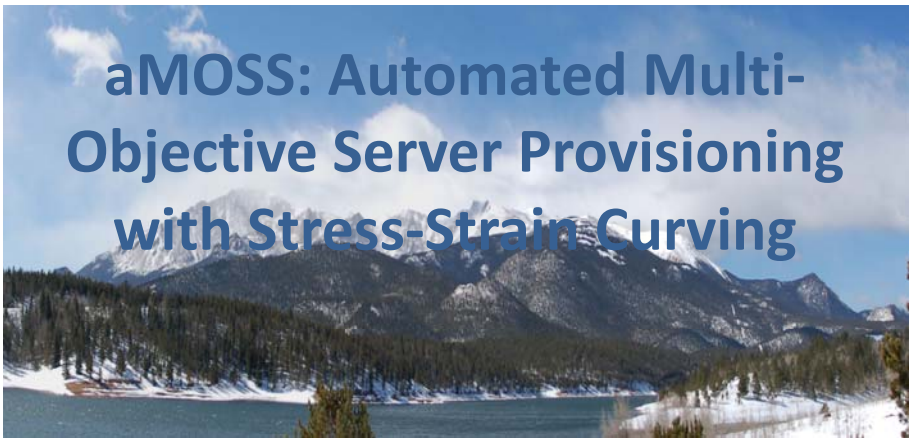


aMOSS: Automated Multi-Objective Server Provisioning with Stress-Strain Curving

Palden Lama and Xiaobo Zhou
University of Colorado at Colorado Springs

Outline

- Background and Motivation
- Challenges
- Related Work
- Proposed Approach
- Performance Evaluation
- Summary

Data Centers

- Data centers are the next computing platform
  - the backbone of a wide variety of services offered via the Internet
  - need to support scalability, availability and flexibility of Internet services
  - associated costs : infrastructure cost, power consumption, and cooling are major components in a data center's cost breakdown.

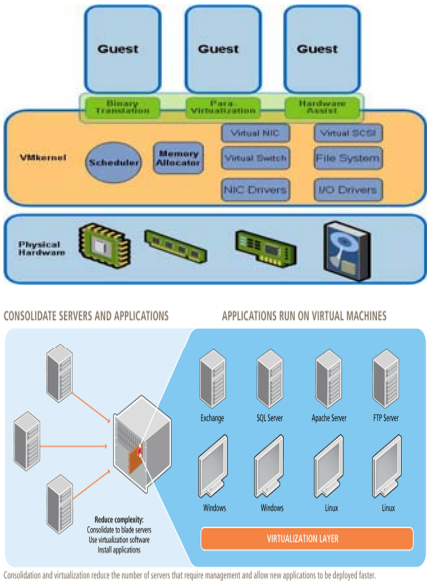
Key Issues:

- low return on investment: most servers in a typical data center run at only 5-10 percent utilization (**over-provisioning**)
- data center electricity usage doubled (11.6~24.2 billion kwh/year) within six years: 2000~ 2006 (US Environmental Protection Agency)
- IEA updated a warning in 5/2009 that ICT energy use could double by 2022, and triple by 2030
- Google search generated ~7g carbon emission; 200m search/day → 70000 car's CO2 emission

3

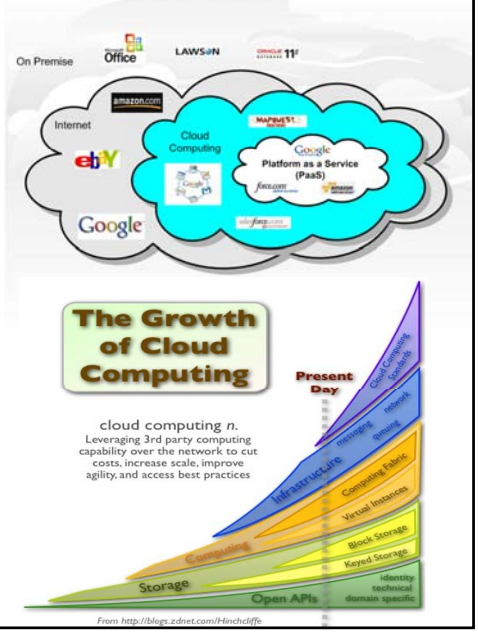
Virtualized Data Centers

- Virtualization
  - resources of physical servers can be abstracted into multiple virtual machines (VMs).
  - VMs can run diverse operating systems and applications as if they were running on physically separate machines.
- Consolidation
  - improves server utilization
  - reduces power consumption



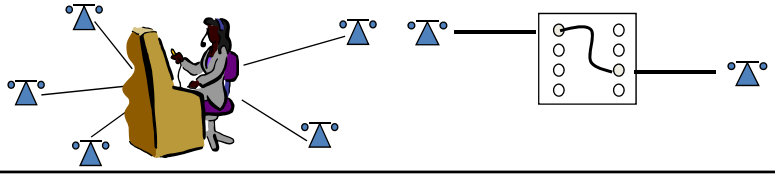
Virtualization enables Cloud Computing

- Flexible and Fine-grained Resource Allocation
  - VM resizing
  - VM migration
- Platform for Cloud Computing
  - outsource computing needs to 3rd party over the Internet
  - on-demand, pay-per-use service



Performance Management Issue

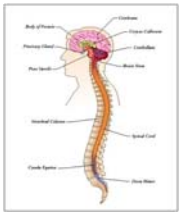
- How to meet SLA with clients, while maintaining resource utilization efficiency and reducing power consumption costs ?
- Difficult to manage highly complex systems such as virtualized data centers.
- Manual efforts require high expertise on workload profiles and underlying computing infrastructure
- Manual intervention may be even infeasible due to unpredictable variability of workload and system dynamics.
- Think about the telephony in 1920s. Automatic branch exchanges were introduced to eliminate the need for human intervention.



Autonomic Computing

- Seeks to improve computing systems with an aim of decreasing human involvement [IEEE/ACM ICAC]
  - "Computing systems have reached a level of complexity where the human effort required to get the systems up and running and keeping them operational is getting out of hand". [ACM Computing Surveys 2008]
- The term "autonomic" comes from biology

  In the human body, the autonomic nervous system takes care of unconscious reflexes, that is, bodily functions that do not require our attention, for example bodily adjustments such as the size of the pupil, the digestive functions of the stomach and intestines, the rate and depth of respiration,…



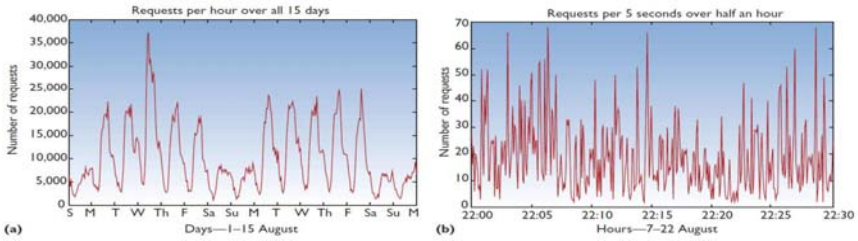
- The term "autonomic computing" describes computing systems that are said to be self-managing [IBM, 2001].
  - self-configuration
  - self-optimization
  - self-healing
  - self-protection

Multi-objective optimization of a virtualized data center

- Data centers have multiple correlated and conflicting objectives. e.g performance vs. resource allocation efficiency.
- Traditionally, utility optimization technique assigns certain weights to each objective expressed as local utility function, and combine them to optimize a global utility function.
  - It is difficult to find suitable weights and local utility functions. [ACM Computing Surveys 2008]
  - We argue that it is even inefficient to apply static weights in the face of dynamic workloads.

Challenges - Workload variability

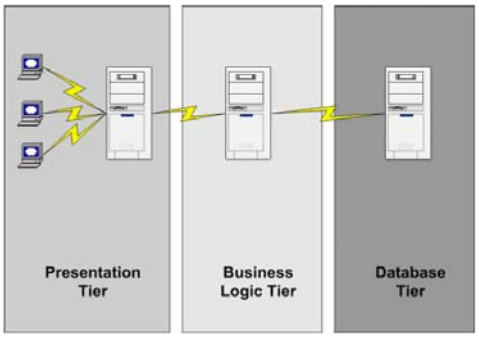
- Workload Variation at multiple time scales



Need for automated decision making for achieving multiple objectives of a data center.

Challenges: Multi-tier architecture

- Data centers host multi-tier applications, with end-to-end performance needs.

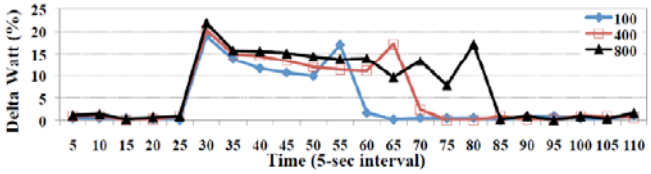


- **Cross-tier dependencies:** inter-tier interaction is generally synchronous. The service rate of upstream tiers is tied to the performance of the downstream tiers.
- **Concurrency limit per tier:** Administrators typically limit the number of threads or processes created in a tier.
- **Bottleneck shifting:**

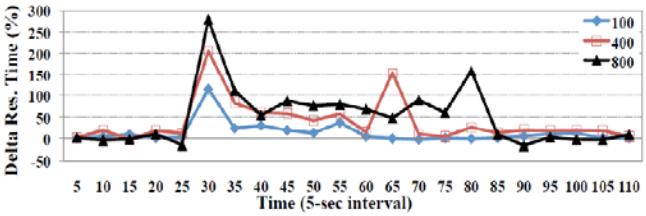
Performance is the result of a complex interaction of workloads in a very complex underlying computer system.

Challenges: Reconfiguration Cost

- Cost of resource reconfiguration



Costs of a single VM migration on power consumption (Jung et al, ICDCS'10)



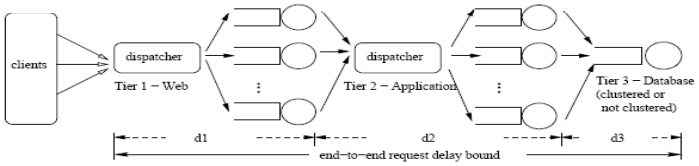
Costs of a single VM migration on response time (Jung et al, ICDCS'10)

RELATED WORK

- D. A. Menasc´e and M. N. Bennani. **Autonomic virtualized environments**. In *Proc. IEEE Int'l Conf. on Autonomic Computing (ICAC), 2006.*
- P. Padala, K.-Y. Hou, K. G. Shin, X. Zhu, M. Uysal, Z. Wang, S. Singhal, and A. Merchant. **Automated control of multiple virtualized resources**. In *Proc. of the EuroSys Conference (EuroSys), pages 13–26, 2009.*
- P. Costa, J. Napper, G. Pierre, and M. Steen. **Autonomous resource selection for decentralized utility computing**. In *Proc. IEEE Int'l Conf. on Distributed Computing Systems (ICDCS), 2009.*
- G. Jung, K. R. Joshi, M. A. Hiltunen, R. D. Schlichting, and C. Pu. **A cost-sensitive adaptation engine for server consolidation of multitier applications**. In *Proc. Int'l Middleware Conference (Middleware), 2009.*
- X. Meng, C. Isci, J. Kephart, L. Zhang, E. Bouillet, and D. Pendarakis. **Efficient resource provisioning in compute clouds via vm multiplexing**. In *Proc. IEEE Int'l Conf. on Autonomic Computing (ICAC), 2010.*

RELATED WORK

- D. Villela, P. Pradhan, and D. Rubenstein. **Provisioning servers in the application tier for e-commerce systems**. *ACM Trans. on Internet Technology, 7(1):1–23, 2007.*
  - *set of application servers modeled as M/G/1 processor sharing queueing systems, optimal server allocation that increase a server provider's profit*
  - *assumes bottleneck at single tier*
- B. Urgaonkar, P. Shenoy, A. Chandra, P. Goyal, and T. Wood. **Agile dynamic provisioning of multi-tier Internet applications**. *ACM Trans. on Autonomous and Adaptive Systems, 3(1):1–39, 2008.*
  - *It decomposes end-to-end delay guarantee into per-tier targets, then per-tier provisioning conducted based on queuing model to meet per-tier delay target.*



  - *no guideline on decomposing delay targets..*

Contributions

- Formulation of a multi-objective optimization problem : to minimize the number of physical machines used, the average response time and the total number of virtual servers allocated for multi-tier applications.
- Service differentiation among competing applications, in case of resource saturation.
- A novel stress-strain curving method to automatically select the most efficient solution from a Pareto-optimal set that is obtained as the result of a non-dominated sorting based optimization technique.
- Reducing server switching cost and improve the utilization of physical machines.
- Implementation of aMOSS on a testbed of virtualized servers.

aMOSS Approach

- Treat each objective as a separate entity and obtain a pareto-optimal set of solutions using any multi-objective optimization technique.
- Explore the tradeoff between performance and resource allocation (Little sacrifice in one objective may provide huge gain in another objective).
- Apply stress-strain curving method on pareto-optimal set to automatically select the most efficient tradeoff between performance and resource allocation

Multi-objective optimization of virtualized data centers

- Objectives:
  - Minimize the total number of physical machines used for all applications.
  - Minimize the average system end-to-end response time, a key QoS metric, for all applications.
  - Minimize the total number of virtual servers allocated to all applications to free up more physical machines and also to reduce virtualization overhead.
- Constraints
  - The average end-to-end response time of each application must be below a given bound according to the service level agreement.
  - The total number of virtual servers running for all applications on one physical machine must not exceed a specified limit due to the concurrency limit.
  - The utilization of a virtual server cannot exceed its resource capacity limit.
  - The number of physical machines available is limited.

$$\text{Minimize } m$$

$$\text{Minimize} \sum_{j=1}^{M}\sum_{k=1}^{K} d_{jk}$$

$$\text{Minimize} \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{K} a_{ijk}$$

Subject to Constraints:

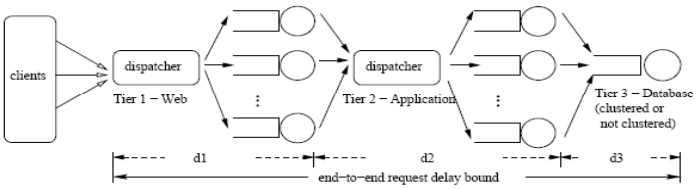
$$\forall j \in [1, M],\ \sum_{k=1}^{K} d_{jk} \le U_j$$

$$\forall i \in [1, N],\ 0 \le \sum_{j=1}^{M}\sum_{k=1}^{K} a_{ijk} \le W$$

$$\forall j \in [1, M], \forall k \in [1, K],\ 0 \le \rho_{jk} < 1$$

$$m \le N.$$

An Analytic Model



Queuing theoretical model for performance analysis of multi-tier applications

$$d_{jk} = E[W_{jk}] + E[X_{jk}] = \frac{\lambda_j (E[T_{jk}^2] + E[X_{jk}^2])}{2\sum_{i=1}^{N} a_{ijk}(1-\rho_{jk})} + E[X_{jk}]$$

Service Differentiation

- Minimize the difference between the normalized end-to-end response time ratio and the desired QoS ratio for each application *j*.

$$\text{Minimize} \quad \sum_{j=1}^{M-1} \left| \frac{\sum_{k=1}^{K} d_{jk}}{\sum_{j=1}^{M}\sum_{k=1}^{K} d_{jk}} - QoS_{ref,j} \right|.$$

Reducing server switching cost

- Server switching by addition and removal of a virtual server at a tier introduces non-negligible latency to a multitier service.
- We incorporate the time required for the reconfiguration of server allocation scheme into calculation of the average response time, while evaluating a group of candidate solutions in our optimization algorithm.

$$T_c = \max_{i\epsilon[1,N]} (\sum_{j=1}^{M}\sum_{k=1}^{K}((b_{ijk}-a_{ijk})^* \cdot T_k^a + (a_{ijk}-b_{ijk})^* \cdot T_k^r)),$$

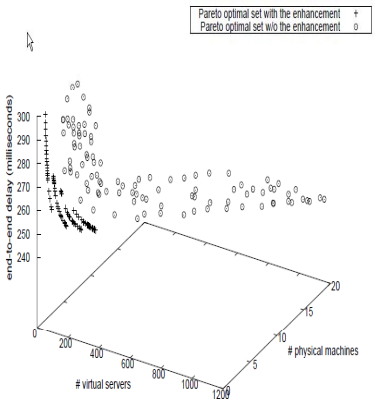
candidate solution

current virtual server configuration

$$D_j = \frac{T_c \cdot D_j^a + (T_s - T_c) \cdot D_j^b}{T_s}, \forall j \in [1, M].$$

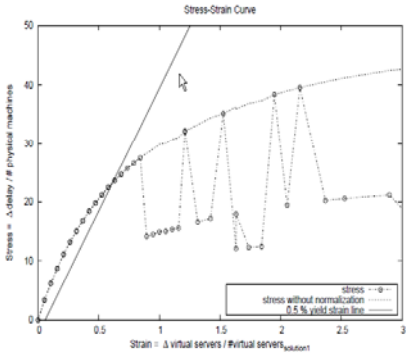
Pareto-Optimal Set

- We apply a computationally fast multi-objective genetic algorithm to obtain multiple Pareto-optimal solutions in one single run.
- We represent a solution to the optimization problem by a "chromosome". It is a string of numbers, coding information about the decision variables.
- Threshold-based enhancement to improve usage of physical machines
  - Initiate algorithm with a small fraction of available physical machines.
  - Increase the number of physical machines in the search space by one if the % evaluations that violated resource constraints exceeds a threshold.



Stress-Strain Curve

- Automatically selecting the solution from a broad range of the Pareto-optimal set
- First, pareto-optimal solutions are sorted in increasing order of #VS
- Calculate stress and strain for each solution in the set.
- Apply yield strain line to find the yield point solution (most efficient tradeoff)
- Any additional allocation at that point will have negligible impact on performance.



$$stress_x = (D_1 - D_x)/PS_x, \forall x \in [1, P].$$

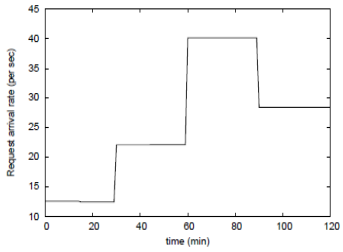
$$strain_x = (VS_x - VS_1)/VS_1, \forall x \in [1, P].$$

Performance Evaluation

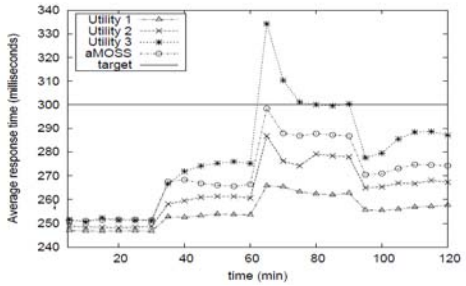
1) Simulation based study using a synthetic workload with Bounded-Pareto distribution of request inter-arrival time and service time.
2) Testbed implementation on a testbed of HP Proliant BL460C G6 blade server modules with Vmware virtual machines.

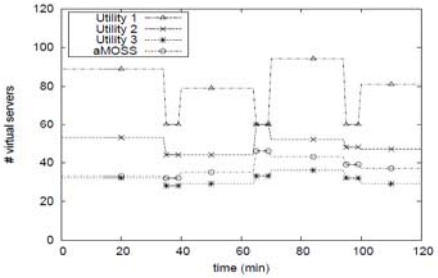
TABLE I
WORKLOAD CHARACTERISTICS.

	WebTier	AppTier	DBTier
$E[X_{jk}]$	64.679 ms	94.78 ms	84.75 ms
$E[X_{jk}^2]$	4191.695	8991.681	7191.683



Comparison with Utility Based Optimization (single application)



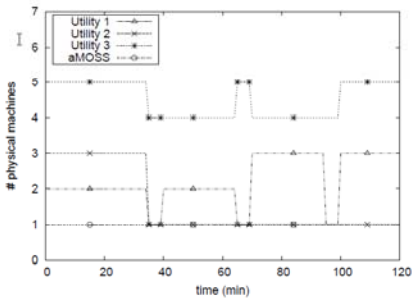


$$U_1 = 0.2 \cdot U_{VS} + 0.2 \cdot U_{PS} + 0.6 \cdot U_D.$$

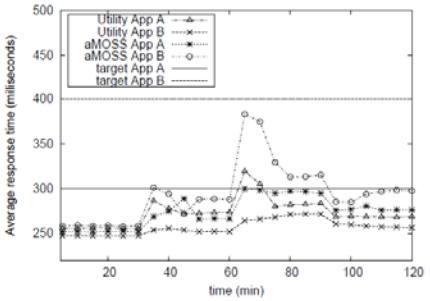
$$U_2 = 0.34 \cdot U_{VS} + 0.33 \cdot U_{PS} + 0.33 \cdot U_D.$$

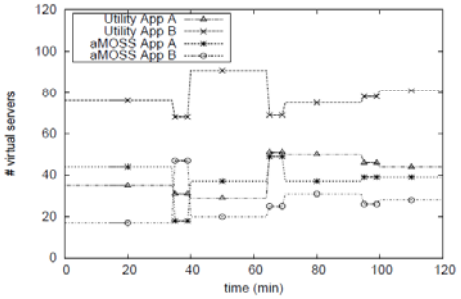
$$U_3 = 0.8 \cdot U_{VS} + 0.1 \cdot U_{PS} + 0.1 \cdot U_D.$$

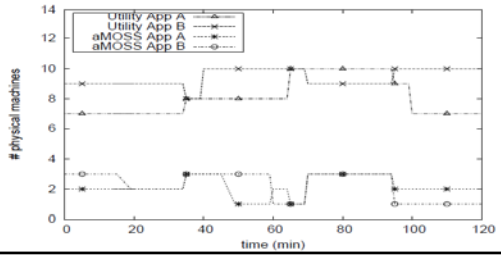
aMOSS Improv.	Response time	VS Alloc.	PS Alloc.
U1	-5.4 %	+54%	+50%
U2	-2.3%	+15%	+29%
U3	+4%	-6.8%	+173%



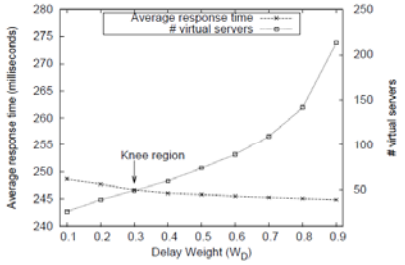
Comparison with Utility Based Optimization (multiple applications)



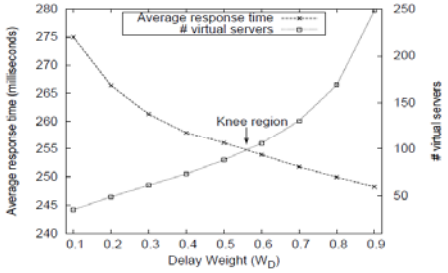




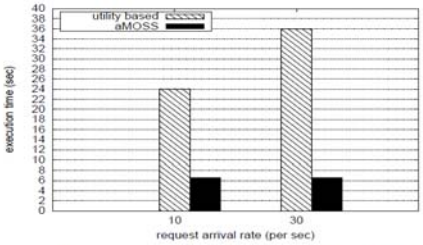
Automation Performance and Overhead



(a) at 10 requests/sec

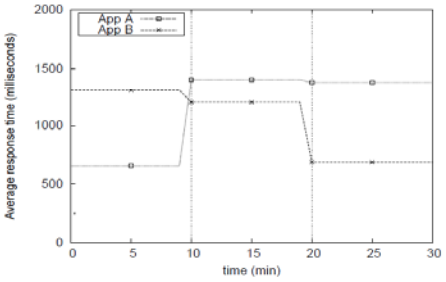


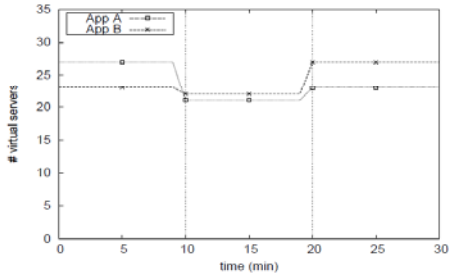
(b) at 30 requests/sec).

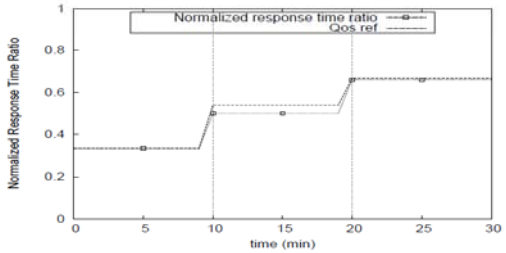


(c) Overhead comparison with aMOSS.

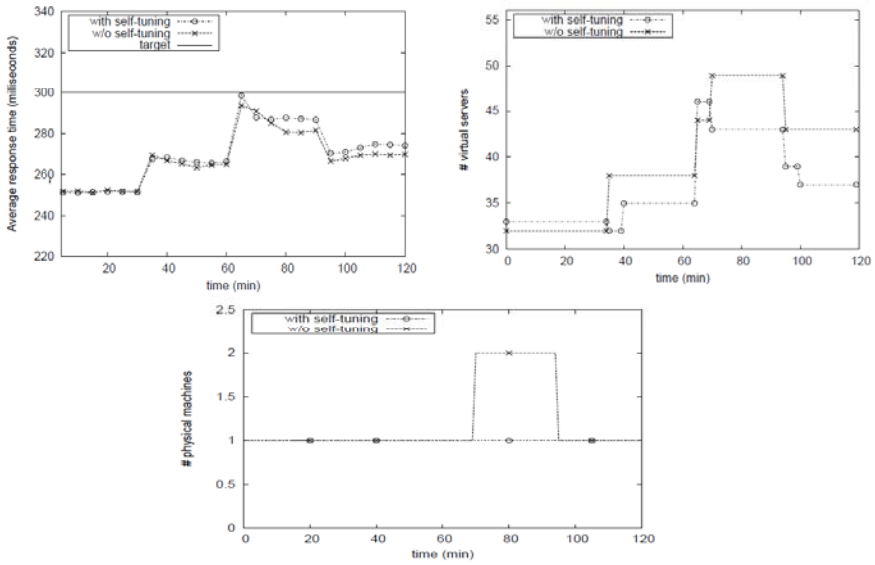
Service Differentiation among Applications



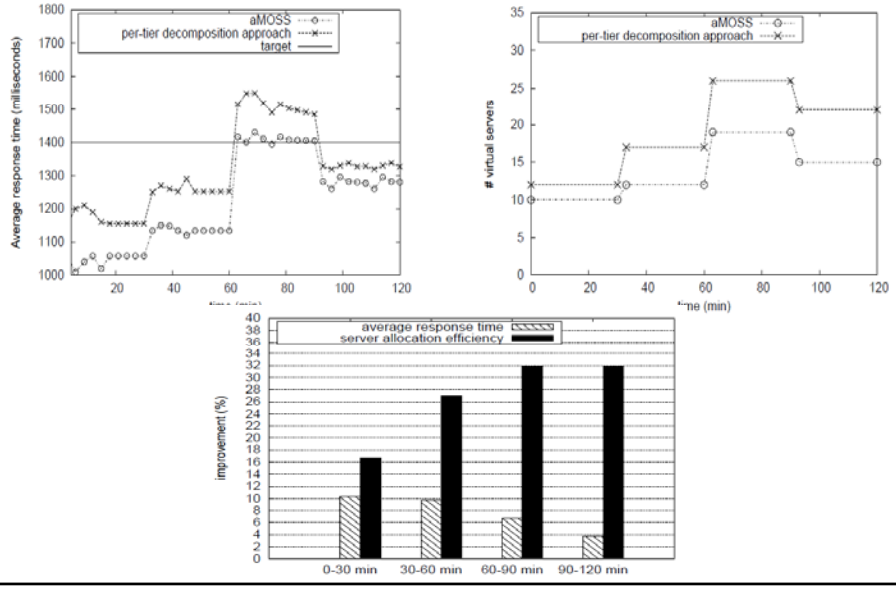




Server switching cost reduction



aMOSS vs. per-tier decomposition approach on a testbed implementation



Summary

- We explored the promise of automated multi-objective optimization for autonomic server provisioning using a novel stress-strain curving method (aMOSS).
- aMOSS automatically chooses the yield-point solution of the Pareto-optimal set, which essentially guarantees the most efficient tradeoff among multiple conflicting objectives.
- The aMOSS approach is generalizable to any objective in a datacenter that can be modeled with sufficient accuracy.

ANY QUESTIONS?