On the Use of Performance Models in Autonomic Computing

Daniel A. Menascé
Department of Computer Science
George Mason University

Motivation for AC

- "…main obstacle to further progress in IT is a looming software complexity crisis." (from an IBM manifesto, Oct. 2001).
  - Tens of millions of lines of code
  - Skilled IT professionals required to install, configure, tune, and maintain.
  - Need to integrate many heterogeneous systems
  - Limit of human capacity being achieved

Motivation for AC (cont'd)

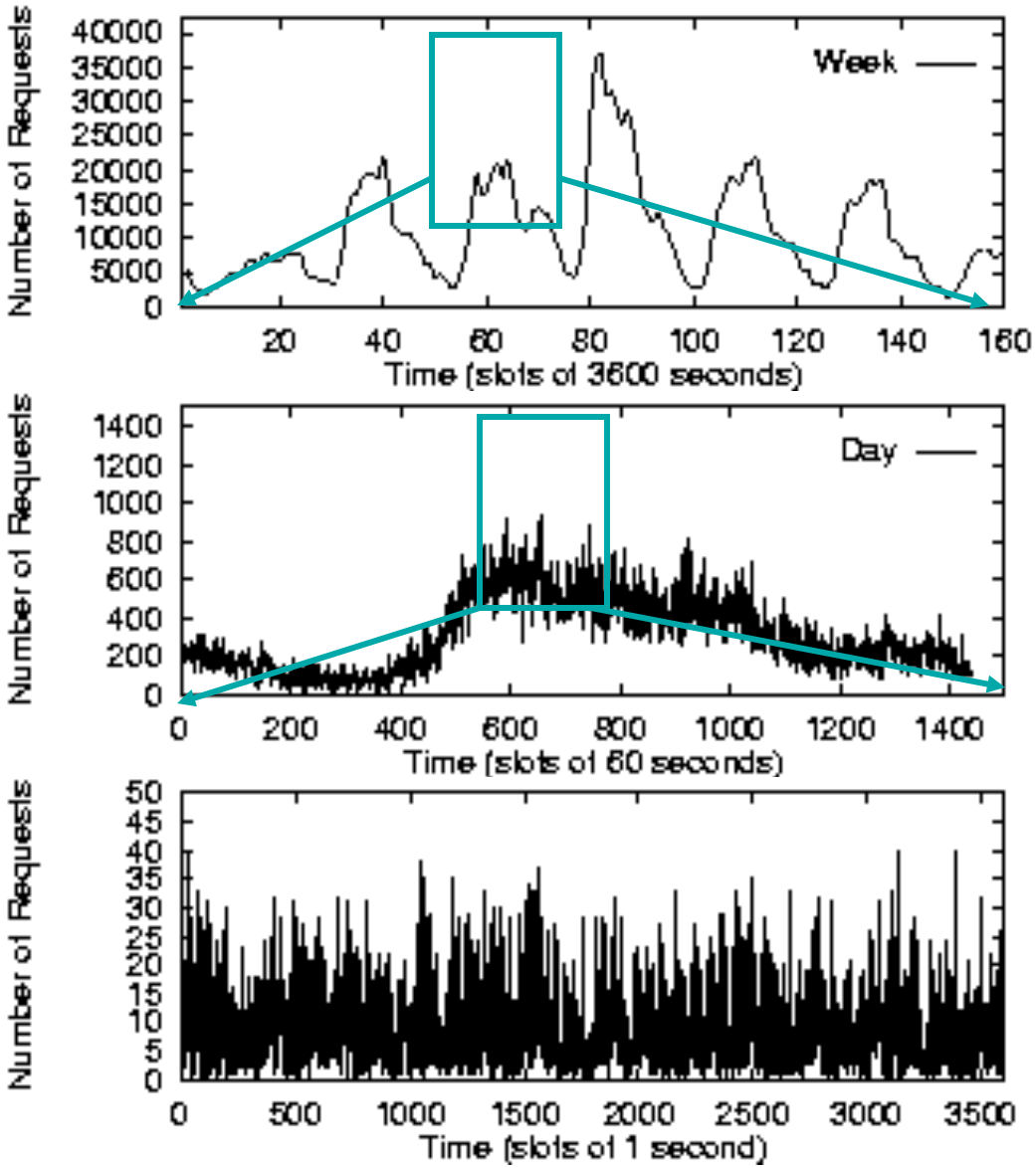
- Harder to anticipate interactions between components at design time:
  - Need to defer decisions to run time
- Computer systems are becoming too massive, complex, to be managed even by the most skilled IT professionals
- The workload and environment conditions tend to change very rapidly with time

Multi-scale time workload variation of a Web Server

3600 sec

60 sec

1 sec



Self-Optimizing Systems

- Complex middleware and database systems have a very large number of configurable parameters.

Web Server (IIS 5.0)	**Application Server (Tomcat 4.1)**	**Database Server (SQL Server 7.0)**
HTTP KeepAlive	acceptCount	Cursor Threshold
Application Protection Level	minProcessors	Fill Factor
Connection Timeout	maxProcessors	Locks
Number of Connections		Max Worker Threads
Logging Location		Min Memory Per Query
Resource Indexing		Network Packet Size
Performance Tuning Level		Priority Boost
Application Optimization		Recovery Interval
MemCacheSize		Set Working Set Size
MaxCachedFileSize		Max Server Memory
ListenBacklog		Min Server Memory
MaxPoolThreads		User Connections
worker.ajp13.cachesize		

Autonomic Computing

- System that can manage themselves given high-level objectives.
  - High-level objectives can be expressed in term of service-level objectives or utility functions.
- Autonomic computing is inspired in the human autonomic nervous system:
  - "The autonomic nervous system consists of sensory neurons and motor neurons that run between the central nervous system and various internal organs such as the: heart, lungs, viscera, glands. It is responsible for monitoring conditions in the internal environment and bringing about appropriate changes in them. The contraction of both smooth muscle and cardiac muscle is controlled by motor neurons of the autonomic system." http://users.rcn.com/jkimball.ma.ultranet/BiologyPages/P/PNS.html
  - The autonomic nervous system functions in an involuntary, reflexive manner.

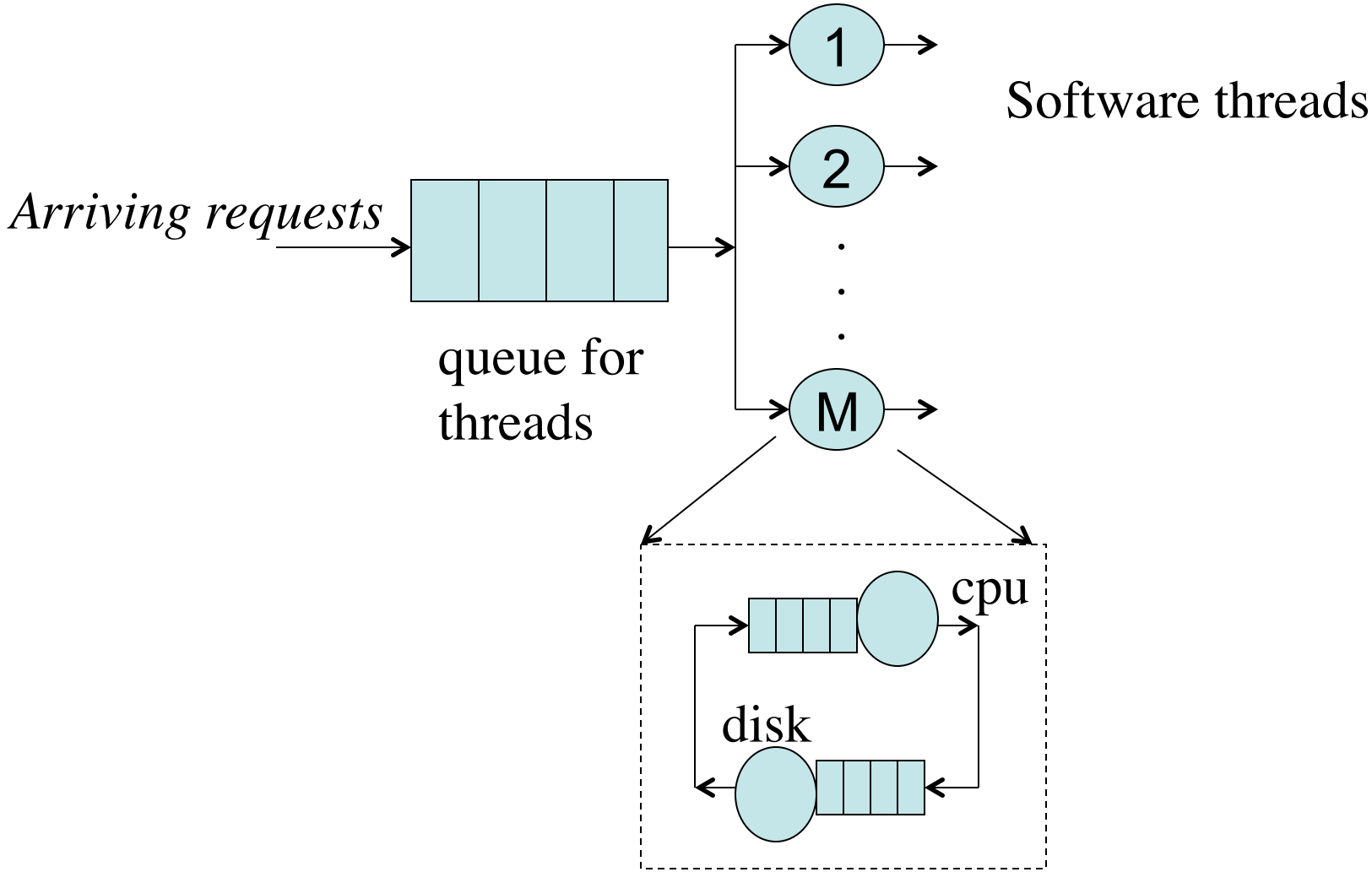
Autonomic Systems

- Self-managing
  - Self-configuring
  - Self-optimizing
  - Self-healing
  - Self-protecting
- Self-* systems

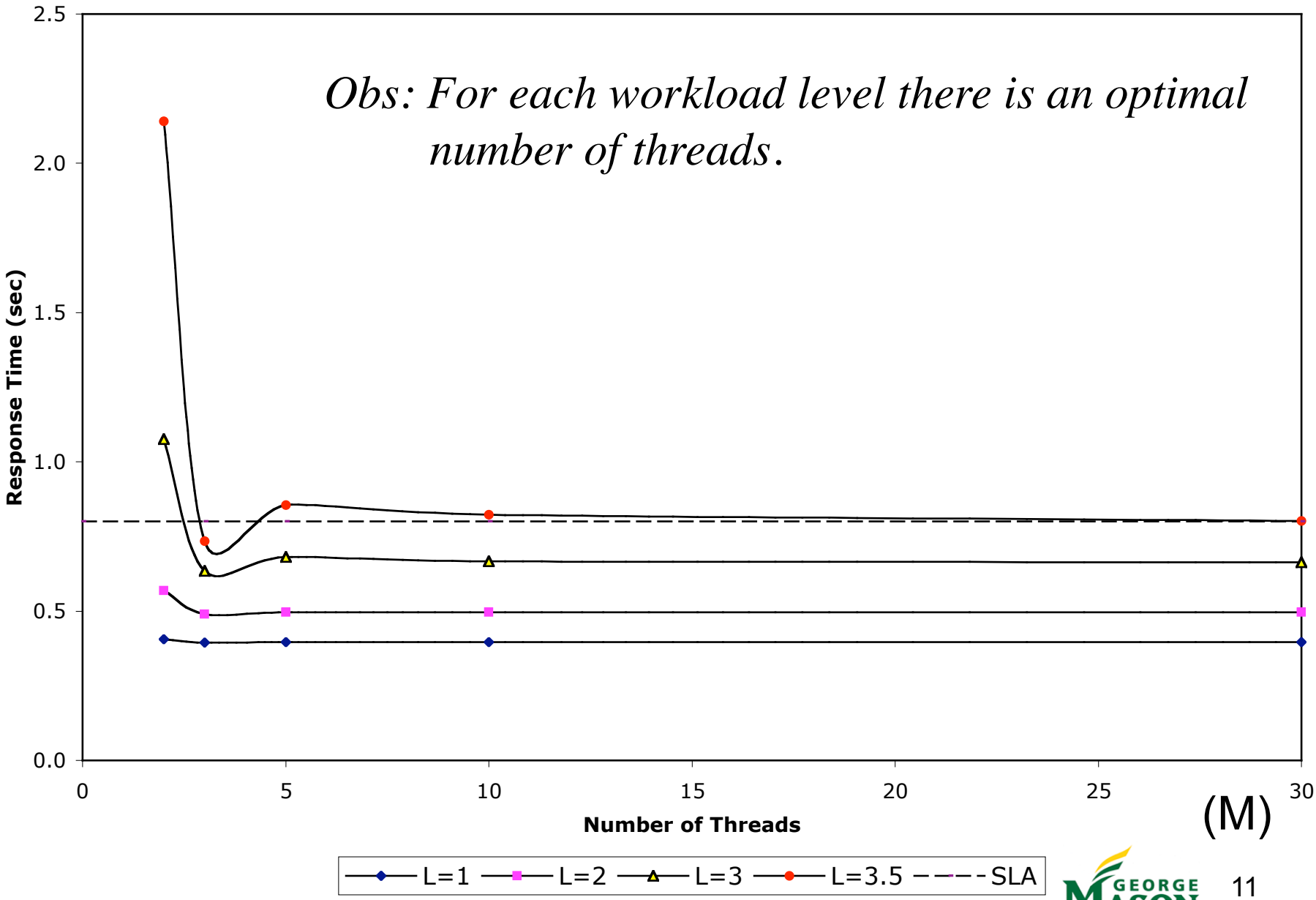
Autonomic Systems

- Self-managing
  - Self-configuring
  - Self-optimizing
  - Self-healing
  - Self-protecting
- Self-* systems

Self-optimizing systems: motivation

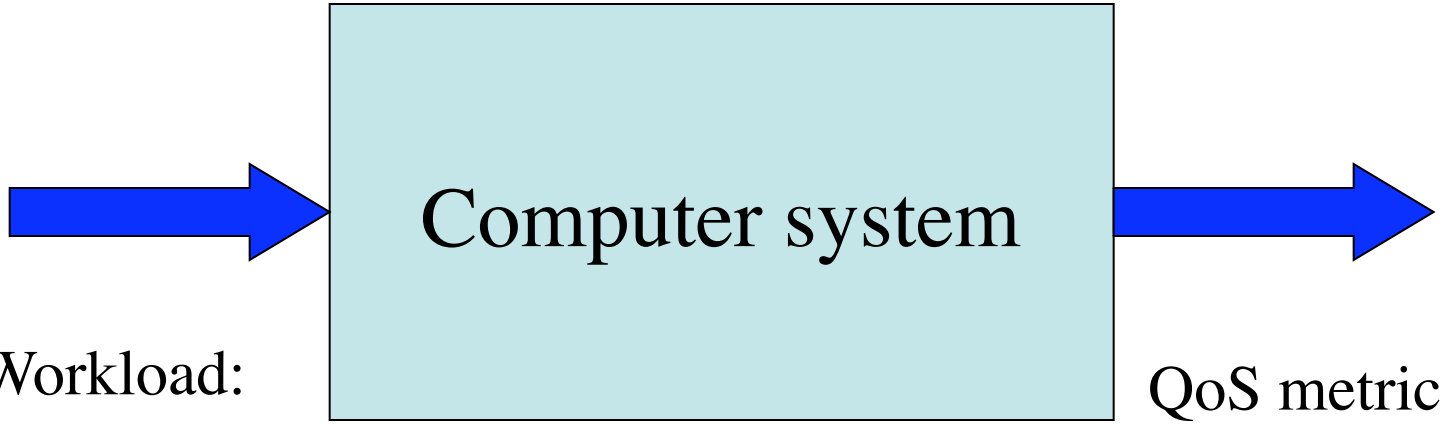


Q: How does the response time vary with the number of software threads, M?



(M)

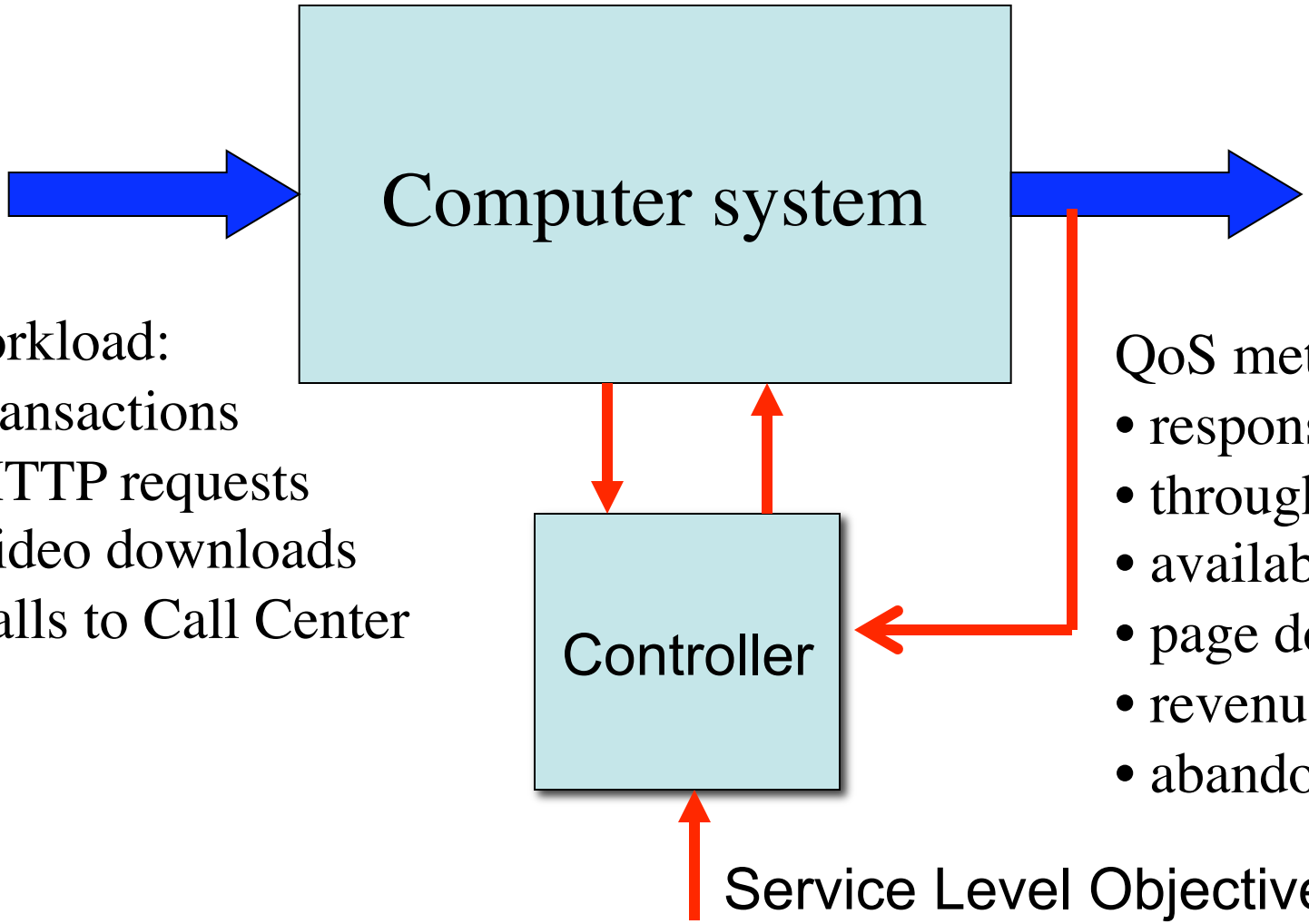
Workload and QoS metrics



Workload:
- transactions
- HTTP requests
- video downloads
- calls to Call Center

QoS metrics:
- response time
- throughput
- availability
- page download time
- revenue throughput
- abandonment rate

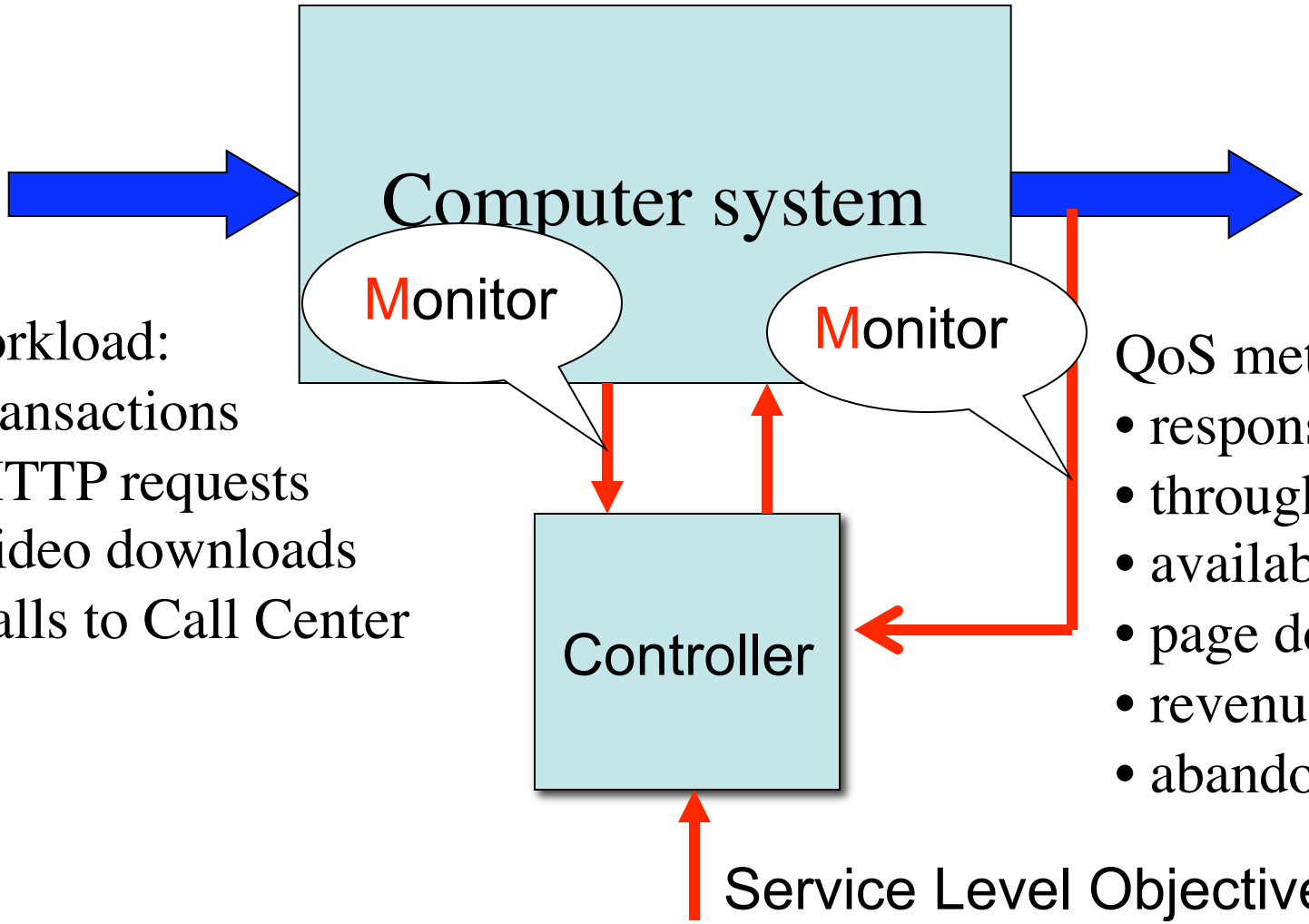
Self-optimizing System



Workload:
- transactions
- HTTP requests
- video downloads
- calls to Call Center

QoS metrics:
- response time
- throughput
- availability
- page download time
- revenue throughput
- abandonment rate

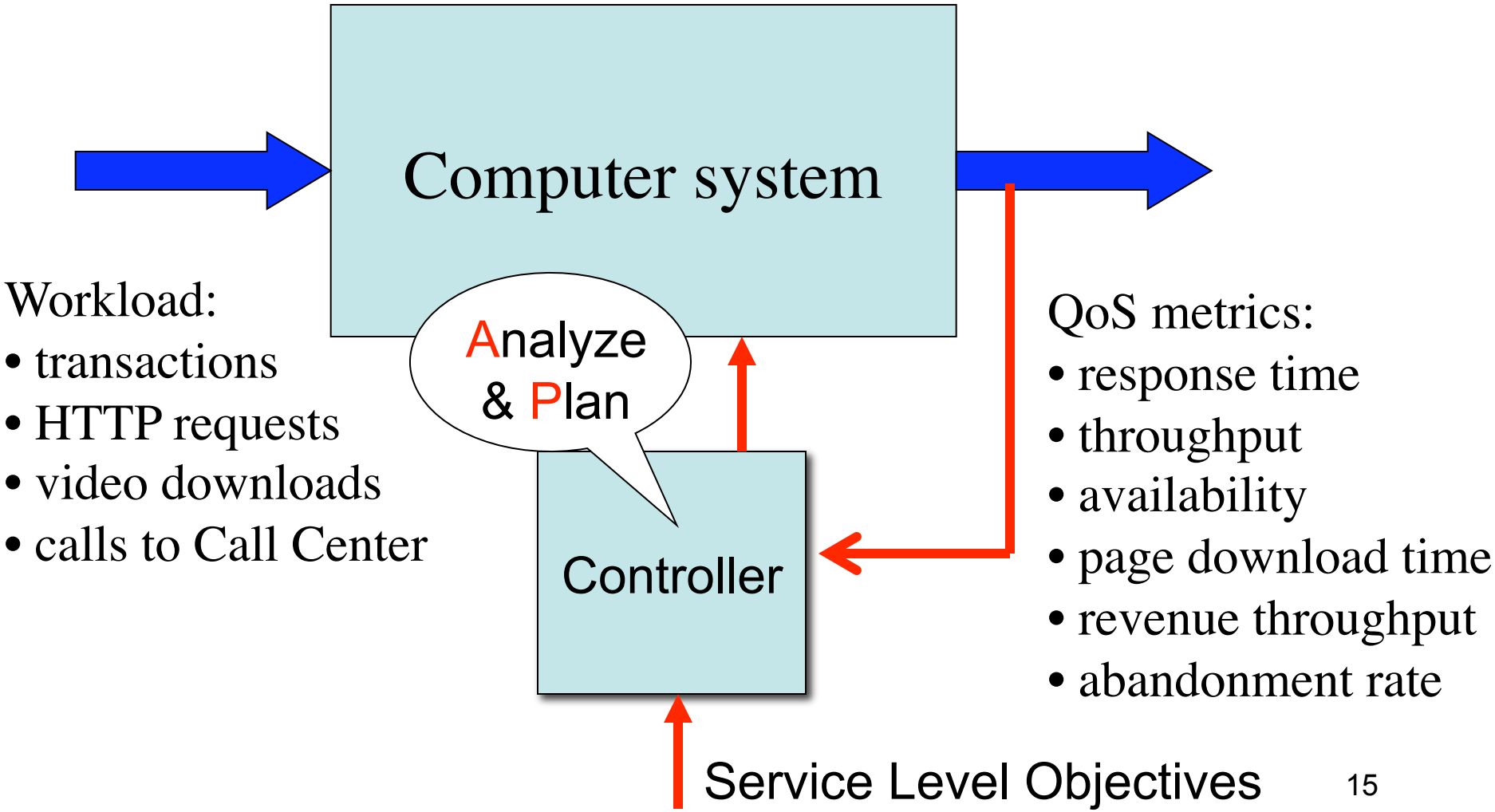
Self-optimizing System



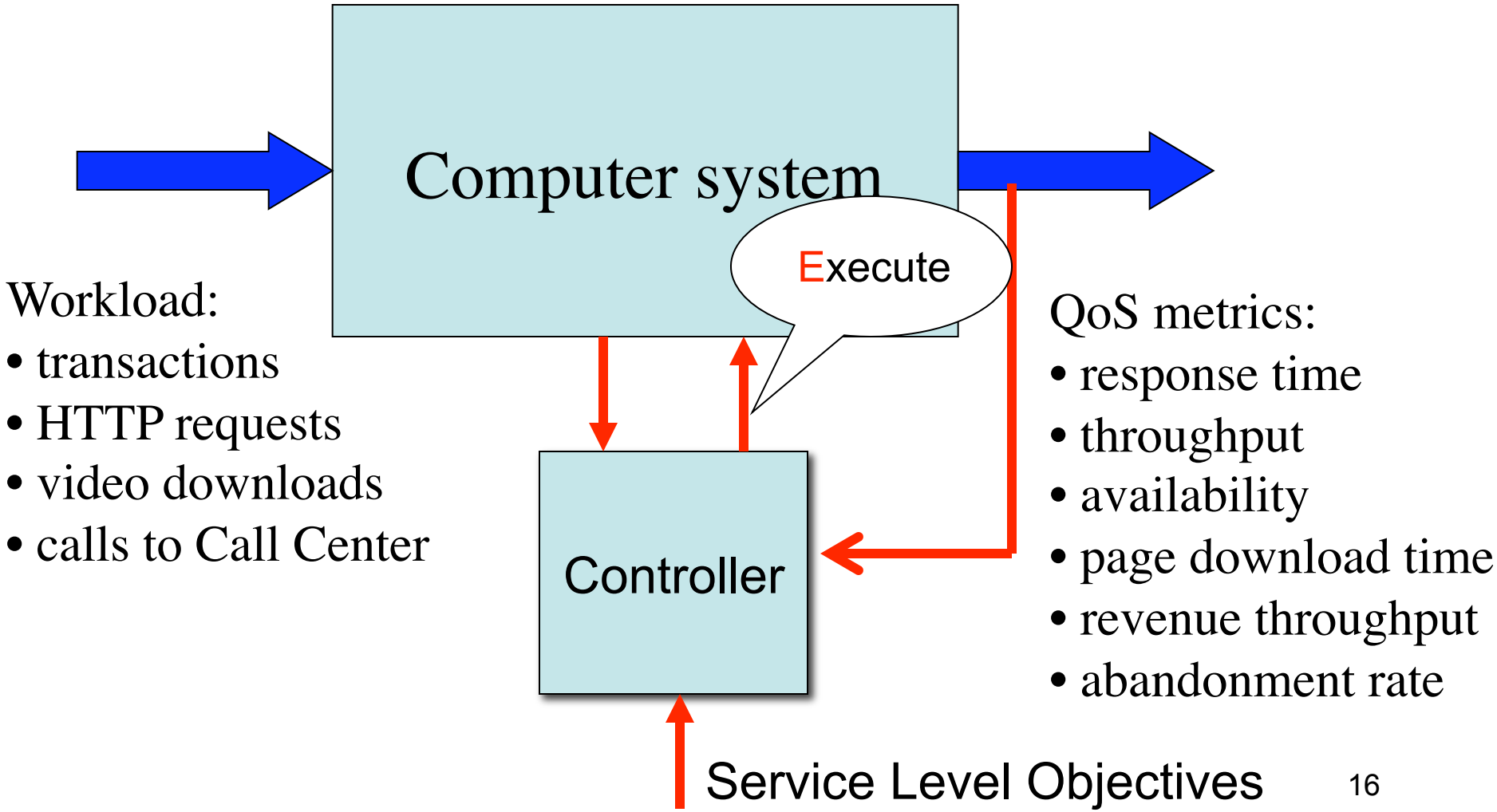
Workload:
- transactions
- HTTP requests
- video downloads
- calls to Call Center

QoS metrics:
- response time
- throughput
- availability
- page download time
- revenue throughput
- abandonment rate

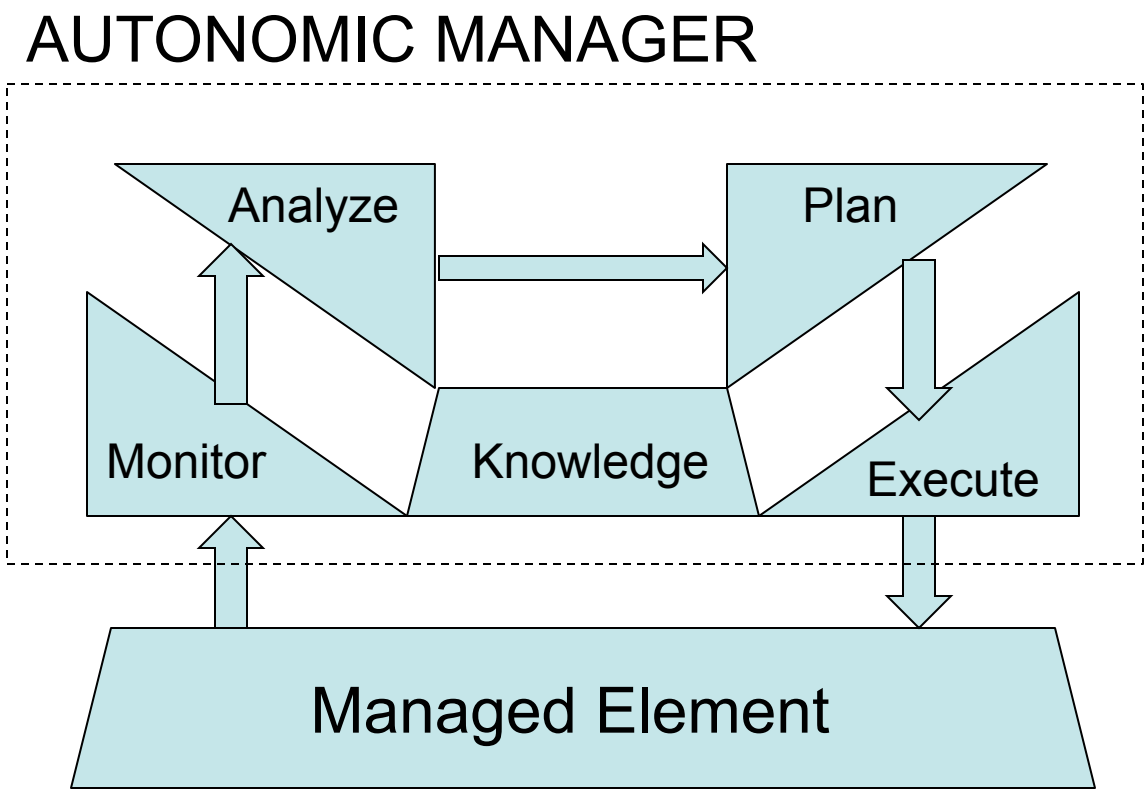
Self-optimizing System



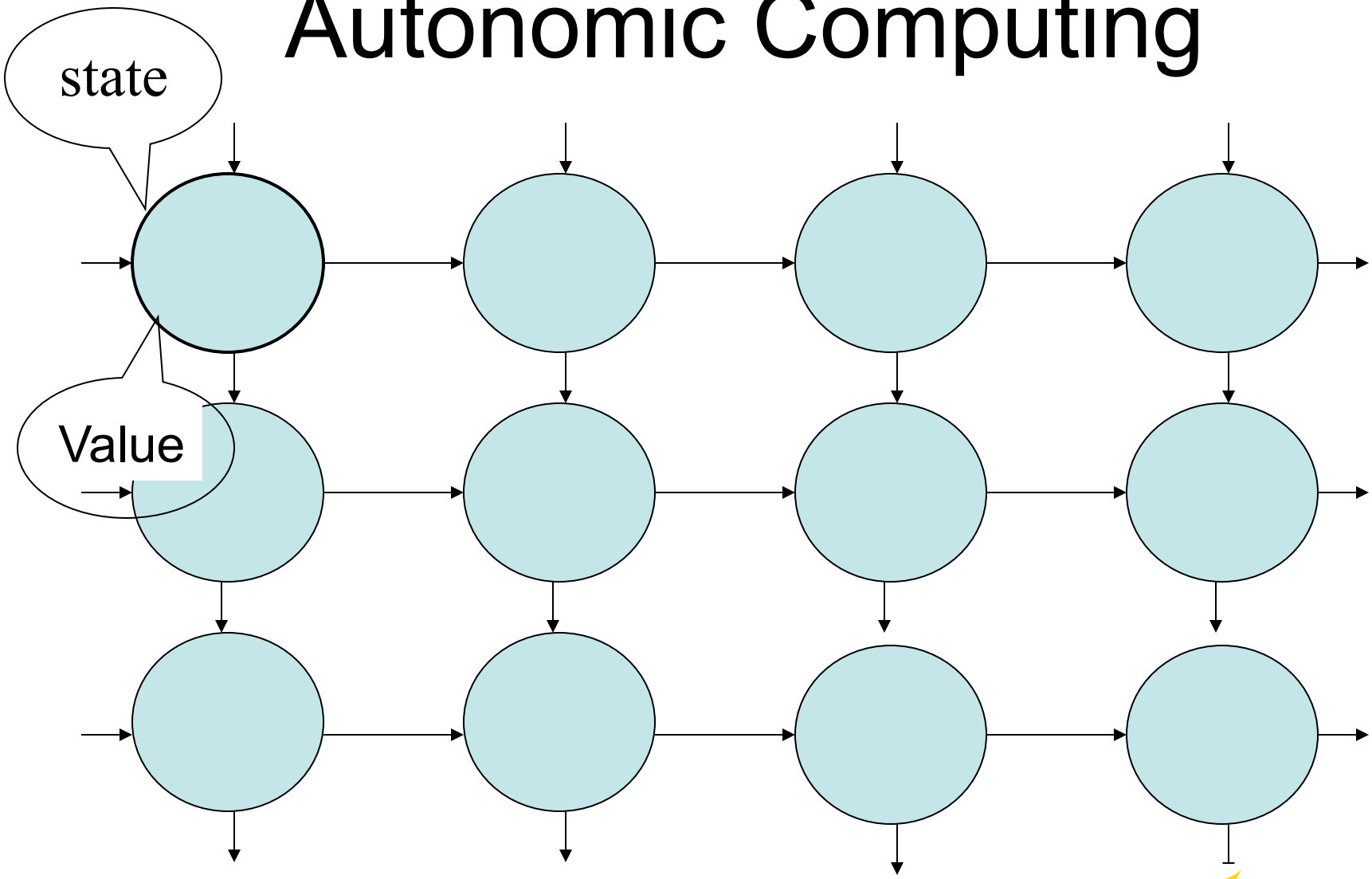
Self-optimizing System



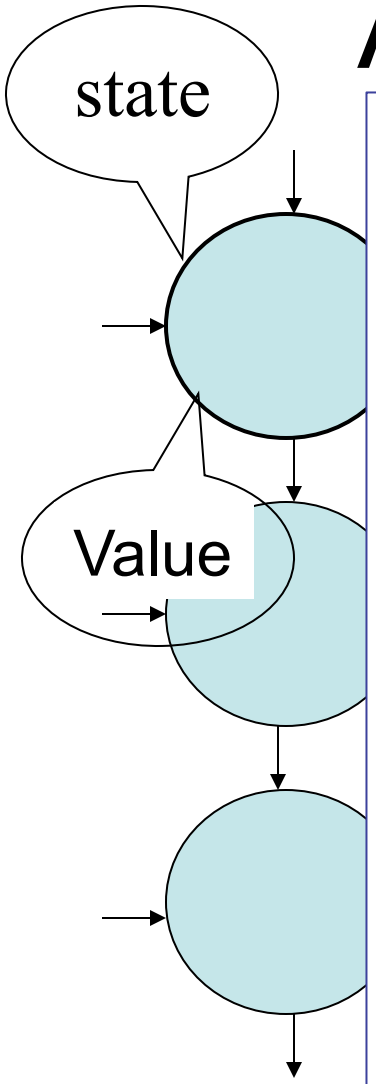
IBM's MAPE-K Model for AC



Performance Model-Based Autonomic Computing



Performance Model-Based Autonomic Computing



State: e.g., set of configuration parameters

Value: e.g., QoS metric, utility function value

Goal: find state that optimizes the value subject to cost constraints

- State space is large
- Objective function does not have a closed form

Use performance models to compute value at each state.

Use combinatorial search techniques to find near-optimal solution.

Examples of Autonomic Computing Systems

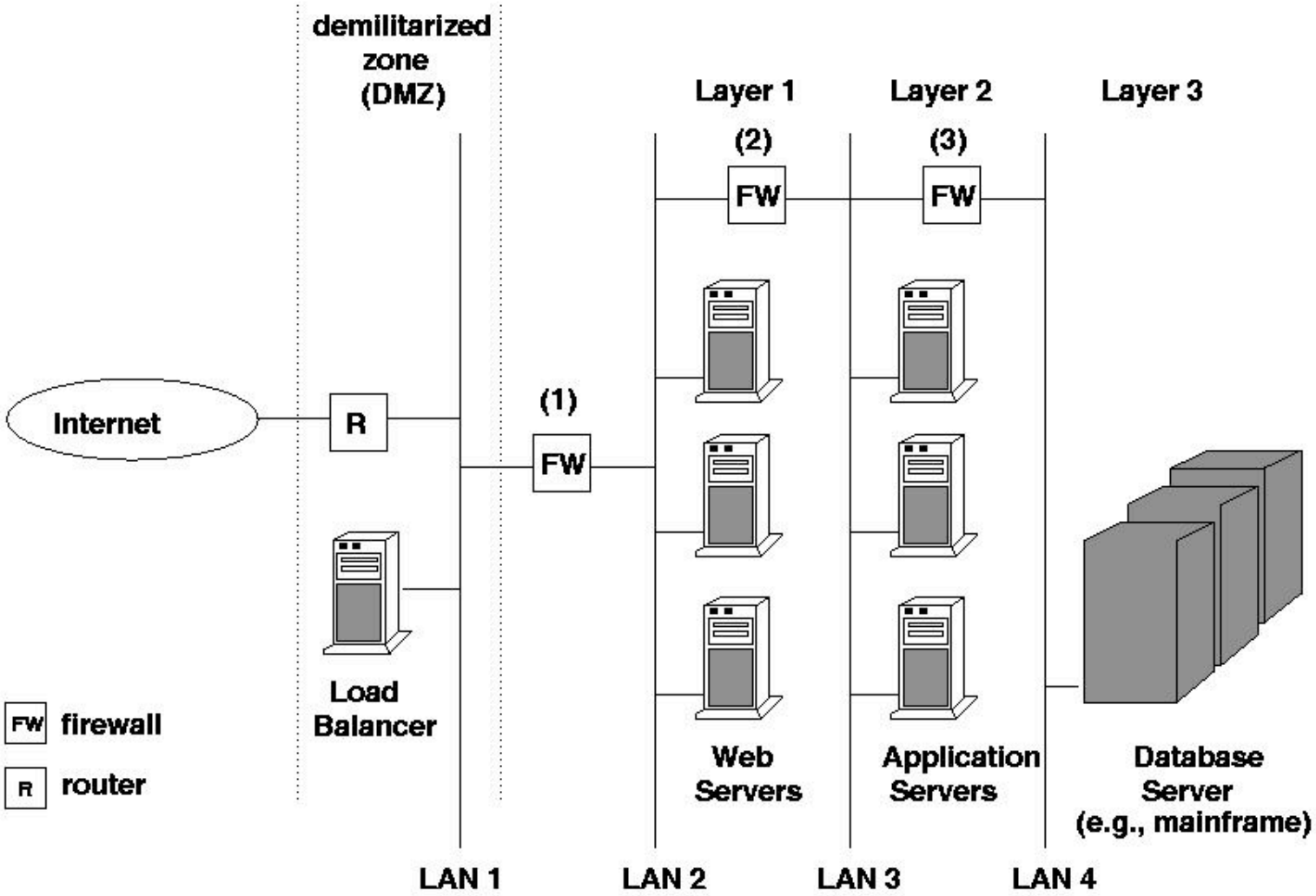
- E-commerce system
- Internet Data Center
- Virtualized Environment
- SOA Software Systems

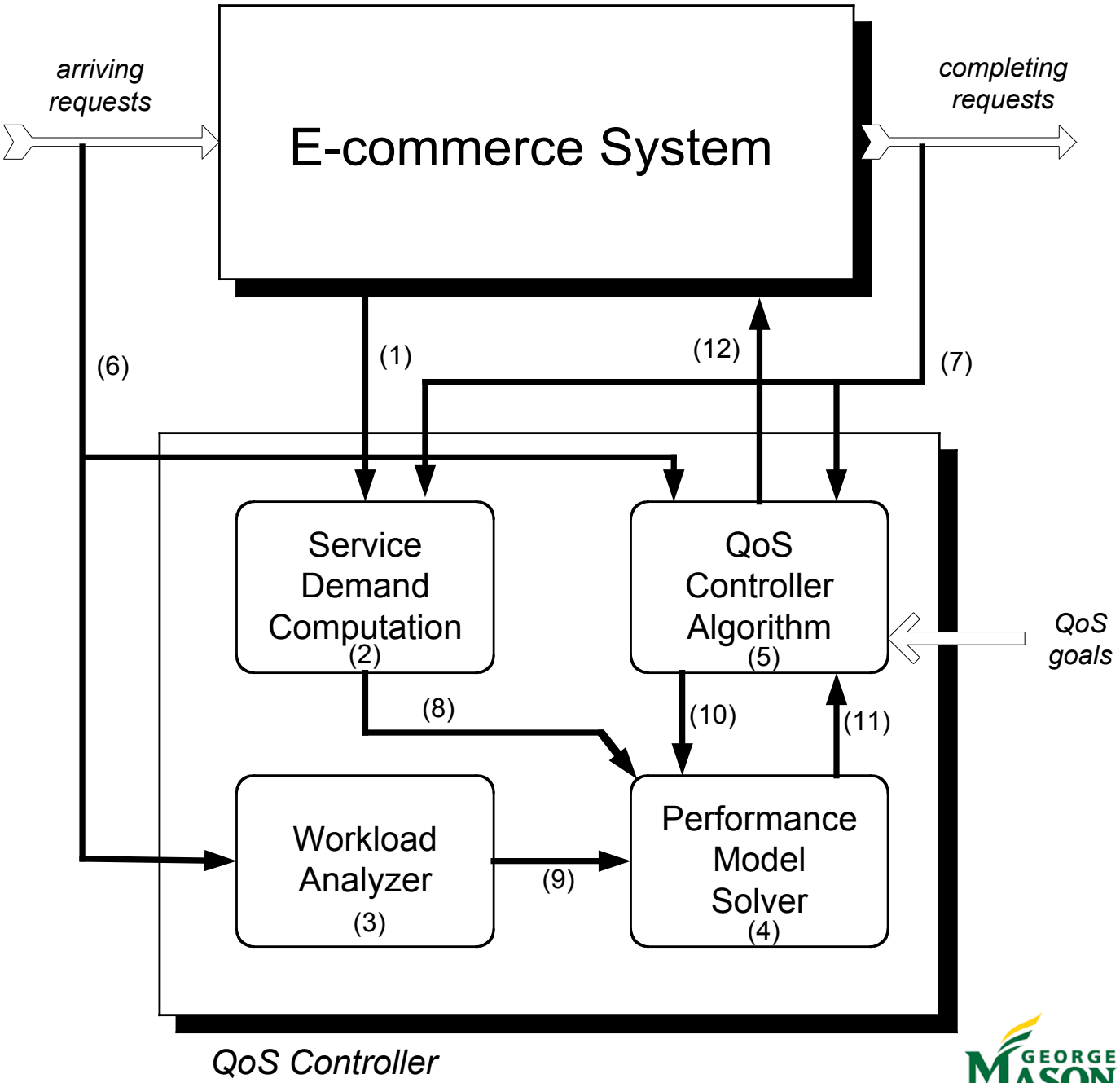
Examples of Autonomic Computing Systems

- E-commerce system
- Internet Data Center
- Virtualized Environment
- SOA Software Systems

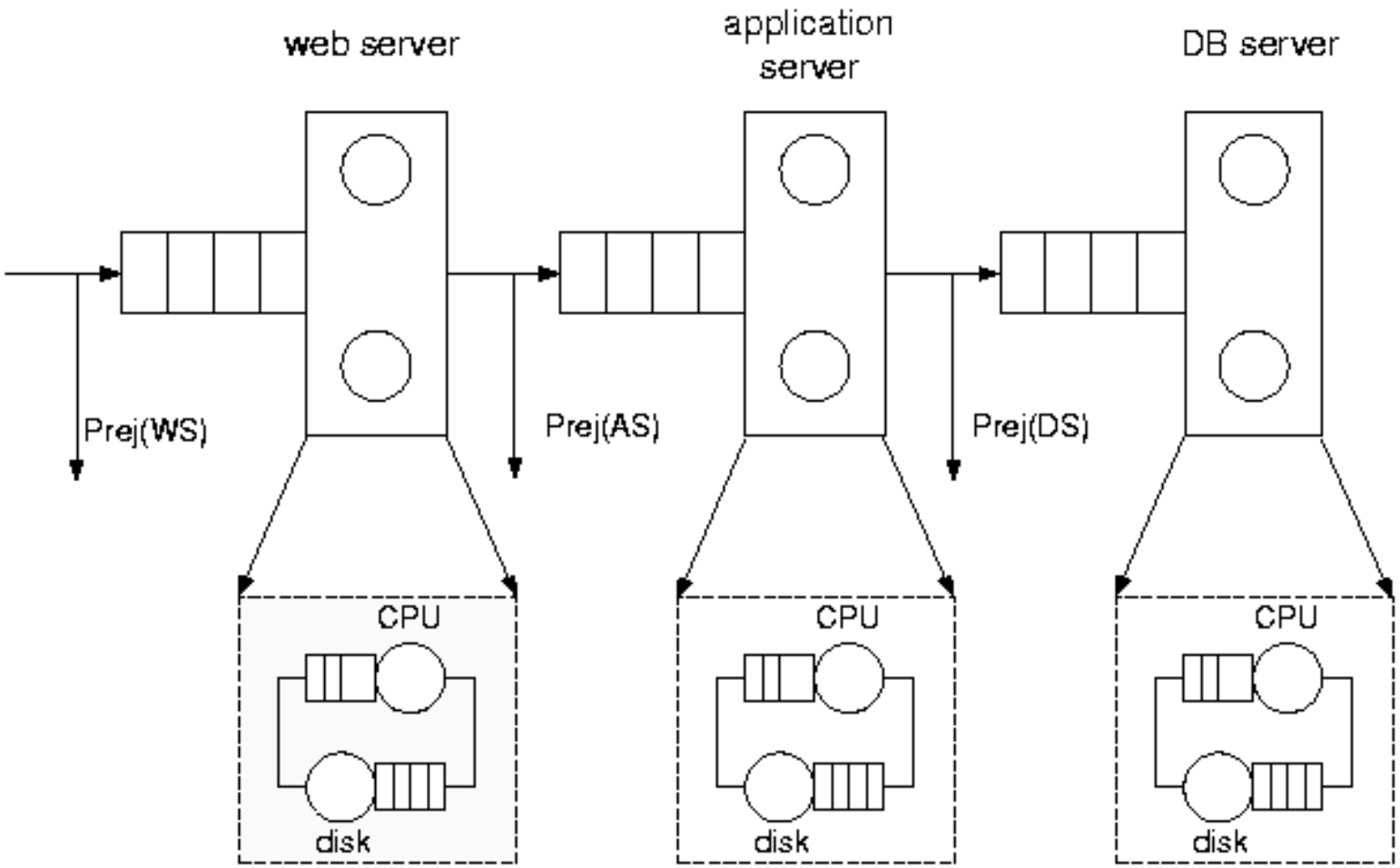
"Preserving QoS of E-commerce Sites Through Self-Tuning: A Performance Model Approach," D.A. Menasce, R. Dodge and D. Barbara, Proc. 2001 ACM Conference on E-commerce, Tampa, FL, October 14-17, 2001

Multi-tier Architecture



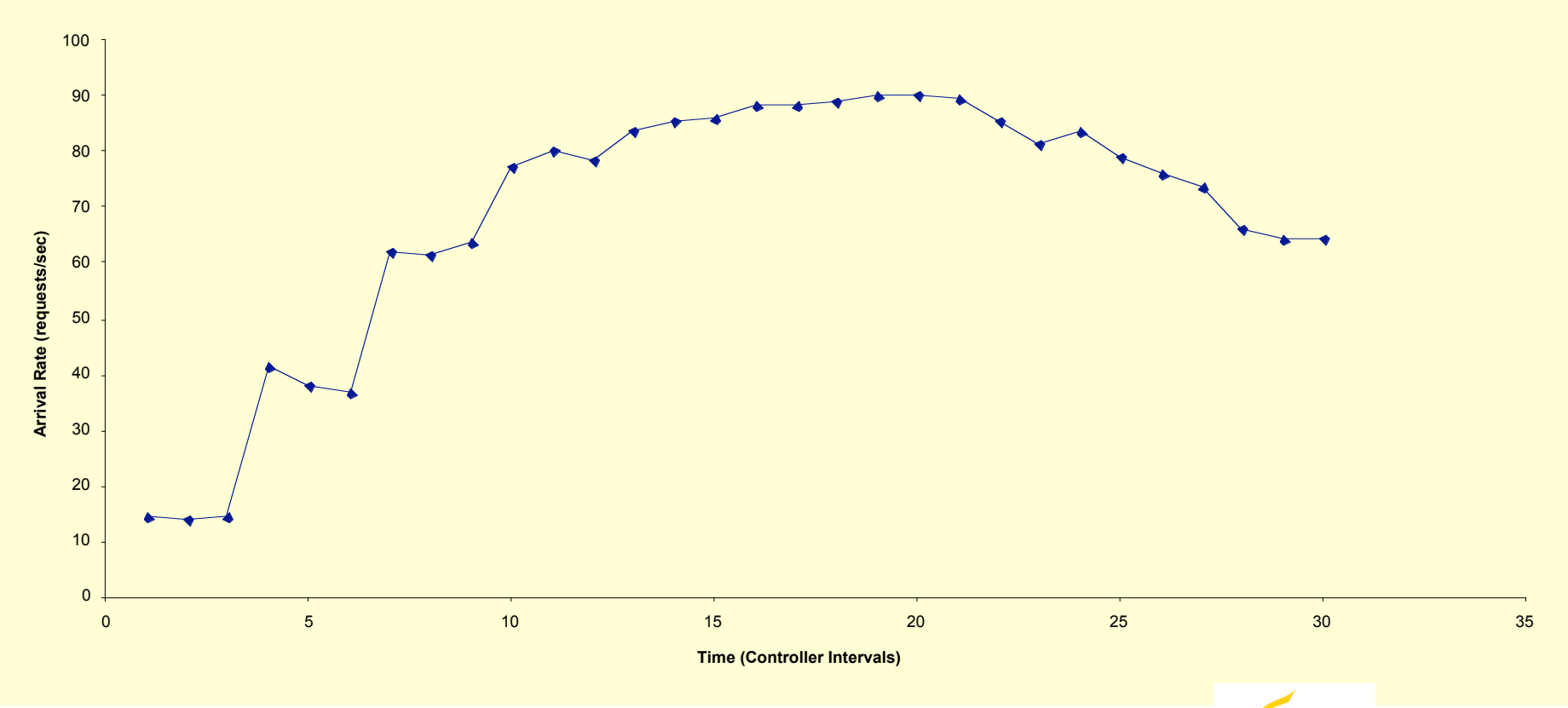


Queuing Model Used to Compute QoS Values

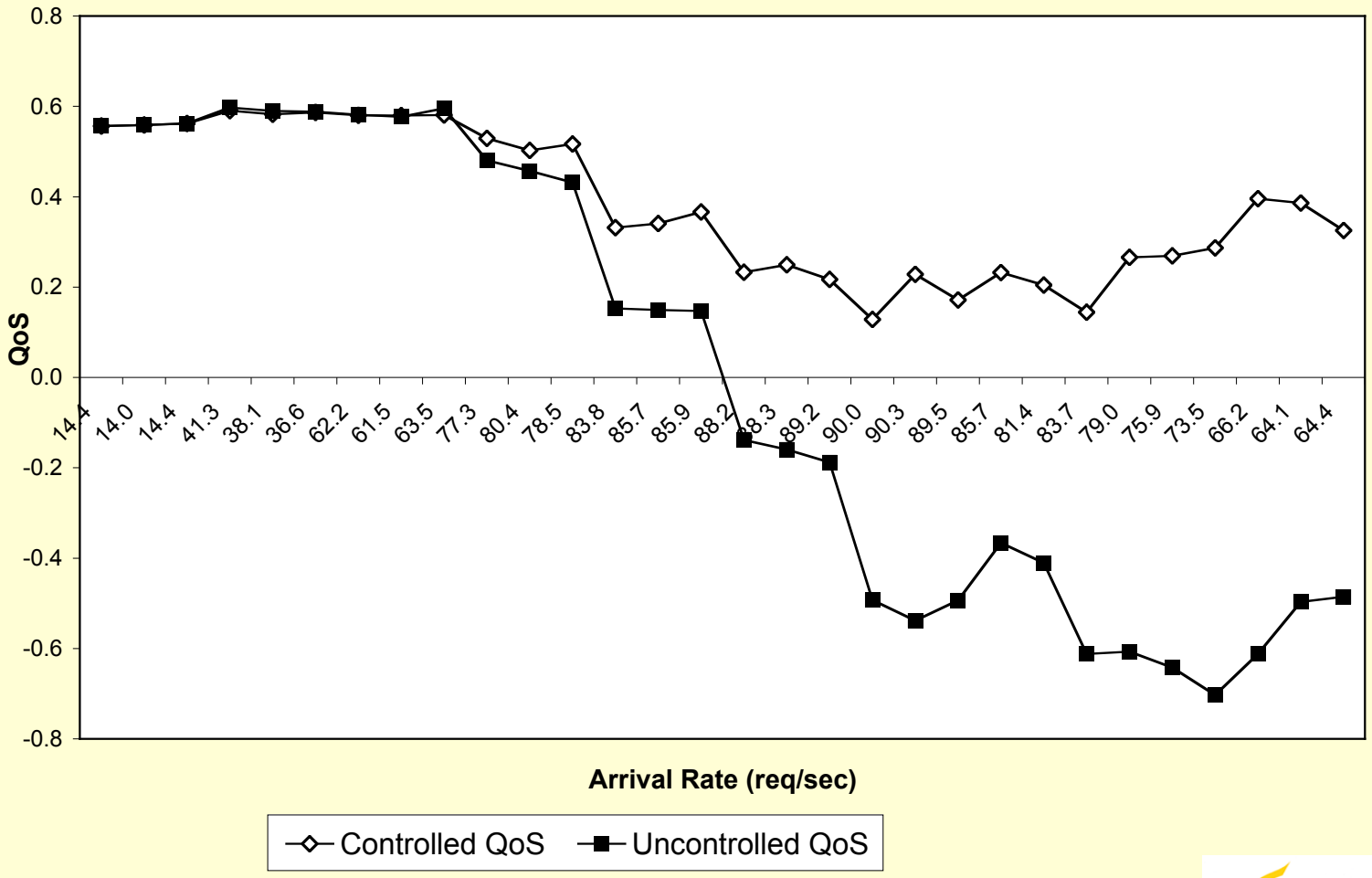


Experimental Results

Arrival rate

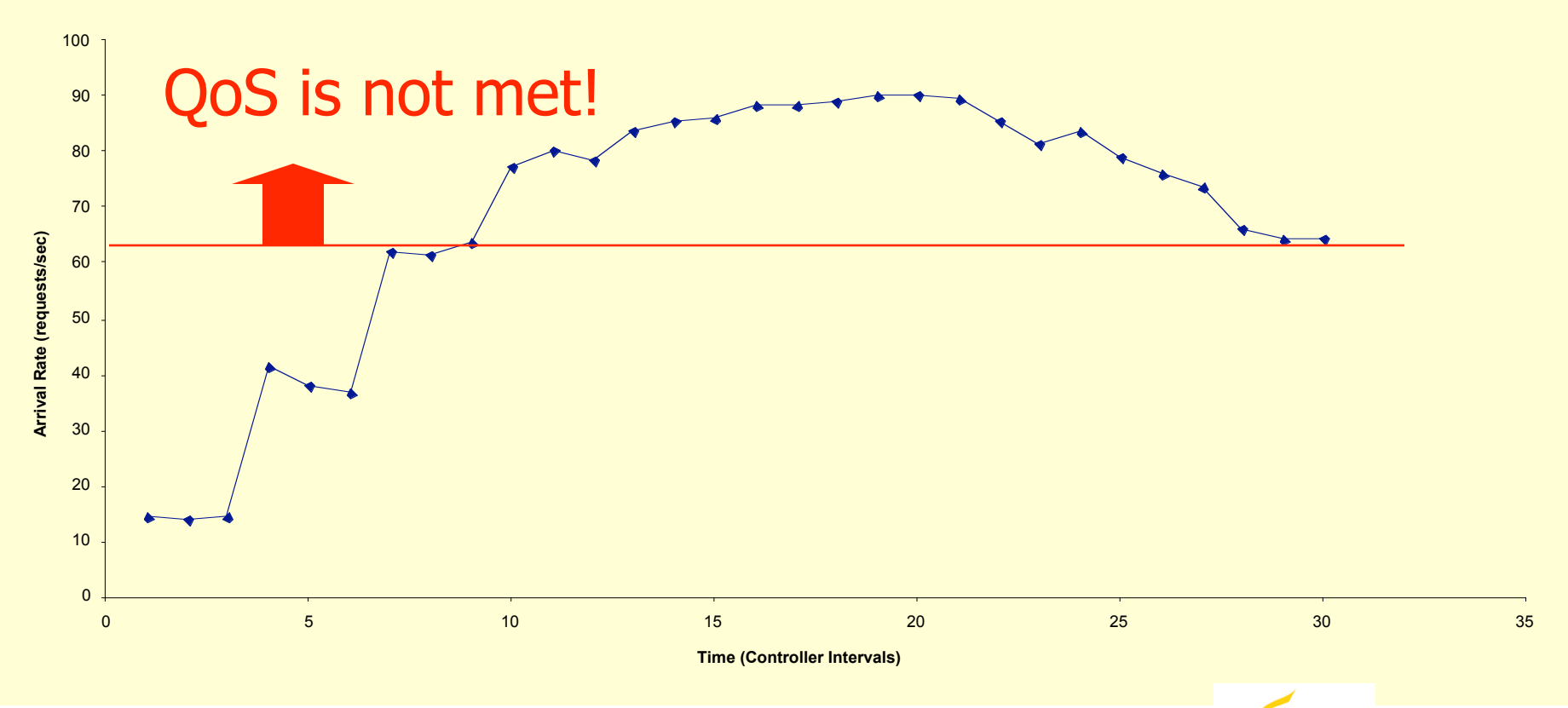


Results of QoS Controller



Experiment Results

Arrival rate



Examples of Autonomic Computing Systems

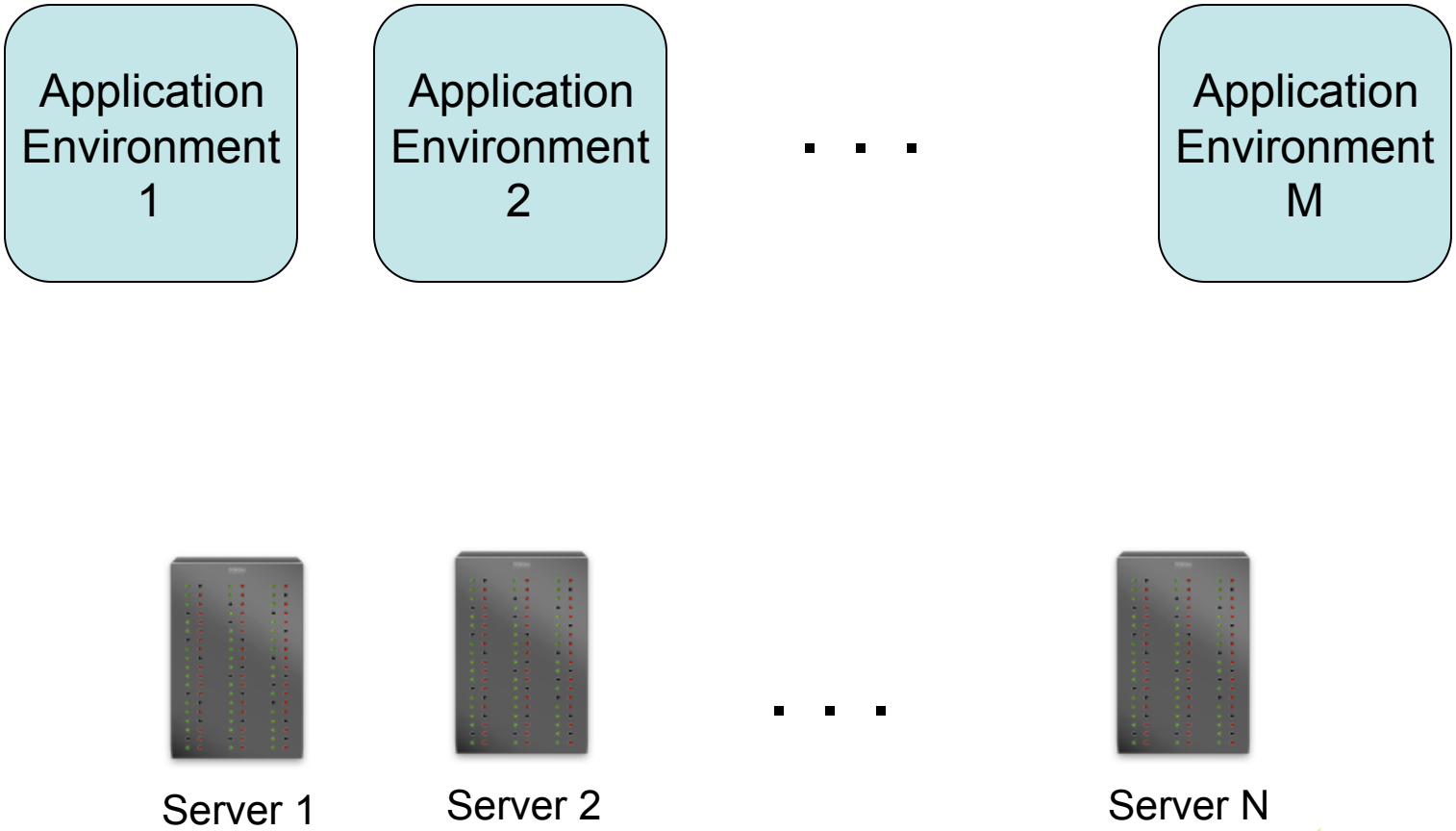
- E-commerce system
- Internet Data Center
- Virtualized Environment
- SOA Software Systems

"Resource Allocation for Autonomic Data Centers using Analytic Performance Models," M. Bennani and D. Menascé, *Proc. 2005 IEEE IntL. Conf. Autonomic Computing (ICAC-05)*, Seattle, WA, June 13-16, 2005.

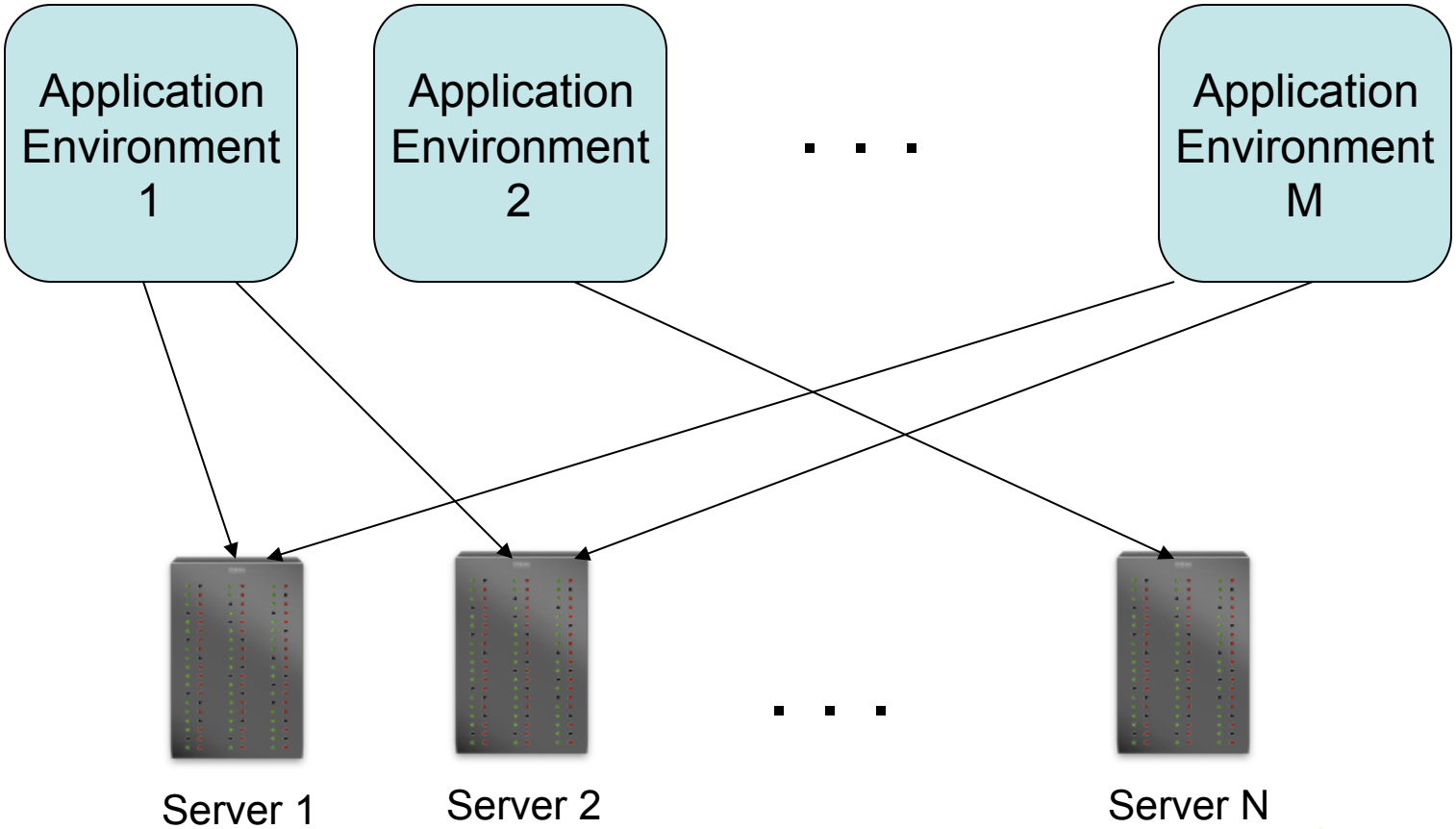
Motivation for Dynamic Resource Allocation in Data Centers

- Large data centers host several application environments (AEs)
- AEs run customers' applications (either online transactions or batch processes) on assigned servers (resources)
- AEs are subject to workloads that exhibit wide and unpredictable variations in their intensities
- Resources need to be redeployed dynamically among AEs to optimize some global utility function

Dynamic Resource Allocation Problem



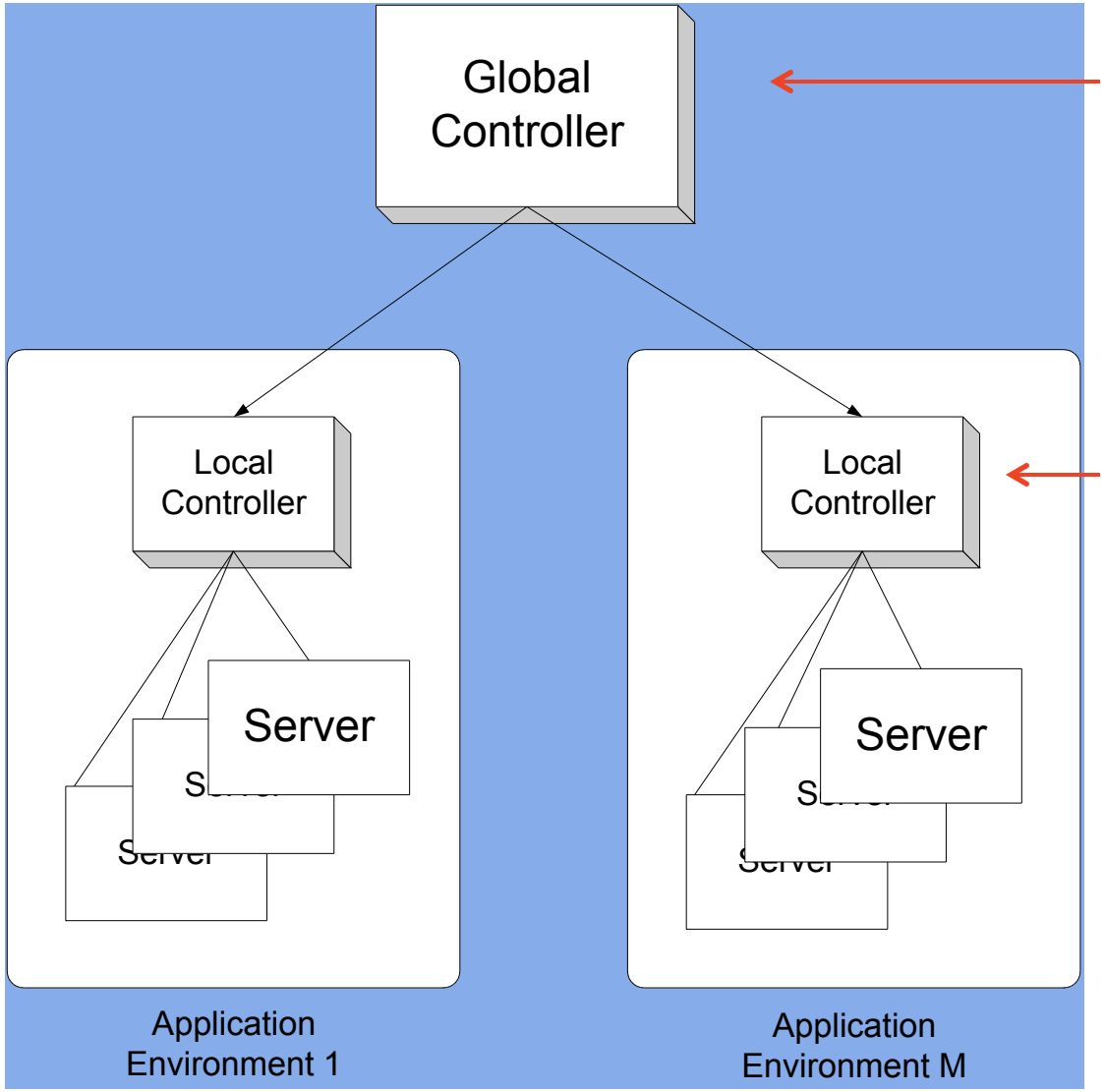
Dynamic Resource Allocation Problem



Dynamic Resource Allocation Problem

- The data center has the following characteristics:
  - M Application Environments (batch and online)
  - A fixed number, N, of physical resources (servers)
  - All servers have the same installed software
  - A server can be assigned to any AE

Dynamic Resource Allocation
Two-level Controllers



Decides how many servers to assign to each AE.

Implements Global Controller's decisions.

Dynamic Resource Allocation Problem – Utility Functions

- The global controller uses a global utility function, Ug, to assess the adherence of the overall data center performance to desired service levels (SLAs)

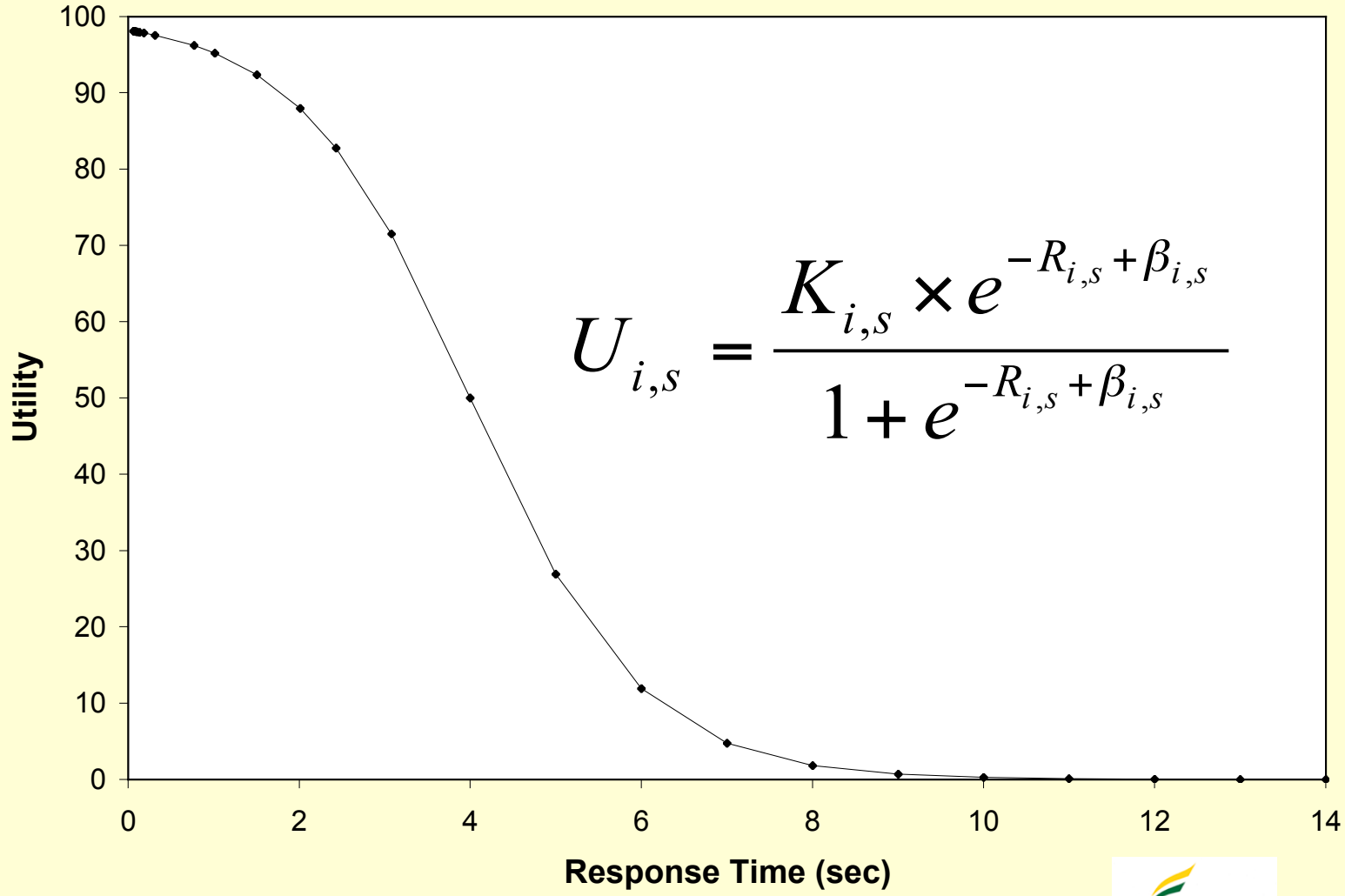
$$U_g = h(U_1, \ldots, U_M)$$

$$U_i = \sum_{s=1}^{S_i} a_{i,s} \times U_{i,s}$$

$$0 < a_{i,s} < 1$$

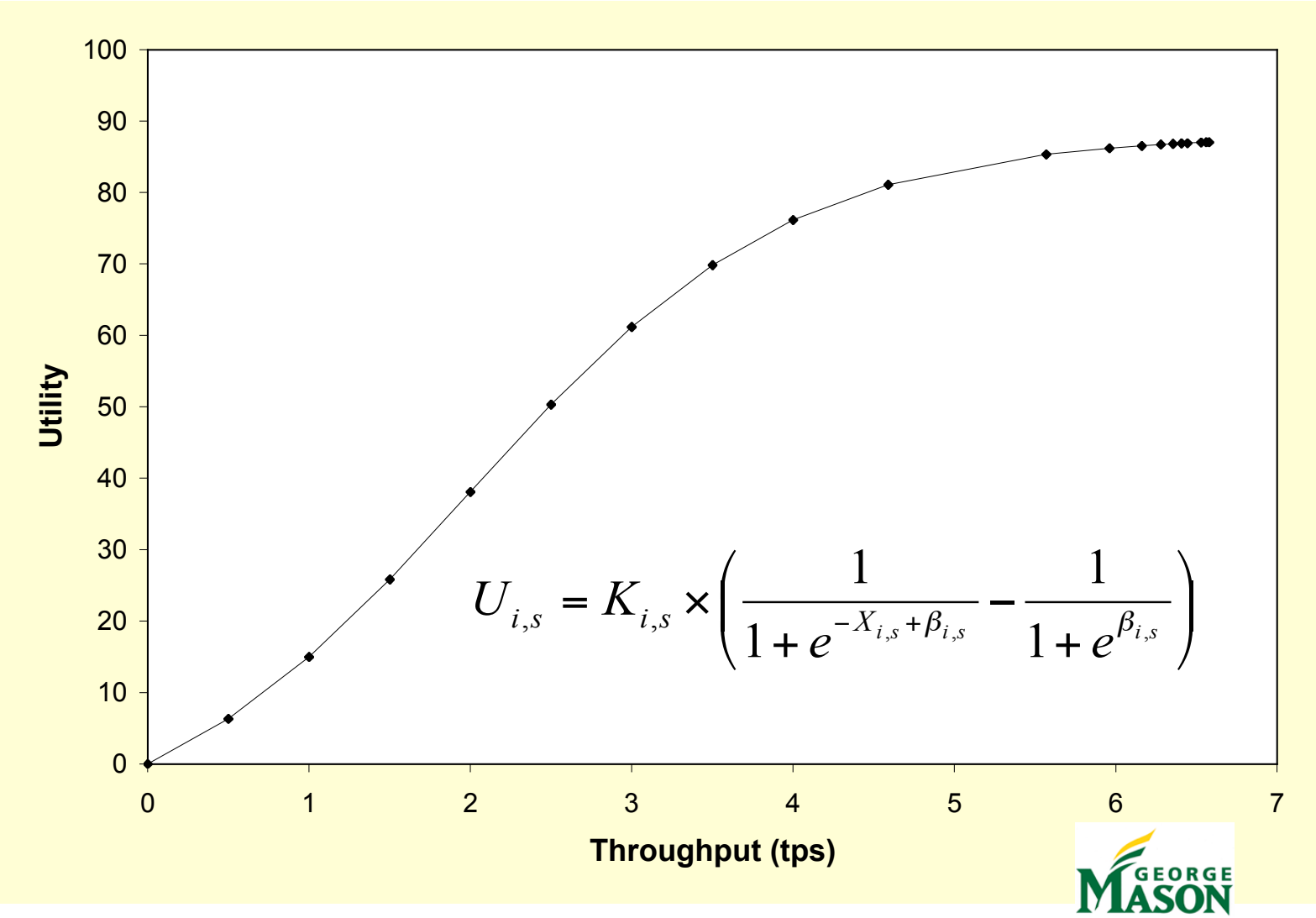
$$\sum_{s=1}^{S_i} a_{i,s} = 1$$

Utility Function as a Function of the Response Time

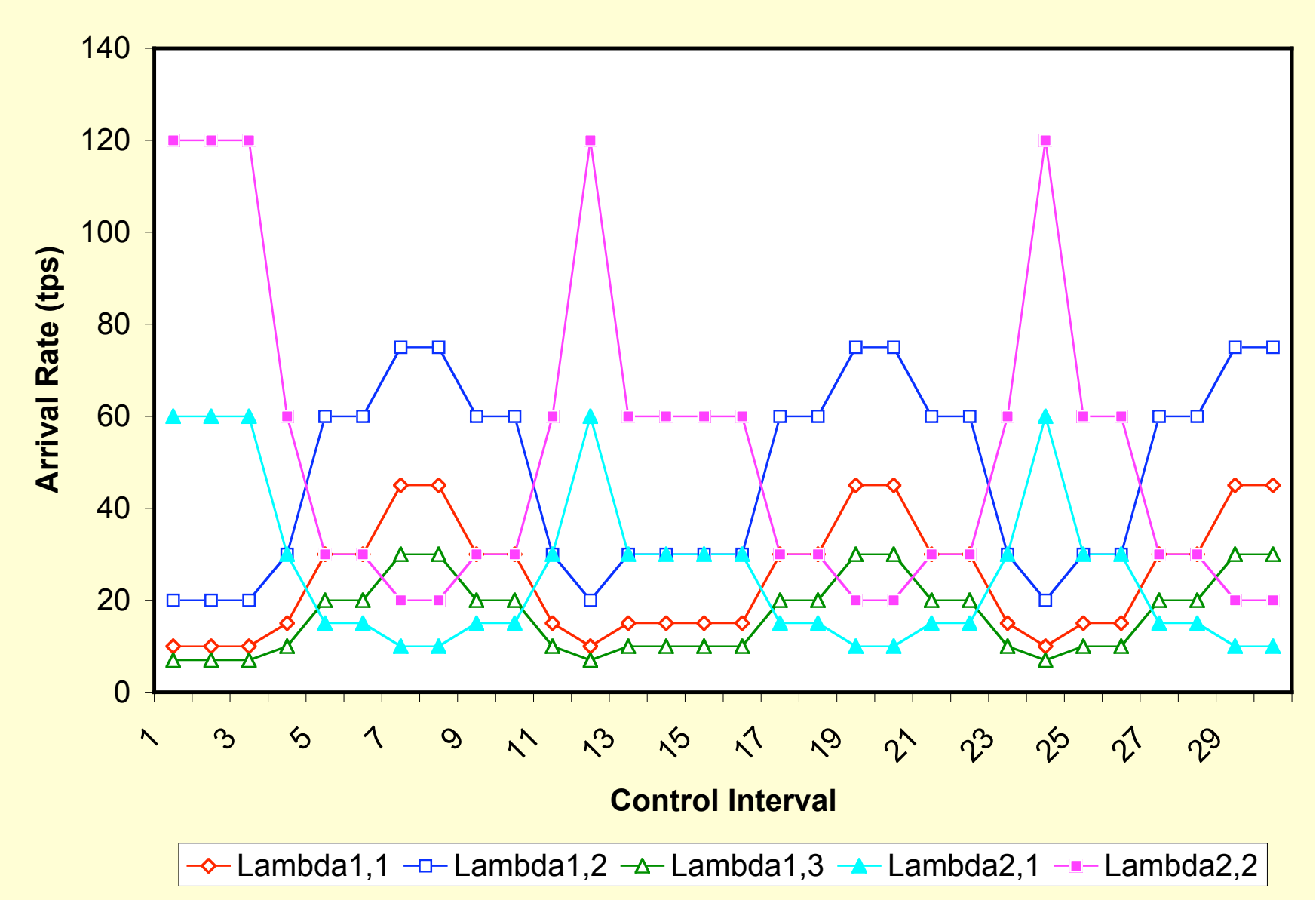


$$U_{i,s} = \frac{K_{i,s} \times e^{-R_{i,s} + \beta_{i,s}}}{1 + e^{-R_{i,s} + \beta_{i,s}}}$$

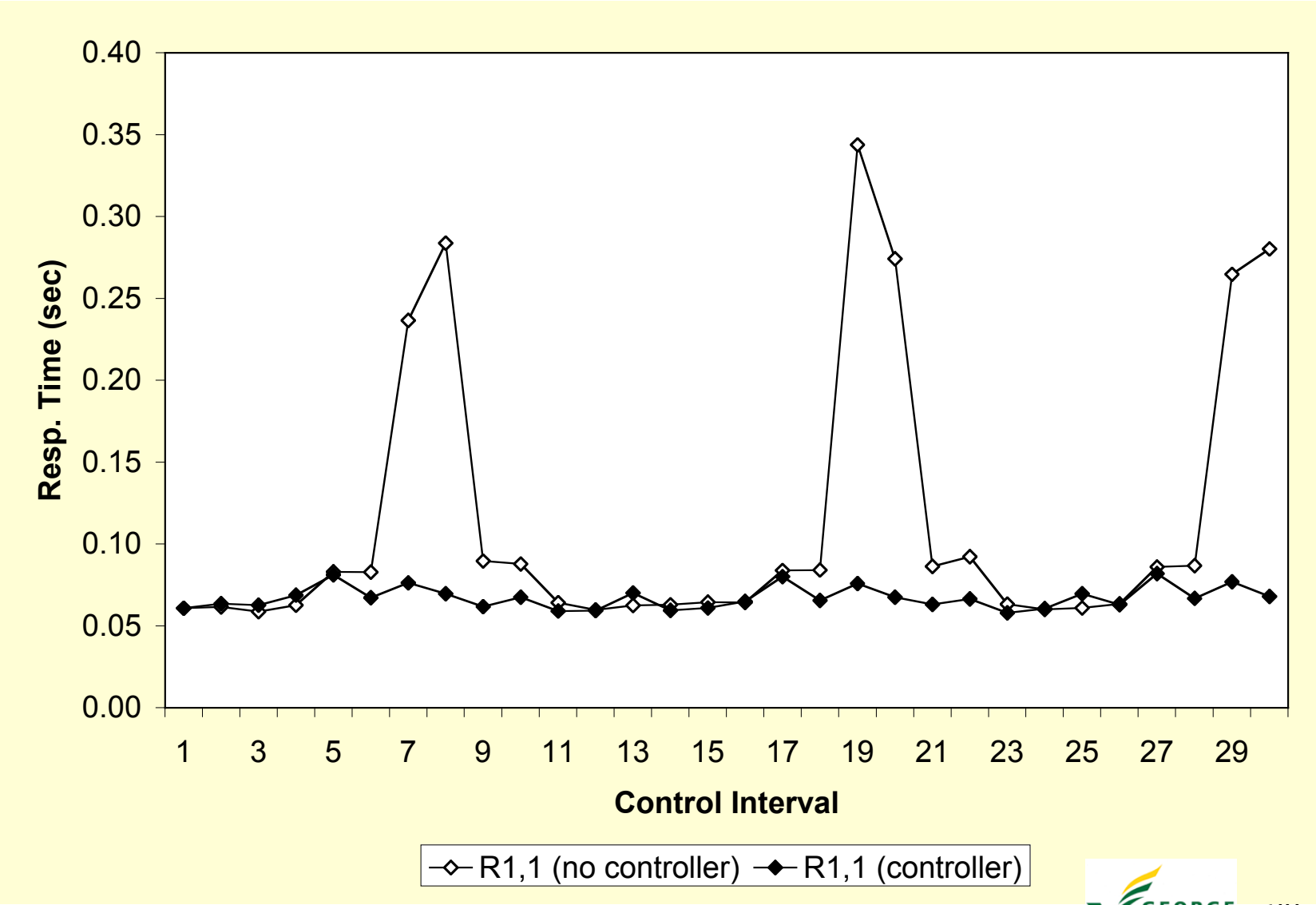
Utility Function as a Function of the Throughput



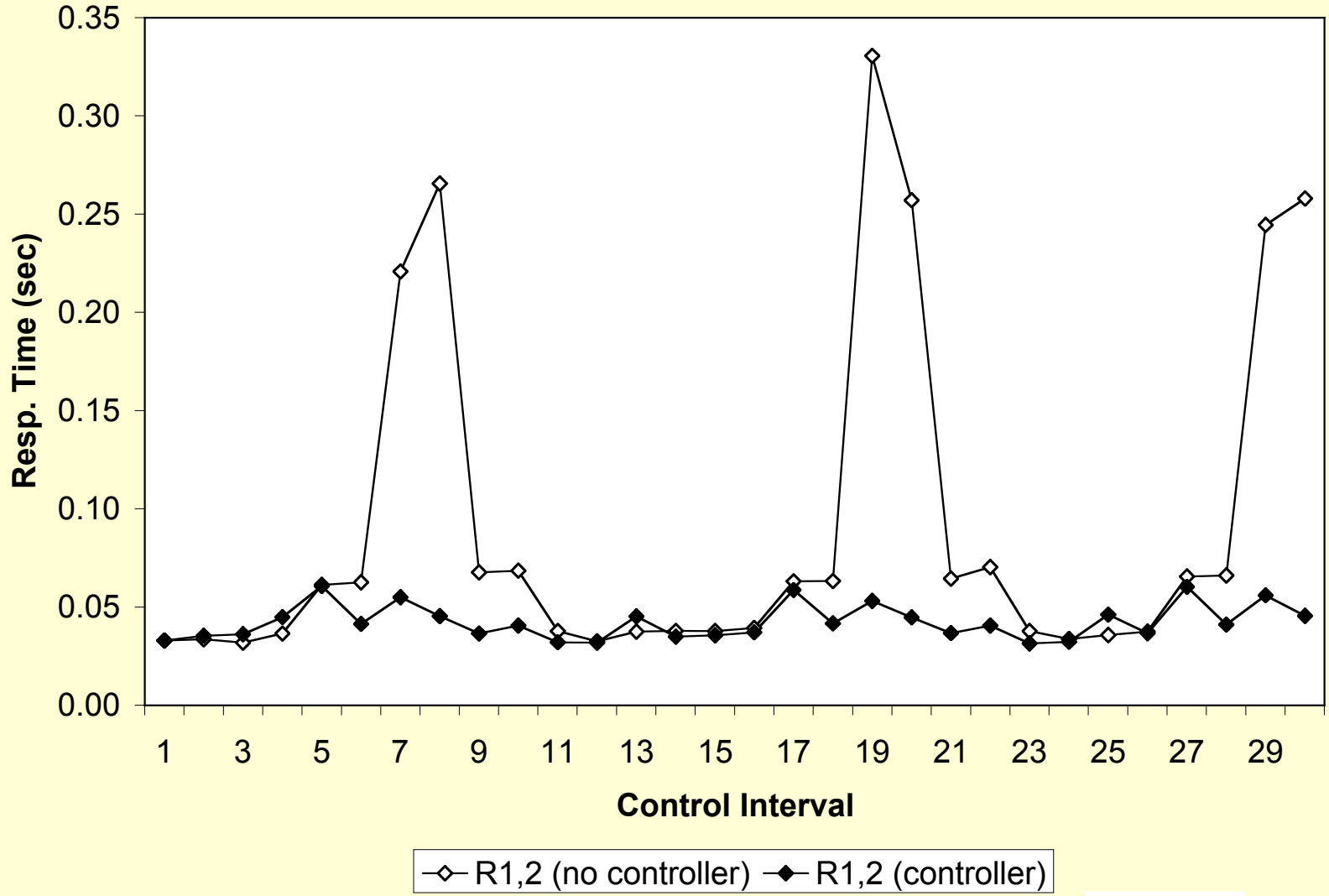
Workload Variation for Online AEs



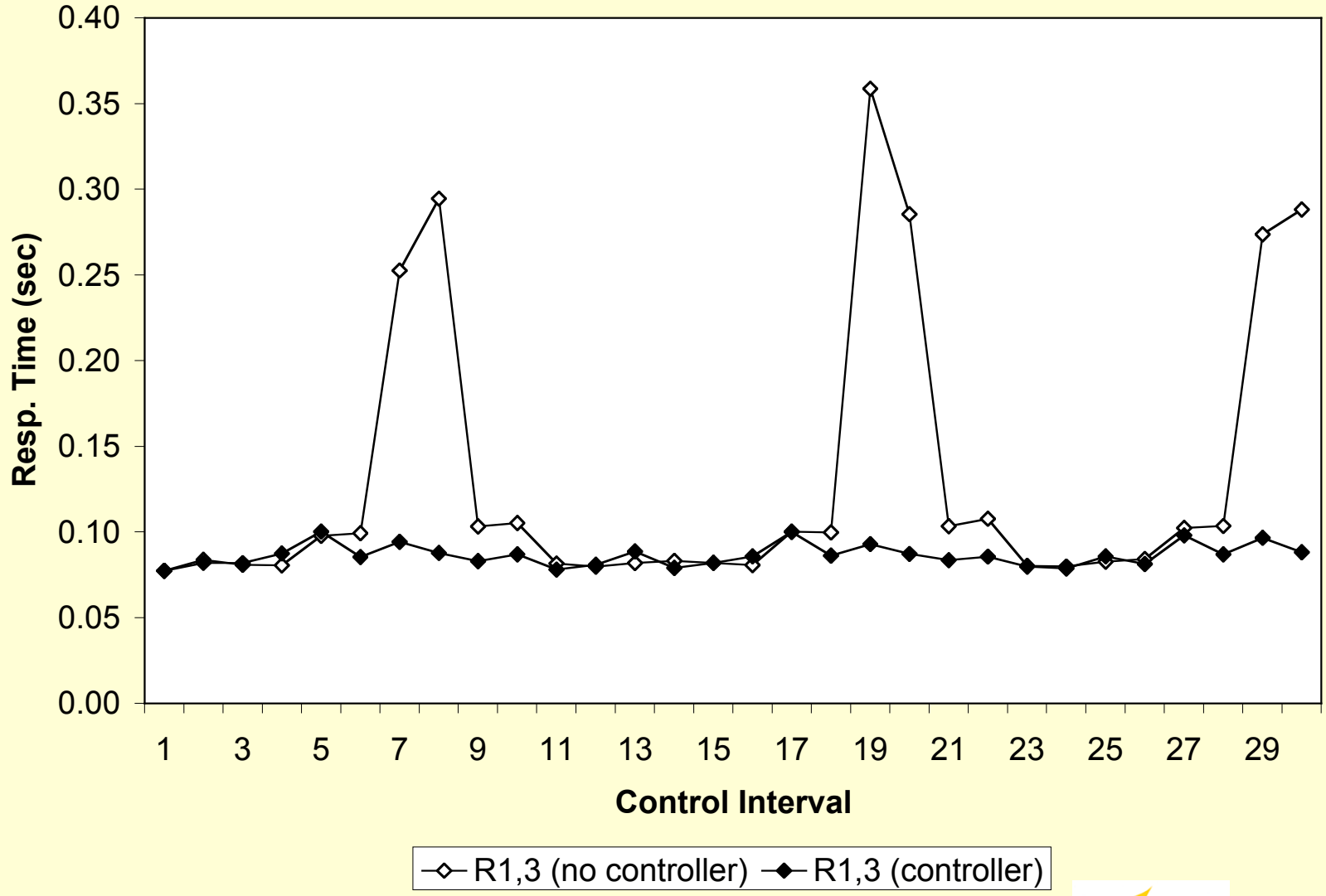
Response Times for Class 1 of AE 1



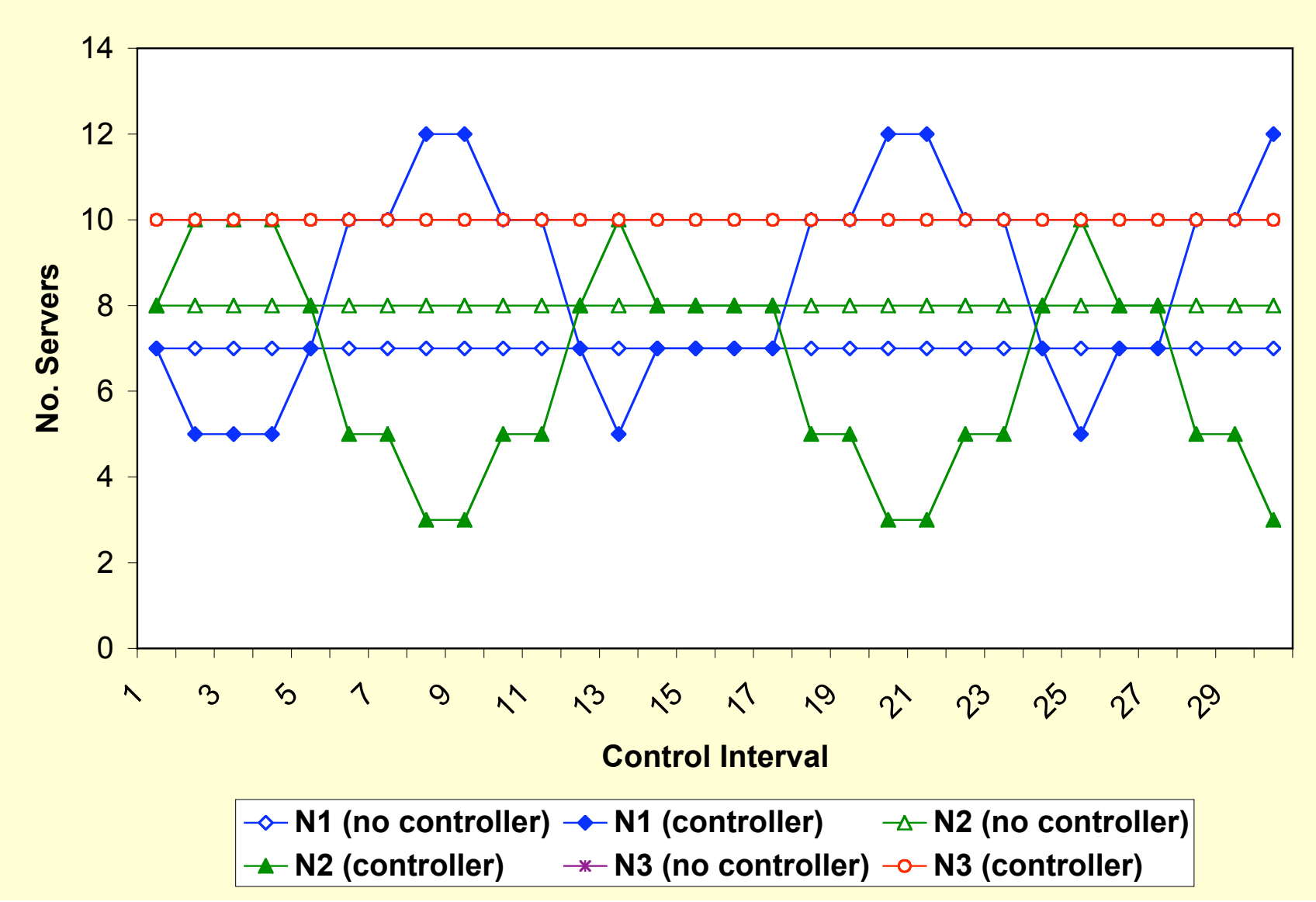
Response Times for Class 2 of AE 1



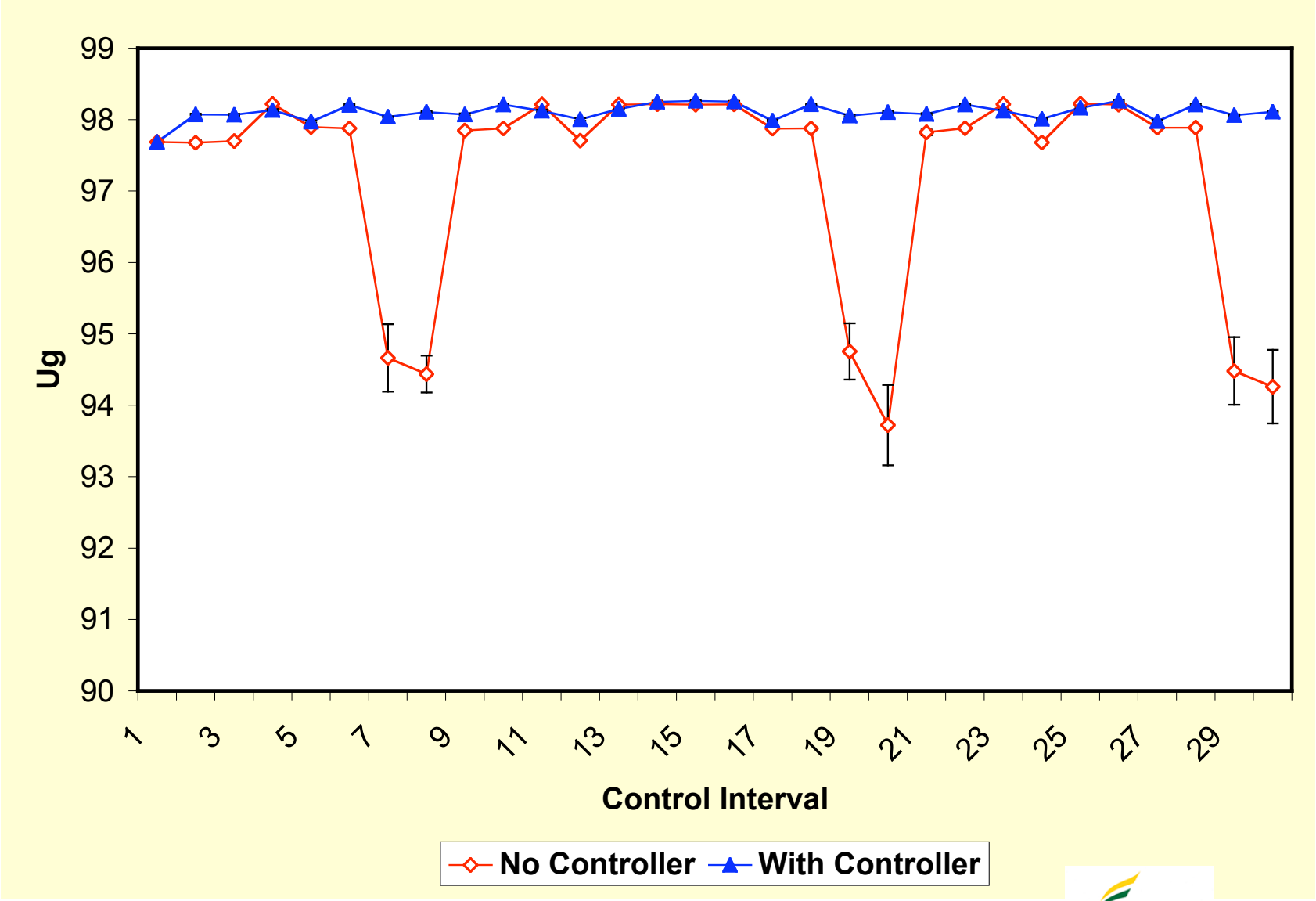
Response Times for Class 3 of AE 1



Variation of the Number of Servers



Variation of Global Utility



Examples of Autonomic Computing Systems

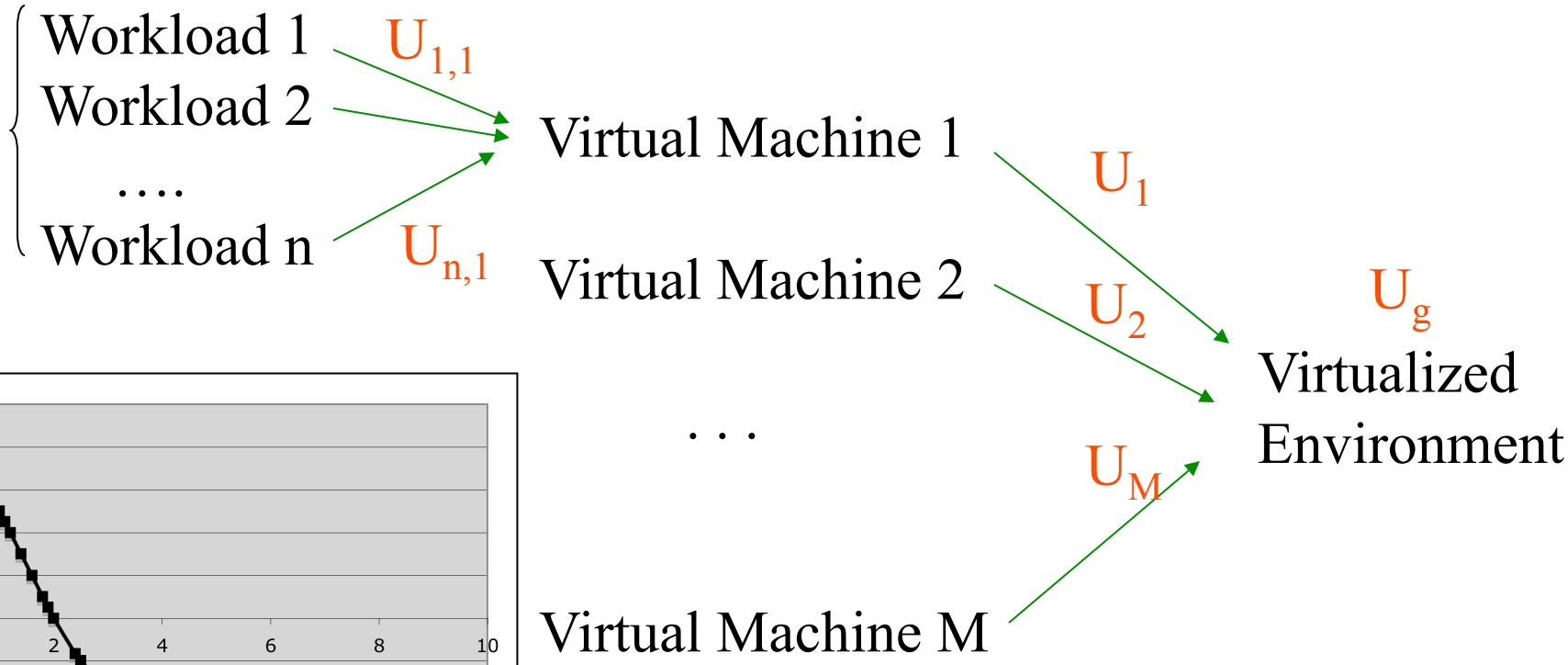
- E-commerce system
- Internet Data Center
- Virtualized Environment
- SOA Software Systems

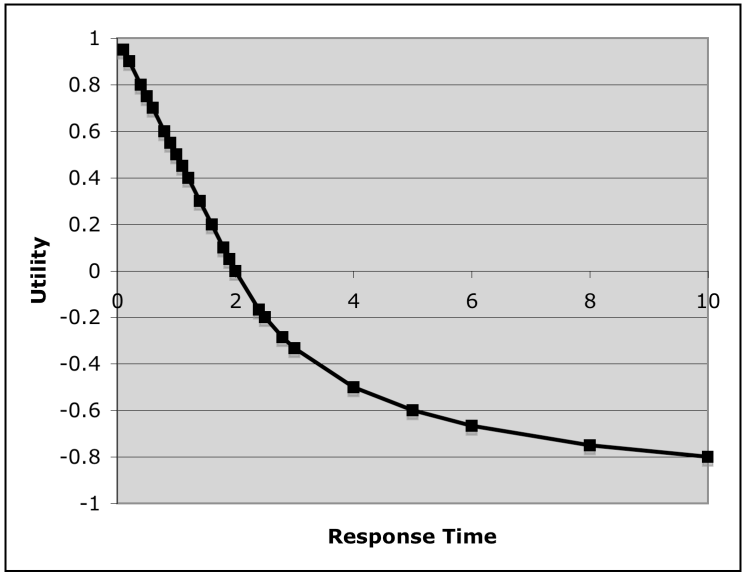
"Autonomic Virtualized Environments," D.A. Menasce and M.N. Bennani, IEEE International Conference on Autonomic and Autonomous Systems, July 19-21, 2006, Silicon Valley, CA, USA

CPU Allocation Problem for Autonomic Virtualized Environments

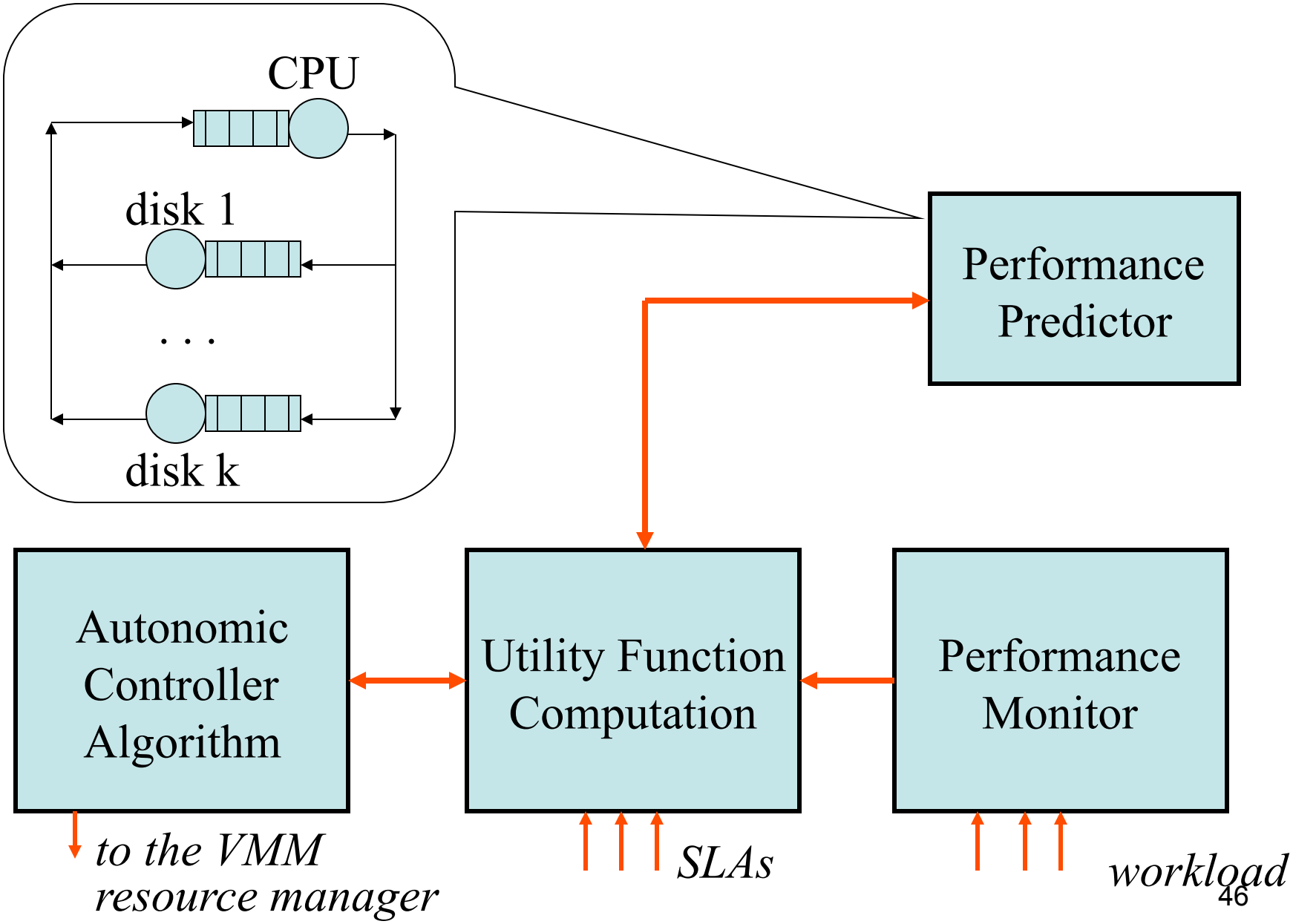
- Existing systems allow for manual allocation of CPU resources to VMs using *CPU priorities* or *CPU shares*.
- Need automated mechanism for the adjustment of CPU shares or the priorities of the virtual machines in order to maximize the global utility of the entire virtualized environment

CPU Allocation Problem for Autonomic Virtualized Environments (Cont'd)

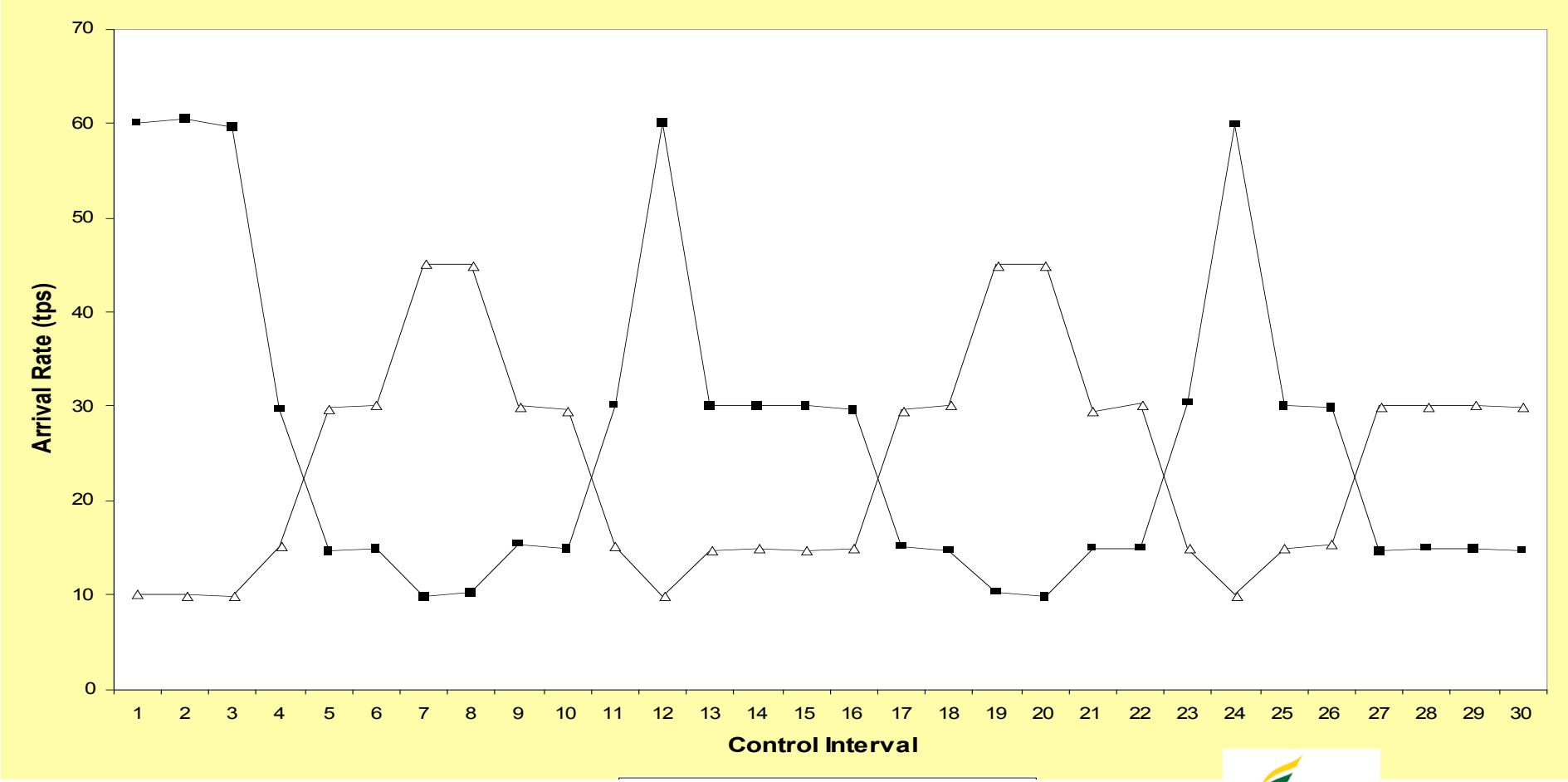




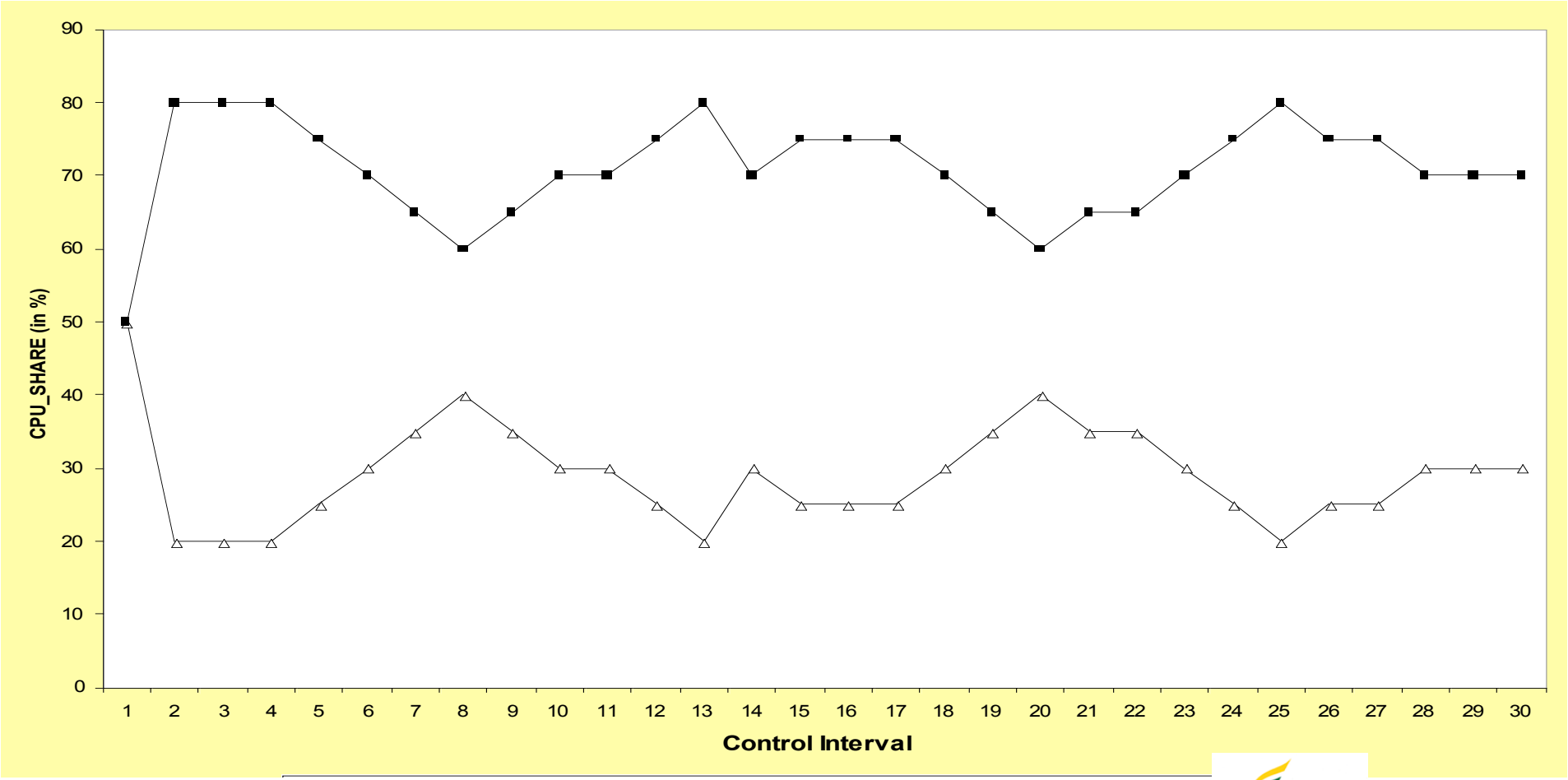
Virtualization Controller Architecture



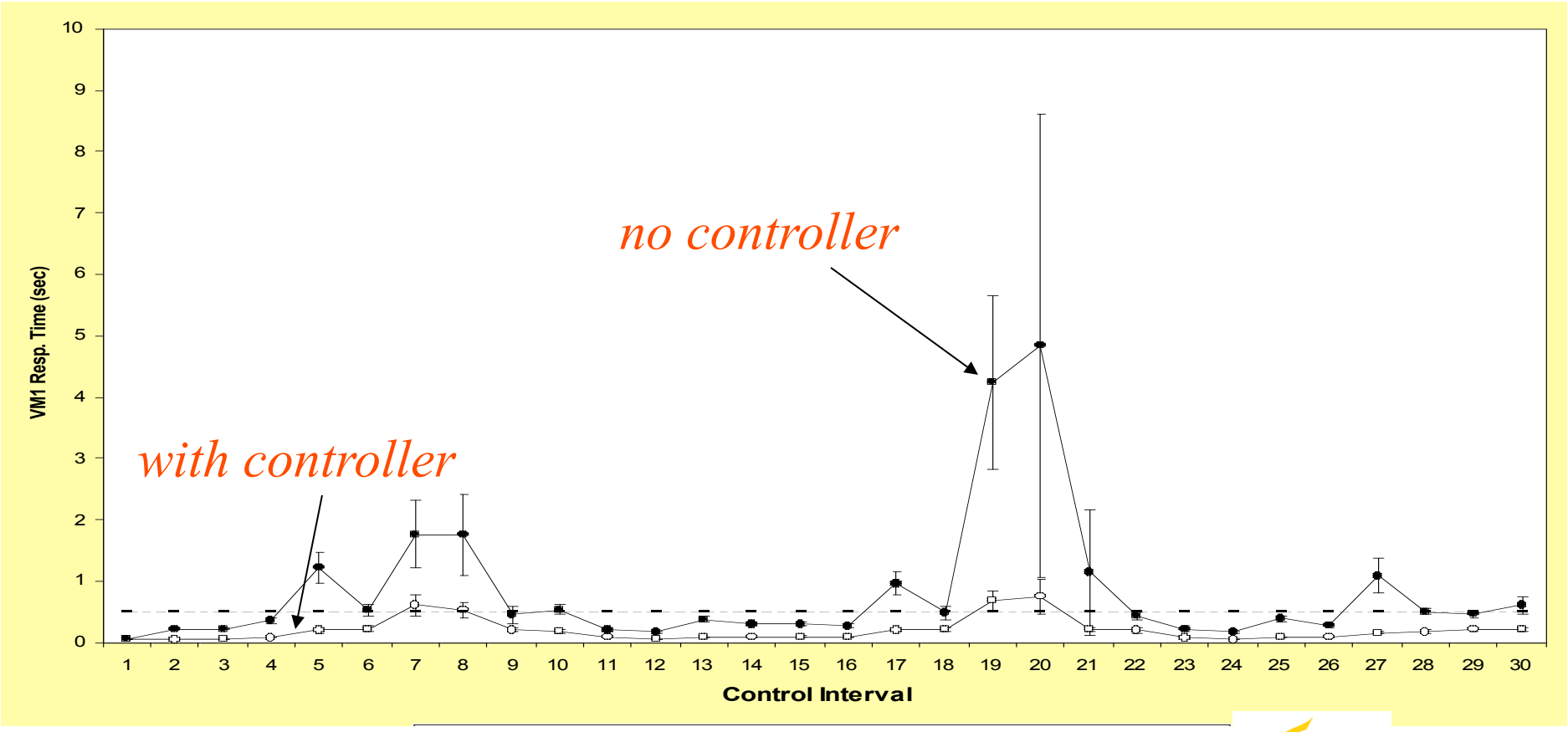
CPU Shares Based Allocation: Workload Variation



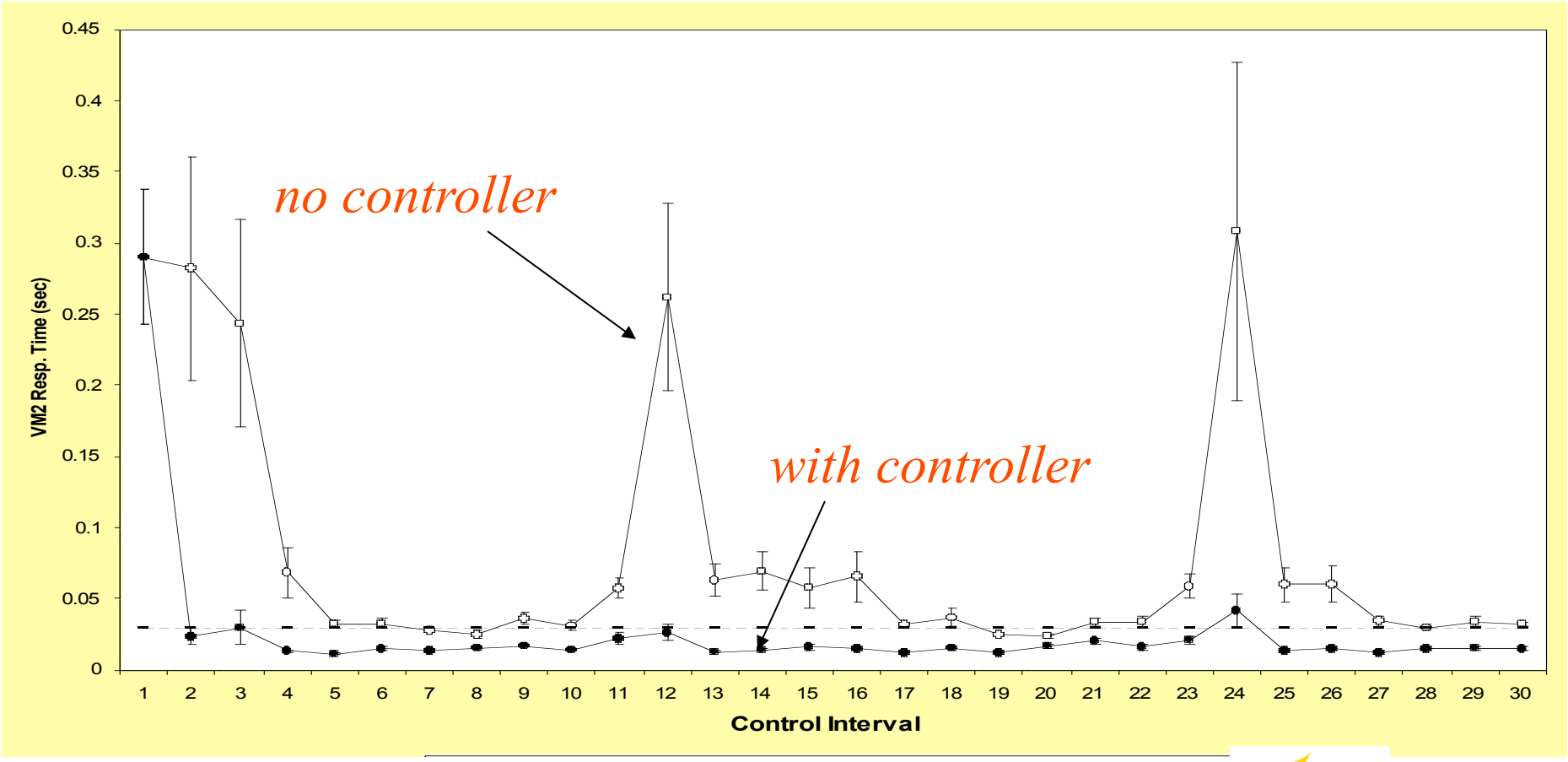
CPU Shares Based Allocation: CPU Shares Variation



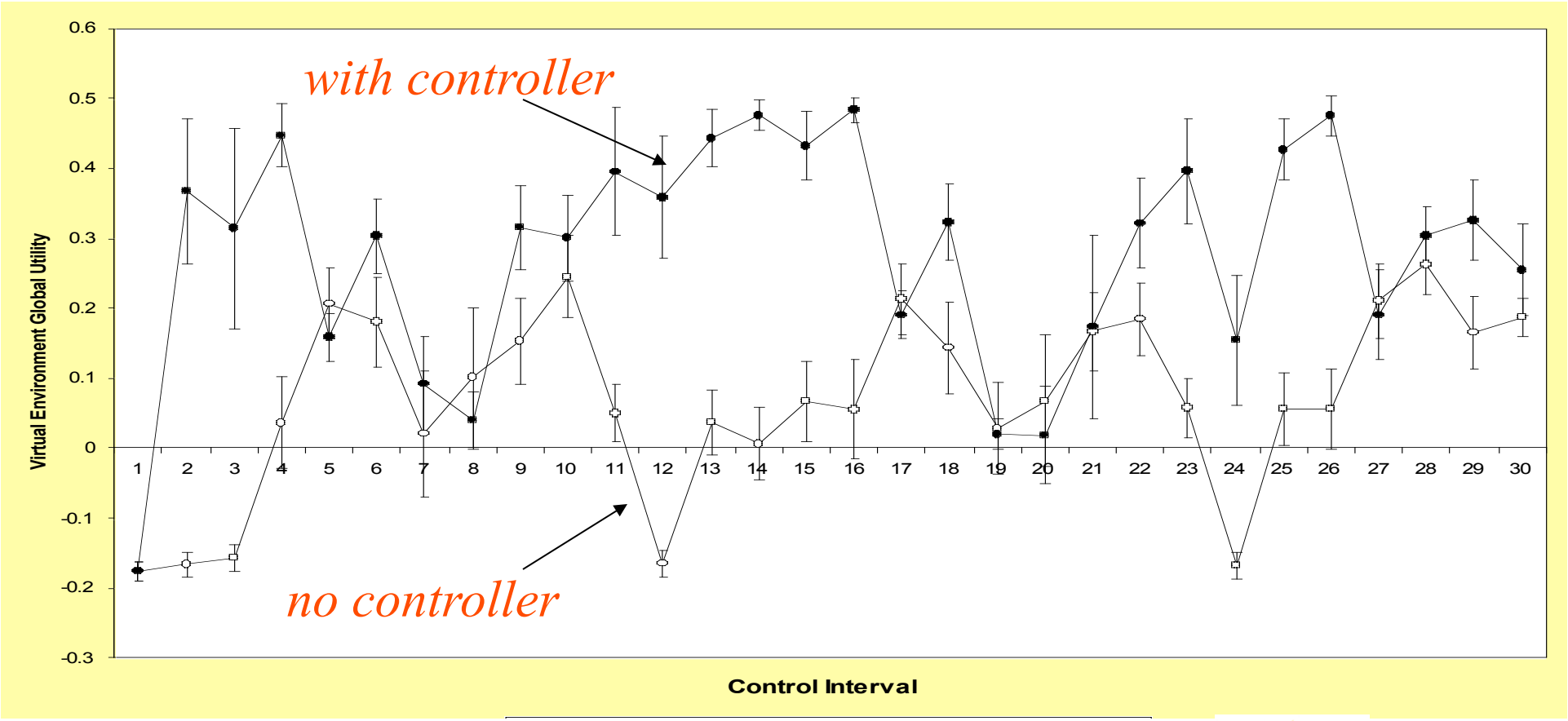
CPU Shares Based Allocation: Response Time for VM1



CPU Shares Based Allocation: Response Time for VM2



CPU Shares Based Allocation: Global Utility



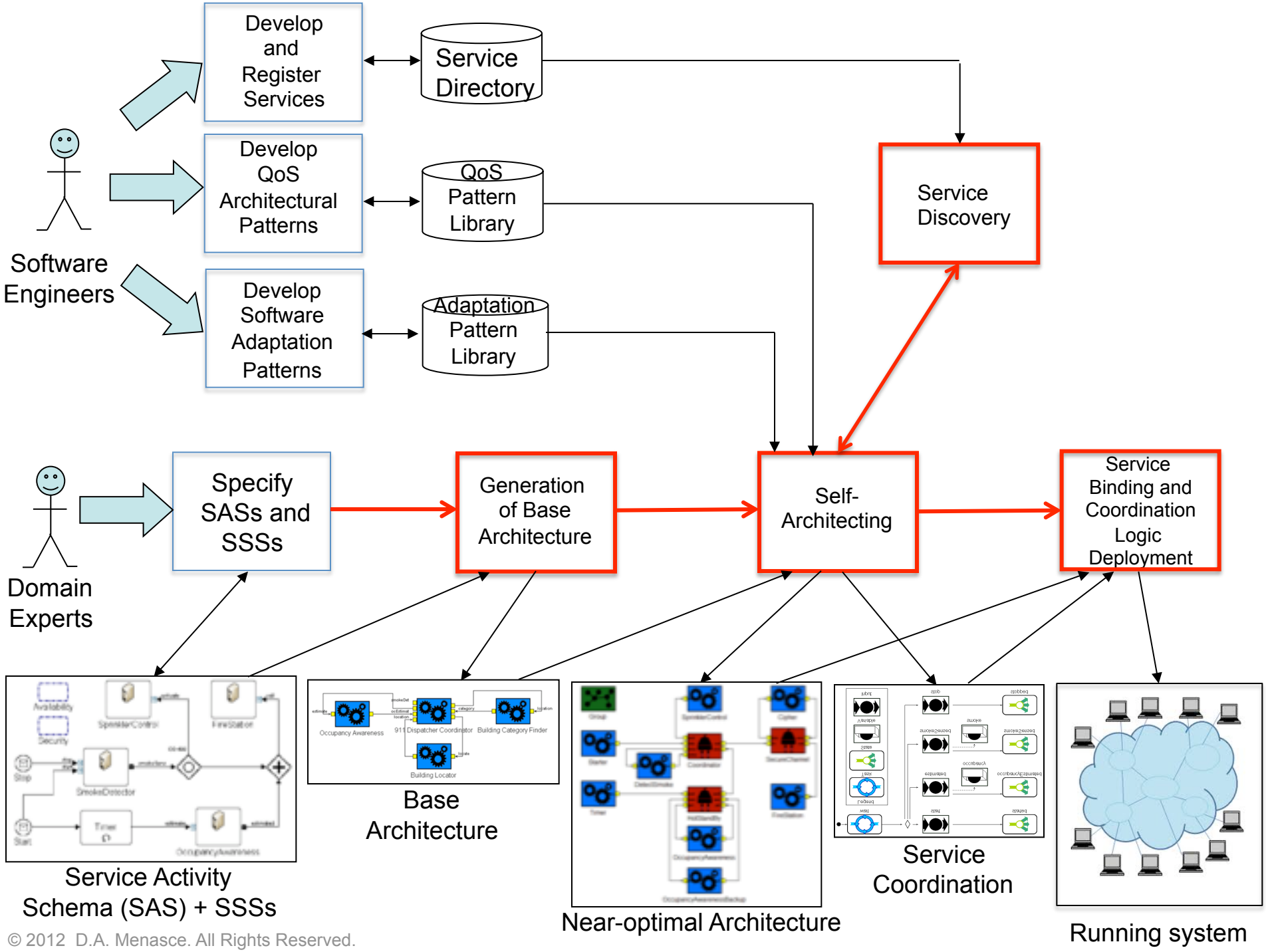
Examples of Autonomic Computing Systems

- E-commerce system
- Internet Data Center
- Virtualized Environment
- SOA Software Systems

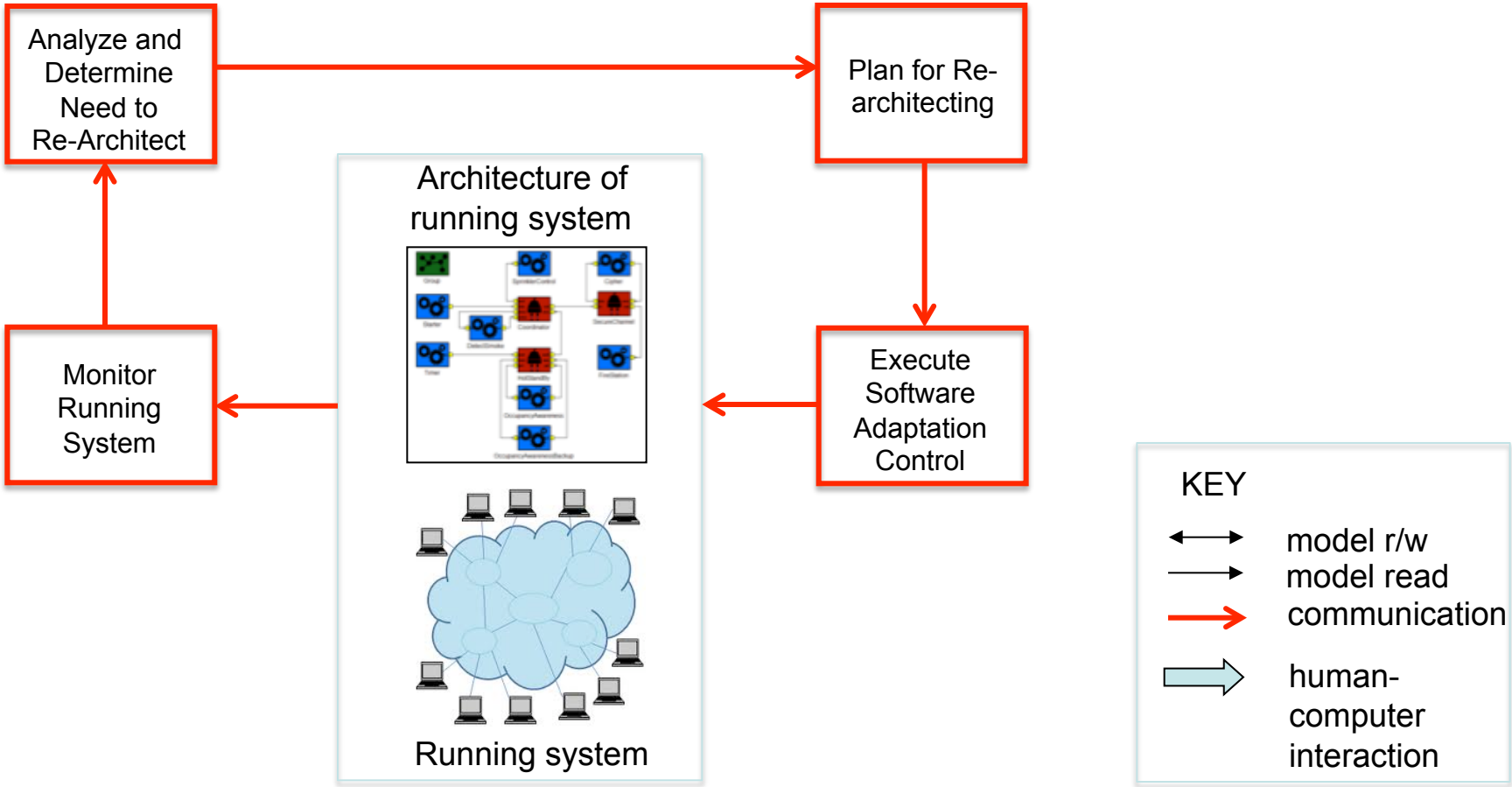
SASSY: A Framework for Self-Architecting Service-Oriented Systems, D.A. Menasce, H. Gomaa, S. Malek, and J.P. Sousa, IEEE Software, Vol. 28, No. 6, November/December 2011.

Self-Architecting SOA Software Systems (SASSY)

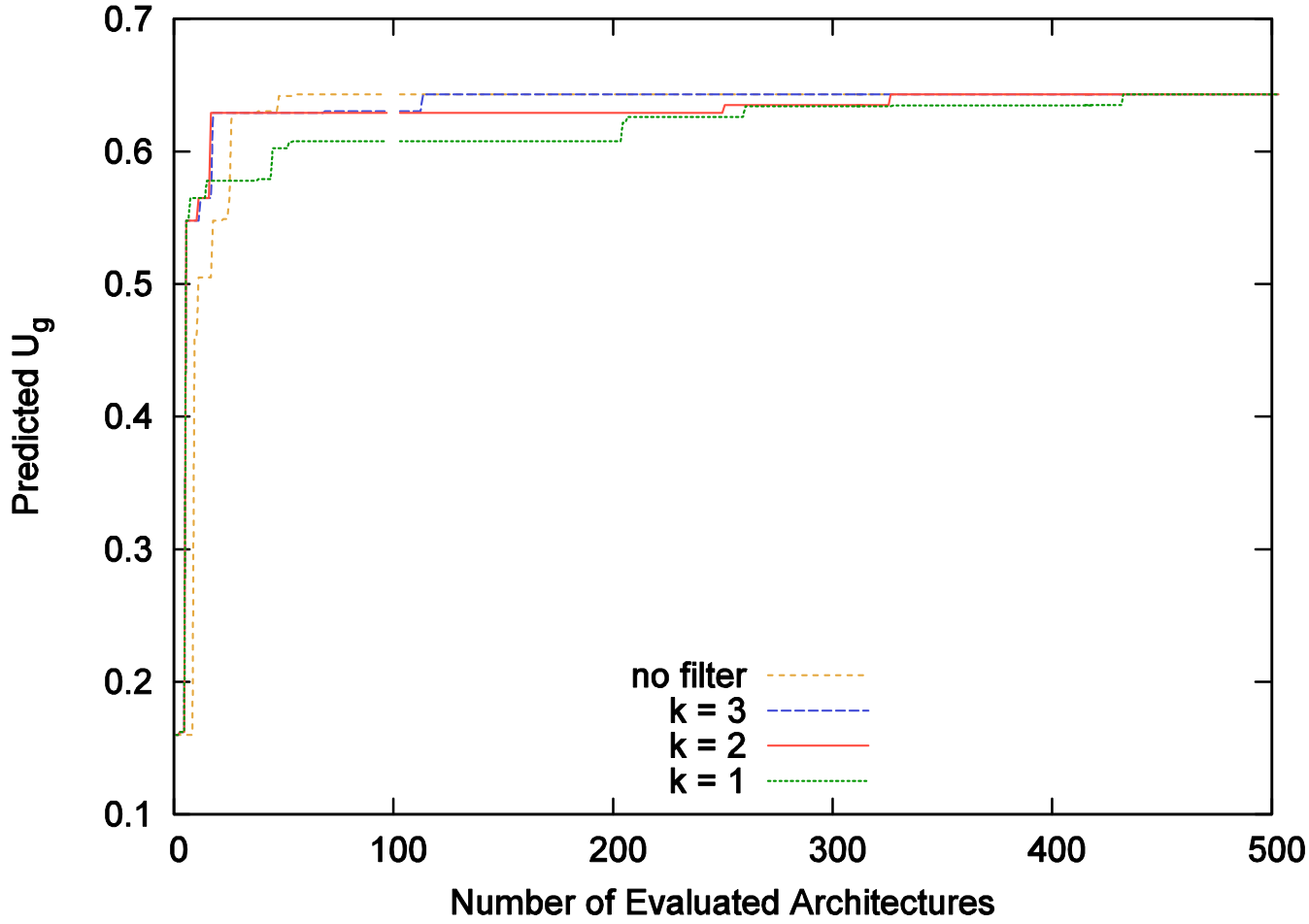
- SASSY allows domain experts to specify requirements using a high-level visual activity language.
- SASSY generates the initial software architecture optimized to maximize a utility function.
- The running system is constantly monitored and SASSY automatically re-architects the system when needed.



SASSY: Run-time Adaptation



Architecture Adaptation Search Trajectories



Concluding Remarks

- Self-optimization is a key attribute of AC systems.
- Performance models and combinatorial search techniques can be useful in dynamically adjusting a system's configuration as the workload changes.

Thanks! Questions?

www.cs.gmu.edu/faculty/menasce.html