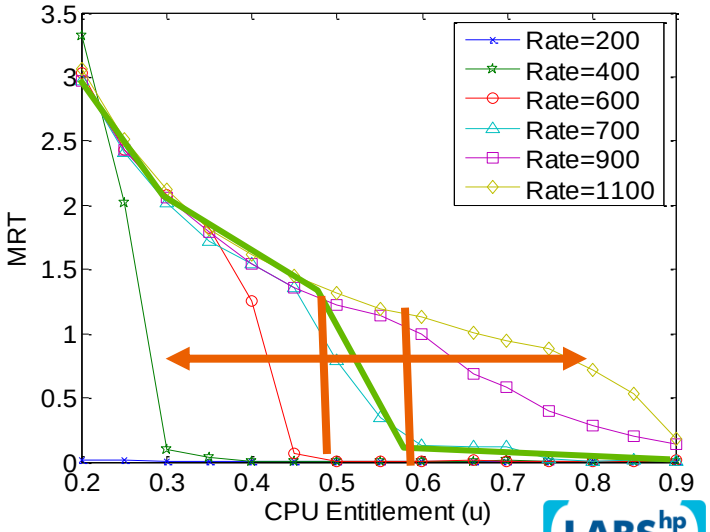


Challenges

Need for adaptive control approaches

- Most applications show multi-modal behavior
  - Certain operating regions "uncontrollable"
    - Metrics of interest do not change with resource quantity
  - Operating regions shift with workload
    - For a given change in metrics, different quantity of resource is needed in the different regions
- Manually tuned or time-invariant techniques are in-appropriate
  - Do not support wide range of workloads
    - A threshold appropriate for workload "valleys" is inappropriate for "peaks"
  - Not amenable to automation
    - humans are in the loop



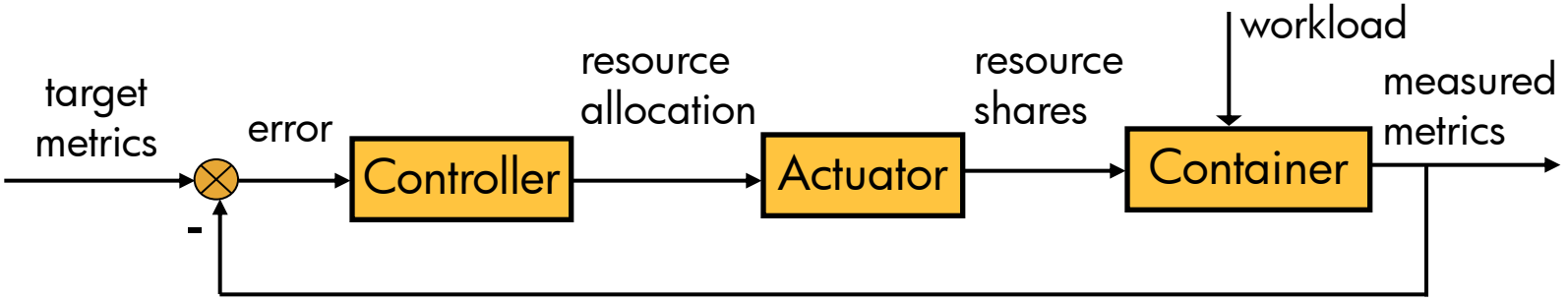


Z. Wang et. al, "Utilization and SLO-based control for dynamic sizing of resource partitions," DSOM'05.

Key technology enablers

Server consolidation and virtualization

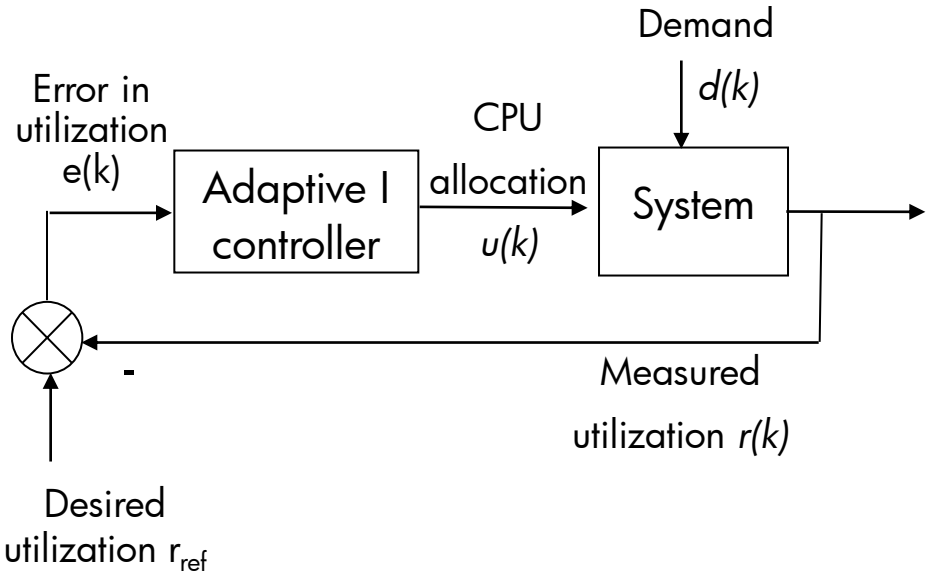
- Container: encapsulates a share of server resources
  - CPU, memory, network I/O, disk I/O
  - Provides performance isolation
- Actuator: APIs for dynamic resource allocation to containers
- Controller: Workload management tools (e.g., HP WLM) can dynamically size a container to maintain a target utilization



Utilization control

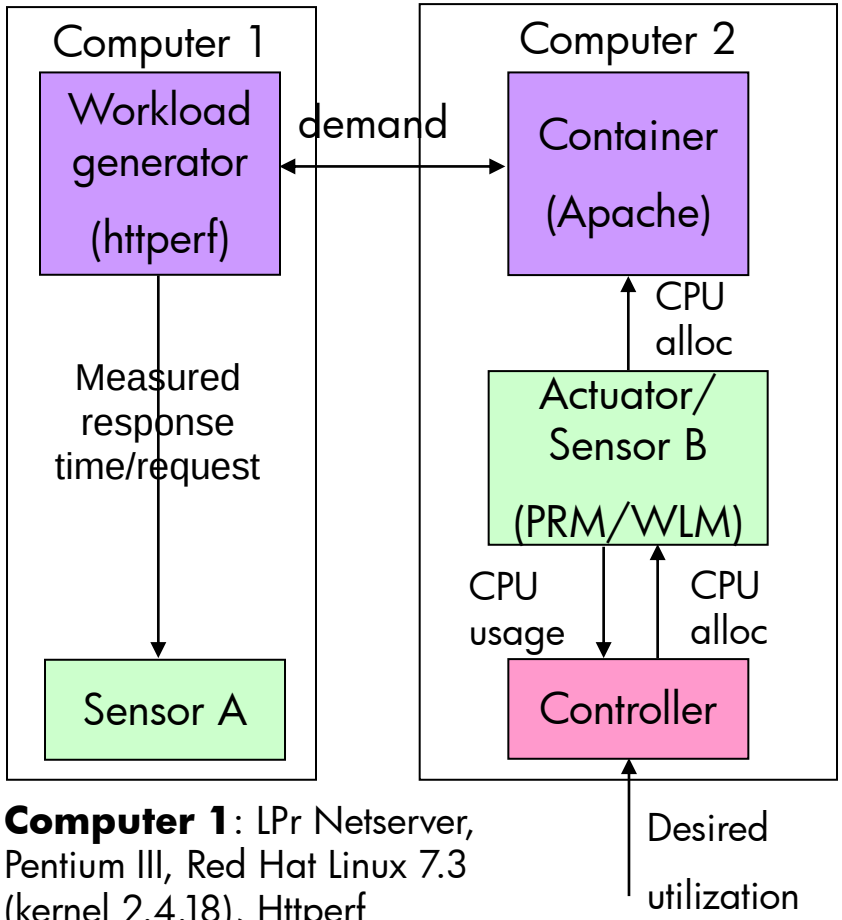
Controller and setup

Control Structure:



Controller: $u(k) = u(k-1) - K_I(k)e(k-1)$

Testbed:



Computer 1: LPr Netserver, Pentium III, Red Hat Linux 7.3 (kernel 2.4.18), Httperf

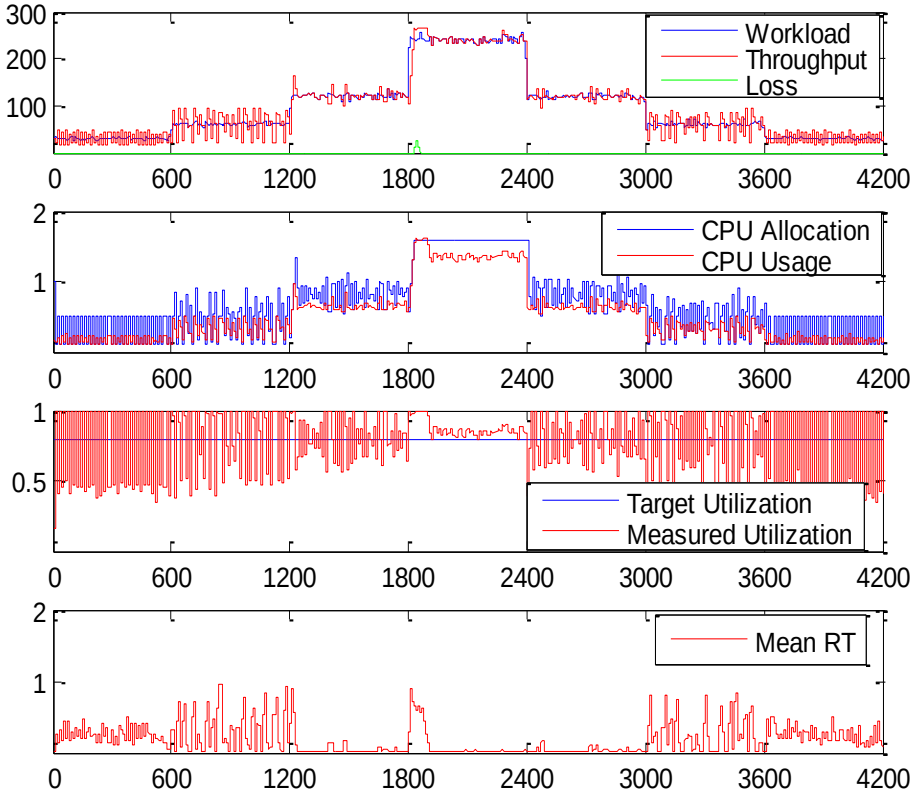
Computer 2: Dual processor, HP 9000-L server, PA-RISC processor, HP-UX B11.11, WLM 3.0, Apache server

Z. Wang et. al, "Utilization and SLO-based control for dynamic sizing of resource partitions," DSOM'05.

Utilization control:

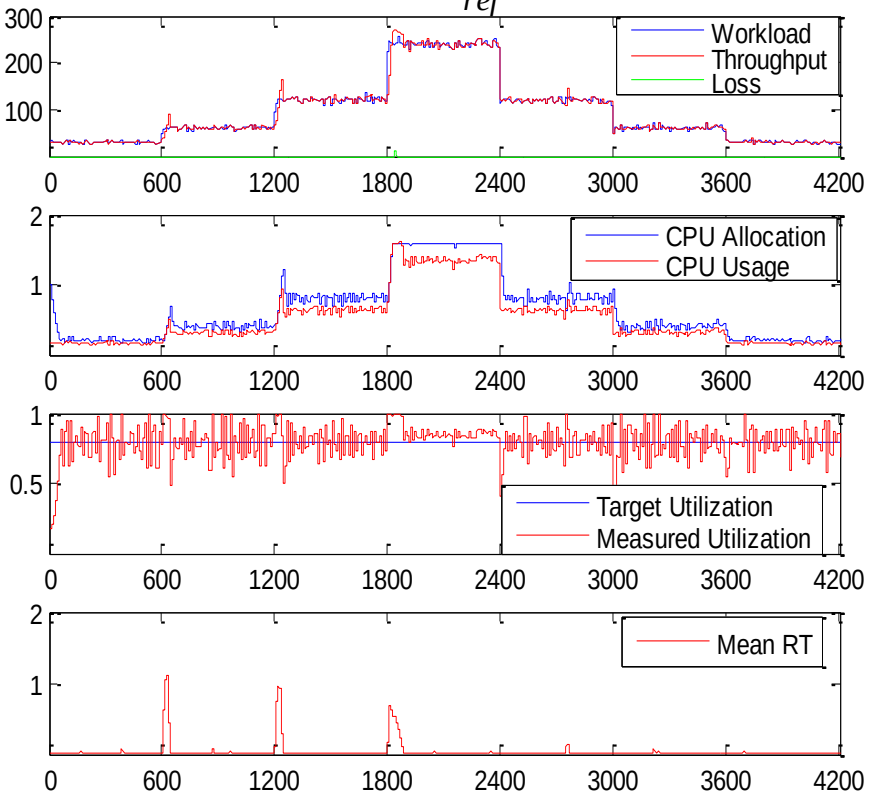
Comparison of feedback control vs adaptive control

$$r_{ref} = 0.8, K_I(k) = 2$$



CPU Alloc	MRT	Std of MRT
0.68	0.18	0.21

$$K_I(k) = 1.5\frac{v(k-1)}{r_{ref}}$$

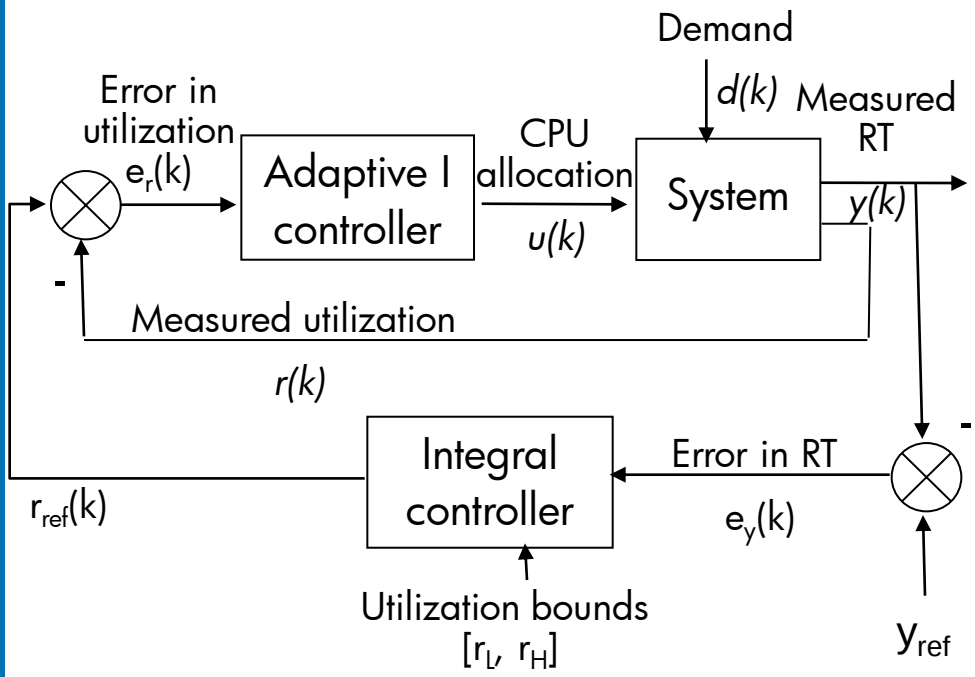


CPU Alloc	MRT	Std of MRT
0.64	0.03	0.13

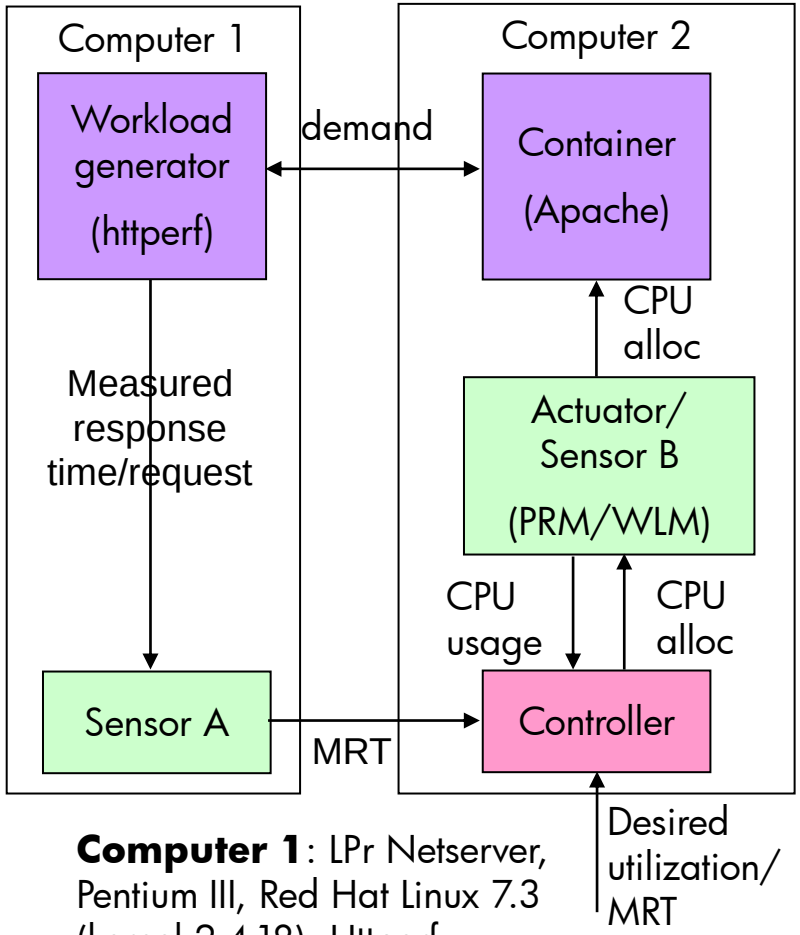
Response time control

Controller and setup

Control Structure:



Testbed:



Inner loop: $$u(k) = u(k-1) - \lambda \frac{v(k-1)}{r_{ref}} \left(r_{ref} - r(k-1) \right)$$

Outer loop: $$\tilde{r}_{ref}(j) = r_{ref}(j-1) + K_{I2} \left(y_{ref} - y(j-1) \right)$$

$$r_{ref}(j) = \min(r_H, \max(r_L, \tilde{r}_{ref}(j)))$$

X. Zhu et. al, "Utility driven workload management using nested control design," ACC'06.

Computer 1: LPr Netserver, Pentium III, Red Hat Linux 7.3 (kernel 2.4.18), Httperf

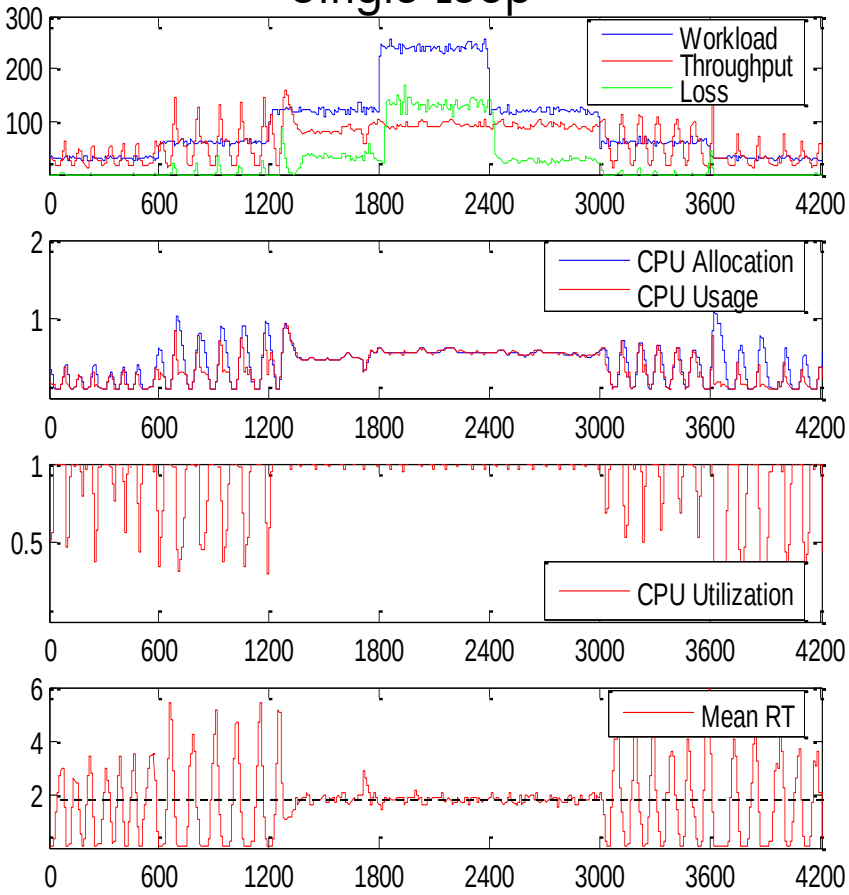
Computer 2: Dual processor, HP 9000-L server, PA-RISC processor, HP-UX B11.11, WLM 3.0, Apache server

LABS hp

Response time control:

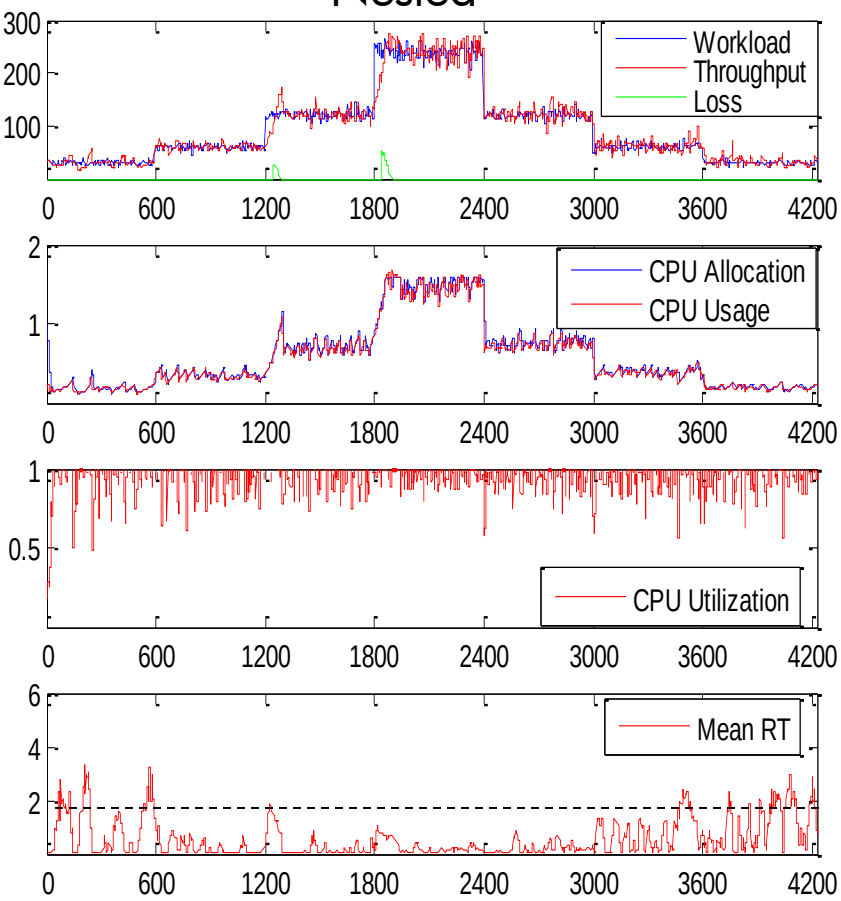
Comparison of feedback control vs nested adaptive control

Single Loop



MRT y_{ref} = 2 sec

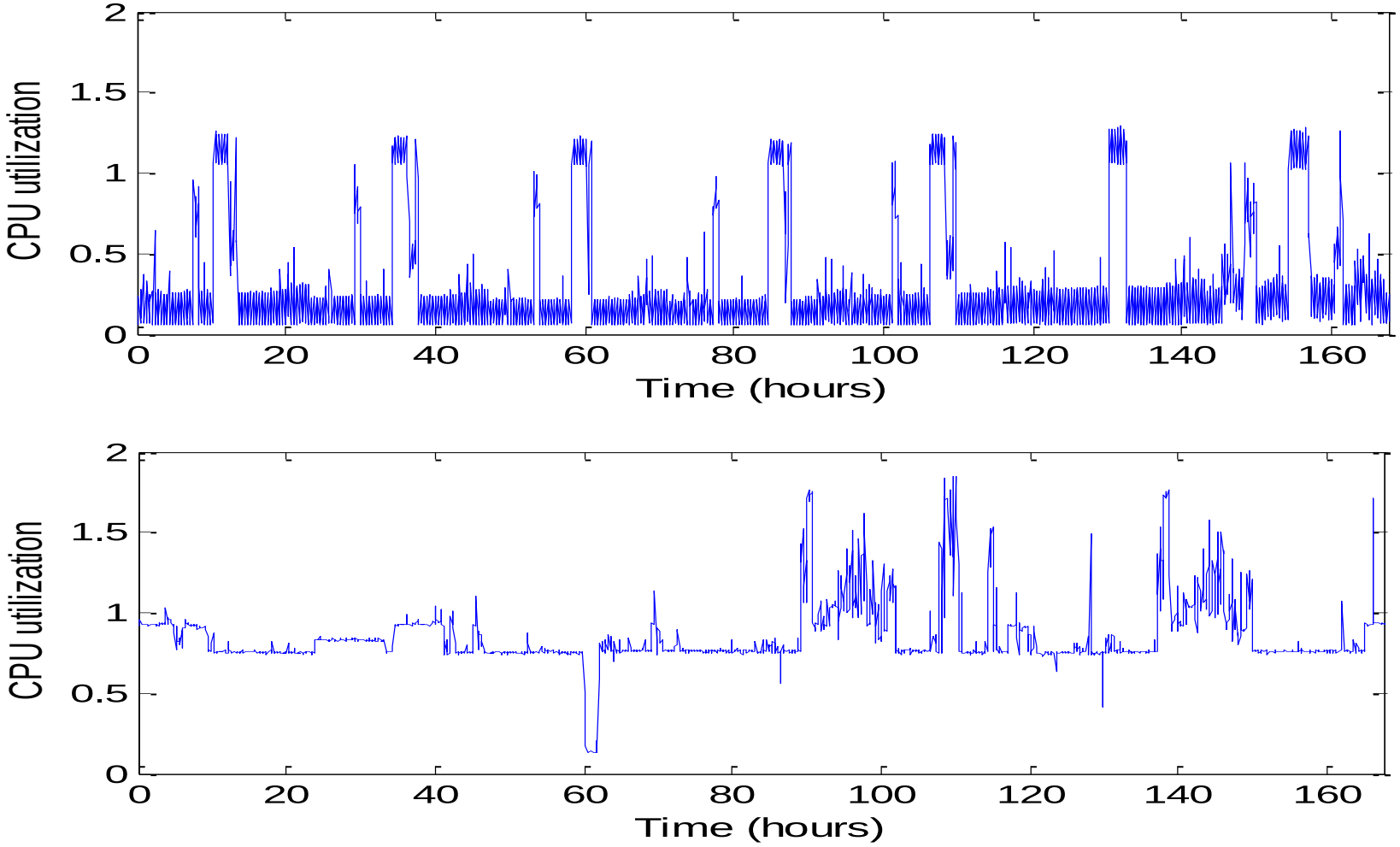
Nested



MRT y_{ref} = 2 sec,
Utilization bounds r_{ref}=[50%, 95%]

Predictive control

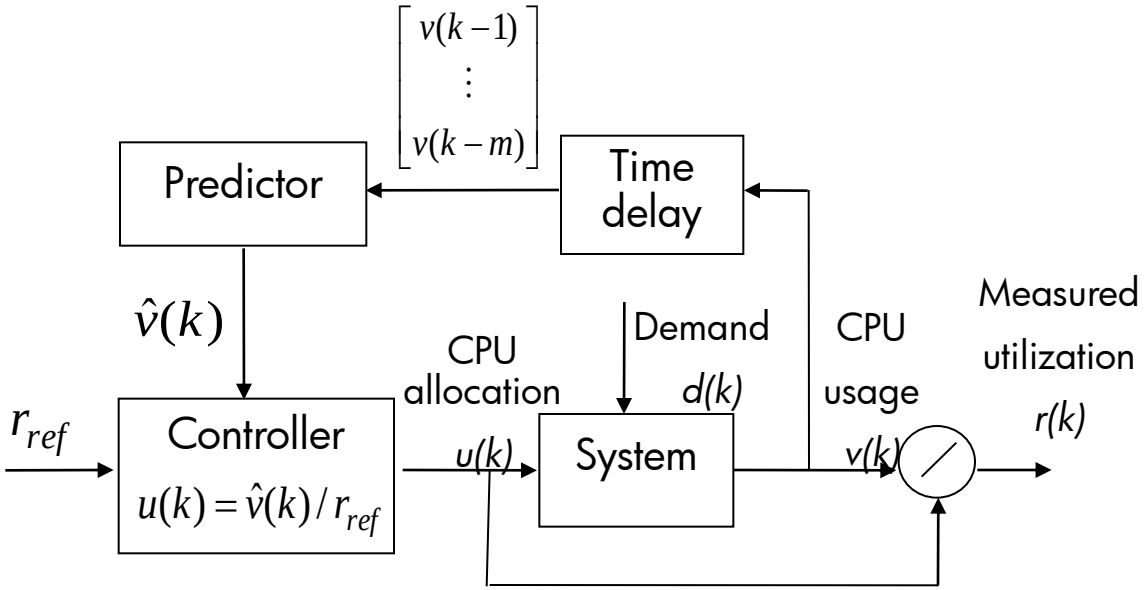
Enterprise workload



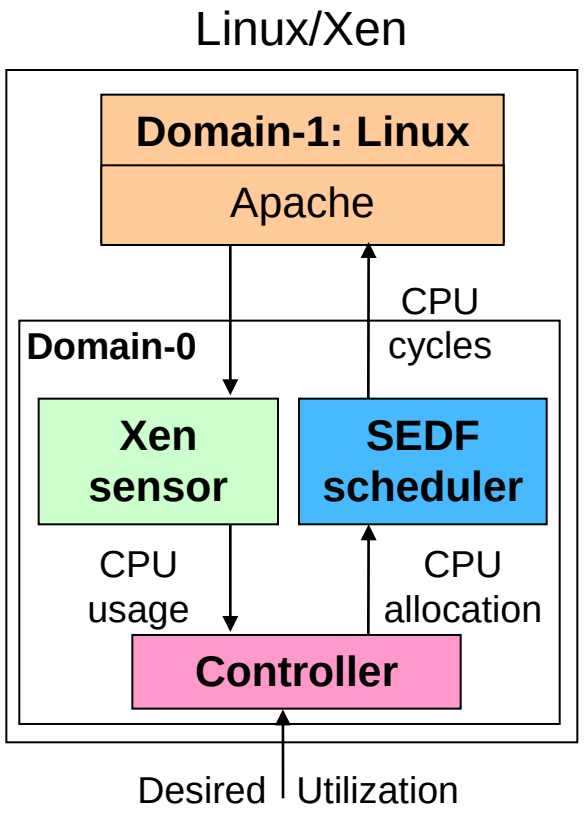
Predictive control (1)

Controller and setup

Control Structure:



Testbed:



Temporal Predictor:

A Multi-Pulse (MP) model for prediction:

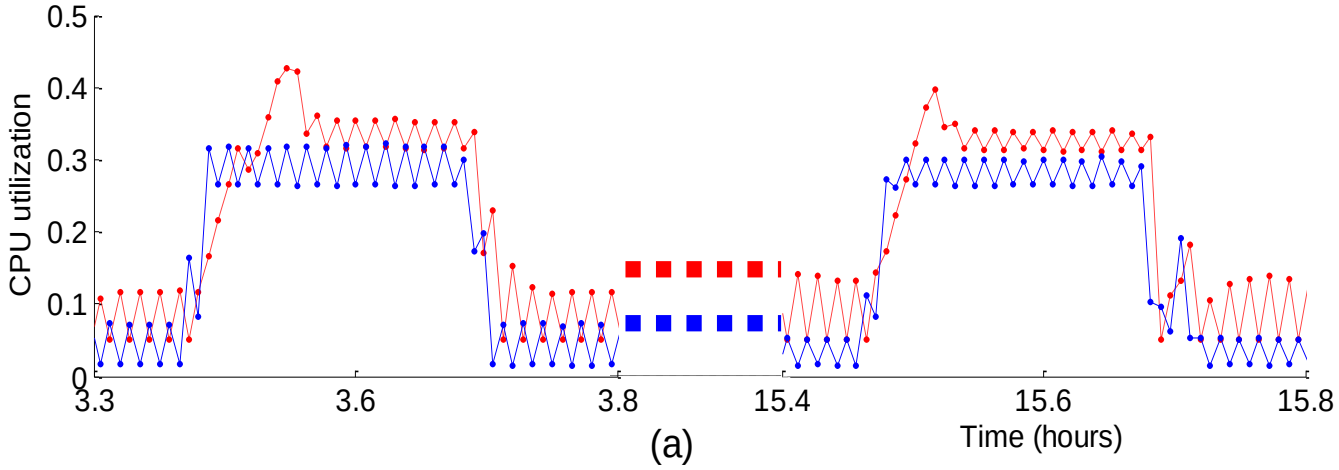
$$\hat{v}(k) = \sum_{i=1}^{m} a_i v(k - n_i)$$

W. Xu et. al, "Predictive control for dynamic resource allocation in enterprise data centers," NOMS'06.

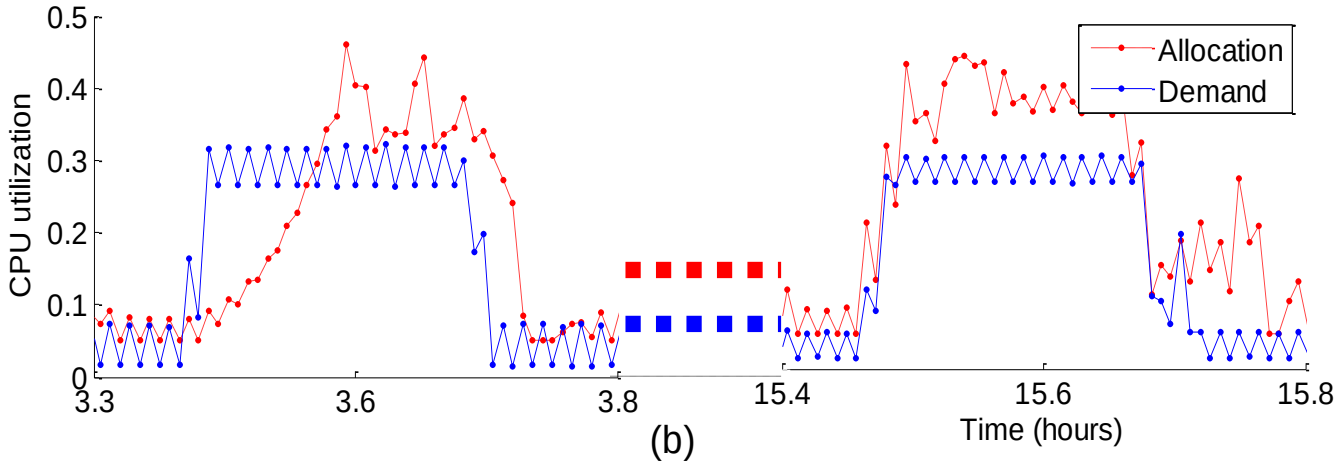
- HP ProLiant DL360 G4 dual Xeon processors
- Xen unstable version
-Apache web server
- Linux kernel 2.6.11
- httperf 0.8

Predictive Control

Comparison of feedback control vs predictive control



PI controller

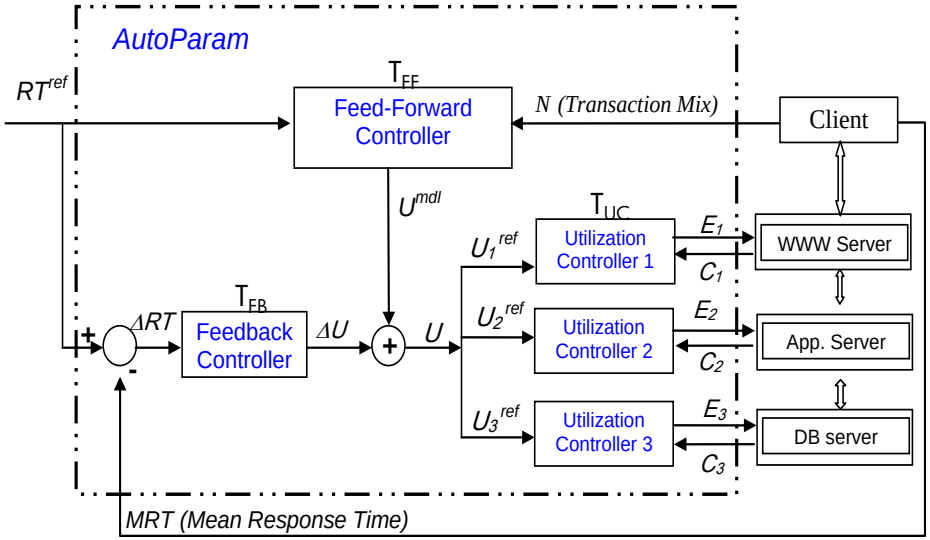


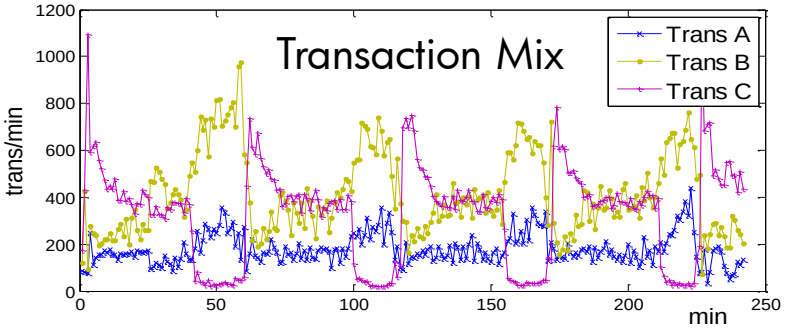
predictive controller

Predictive control (2)

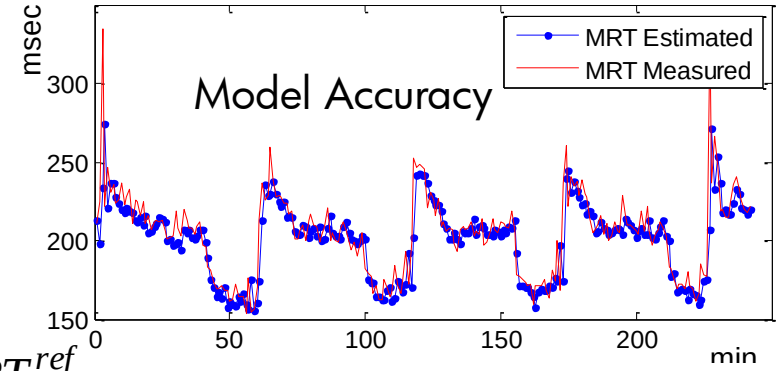
Controller and Model

Control Structure:





- Top 3 transaction types in an enterprise application
- Average workload intensity -- 25 transactions/sec
- These three types account for 62% of all transactions



$$E_t(k) = E_t(k-1) + G_t(k)(U_t^{ref} - U_t(k-1))$$

$$G_t(k) = -\beta C_t(k-1)/U_t^{ref}, \quad 0 < \beta < 2$$

$$\Delta U(m) = \Delta U(m-1) + G^{FB}(RT^{ref} - RT(m-1))/RT^{ref}$$

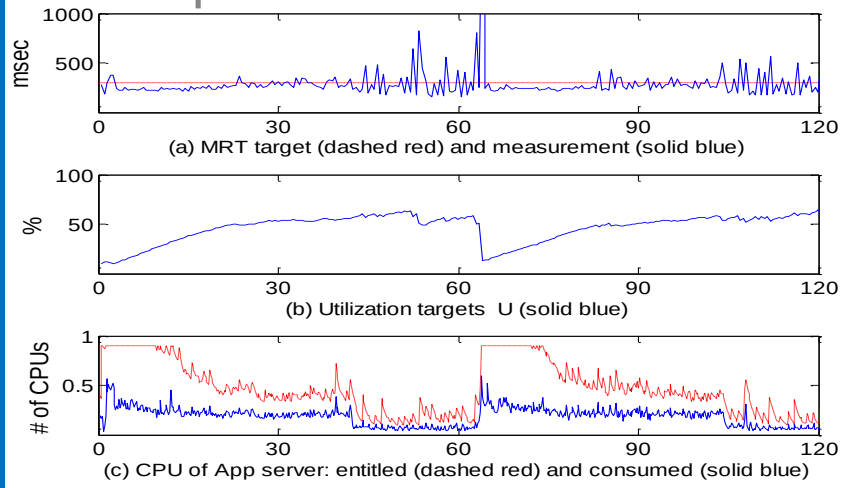
$$RT^{ref} = \frac{\sum_j \alpha_j N_{ij}}{\sum_j N_{ij}} + \frac{T}{\sum_j N_{ij}}\left(\sum_t \frac{U_{it}^{mdl2}}{1-U_{it}^{mdl}}\right)$$

$T_{UC} < T_{FB} < T_{FF}$ (10, 30, 90 sec)

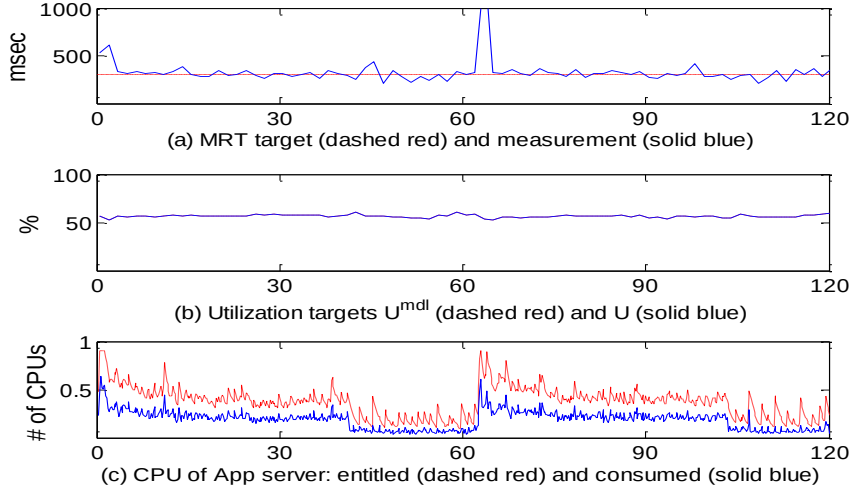
Z. Wang et. al, "AutoParam: Automated control of application level performance in virtualized server environments," FeBID'07.

Predictive Control (2):

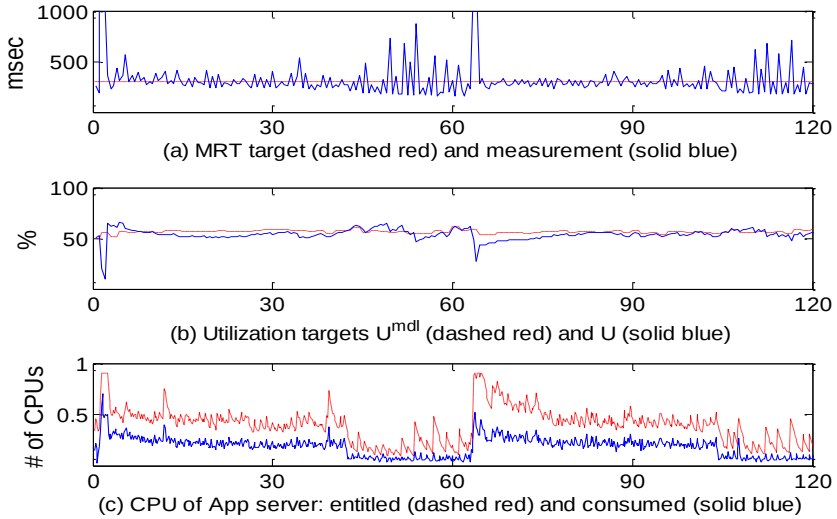
Comparison of feedback control vs feedforward control



Feedback only



Feedforward only



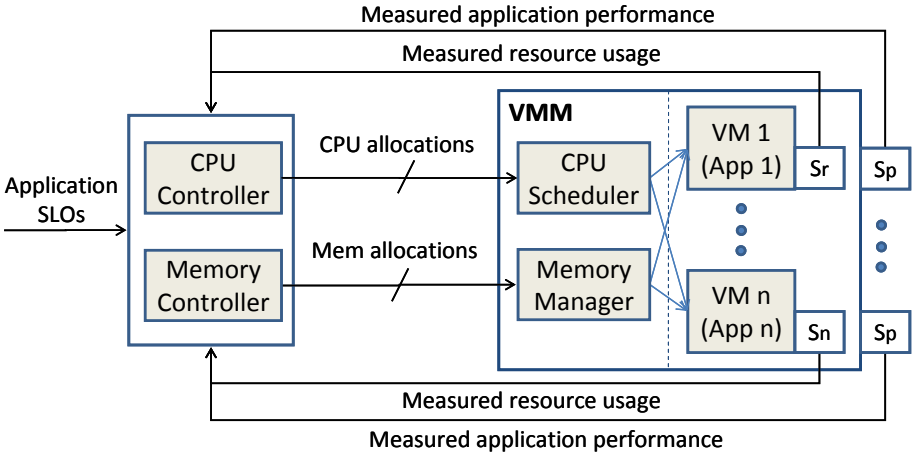
Feedback+Feedforward

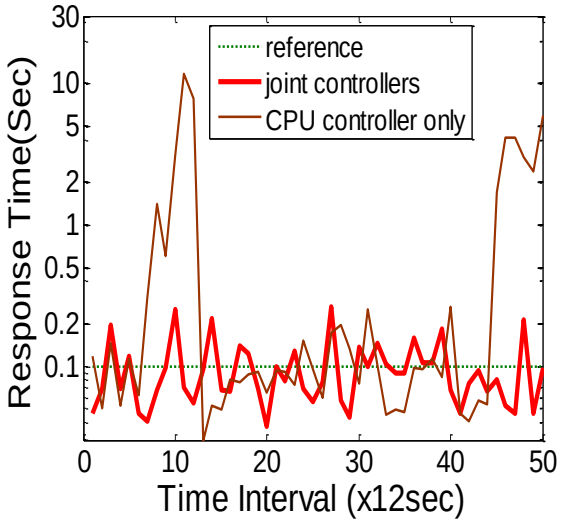
Controller designs (target, model)	MRT (ms)	Total entitlement (# of CPUs)
FF + UC (300ms, good model)	329	0.66
FF + UC (300ms, bad model)	357	0.67
AutoParam (300ms, good model)	320	0.70
AutoParam (300ms, bad model)	314	0.72

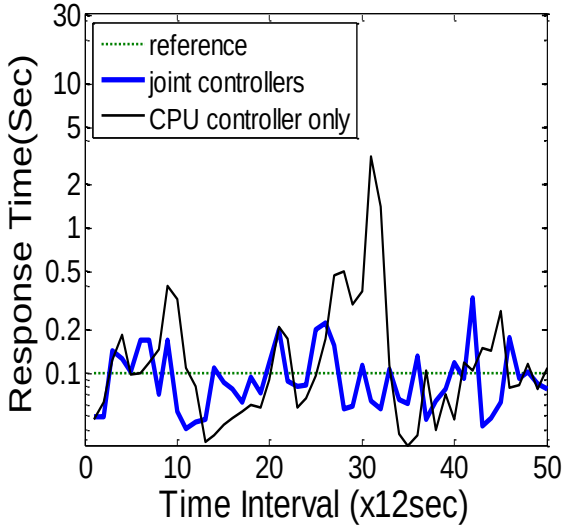
LABS hp

Multi-Input Multi-Output Control

Joint control of memory and cpu







Memory Controller

$$u_{mem}(k+1) = u_{mem}(k) - \lambda_{mem} v_{mem}(k)(r_{mem}^{ref} - r_{mem}(k))/r_{mem}^{ref}$$

CPU Controller

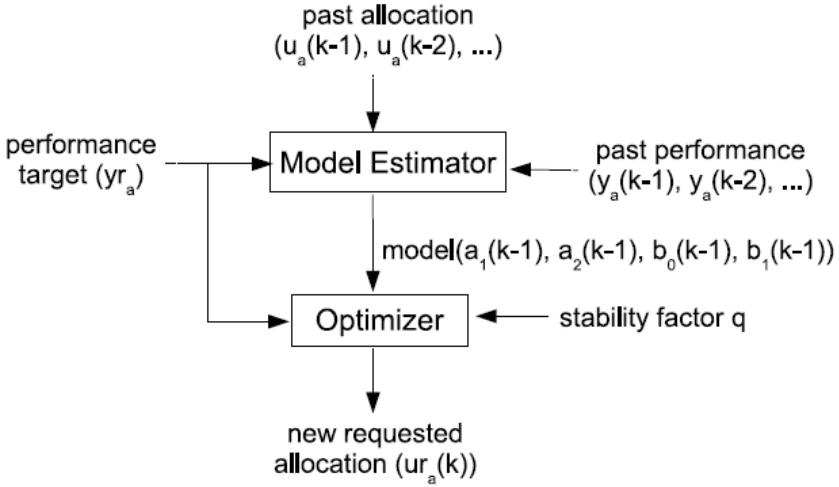
$$u_{cpu}(k+1) = u_{cpu}(k) - \lambda_{cpu} v_{cpu}(k)(r_{cpu}^{ref} - r_{cpu}(k))/r_{cpu}^{ref}$$

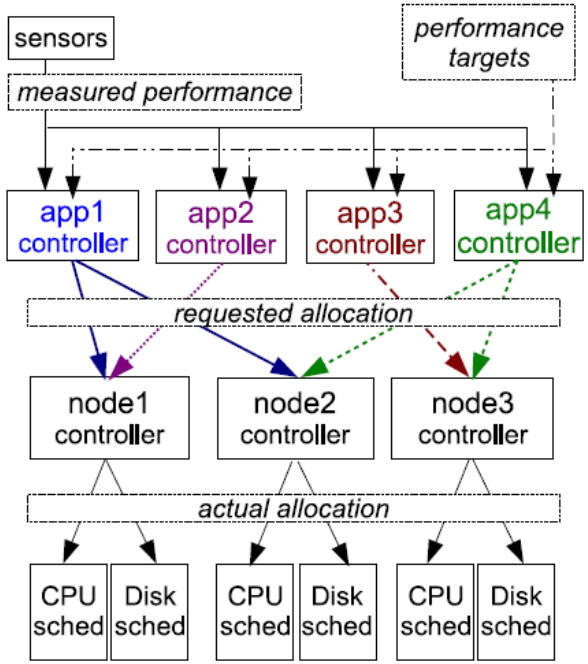
$$r_{cpu}^{ref}(i+1) = r_{cpu}^{ref}(i) + \beta(RT^{ref} - RT(i))/RT^{ref}$$

J. Heo et. al, "Memory overbooking and dynamic control of Xen virtual machines in consolidated environments," IM'09 mini-conference MC07.

Multi-Input Multi-Output Control (2)

Joint control of disk and cpu





Model

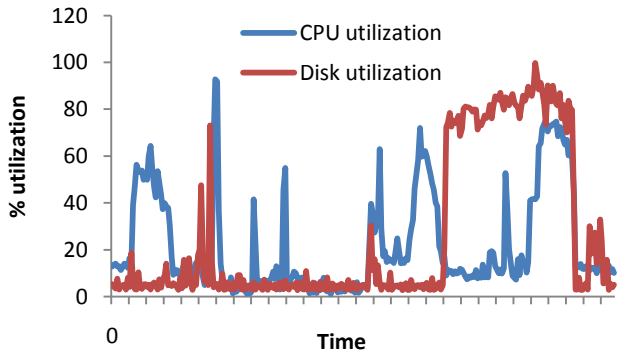
$$y_a(k) = a_1(k)y_a(k-1) + a_2(k)y_a(k-2) + \mathbf{b}_0^T(k)\mathbf{u}_a(k) + \mathbf{b}_1^T(k)\mathbf{u}_a(k-1)$$

Cost Function

$$J_a = (y_a(k)-1)^2 + q\left\|\overline{\mathbf{u}}_a(k) - \mathbf{u}_a(k-1)\right\|^2$$

Solution

$$\overline{\mathbf{u}}_a^*(k) = (\mathbf{b}_0(k)\mathbf{b}_0^T(k) + q\mathbf{I})^{-1}((1 - a_1(k)y_a(k-1) - a_2(k)y_a(k-2) - \mathbf{b}_1^T(k)\mathbf{u}_a(k-1))\mathbf{b}_0(k) + q\mathbf{u}_a(k-1))$$

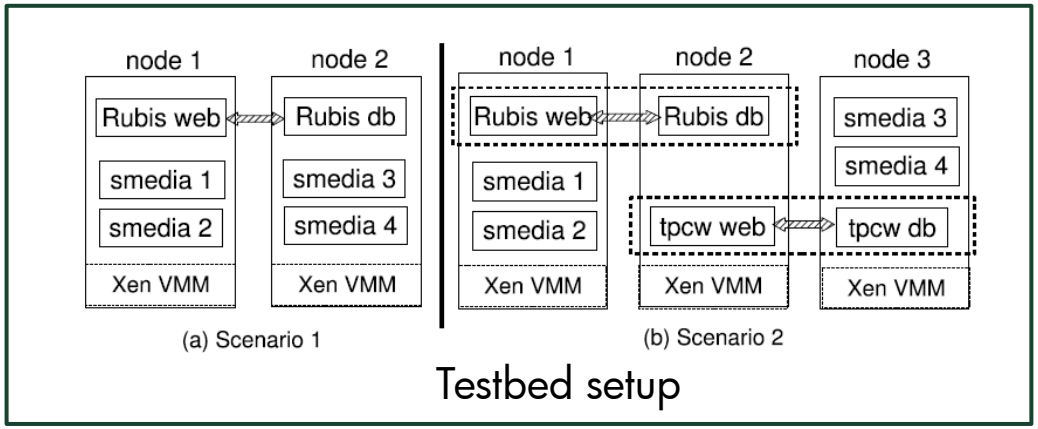


P. Padala et. al, "Automated control of multiple virtualized resources," Eurosys'09.

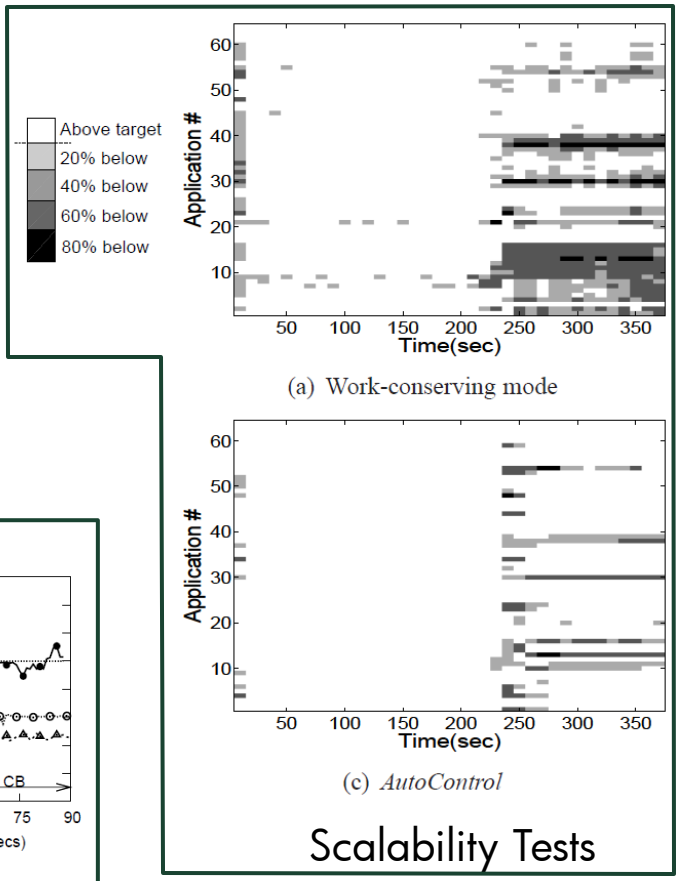
LABS hp

Multi-Input Multi-Output Control (2)

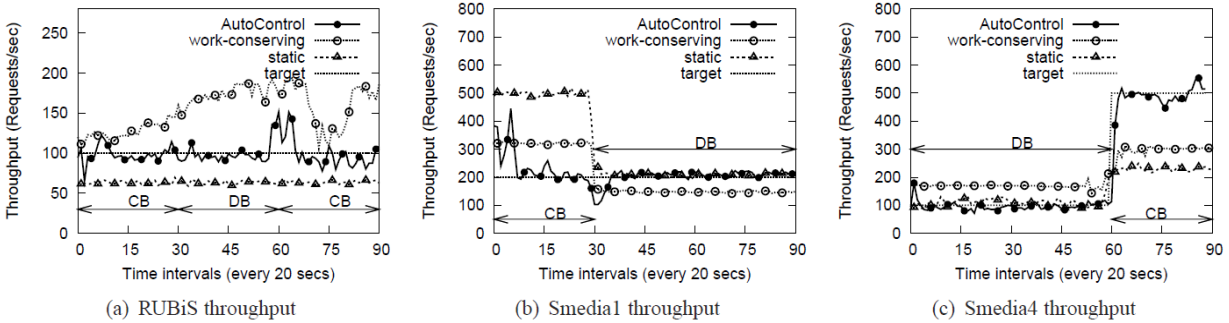
Joint control of disk and cpu



Testbed setup



Scalability Tests



Managing bottleneck changes

P. Padala et. al, "Automated control of multiple virtualized resources," Eurosys'09.

LABS hp

Summary

- Managing application performance in virtualized environments is a challenging problem
- Control theory based approaches offer solutions
  - On-line adaptation to environmental changes
  - Rapid response to transient disturbances
  - General techniques take advantage of significant body of literature
- Proven using case studies
  - Utilization control
  - Application response times and throughput
  - Performance differentiation
- Integration with other workload placement, admission control, and capacity planning techniques


LABS hp