

Basic Virtualization Syllabus

Cover the techniques for virtualizing and managing the hardware components of a single computer system.

Academics and researchers: With your help, we can expand and enhance this syllabus. If you would like to collaborate on classroom lab or assignment courseware, or if you have your own previously developed courseware items to share, please [contact](/contact.jspspa) the govirtual team.

1. CPU Virtualization

Cover techniques for virtualizing a CPU, including classic trap-and-emulate and binary translation. The goal of CPU virtualization is to allow the instructions of a virtual machine to execute natively on the physical CPU whenever possible.

Lecture Slides: [CPU Virtualization](#)

Lecture Notes/Guide: This lecture covers the most widely-used techniques for virtualizing a CPU. It contrasts the user and system parts of a CPU's instruction set architecture (ISA), and discusses interpretation, trap-and-emulate, and binary translation methods of virtualizing the system ISA.

Reading List:

Rosenblum et al., "[Complete Computer System Simulation: The SimOS Approach](#)," IEEE Parallel and Distributed Technologies, 1995.

Describes several strategies for simulating or virtualizing processors.

Popek and Goldberg, "[Formal Requirements for Virtualizable Third Generation Architectures](#)," CACM 1974.

Describes the properties that an instruction set must have to be classically virtualizable through trap-and-emulate.

Smith and Nair, "Virtual Machines: Versatile Platforms for Systems and Processes", Morgan Kaufmann Publishers, 2005, Chapter 2.

Describes different ways to emulate instructions.

Barham et al., "[Xen and the Art of Virtualization](#)," SOSP 2003.

Describes a virtual machine monitor that uses paravirtualization.

Keith Adams, Ole Agesen, "[A Comparison of Software and Hardware Techniques for x86 Virtualization](#)", ASPLOS 2006.

Evaluates the two main ways to virtualize the x86 architecture.

Rich Uhlig, Gil Neiger, Dion Rodgers, Amy L. Santoni, Fernando C. M. Martins, Andrew V. Anderson, Steven M. Bennett, Alain Kagi, Felix H. Leung, Larry Smith, "[Intel Virtualization Technology](#)", IEEE Computer, May 2005.

Describes Intel's extensions to the x86 architecture to support virtualization.

Class Assignments:

[Homework 1: E6998 - Virtual Machines](#)

Extend the emulation of some privileged instruction in KVM. For example, modify the emulation of the HLT instruction to maintain the cumulative idle time of that virtual machine.

2. Memory Virtualization

Cover techniques for virtualizing the memory management unit of a modern CPU. The goal of memory virtualization is to allow the memory of a virtual machine to map directly to physical memory using the natively available page-translation hardware.

Lecture Slides: [Memory Virtualization](#)

Lecture Notes/Guide: This lecture introduces the background, implementation, and challenges of memory virtualization. It begins with traditional MMU-based address translation and adds an extra level of indirection (due to virtualization). The basic implementation is summarized and techniques for efficiently keep page tables consistent are discussed. Other topics include: memory tracing (for a variety of purposes), hiding the monitor (so that the guest cannot detect or interfere with it), and nested paging (to reduce the need for VMM involvement in page table management).

Reading List:

Waldspurger, C.A. "[Memory Resource Management in VMware ESX Server](#)," In Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI'02). This paper introduces and exploits the unique opportunities presented by virtualization, allowing for multiple virtual machine workloads to over commit memory (i.e., the sum of all guest physical memory is greater than the total amount of real machine memory) at little or no cost. A technique called ballooning is used to reclaim guest memory that is under utilized, and content-based page sharing is used to share common data between different virtual machines. Both these mechanisms and the policies that govern them are presented and experimentally evaluated.

Ravi Bhargava, Ben Serebrin, Francesco Spadini, Srilatha Manne, "[Accelerating Two-Dimensional Page Walks for Virtualized Systems](#)", ASPLOS 2008. Nested paging is a hardware technique to reduce the involvement of VMM in guest page table management, but it comes at the cost of increasing the amount of work that must be done in hardware to translate address (effectively requiring a two-dimensional page walk). This paper describes

this problem, presents micro architectural techniques (page walk cache and large pages) for reducing the overhead induced by this problem, and evaluates their impact on performance.

Class Assignments:

Using an open source virtualization system, e.g. KVM or VirtualBox, (a) Optimize shadow MMU for large working sets. (b) Implement a shared MMU for SMP VMs. (c) Restructure MMU for highly efficient tracing (e.g., in support of VM live migration). (d) Implement tracing. (f) Implement content-based page sharing between multiple VMs (or between guest/host).

3. I/O Virtualization

Cover techniques for presenting I/O devices and virtual disks to a guest OS in a virtual machine. Unlike CPU and Memory virtualization, which try to map directly to physical hardware whenever possible, I/O devices are typically emulated or paravirtualized to present a uniform device abstraction above the wide variety of I/O hardware found in today's computer systems. Discuss useful I/O features that can be provided by the virtualization layer.

Lecture Slides: [Device Virtualization](#)

Lecture Notes/Guide: This lecture focuses on three methods of presenting virtual devices to a virtual machine, (direct-access, emulated, and paravirtualized) and their tradeoffs. It also discusses how virtual disks can provide useful functionality, such as copy-on-write, that is not available with physical disks.

Additional Slides:

[I/O Architectures for Virtualization](#)

[I/O Virtualization for Dummies](#)

Reading List:

Sugerman et al., "[Virtualizing I/O Devices on VMware Workstation's Hosted Virtual Machine Monitor](#)," In Proceedings of 2001 USENIX Annual Technical Conference.

Describes a hosted or split I/O virtualization model where physical I/O is performed by a host OS that is separate from the guest OS. Measures the I/O virtualization overheads and presents several techniques to reduce those overheads.

Apparao, P., Makineni, S., and Newell, D. "[Characterization of Network Processing Overheads in Xen](#)". In Proceedings of the 2nd International Workshop on Virtualization Technology in Distributed Computing. (VTDC)

Measures and analyzes network I/O virtualization overheads in the Xen hypervisor.

Menon, A., Cox, A. L., and Zwaenepoel, W. "[Optimizing Network Virtualization in Xen](#)". In Proceedings of 2006 USENIX Annual Technical Conference.

Presents several techniques for reducing network I/O virtualization overhead in the Xen hypervisor.

Liu, J., Huang, W., Abali, B., and Panda, D. K. "[High Performance VMM-Bypass I/O in Virtual Machines](#)," In Proceedings of 2006 USENIX Annual Technical Conference.

Shows how user-level communication support in high-performance computing interfaces, such as Infiniband, allows a guest OS to directly interface with the device for time-critical operations, thus reducing the overhead of I/O device virtualization.

Lagar-Cavilla, H. A., Tolia, N., Satyanarayanan, M., and de Lara, E. "[VMM-Independent Graphics Acceleration](#)". In Proceedings of the 3rd international Conference on Virtual Execution Environments. (VEE '07).

Describes how to provide accelerated virtual machine graphics by virtualizing and intercepting graphics operations at the graphics API layer instead of emulating proprietary graphics hardware.

Class Assignments:

Using an open source virtualization package such as QEMU, provide an alternative back-end for an existing virtual device, for example, use named pipes as a back-end for a virtual serial port. For a more challenging project, implement a new virtual device.

4. Resource Management

Cover techniques for allocating physical resources among virtual machines. The goal of resource management is to maximize hardware utilization while maintaining performance isolation and service-level guarantees. Cover CPU and memory management within a single physical machine.

Lecture Slides: [Resource Management for Virtualized Systems](#)

Lecture Notes/Guide: This lecture introduces the basic concepts and units of resource management in a virtualized system. It describes hierarchical proportional-share scheduling of CPU resources and the allocation, de-duplication, and reclamation of memory resources. It also addresses specific challenges of multiprocessor VM scheduling, scheduling on hyper threaded CPUs, and memory management of NUMA systems.

Reading List:

Waldspurger, C. A. and Weihl, W. E. "[Lottery Scheduling: Flexible Proportional-Share Resource Management](#)". In Proceedings of the First Symposium on Operating System Design and Implementation (OSDI '94).

This paper describes lottery scheduling, a flexible and modular implementation of proportional-share scheduling.

Cherkasova, L., Gupta, D., and Vahdat, A. "[Comparison of the Three CPU Schedulers in Xen](#)". SIGMETRICS Perform. Eval. Rev. 35, 2 (Sep. 2007), 42-51.

This paper describes the concept of proportional-share schedulers and compares the performance of three available CPU schedulers in the Xen hypervisor.

Waldspurger, C. A. "[Memory Resource Management in VMware ESX Server](#)," In Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI'02).

This paper introduces and exploits the unique opportunities presented by virtualization, allowing for multiple virtual machine workloads to over commit memory (i.e., the sum of all guest physical memory is greater than the total amount of real machine memory) at little or no cost. A technique called ballooning is used to reclaim guest memory that is under utilized, and content-based page sharing is used to share common data between different virtual machines. Both these mechanisms and the policies that govern them are presented and experimentally evaluated.

Ongaro, D., Cox, A. L., and Rixner, S. "[Scheduling I/O in Virtual Machine Monitors](#)". In Proceedings of the Fourth ACM SIGPLAN/SIGOPS international Conference on Virtual Execution Environments (VEE'08).

This paper investigates the impact of virtual machine scheduling algorithms on I/O performance in the Xen hypervisor.

Class Assignments:

Experiment with various synthetic VM workload distributions and determine how a virtual machine system reacts to different CPU and memory loads.

Click below to download the entire virtualization course:

[Complete Virtualization Course Download](#)|| |Introductory||