Virtualization: Concepts, Applications, and Performance Modeling

Daniel A. Menascé, Ph.D.

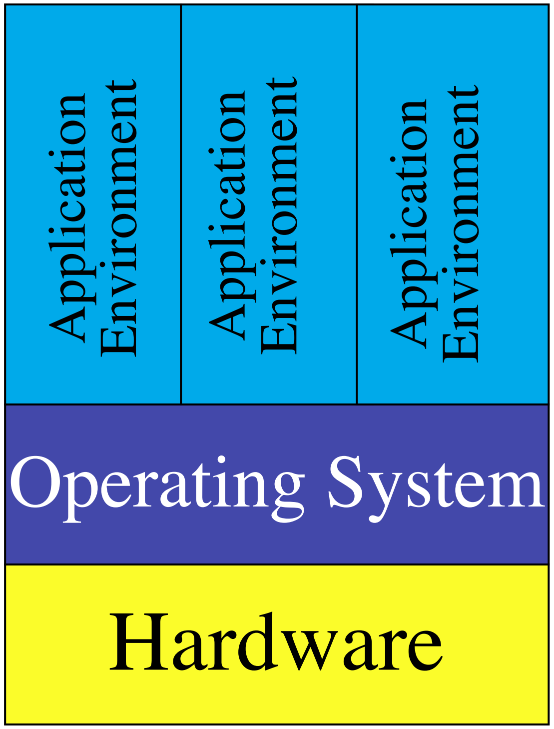
The Volgenau School of Information Technology and Engineering

Department of Computer Science

George Mason University

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

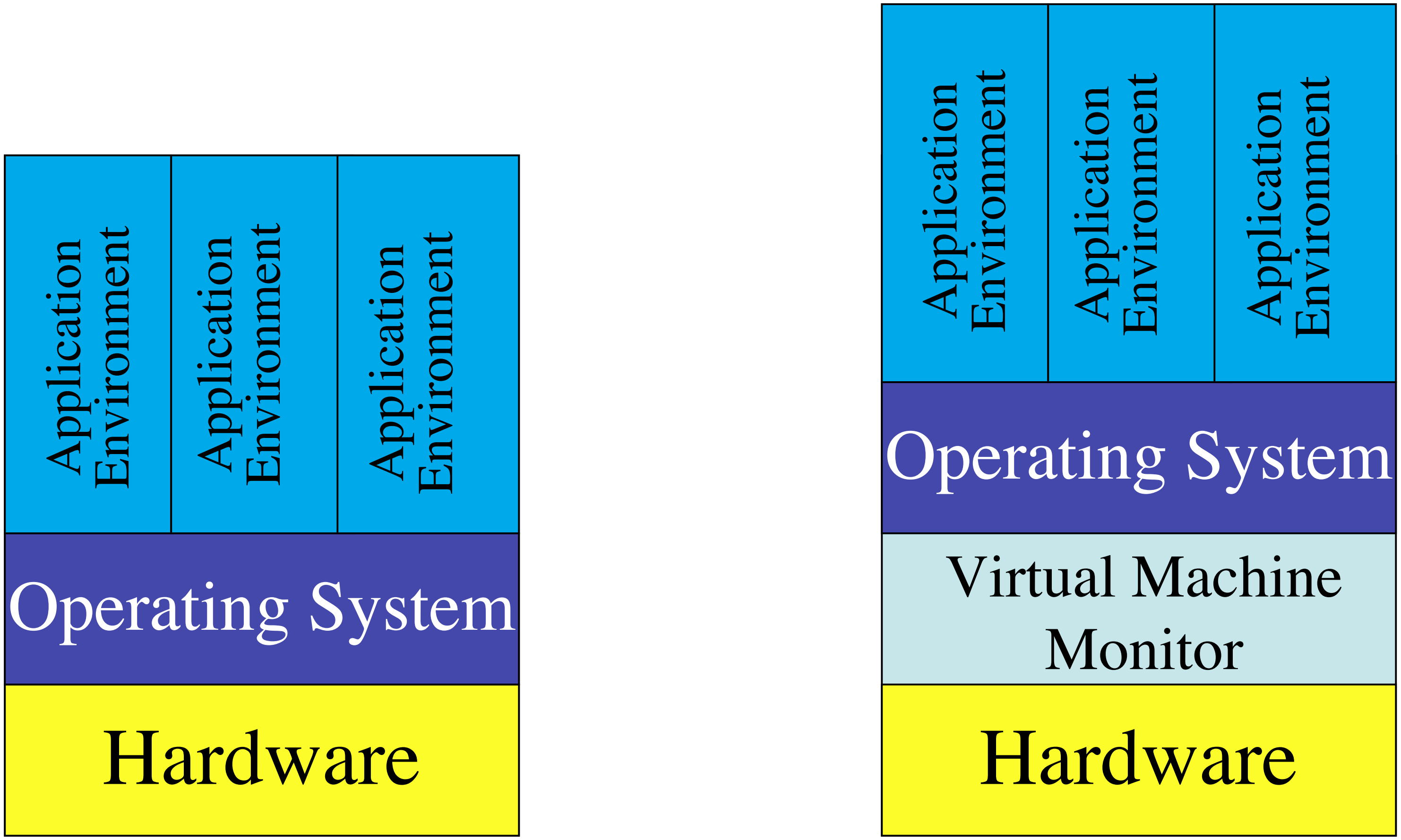
Basic Concepts in Virtualization



Non-virtualized system

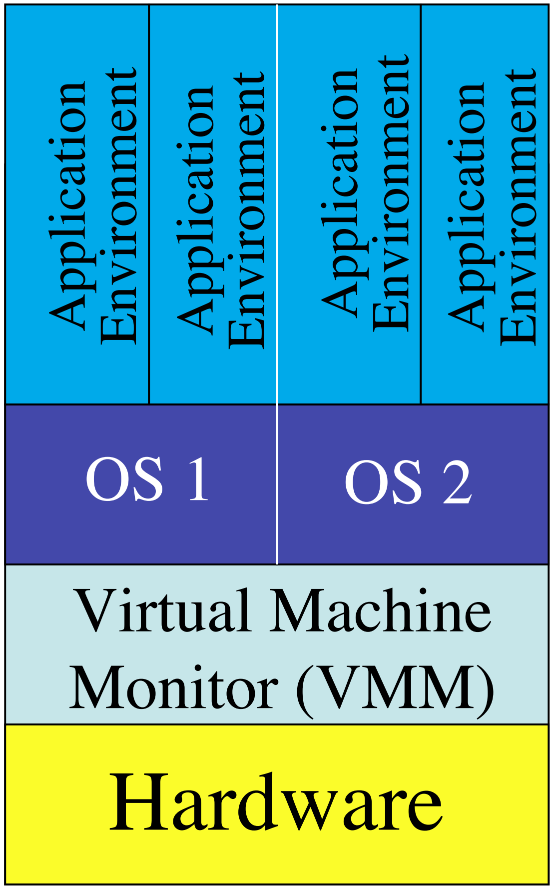
- The OS controls access to the hardware resources.
- The instruction set is divided into privileged and non-privileged.
- The machine can be in two modes of operation: user and supervisor.
- Only non-privileged instructions can be executed in user mode.
- Any instruction can be executed in supervisor mode.
- Application environments execute in user mode and the OS in supervisor mode.

Basic Concepts in Virtualization



Non-virtualized system *Virtualized system*

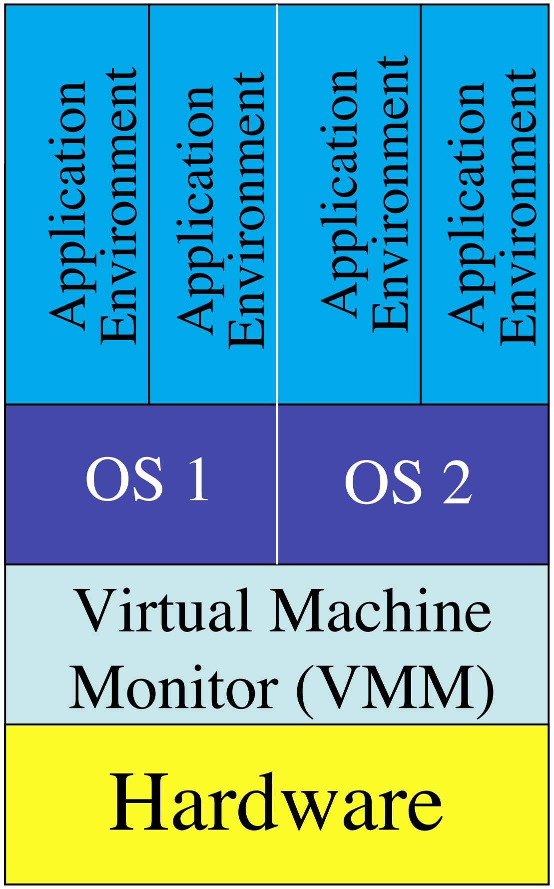
Basic Concepts in Virtualization



Virtualized system

- The VMM controls access to the hardware resources.
- OSs (called guest OSs) execute in user mode and the VMM in supervisor mode.
- The VMM interprets in software privileged instructions that would be executed by an OS.
- Any non-privileged instruction issued by an OS or Application Environment is executed directly by the machine.
- This is called virtualization by direct execution or full virtualization.

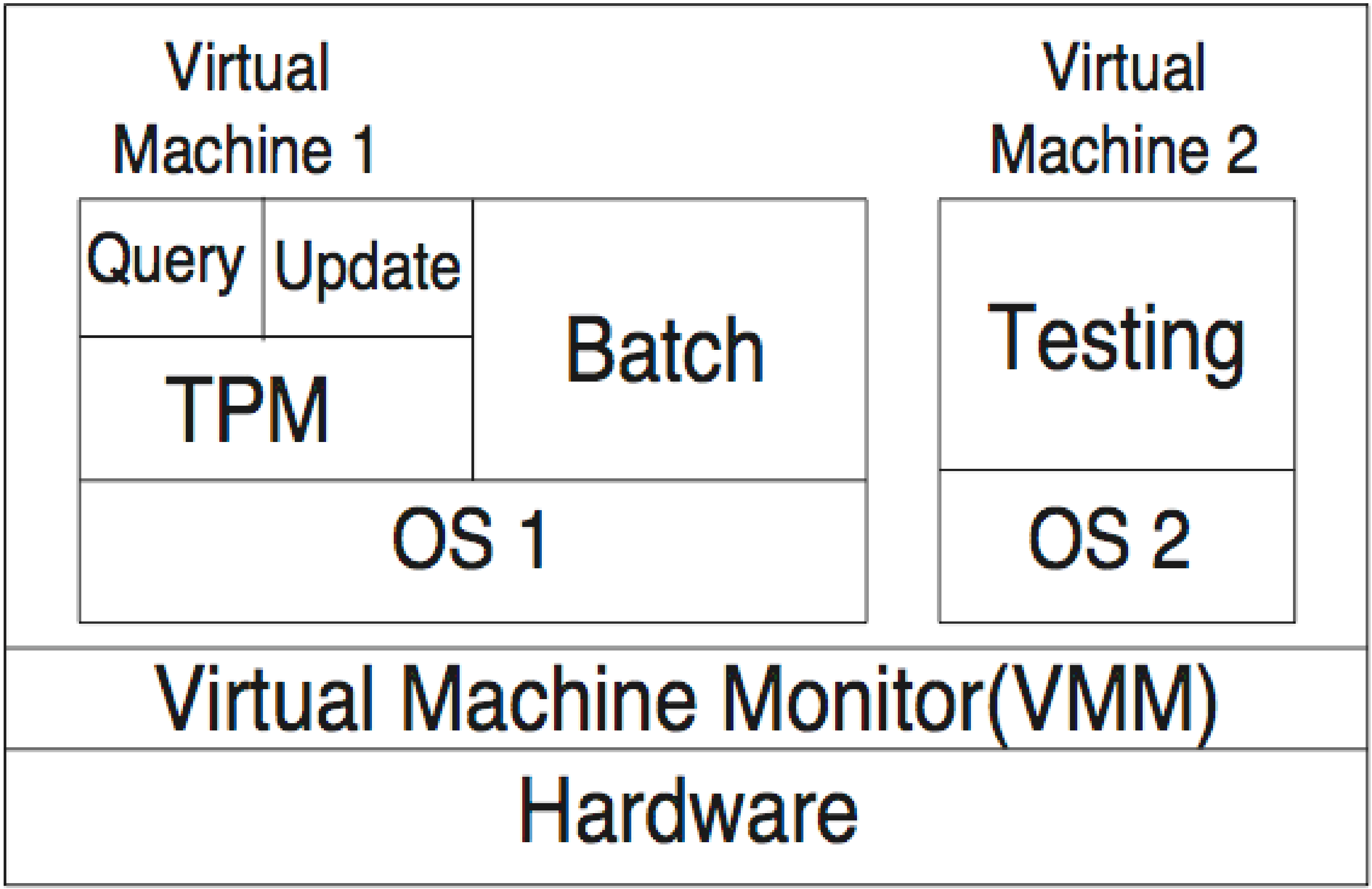
Basic Concepts in Virtualization: the x86 architecture



Virtualized system

- The x86 architecture provides four levels of privilege: rings 0 through 3.
- In a non virtualized environment, the OS executes in ring 0 and the applications in ring 3.
- In a virtualized environment that uses paravirtualization (e.g., Xen) the VMM runs at ring 0, the guest OS at ring 1 and the applications at ring 3.

Basic Concepts in Virtualization



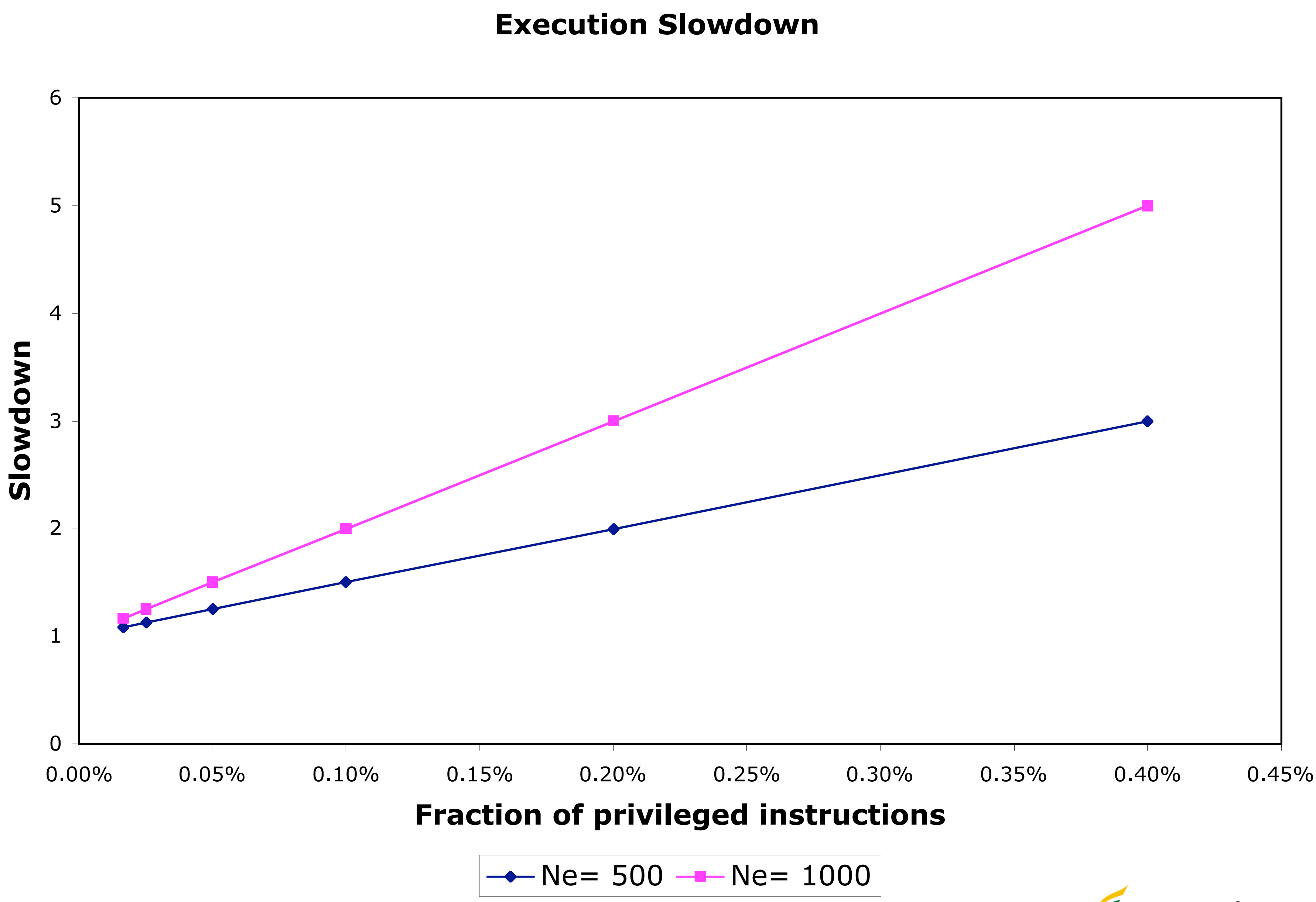
Virtualization Slowdown, S_v

- In the direct execution approach, privileged instructions have to be emulated by the VMM instead of being executed by the hardware.

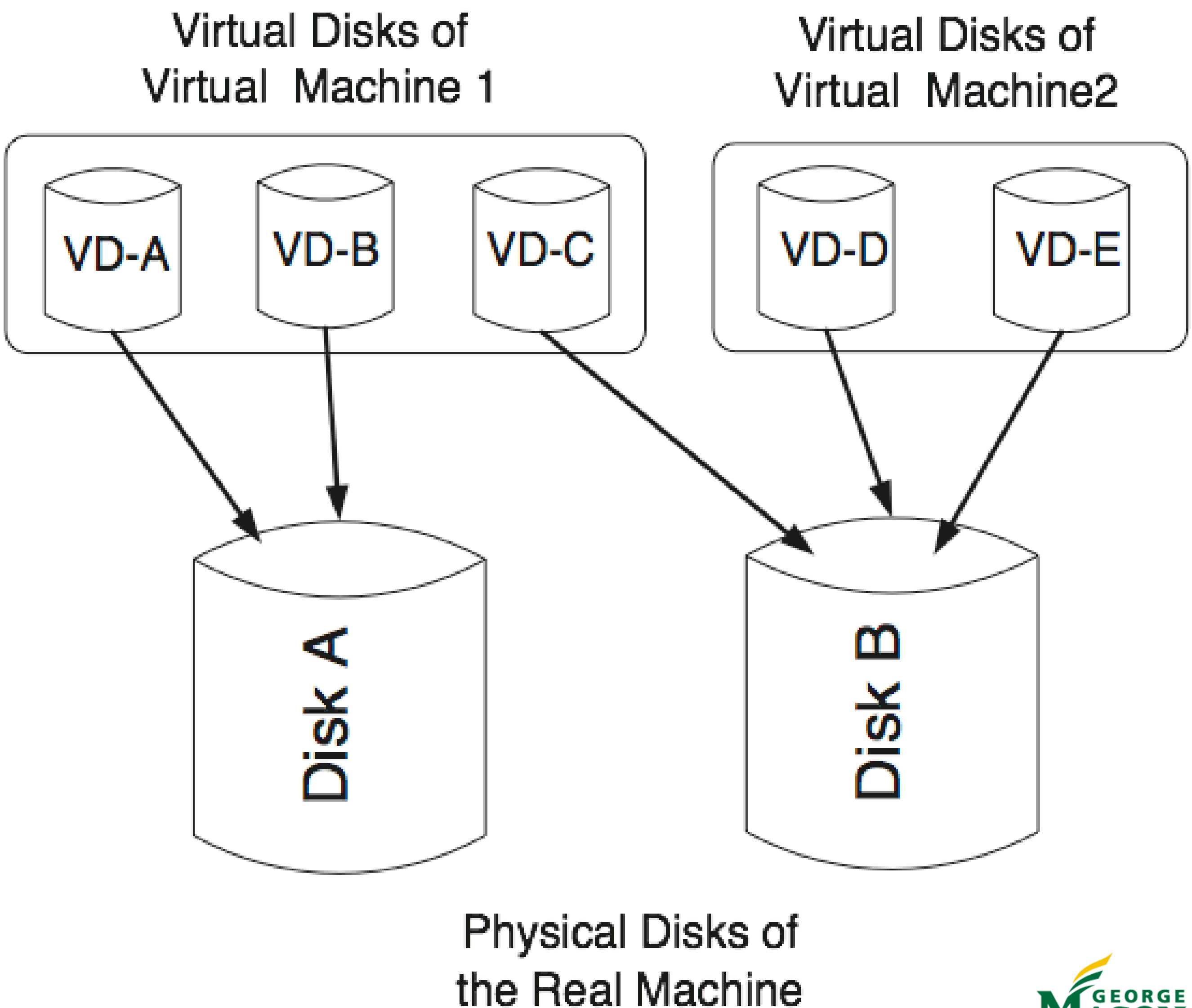
$$S_v = f_p \times N_e + (1 - f_p)$$

f_p : fraction of privileged instructions executed by a VM.

N_e : average number of instructions required by the VMM to emulate a privileged instruction.


Execution Slowdown
Slowdown
0
1
2
3
4
5
6
0.00%
0.05%
0.10%
0.15%
0.20%
0.25%
0.30%
0.35%
0.40%
0.45%
Fraction of privileged instructions
Ne= 500
Ne= 1000

Mapping of Virtual to Physical Disks



Some History

- Over 30 years ago: IBM's VM/370
- 80's: decrease in hardware cost caused migration away from mainframes and virtualization faded away.
- More recent networked environments brought problems such as reliability, security, increased administration cost and complexity, thermal dissipation.
  - Virtualization is poised to address these problems.

Advantages of Virtualization

- Security
  - Compartmentalized environments allow for better choice of guest operating system for each environment (e.g., run Apache on top of Linux and MS SQL on top of Windows XP)
- Reliability and Availability
  - A software failure in one VM does not affect other VMs
- Cost
  - Server consolidation can bring cost reductions from hardware economies of scale, personnel cost reductions, floor space, and software licenses.
    - Typical savings: 29% to 64%.

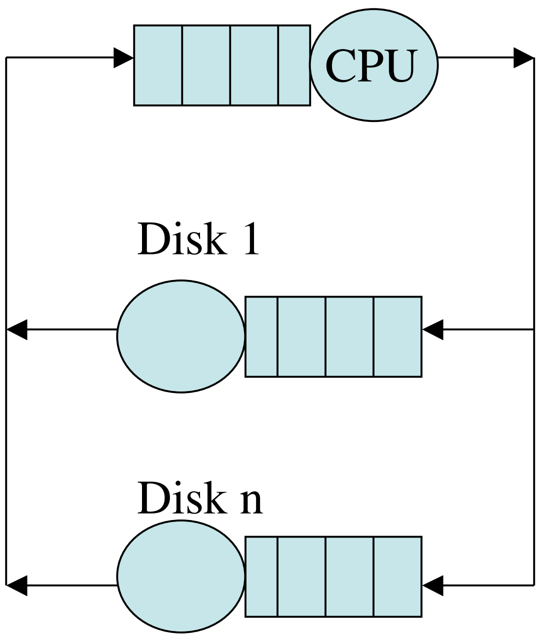
Advantages of Virtualization (cont'd)

- Adaptability to Workload Variations:
  - Changes in workload intensity levels can be taken care by dynamically shifting resources and priority allocations among VMs using autonomic computing techniques.
- Load Balancing
  - The state of a VM is completely encapsulated in the VMM: easy to migrate VMs to other platforms to improve performance.
- Legacy applications
  - Legacy applications can continue to run on old OSs that run as a guest operating systems on VMs.

Performance Modeling of Virtualized Environments

- Use Queuing Network (QN) models:
  - Customer classes
  - Parameters: service demands and workload intensities
  - Service Demand Law: $D_{i,r} = U_{i,r} / X_{0,r}$

Queuing Network Models



Input parameters per class:

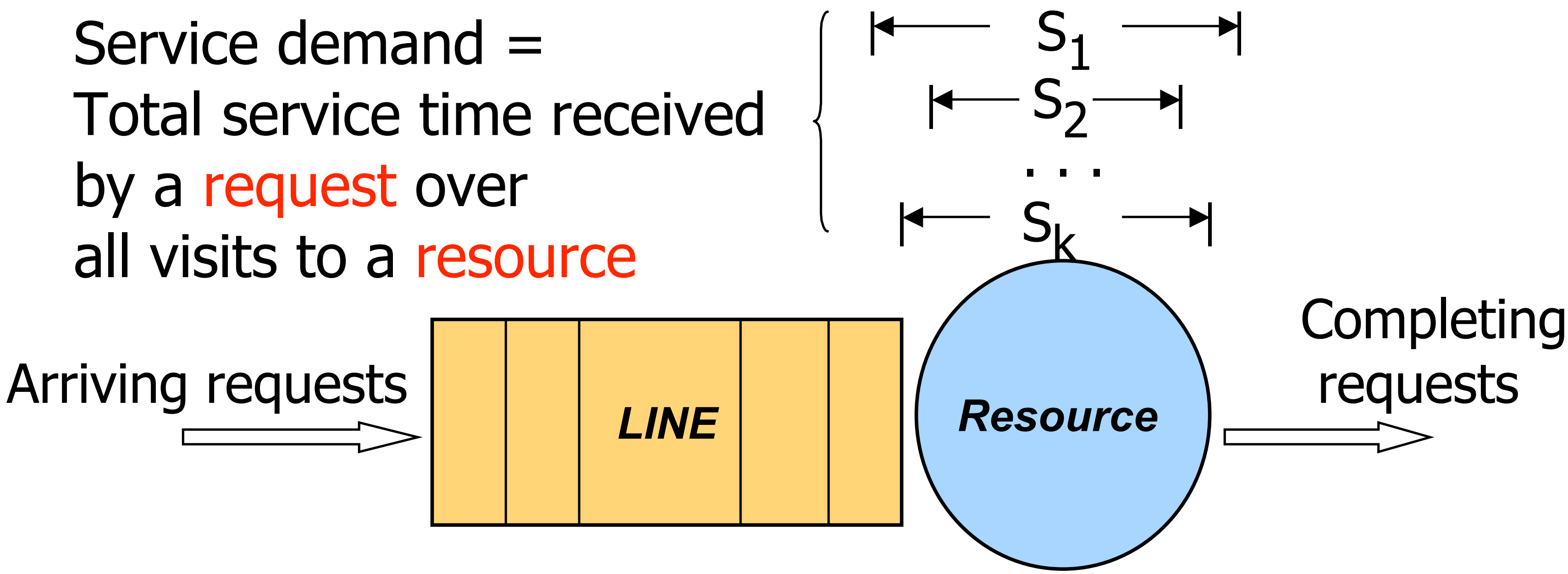
- workload intensity (e.g., arrival rates, concurrency levels)
- service demands per device (i.,e, total time spent by a transaction receiving service from that device)

Solution techniques:

- "Performance by Design," Menasce, Almeida, and Dowdy, Prentice Hall, 2004.

Service Demand (D)

Service demand =
Total service time received
by a request over
all visits to a resource



S_k: Service time received by the request during visit k
D: Service demand = S1 + S2 + …+ Sk

Computing Service Demands using the Service Demand Law

- The service demand D_i is given by:

$$D_i = U_i / X_o$$

where U_i is the utilization of resource i and X_o the system throughput.

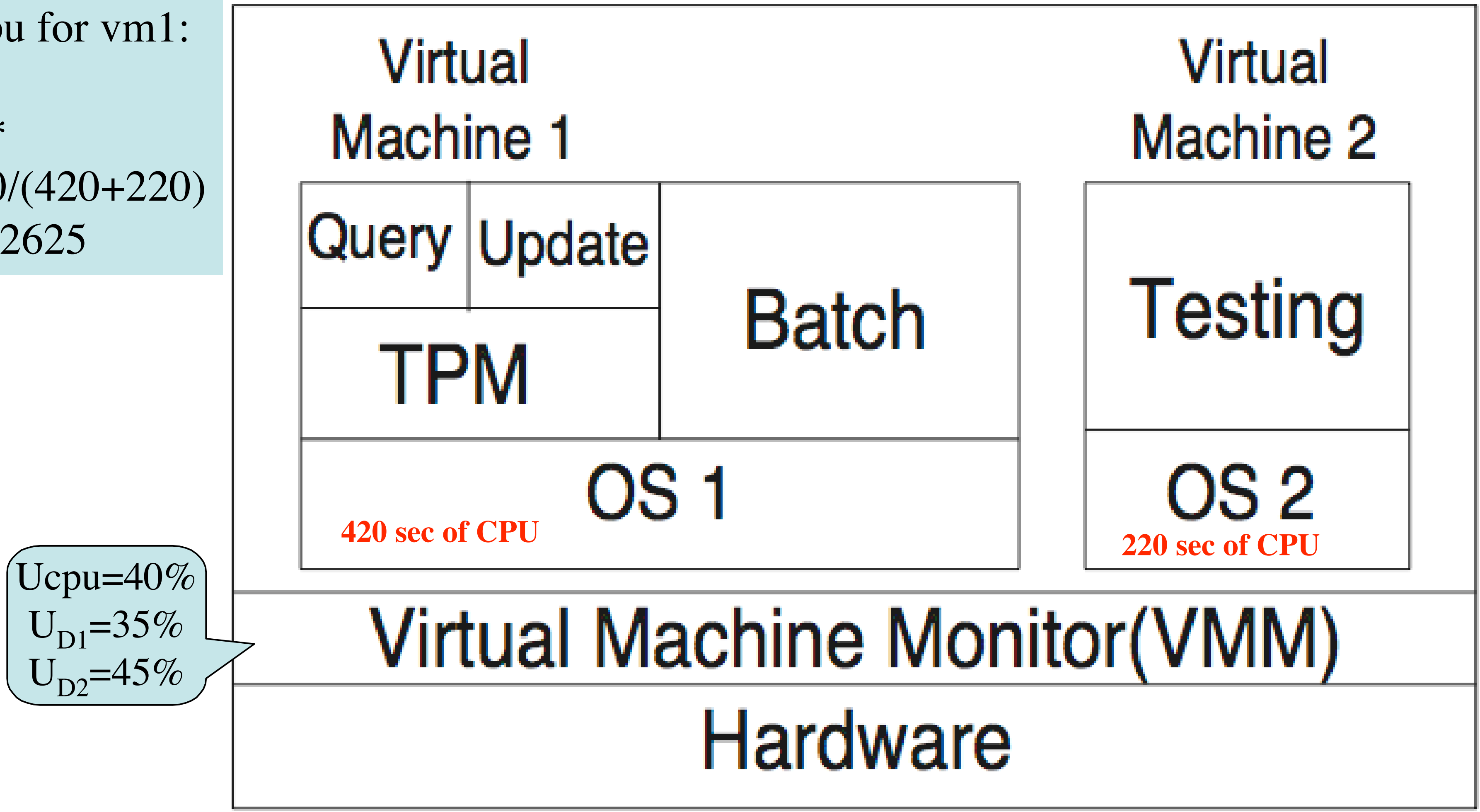
Performance Modeling of Virtualized Environments

- Use Queuing Network (QN) models:
  - Customer classes
  - Parameters: service demands and workload intensities
  - Service Demand Law: $D_{i,r} = U_{i,r} / X_{0,r}$
- Issue: How to combine measurements obtained from measurements tools at different software layers of a virtualized environment?

Parameter Computation

Ucpu for vm1:

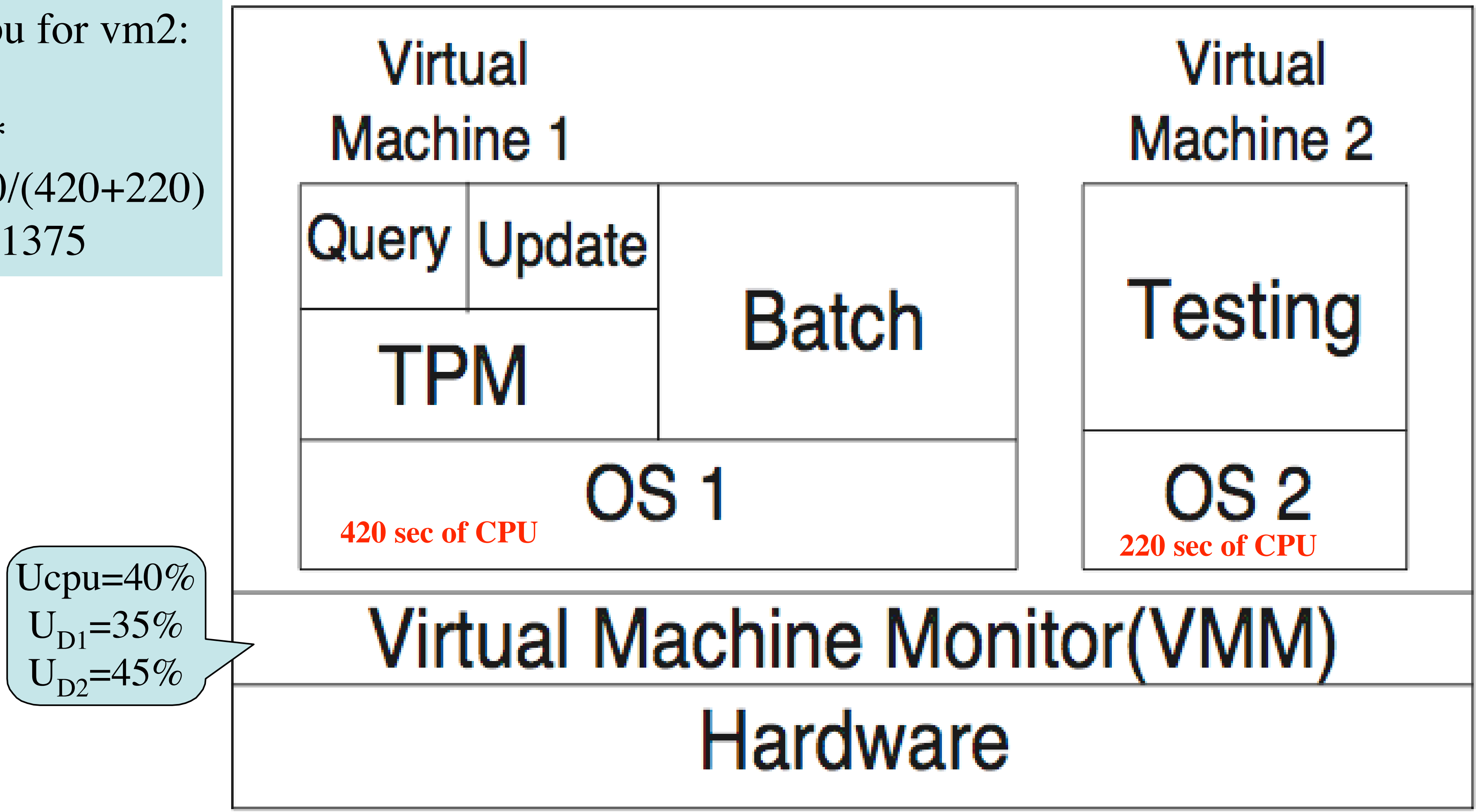
0.4* 420/(420+220) = 0.2625



Parameter Computation

Ucpu for vm2:

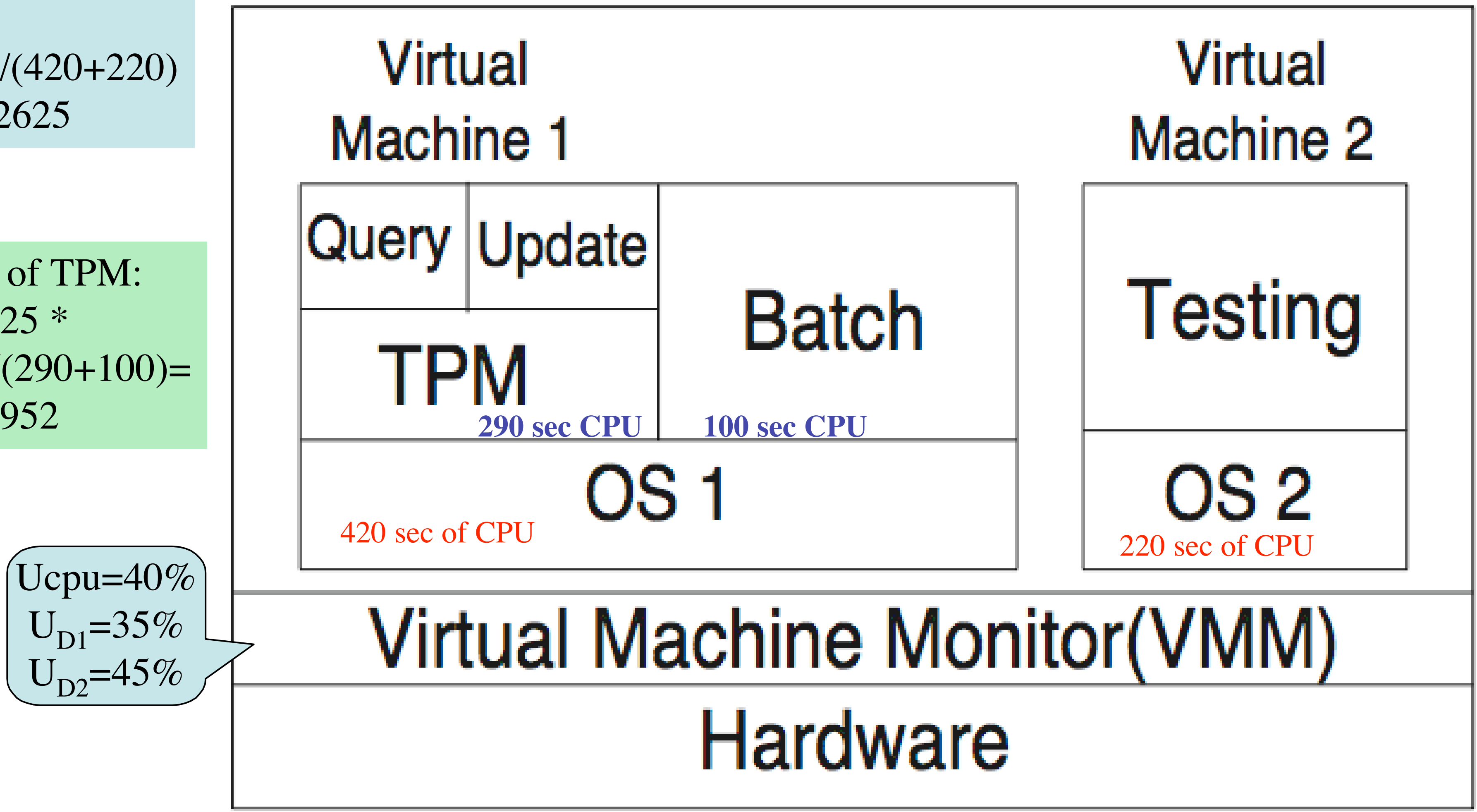
0.4* 220/(420+220) = 0.1375



Parameter Computation

U_{cpu} for vm1:
0.4*
420/(420+220)
= 0.2625

U_{cpu} of TPM:
0.2625 *
290/(290+100)=
0.1952



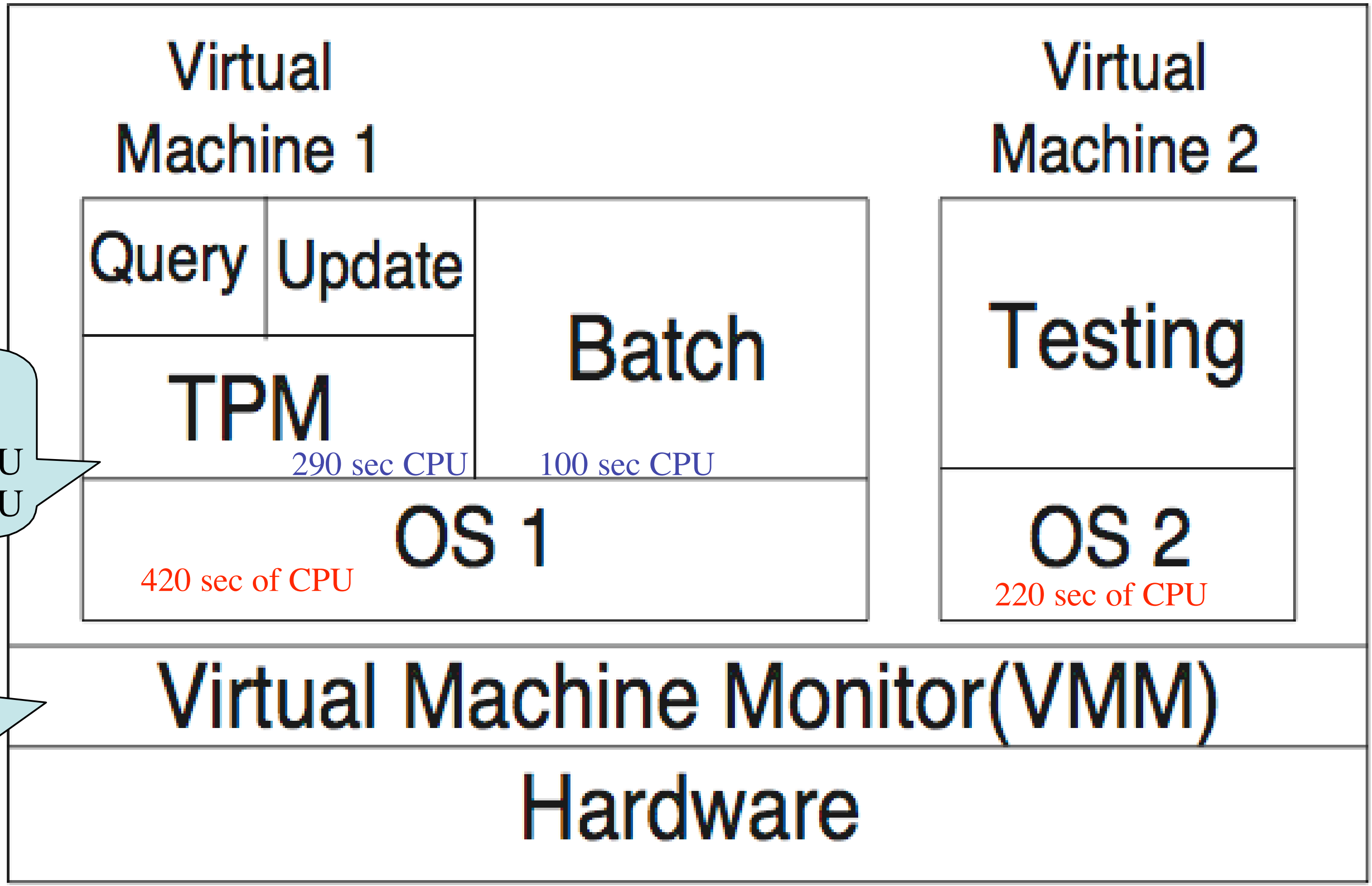
Parameter Computation

U_{cpu} of TPM:
0.2625 * 290/(290+100)= 0.1952

U_{cpu} of Query:
0.1952 * 110/(110+150)= 0.0826

5400 Q
1200 U
Q: 110 sec CPU
U: 150 sec CPU

Ucpu=40%
U_{D1}=35%
U_{D2}=45%



GEORGE MASON UNIVERSITY

Parameter Computation

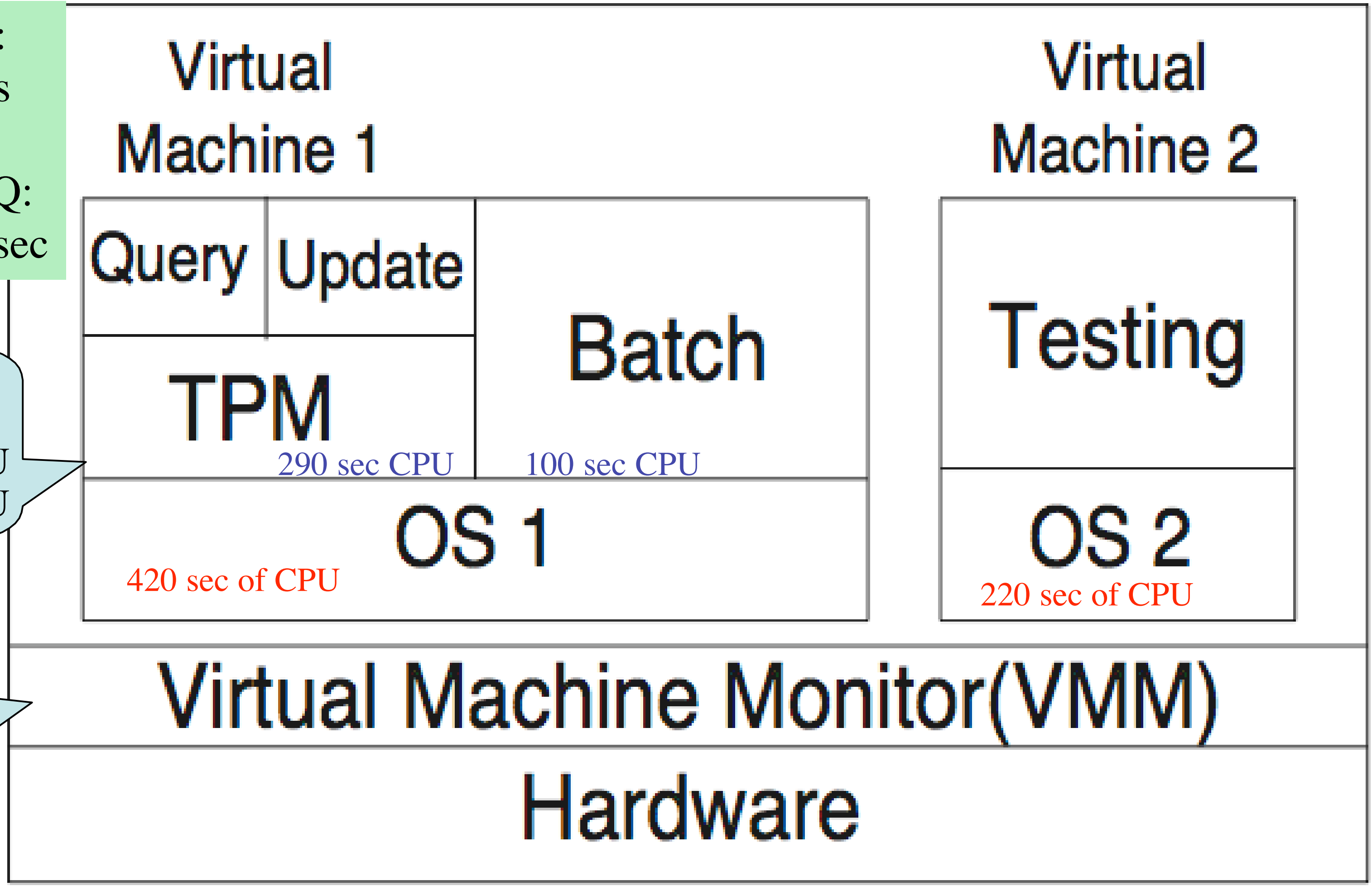
U_{cpu} of Query:
0.1952 * 110/(110+150)= 0.0826

Query throughput:
5400 / 1800 = 3tps

CPU demand for Q:
0.0826/3=0.0275 sec

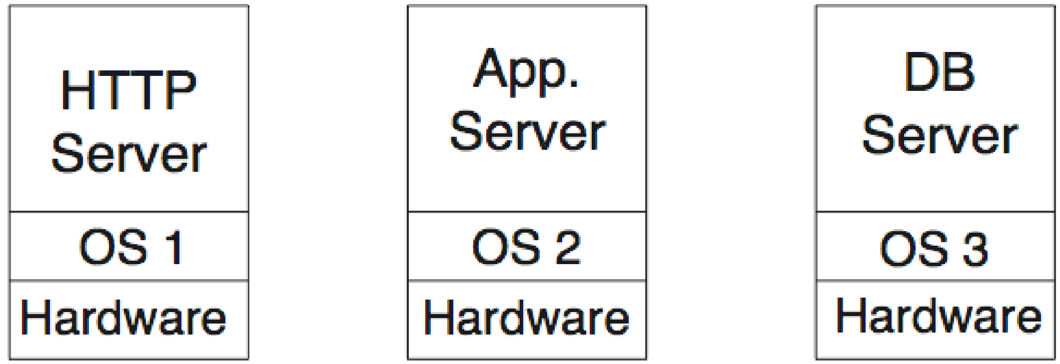
5400 Q
1200 U
Q: 110 sec CPU
U: 150 sec CPU

Ucpu=40%
U_{D1}=35%
U_{D2}=45%



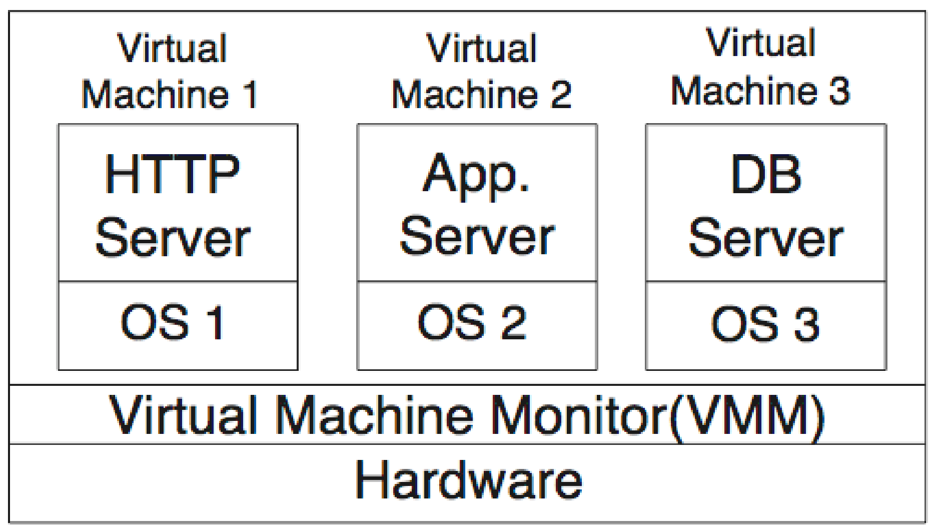
Server Consolidation

3-server scenario



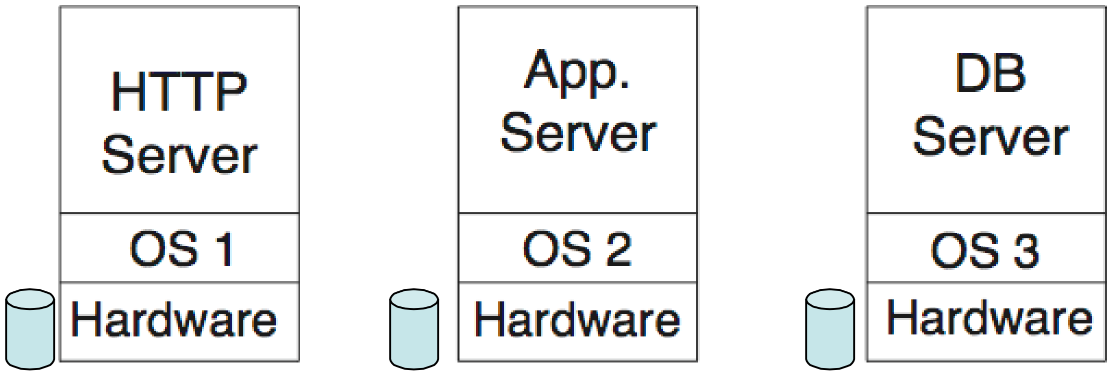
(a)

consolidated server scenario



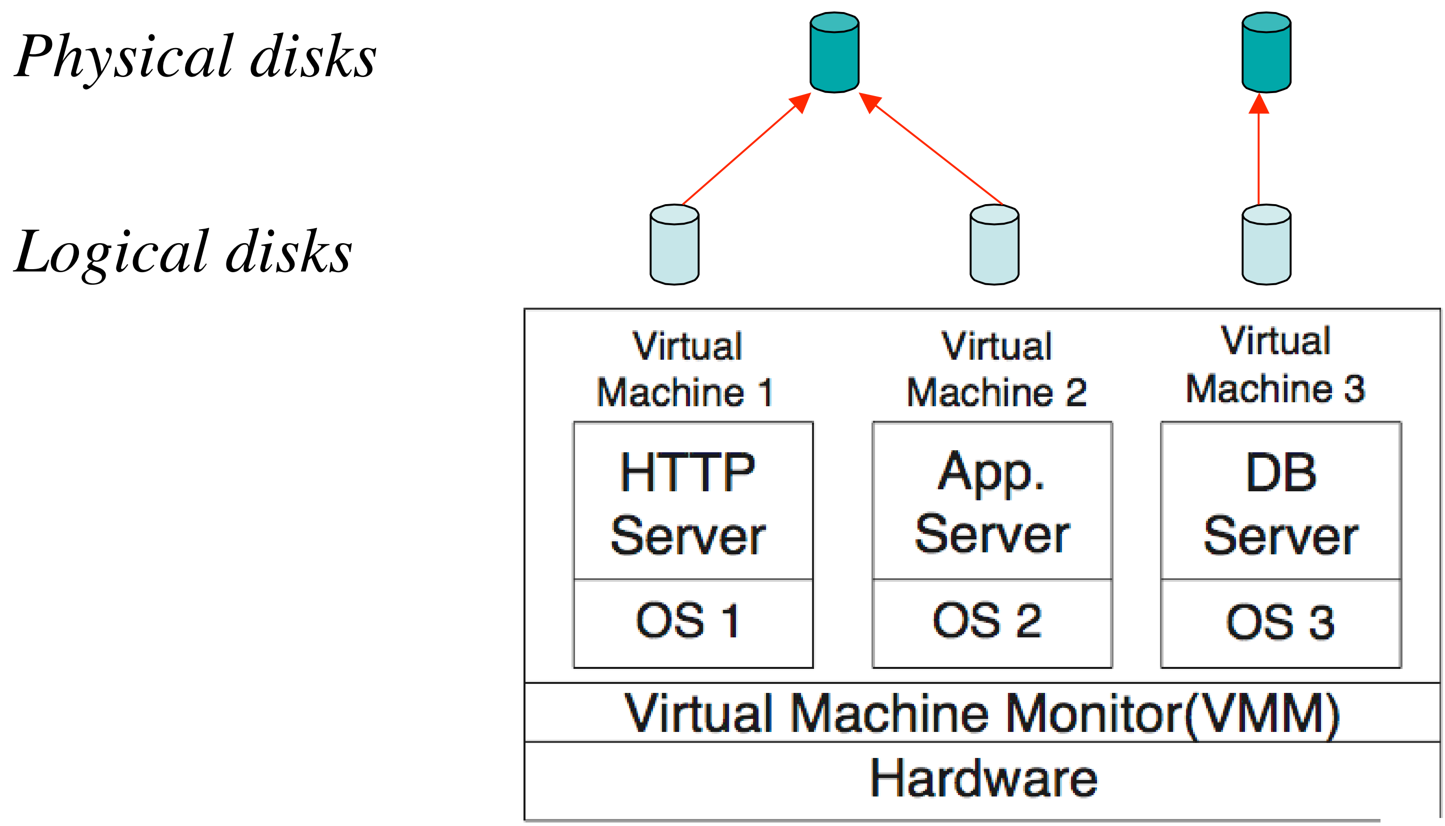
(b)

Server Consolidation



(a)

Server Consolidation



(b)

Server Consolidation Computing Service Demands

Number of original servers

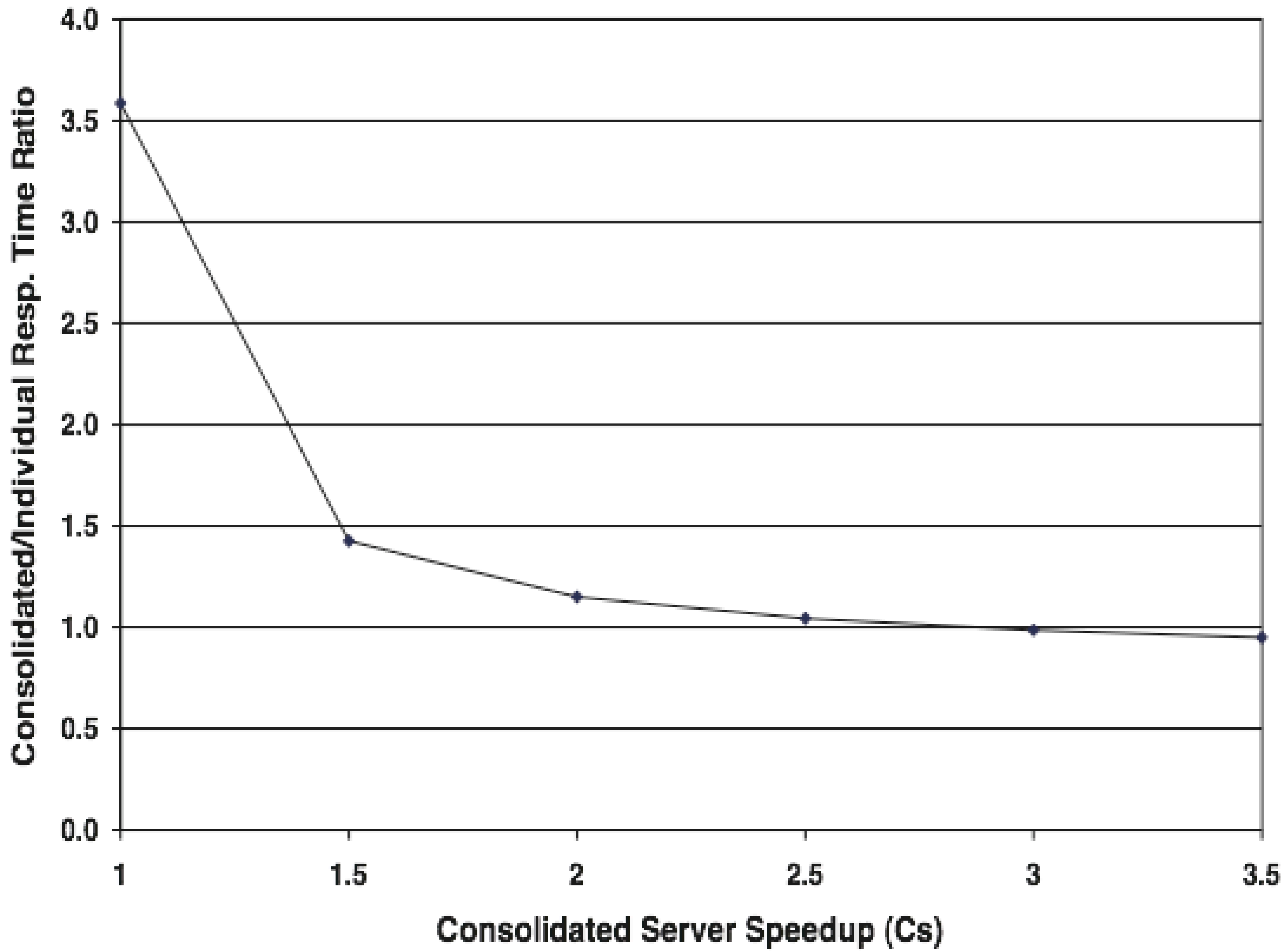
Slowdown factor

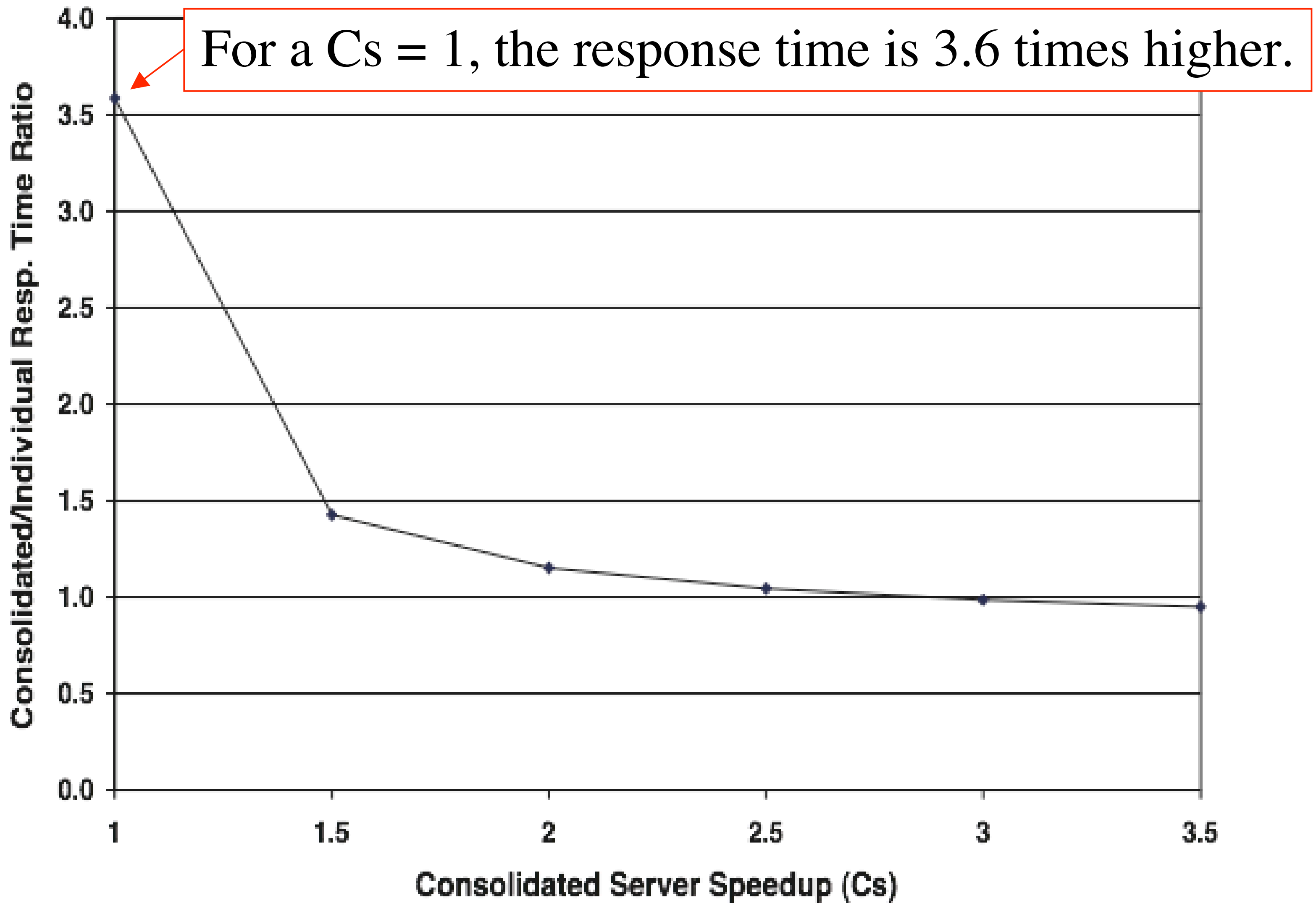
$$D_{CPU,r}^{cons} = \sum_{s=1}^{S} D_{CPU_s,r} \times S_v / C_s$$

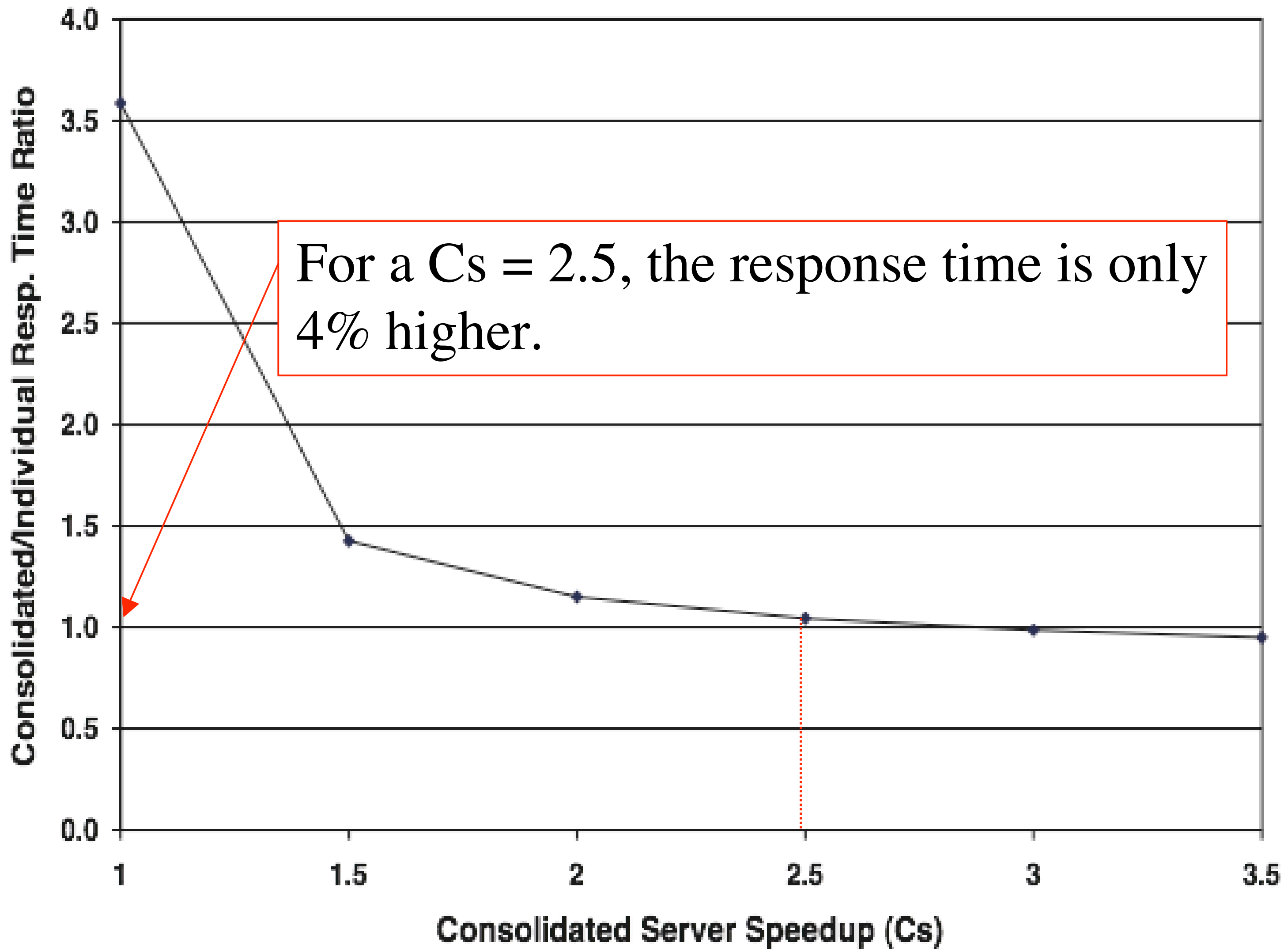
Service demand at the CPU of the consolidated server for class r applications.

Service demand at the CPU of original server s for class r applications.

Speedup of consolidated server relative to individual server s


4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Consolidated/Individual Resp. Time Ratio
1
1.5
2
2.5
3
3.5
Consolidated Server Speedup (Cs)


For a Cs = 1, the response time is 3.6 times higher.
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Consolidated/Individual Resp. Time Ratio
1
1.5
2
2.5
3
3.5
Consolidated Server Speedup (Cs)


4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Consolidated/Individual Resp. Time Ratio
1
1.5
2
2.5
3
3.5
Consolidated Server Speedup (Cs)
For a Cs = 2.5, the response time is only 4% higher.

Concluding Remarks

- Many vendors are adopting virtualization solutions:
  - Sun's zones in Solaris 10
  - IBM's Logical Partitioning (LPAR)
  - Intel's Virtualization Technology: hardware support for virtualization.
    - Additional instructions that can be used by the VMM to create and support VMs.
    - The VMM runs one level below ring 0 and the VMs execute at ring 0
  - Xen (Open Source)
  - EMC's VMWare
  - Microsoft Virtual Server 2005 R2

Concluding Remarks (cont'd)

- Two main directions in virtualization:
  - Full virtualization: the VMM provides an identical abstraction of the underlying hardware. However, not all architectures are virtualizable (e.g., x86).
    - No changes required to the guest OSs
    - Example: VMware
  - Paravirtualization: the VMM provides an "almost" identical abstraction of the underlying hardware. The abstraction implements some new instructions to make the machine virtualizable.
    - Guest OSs need to be modified.
    - Better performance than full virtualization.
    - Examples: Xen and Denali

Questions?

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