Measuring and Modeling the Performance of the Xen VMM

Jie Lu, Lev Makhlis, Jianjiun Chen

Virtualization background

- **Server virtualization**

  Server virtualization is the masking of server resources, including the number and identity of individual physical servers, processors, and operating systems, from server users.

  - Provides an alternative for server consolidation by creating a set of isolated logical resources that share underlying physical resources
  - Guest operating systems run on Virtual Machine, which is simulated in a contained software environment by the host system.

- **Virtual servers**
  - Enable dynamic tuning
  - Make better use of resources
  - Provide high availability
  - Reduce overall cost
  - Increase flexibility

Virtualization background

- **Key terminology**
  - Virtual machine

    A representation of real machine using software that provides an operating environment which can run or host a guest operating system.
  - Guest operating system

    An operating system running in a virtual machine environment that would otherwise run directly on a separate physical system.
  - Hypervisor

    A thin layer of software that generally provides virtual partitioning capabilities which runs directly on hardware, but underneath higher-level virtualization services.
  - Virtual machine monitor (VMM)
    - Software that runs in a layer between a hypervisor or host operating system and one or more virtual machines.
    - It provides the virtual machine abstraction to the guest operating systems.

Virtualization types

- **Full virtualization (native virtualization, hardware-level virtualization)**
  - The virtualization layer exports the virtual machine abstraction identical to a physical machine.
- **Para-virtualization**
  - A technique that presents a software interface to virtual machines that is similar but not identical to that of the underlying hardware.
  - It requires operating systems to be explicitly modified and ported to run.
- **Emulation (hosted virtualization)**
  - An approach where partitioning and virtualization services run on top of a standard operating system (the host).
  - The virtualization software relies on the host operating system to provide the services to talk directly to the underlying hardware
  - The simulated virtual hardware is independent of the nature of the host computer

Virtualization types

- **OS-level virtualization**
  - The virtualization layer sits between the operating system and the application programs.
  - The virtual machine runs applications that are written for the particular operating system being virtualized.
  - The guest OS uses the same kernel as the host system.

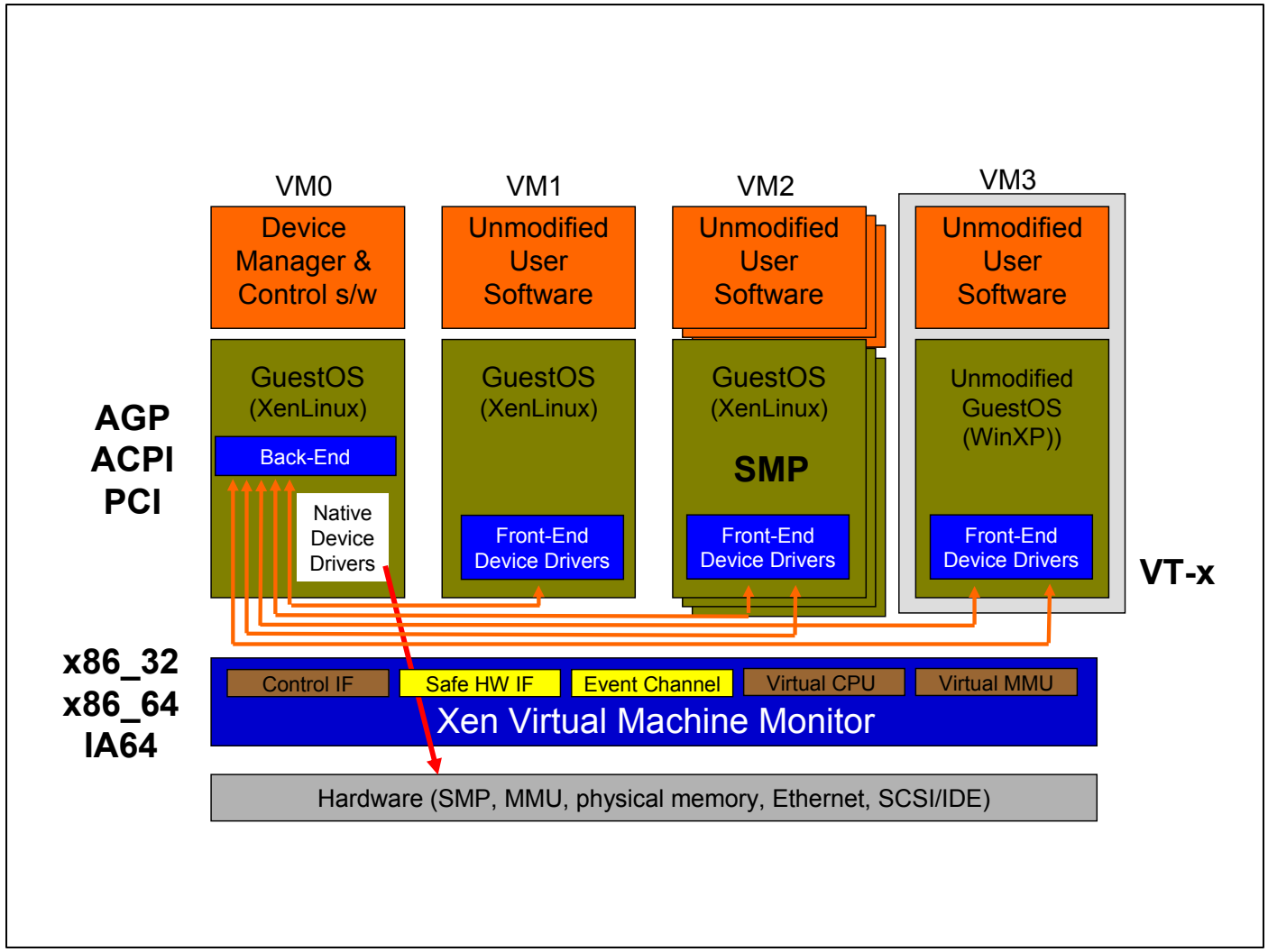
Commercial virtualization products

Product	Type	Hypervisor	Hardware platform	Guest OS	Modify guest OS
IBM micro-partition (SPLPAR)	System	Firmware	Power 5	AIX, Linux, i5/OS and Windows	Yes
VMware ESX Server	Full	Software	x86	Windows and Linux	No
Xen	Para, Full	Software	x86	Linux and Windows	Yes (for Para)
MS Virtual Server	Emulation	No	x86	Windows and Linux	No
Solaris Container	OS	No	SPARC and x86	Solaris	N/A

Xen

- **Open source virtual machine monitor (VMM) for x86**
  - Developed by the Computer Laboratory at University of Cambridge
- **Para-virtualization for traditional x86**
  - Requires the modifications on the guest OS kernel (XenLinux)
  - No modification on guest applications
  - Guest OS run at a lower privilege level
  - Privileged instructions are validated and executed within Xen
- **Support full virtualization for Intel VT or AMD Pacifica**
  - No modification is required on the guest OS kernel

Xen 3.0 architecture



Xen data visibility

- **Traditional performance measures are obtained from OS**
  - The resource usage metrics are virtual based
    - Full virtualization
    - Para-virtualization (before 3.0.2)
  - XenLinux implements “steal” (involuntary wait) time accounting (since 3.0.2)
    - Tracks real CPU time
  - Effective modeling requires instrumentation on every guest OS.
    - Each OS only has a partial view of the physical system.
    - Incur considerable overhead
- **Hypervisor keeps track of the actual resource usage by each individual guest**
  - Most reliable statistics
  - Provide complete view

Xen data visibility

- **User level**
  - xm
- **Low level**
  - Xen daemon (Xend)
    - SXP over a UNIX domain socket
      `/var/lib/xend/xend-socket`
    - SXP over TCP (port 8000)
    - XML-RPC over a UNIX domain socket
      `/var/run/xend/xmlrpc.sock`
    - XML-RPC over TCP (port 8005)
  - XenStore
    - Through procfs, using the file `/proc/xen/xenbus`
    - Through xenstored, using UNIX domain socket
      - `/var/run/xenstored/socket`
      - `/var/run/xenstored/socket_ro`

Xen data visibility

- **Low level**
  - Hypervisor calls
    - DOM0_PHYSINFO
      - CPU topology
      - CPU speed in KHz
      - Memory
    - DOM0_GETDOMAININFOLIST
      - Numeric ID
      - Status (dying | shut down | paused | blocked | running)
      - Memory pages allocated to the domain and its memory limit
      - Number of virtual CPUs (VCPUs)
      - Aggregate CPU usage (in nanoseconds)
    - DOM0_GETVCPUINFO
      - CPU usage for each individual CPU

Xen data visibility

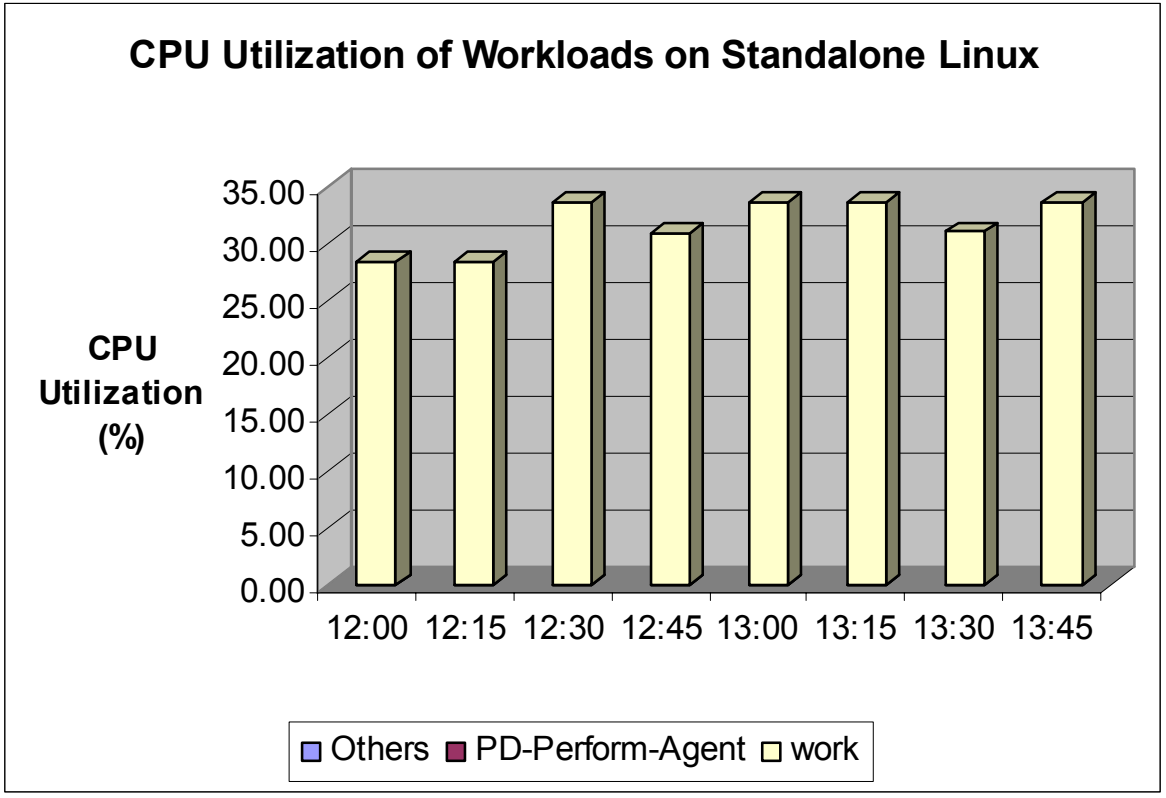
- **Device data**
  - Only driver domain can assess to the PCI hardware
    - Report physical I/O statistics
  - Other domains see virtual devices
    - Report virtual I/O statistics

Modeling at guest level

- Collect regular performance data from each guest OS
  - System configuration
  - System statistics
  - Process statistics
- Measure actual response time of controlled workload
- Estimate response time using existing performance modeling tool

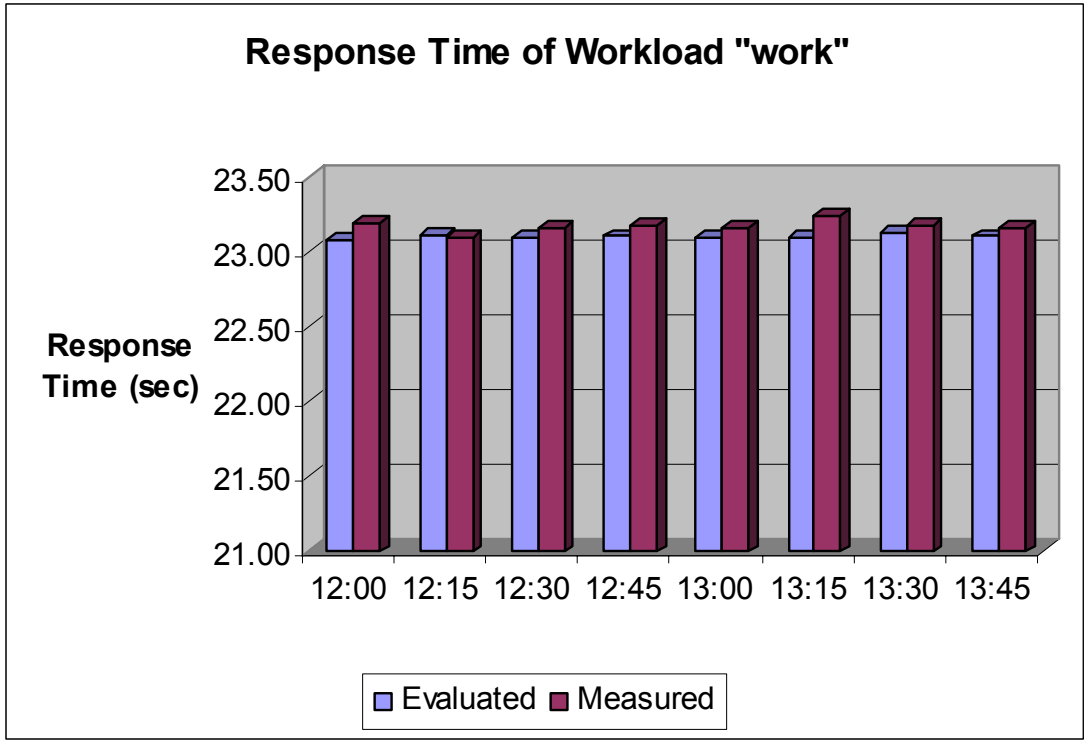
Modeling at guest level

- **Baseline**
  - Controlled workload running on a dedicated Linux system



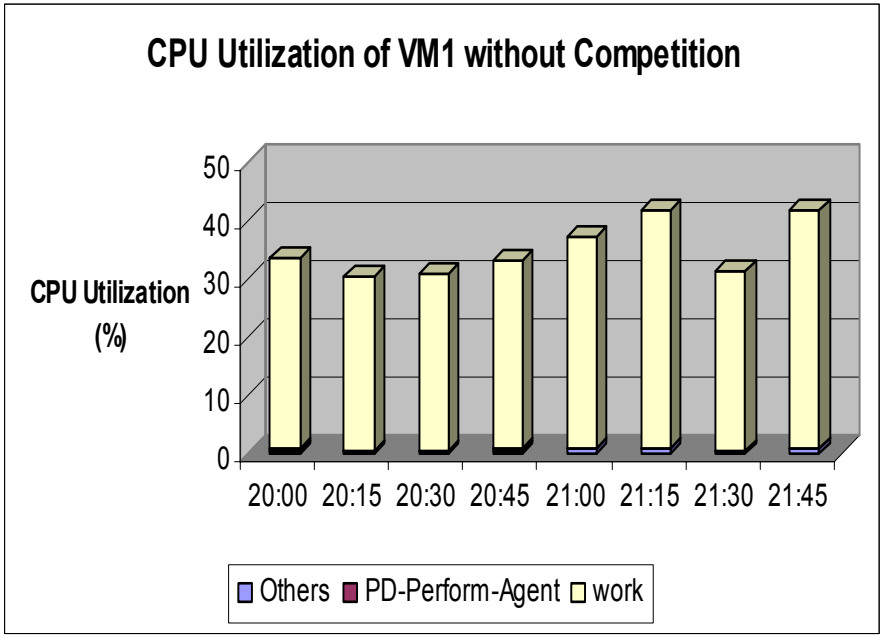
Modeling at guest level

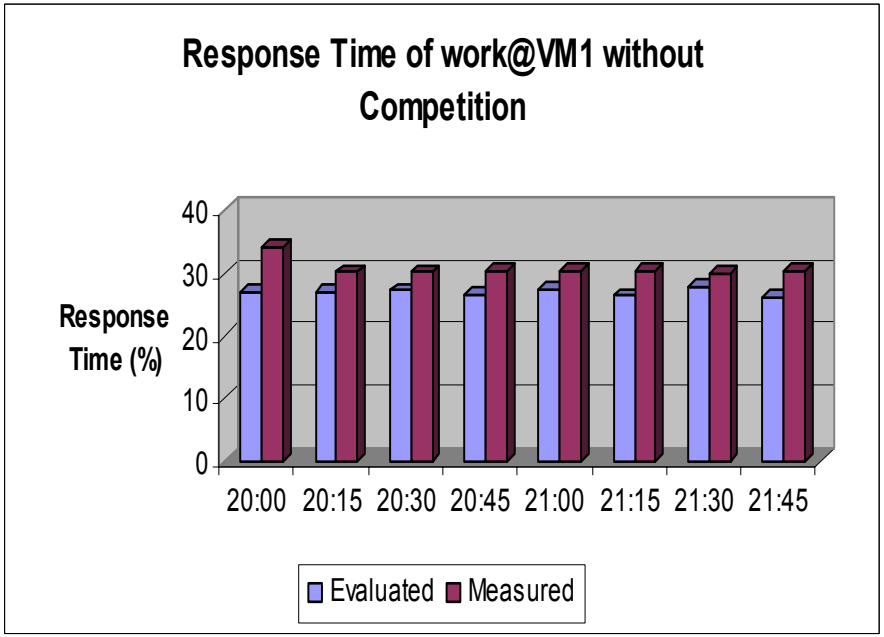
- **Baseline**
  - Estimated response time well matches the measured response time
  - Validated traditional performance modeling tool



Modeling at guest level

- **Case 1: Without competition**
  - One guest runs the same workload as baseline, and one guest is ilde.
  - CPU utilization is same as the standalone system
  - Estimated response time roughly matches the measured response time

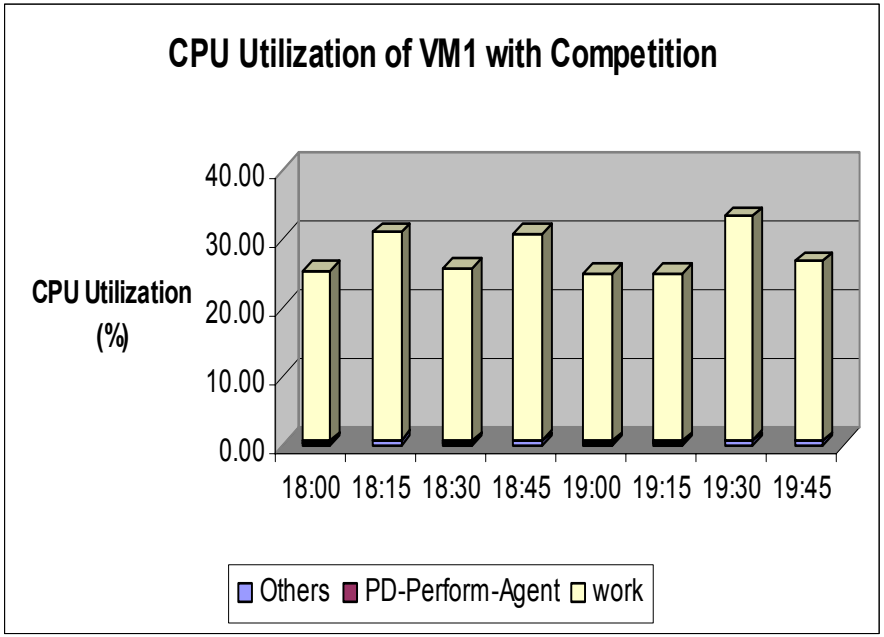


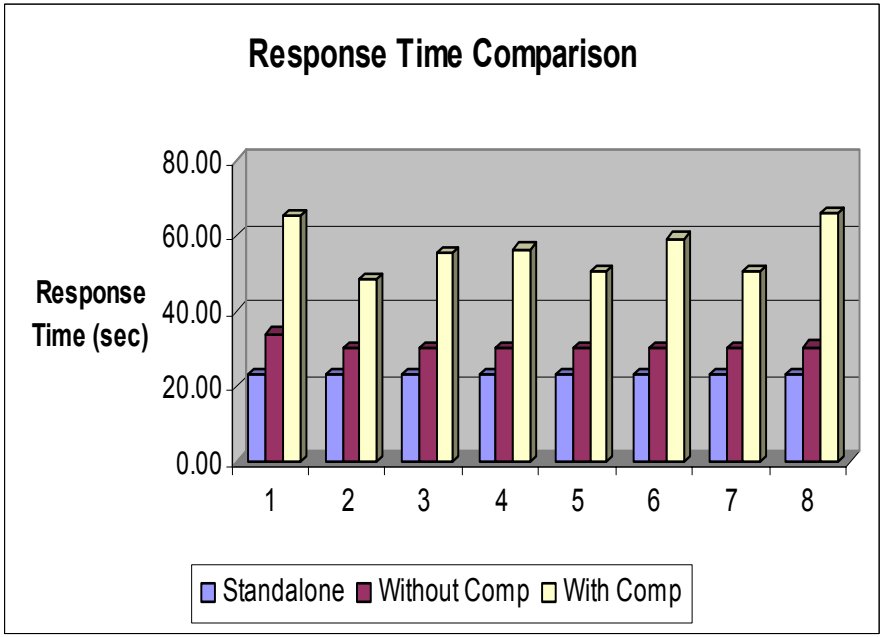


Modeling at guest level

› Case 2: With competition

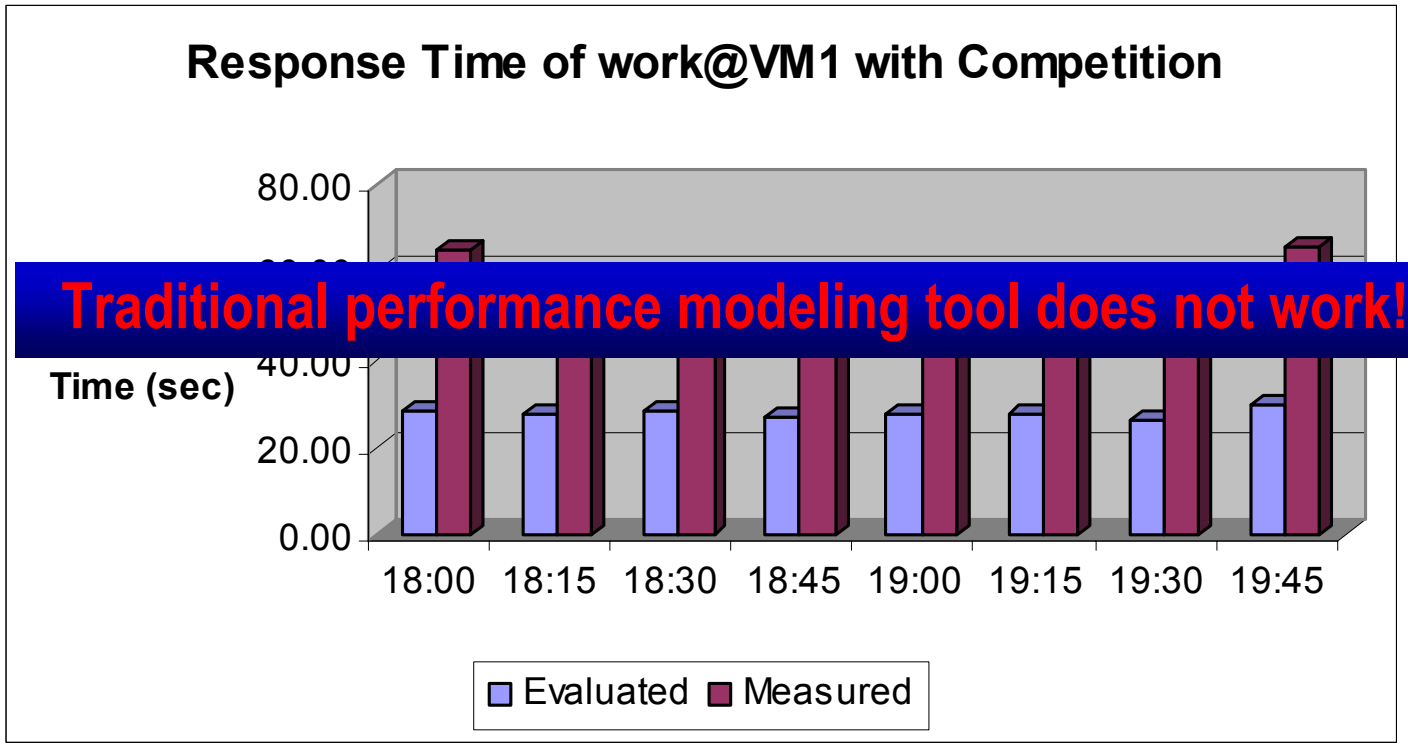
- Both guests run the same workload as baseline
- CPU utilization decreases
- Measured response time increases significantly





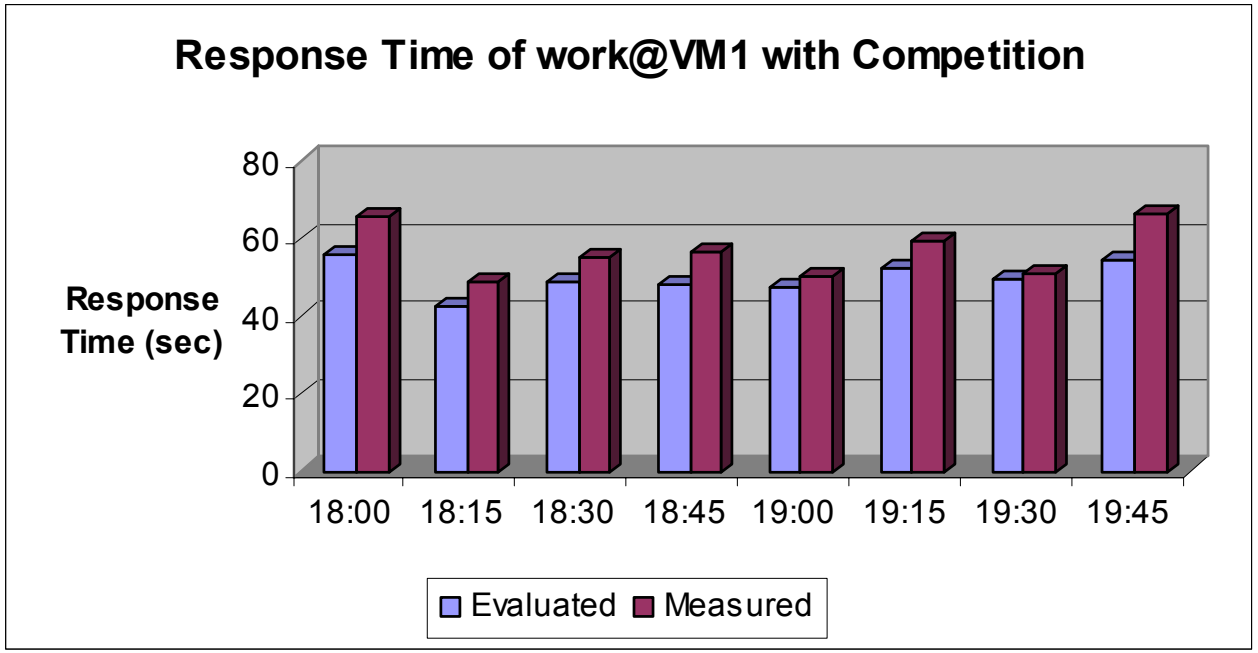
Modeling at guest level

- **Case 2: With competition**
  - Estimated response time is way off the measured response time



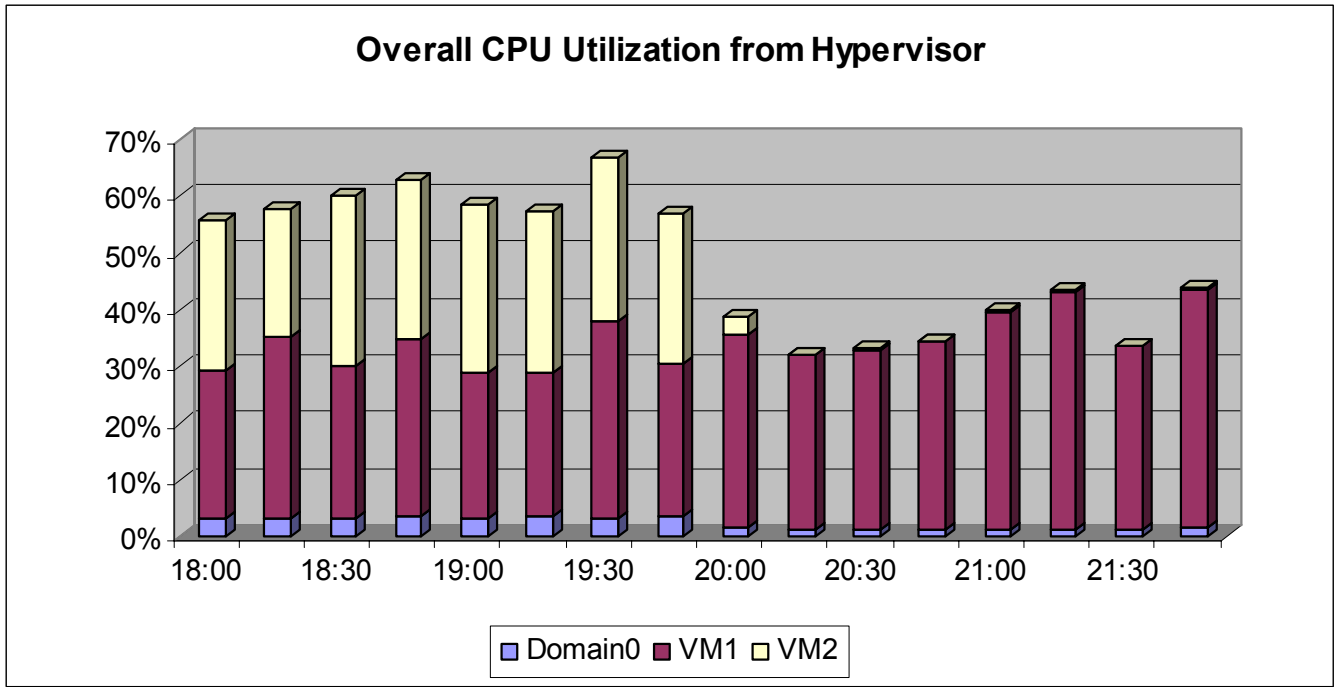
Modeling at guest level

- **Workaround**
  - Build performance model with traditional tool
  - Modify configuration of Domain0 to have it represent physical server
  - Consolidate all workloads on to the updated Domain0.
  - Produce reasonable results

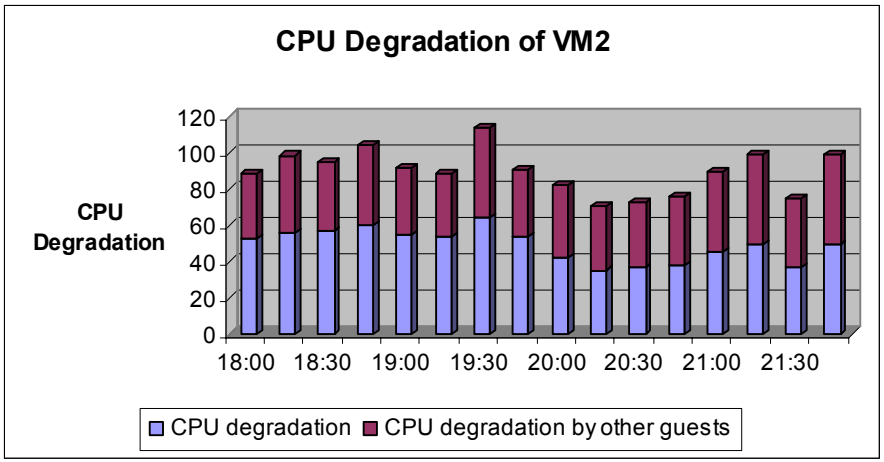


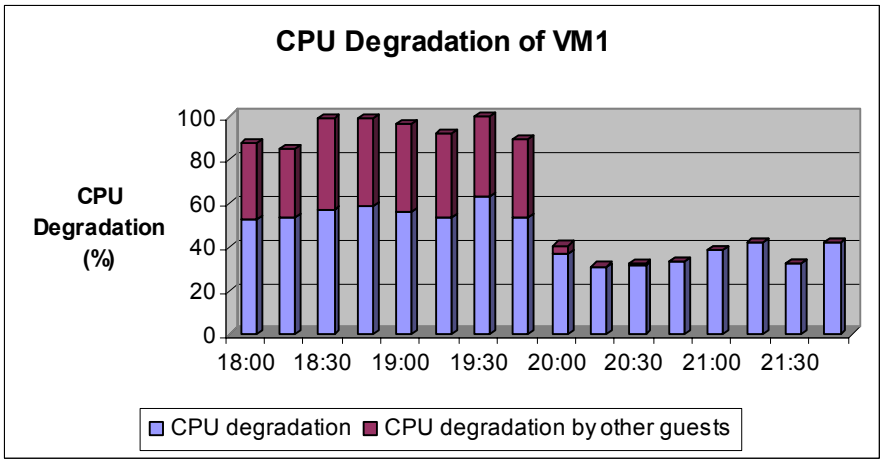
Modeling at hypervisor level

- Collect Xen specific metrics via hypervisor calls in Domain0
  - Configuration of each domain
  - Statistics of each domain
- Use existing performance tool for VMware ESX virtual server



Modeling at hypervisor level





Conclusion

- Xen provides an alternate virtualization solution
- Conventional performance modeling method does not work in a virtualized environment
- Performance can be measured at both hypervisor level and guest level
- In para-virtualized Xen, the data from hypervisor and from guest can be correlated to provide both top down and bottom up views of performance aspects
- Existing tool can be used with workaround

Thank You !