Adaptive Virtual Resource Management with Fuzzy Model Predictive Control

Lixi Wang, Ming Zhao

Computing & Information Sciences
Florida International University
Miami, FL

Jing Xu, José Fortes

Electrical & Computer Engineering
University of Florida
Gainesville, FL

Outline

- Introduction
- Background
- Approach
- Evaluation
- Related work
- Conclusion

Introduction

- Resource management in virtualized systems (e.g., utility datacenters, clouds)
  - Multiple application VMs compete for shared resources
  - Application-level objectives: meet QoS targets
  - System-level objectives: maximize total revenue

- Goals
  - Accurately model system behaviors
  - Adaptively optimize resource allocations

Challenges

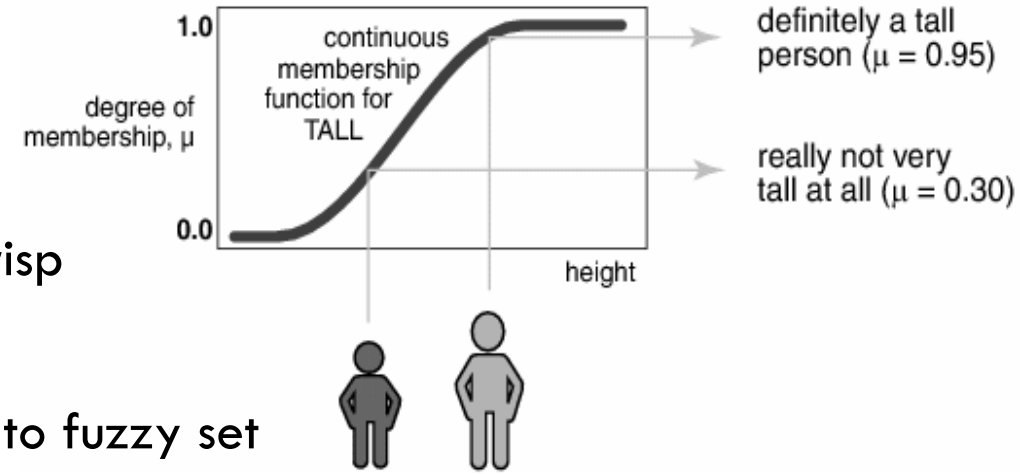
- Dynamic and complex resource usage behaviors in virtualized systems
  - Dynamics in application workloads
    - E.g., a database workload's both intensity and composition can change over time
  - Interference among co-hosted VMs
    - E.g., competition for shared LLC and I/O
  - Changes in application- and system-level objectives
    - E.g., SLOs can change over time

Proposed solution

- Fuzzy Model Predictive Control (FMPC)
  - Fuzzy-logic-based modeling
    - Efficiently capture complex resource usage behaviors
  - Predictive control
    - Quickly adapt to changes and sustain optimization goals

Background – Fuzzy modeling

- Fuzzy logic [4]
  - Fuzzy set
    - Vague linguistic term, not crisp
  - Membership function (MF)
    - Degree of the membership to fuzzy set
  - Fuzzy rule
    - If *x* is *A* then *y* is *B*
    - *A*, *B* are fuzzy sets associated with MFs
    - Different from event-condition-action rules



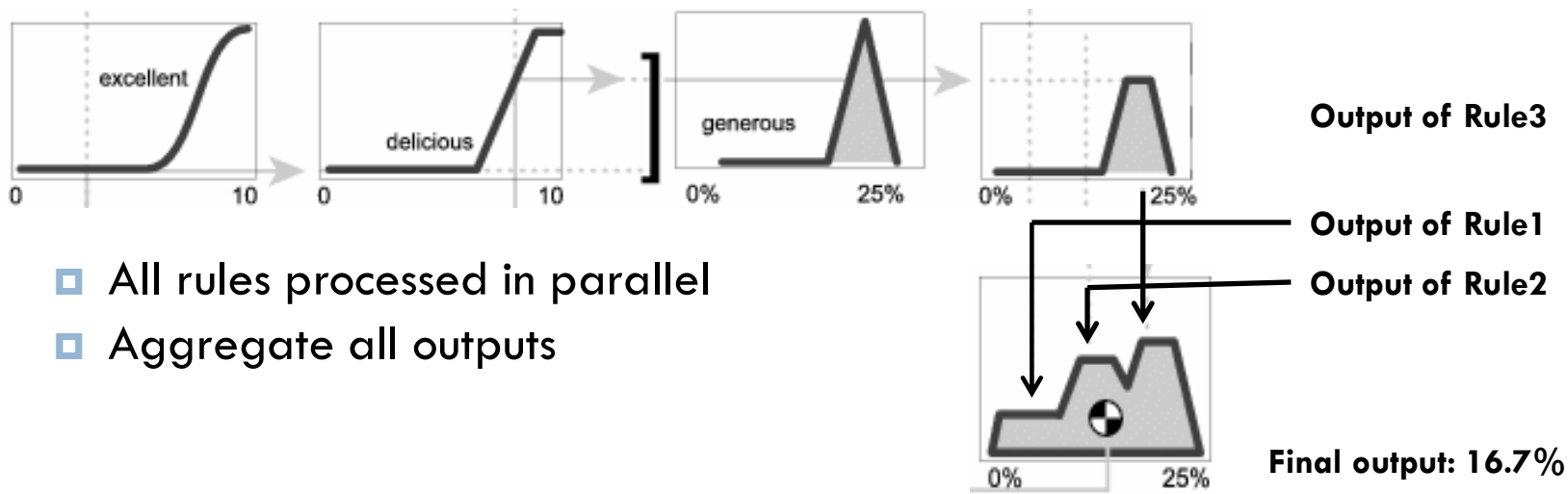
A fuzzy set with its MF[3]

[4] L. A. Zadeh, "Fuzzy Sets", Information and Control, 1965

Background – Fuzzy modeling

- Fuzzy model – a collection of fuzzy rules
  - E.g., A model for tipping problem[3]
    - Rule1: If service is poor or the food is rancid, then tip is cheap
    - Rule2: If service is good, then tip is average
    - Rule3: If service is excellent or food is delicious, then tip is generous

- Fuzzy inference – mapping from a given input to an output on a fuzzy rule base
  - Given input <service=3, food=8>, inference process of rule 3



  - All rules processed in parallel
  - Aggregate all outputs

[3] http://www.mathworks.com/help/toolbox/fuzzy/bp78l6_-1.html#bp78l70-2

Background – Fuzzy modeling

- Fuzzy-logic based modeling of virtualized system
  - Model the relationship between resource allocation and application performance
  - Automatically learn complex system behaviors without assuming any a priori knowledge about the system
  - Provide input for a resource controller to determine the optimal resource allocation

Background – Model predictive control

- Model predictive control [1]
  - Basic phases
    - Modeling: Build a model to capture the relationship between control input (e.g., resource allocation) and system output (performance)
    - Optimization: Take the control action that optimizes an objective function (e.g., maximized total revenue) according to the predictions using the model
  - Performed iteratively online to adapt to changes in system behaviors

[1] J. Maciejowski, "Predictive Control with Constraints," Prentice Hall, 1 edition, 2002.

Background – Model predictive control

- General model in Modeling

$$y(t) = \Phi(u(t), \ldots, u(t-m), y(t-1), \ldots, y(t-n))$$

  - Control input: $\boldsymbol{u}$ – resource allocation
  - Output: $\boldsymbol{y}$ – application performance
  - Capture system behaviors and predict performance
- Objective function in optimization

$$J(t) = \sum_{i=1}^{P} \left\| \Delta y(t+i|t) \right\|^2 Q(i) + \sum_{i=0}^{M-1} \left\| \Delta u(t+i|t) \right\|^2 R(i)$$

Accuracy factor

Stability factor

Prediction error

Control effort

Background – Model predictive control

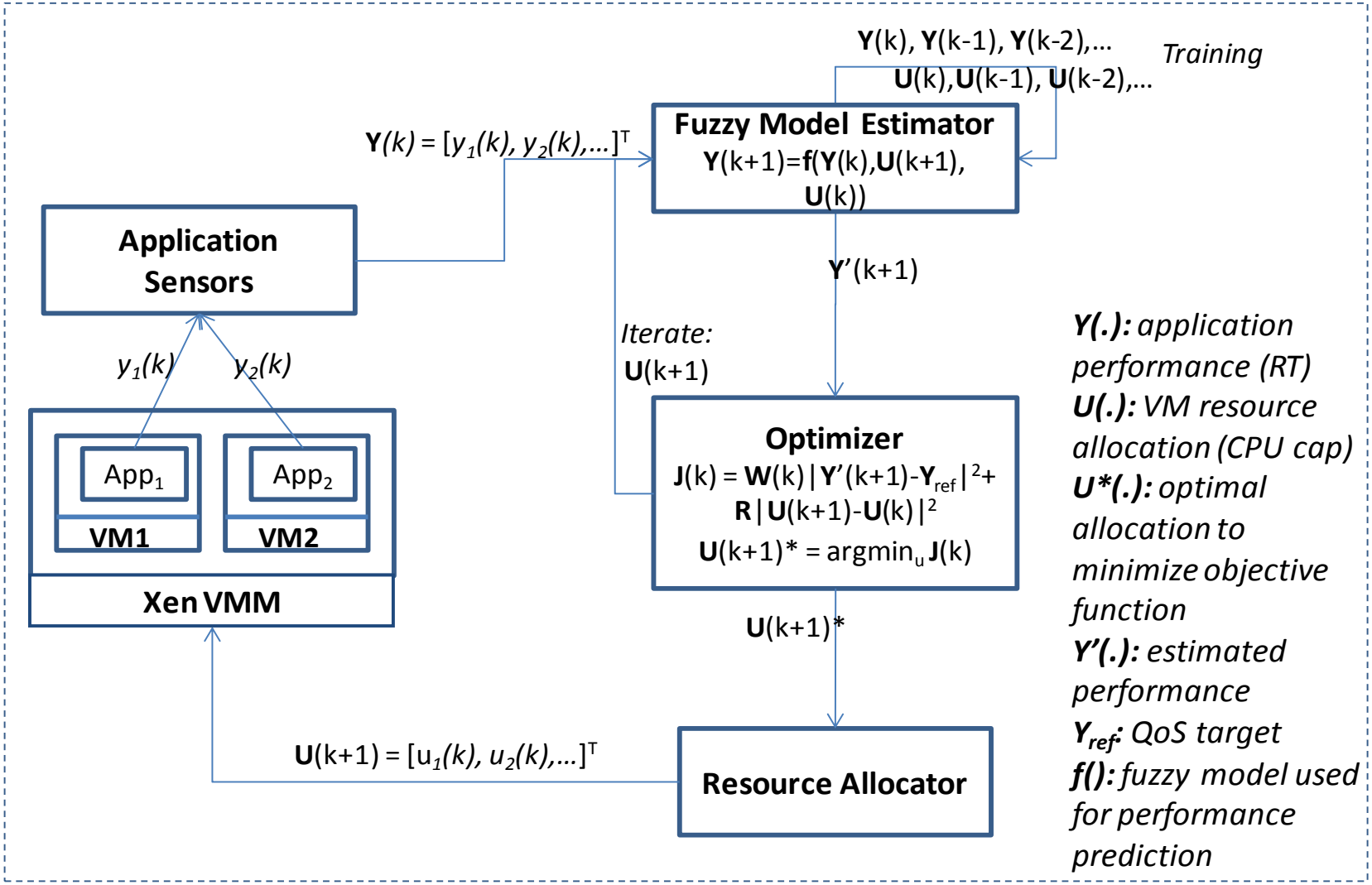
- Most approaches use linear models
  - Can only approximate nonlinear system behaviors within a small region of the operation space
  - Need to be updated when the operating point changes

- We propose Fuzzy Model Predictive Control (FMPC)
  - Effectively capture the complexity in system behaviors
  - Perform optimized control over the entire operating space
  - Inherit the benefits of traditional predictive control

FMPC based virtual resource management

- Fuzzy model estimator
  - Create fuzzy model to capture the resource usage behavior of VMs
  - Predict the performance of virtualized applications for given resource allocations
- Optimizer
  - Produce the resource allocation solution that optimizes predefined objective function
    - E.g., maximize the total revenue of the virtualized system

Architecture



FMPC – Fuzzy model estimator

- Problem statement
  - Given a resource provider that hosts q applications by multiplexing p types of resources via VMs
  - A general MIMO performance model

$$y(t) = \Phi(u(t), \ldots, u(t-m), y(t-1), \ldots, y(t-n))$$

  - Input: resource allocation of p types of resources at time t

$$u(t) = [u_1(t), u_2(t), \ldots, u_N(t)]^T \quad (N = pq)$$

  - Output: predicted performances of q applications at time t

$$y(t) = [y_1(t), y_2(t), \ldots, y_q(t)]^T$$

[5] T. Takagi, M. Sugeno, "Fuzzy identification of systems and its application to modeling and control," *TSMC*, 1985.

Fuzzy model estimator

- Linear model in MPC

$$y(t) = \Phi(u(t), y(t-1)) = au(t) + by(t-1)$$

  - ***a, b*** are fitting parameters
  - Approximate the nonlinear behaviors around the current operating point

- Fuzzy model in MPC
  - A set of Takagi-Sugeno[5] type fuzzy rules

$$R^i : If\ u(t)\ is\ A^i\ and\ y(t-1)\ is\ B^i$$

$$then\ y^i(t) = a_i u(t) + b_i y(t-1)$$

T. Takagi, M. Sugeno, "Fuzzy identification of systems and its application to modeling and control," *TSMC*, 1985

Model creation

- Data clustering
  - Cluster training data to build a concise rule base [8]
    - Subtractive clustering — efficient, one-pass algorithm without assuming the number of clusters
  - Each cluster exemplifies a representative characteristic of the system behaviors and determines a fuzzy rule
  - Can be done adaptively online
    - Both the number of clusters and their centers can be adapted

1. S. Chiu, "Fuzzy Model Identification Based on Cluster Estimation," Journal of Intelligent and Fuzzy Systems, 1994.

Model creation

- Fuzzy rule base

Control inputs

Fuzzy sets

$$R^i : If\ u(t)\ is\ A^i\ and\ y(t-1)\ is\ B^i$$

$$then\ y^i(t) = a_i u(t) + b_i y(t-1)$$

Fitting parameters

  - Collect the control inputs online
  - Clustering data sample
  - Each cluster defines a fuzzy rule with fuzzy sets associated with Gaussian MFs
  - Train the fitting parameters based on collected data

Model prediction

- Invoked by the *Optimizer* at control period t -1
- Given a specific control input < $\boldsymbol{u}(t)$, $\boldsymbol{y}(t\text{-}1)$ >, apply *fuzzy inference* to generate output $\boldsymbol{y(t)}$
  - Calculate membership grade $\boldsymbol{\delta}^i$ for each rule
  - Compute consequent output $\boldsymbol{y}^{\boldsymbol{i}}(t)$ for each rule
  - Generate final output $y(t) = \sum_{i=1}^{S} \delta^i y^i(t)$

FMPC – Optimizer

- Work with *Estimator* in closed-loop
- Call *Estimator* iteratively to search for the optimal $\boldsymbol{u}(t)$ at the end of each control period t-1
- Optimize the objective function
  - Can consider both system- and application-level objectives

Objective function

$$J(t) = Q\left\| w \bullet (y(t+1) - y_{ref} \right\|^2 + R\left\| (u(t+1) - u(t) \right\|^2$$

$$= Q\sum_{i=1}^{q}\left[w_i(y_i(t+1) - y_{refi}\right]^2 + R\sum_{i=1}^{N}\left[(u_i(t+1) - u_i(t)\right]^2$$

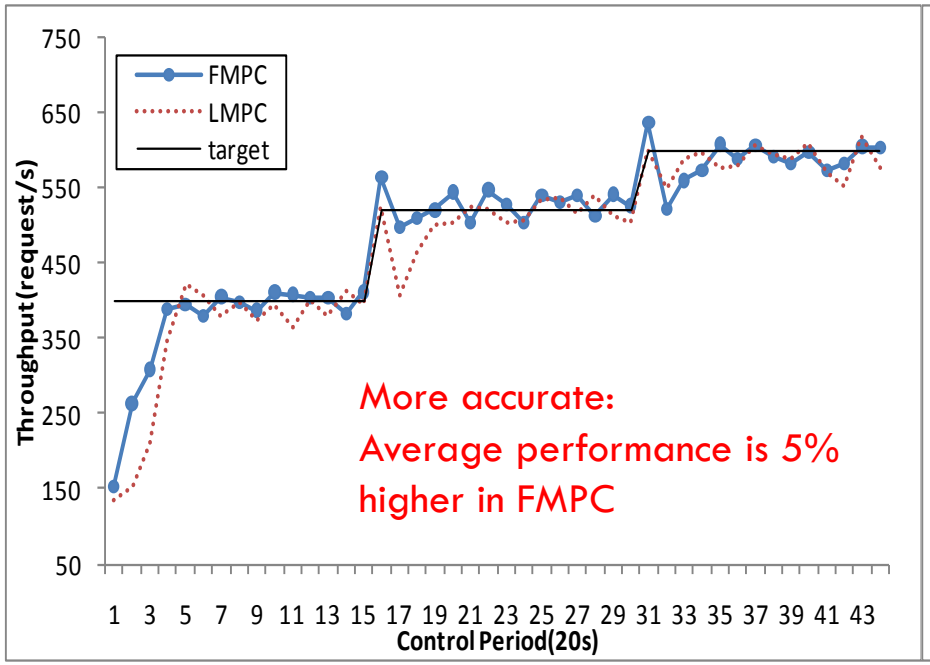
- *Prediction error* factor Q: track accuracy in performance targeting
- *Control effort* factor R: maintain stability in control operation
- $\boldsymbol{w} = [w_1(t), w_2(t), \ldots w_q(t)]^{\mathrm{T}}$: preferences given to the applications
- $\boldsymbol{y}_{ref}$: QoS targets for applications

Evaluation

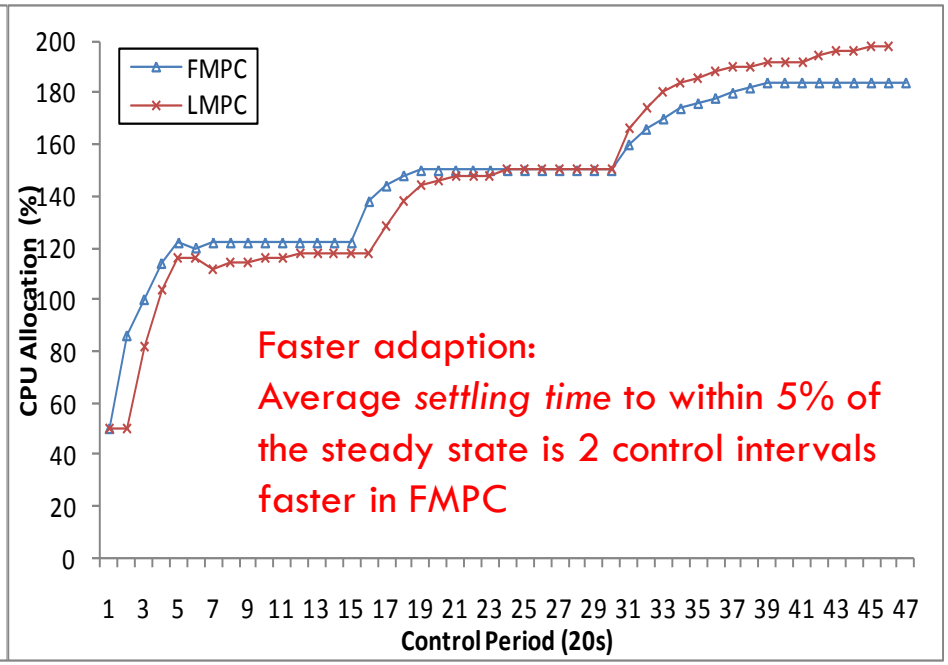
- Setup
  - Test bed: a quad-core Intel Q6600 2.4GHz physical machine, 4GB RAM and 142GB SATA disk storage
  - Virtualized environment: Xen 3.3.1
  - DomU runs the benchmark
    - 2 virtual CPUs, 1.7G RAM
    - RUBiS web-tier and database-tier
  - Dom0 runs the resource management system
  - FMPC vs. LMPC (linear model predictive control)

Experiments with changing workload

- Setup: A workload with dynamically changing intensity
- Results: FMPC outperforms LMPC in accuracy and speed



Performance for workload with changing intensity

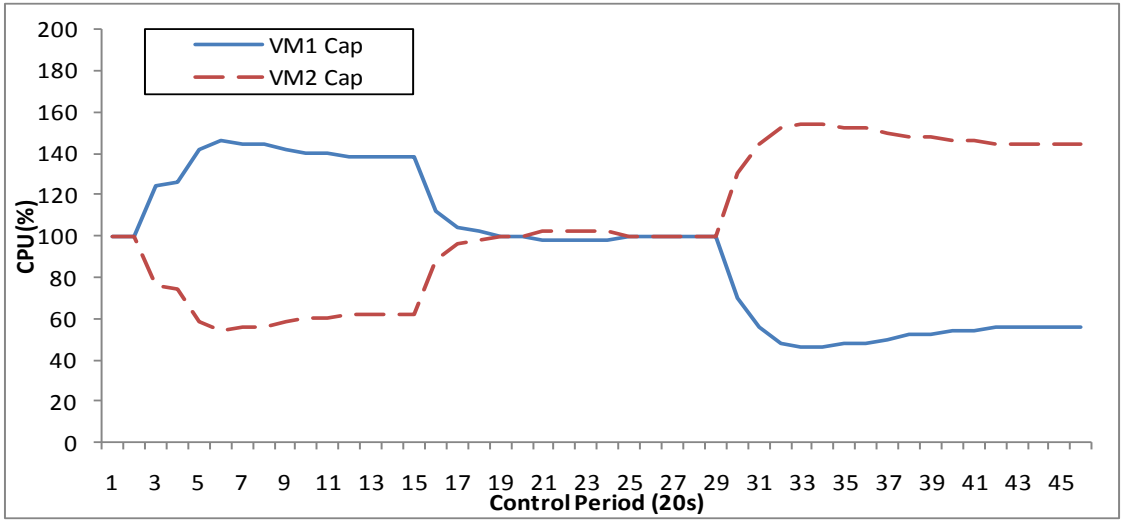


CPU allocation for workload with changing intensity

Experiments with interfering VMs

- Interference
  - Two independent RUBiS VMs running concurrently
  - Neither VM can achieve the same level performance as when running alone
- Setup:
  - Fix workload intensity but vary VMs' weights w_1, w_2
- Allocation schemes:
  - FMPC-based
  - LMPC-based
  - Proportionally to the weights

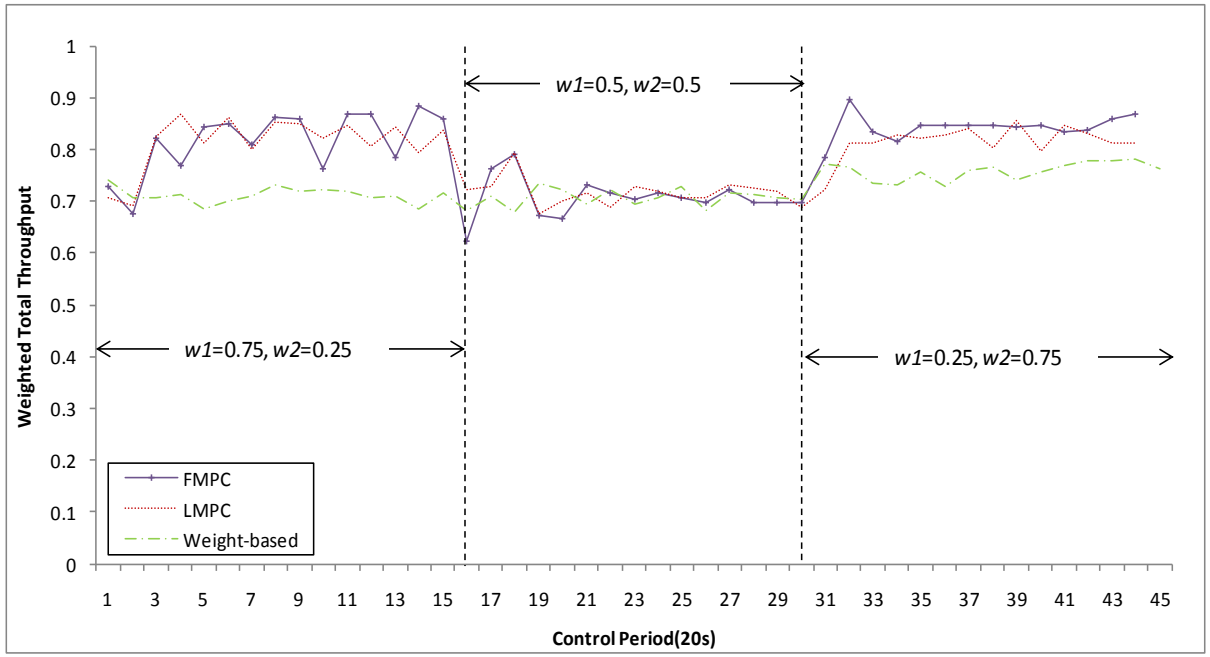
Experiments with interfering VMs



CPU allocations for interfering VMs using FMPC

- Nonlinearity: Cap1 and Cap2 is not proportional to their weight ratio; 140% : 60% ≠ 3: 1

Experiments with interfering VMs



Weighted total throughput for interfering VMs

- Outperform the weight-based scheme in all phases
- Achieve about 4.7% more throughput in the third phase than LMPC

Related work

- **Modeling-based resource management**
  - Regression analysis for memory allocation
    - J. Wildstrom et al., "CARVE: A Cognitive Agent for Resource Value Estimation", ICAC 2008
  - Reinforcement learning for CPU and memory configurations
    - J. Rao et al., "VCONF: A Reinforcement Learning Approach to Virtual Machines Auto-configuration", ICAC09
  - Artificial NN for resource allocation and interferences between VMs
    - S. Kundu et al., "Application Performance Modeling in a Virtualized Environment," HPCA 2010
  - **Our approach can adapt to changes more quickly**

- **MPC-based resource management**
  - Use MIMO model to achieve QoS differentiations
    - X. Liu et al, "Optimal Multivariate Control for Differentiated Services on a Shared Platform," CDC 2007
  - Use ARMA model for each application
    - P. Padala et al., "Automated Control of Multiple Virtualized Resources," EuroSys 2009
  - Capture the LLC interference between VMs
    - R.Nathuji et al., "Q-Clouds: Managing Performance Interference Effects for QoS-Aware Clouds," EuroSys, 2010
  - **Our approach can capture nonlinearity more effectively**

Conclusion

- A new fuzzy modeling based predictive control (FMPC) approach
  - Fuzzy-logic based modeling for capturing complex system behaviors
  - MPC-based resource control for agile system optimization and adaption to changes in the system
- Accurately estimate the resource allocation for a VM hosting changing workload and meet the desired QoS
- Effectively handle resource competing VMs and optimize their resource allocations

References

1. J. Maciejowski, "Predictive Control with Constraints," Prentice Hall, 1 edition, 2002.
2. P. Padala et al., "Automated Control of Multiple Virtualized Resources," Proceedings of EuroSys, 2009.
3. Matlab fuzzy logic URL: http://www.mathworks.com/help/toolbox/fuzzy/bp78l6_-1.html#bp78l70-2
4. L. A. Zadeh, "Fuzzy Sets", Information and Control, 1965.
5. T. Takagi, M. Sugeno, "Fuzzy identification of systems and its application to modeling and control," *TSMC*, 1985.
6. X. Liu et al, "Optimal Multivariate Control for Differentiated Services on a Shared Hosting Platform," CDC, 2007.
7. R.Nathuji et al., "Q-Clouds: Managing Performance Interference Effects for QoS-Aware Clouds," EuroSys, 2010.
8. S. Chiu, "Fuzzy Model Identification Based on Cluster Estimation," Journal of Intelligent and Fuzzy Systems, 1994.
9. J. Wildstrom et al., "CARVE: A Cognitive Agent for Resource Value Estimation", ICAC, 2008
10. J. Rao et al., "VCONF: A Reinforcement Learning Approach to Virtual Machines Auto-configuration", ICAC, 2009.
11. S. Kundu et al., "Application Performance Modeling in a Virtualized Environment," HPCA, 2010

Acknowledgement

- Research teams
  - VISA lab at FIU: Lixi Wang, Dr. Ming Zhao
  - ACIS lab at UF: Jing Xu, Dr. José Fortes
- Sponsors
  - National Science Foundation grants: CCF-0938045, OCI-0910812, IIP-0932023, CNS-0855123, IIP-0758596
  - Department of Homeland Security grant: 2010-ST-062-000039

- Thank you!