An Overview of Virtualization Techniques

Renato Figueiredo
Advanced Computing and Information Systems (ACIS)
Electrical and Computer Engineering
University of Florida
NCN/NMI Team

Work supported by SCI-0438246

Outline

- Resource virtualization: Motivations and techniques
- Virtual Machines: past, present, future
- Applications in Grid Computing

Work supported by SCI-0438246

Physical computer

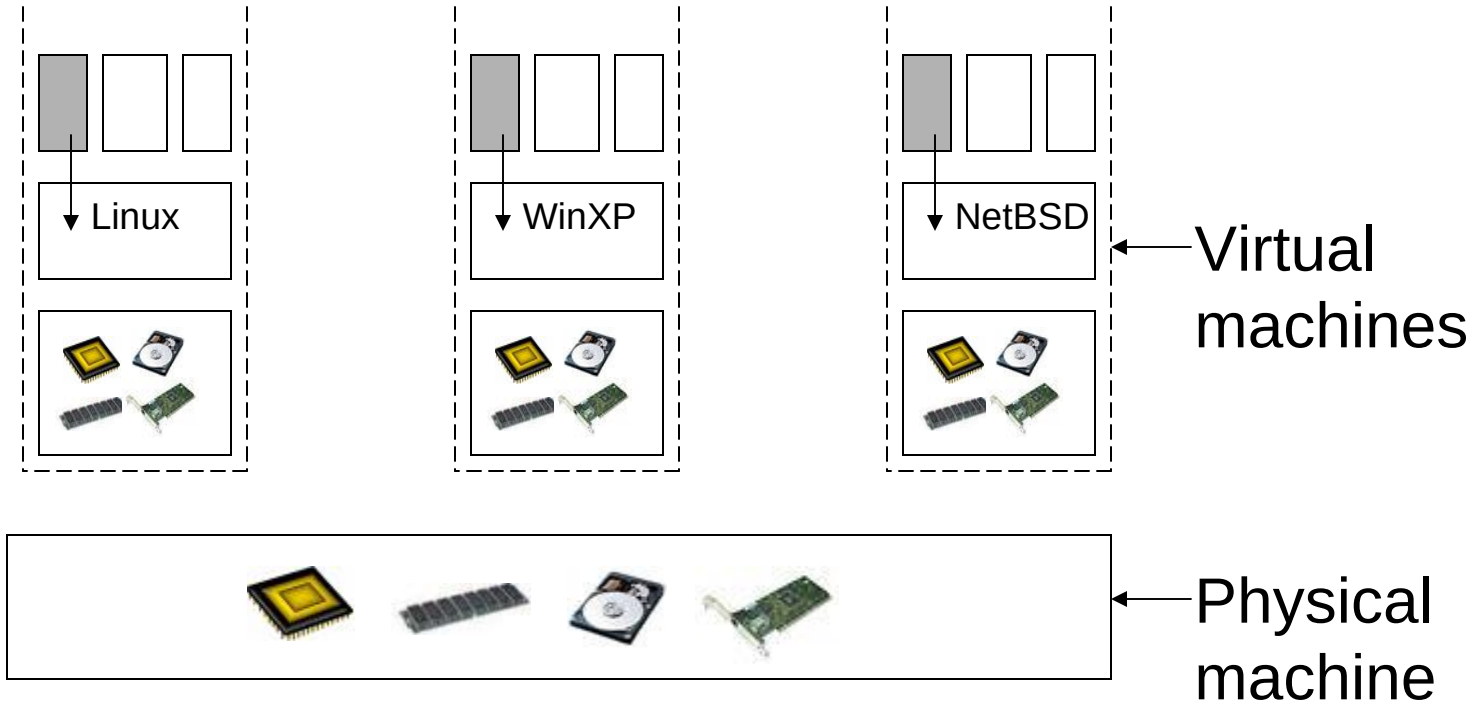
- Processor, memory, and I/O
  - Hard disk and network interfaces are key components of the I/O subsystem

- Processor runs instructions.
- Memory holds data for fast operation.
- Disk holds data in persistent state (files, directories).
- Network supports communication




Work supported by SCI-0438246

- A virtual computer has the same "look and feel" as a physical computer
  - Virtual processor, virtual memory, virtual disk and virtual network card
  - Can run exactly the same software (e.g. Windows, Linux)
  - But *multiple virtual computers can time-share a single physical computer!*

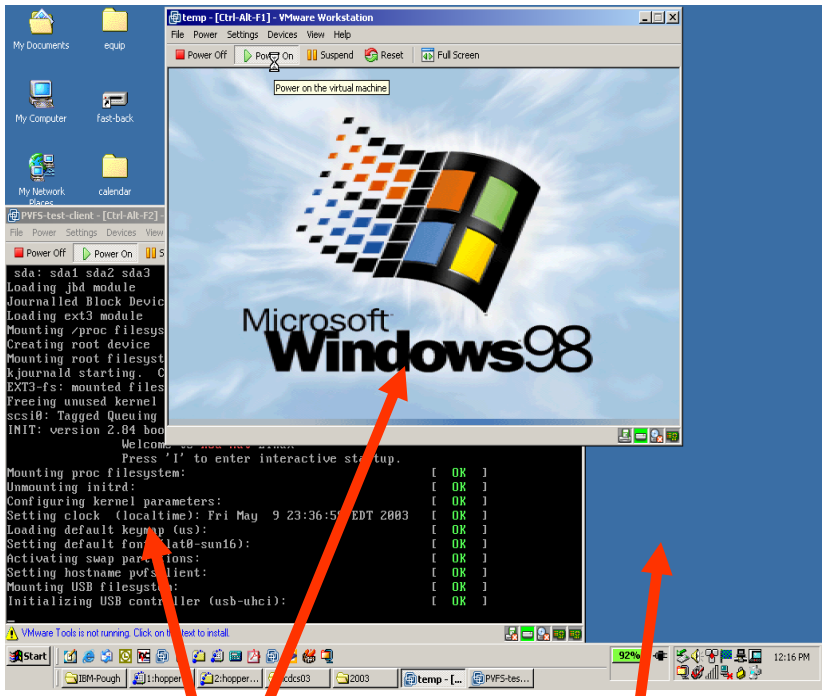


Work supported by SCI-0438246

Capable of running full-blown O/S within VM "guest"

- VMware
- Xen
- Microsoft Virtual PC

- Support legacy software
  - Advantage over Java VM, C# CLR



Work supported by SCI-0438246

Why virtual Computers?

- Motivations

- Resource *sharing* and consolidation; security and isolation
  - Familiar example: virtual memory in multi-task O/Ss
  - Physical memory shared by multiple processes; each process has its own, independent address space

- Ease of management
  - Virtual (software) vs. physical (hardware)
  - Copying, migration, archival, versioning

- Virtualization adds an extra layer of security and isolation
  - A user can “crash” or reboot a VM without compromising its host, or other VMs

- VM is not nearly as complex as O/S
  - Because they are smaller and simpler, they are harder to ‘hack’ than an O/S
  - Disco VM: 50K lines of code
  - IRIXO/S : millions of lines

Work supported by SCI-0438246

Virtual Machines - VMs

- "Classic" VMs
  - Technology traces back to the early 70s
  - Legacy O/Ss which would share expensive mainframes
  - IBM 370 - zSeries

- Virtual machine defined:

"A system…which…is a hardware-software duplicate of a real existing machine, in which a non-trivial subset of the virtual machine's instructions execute directly on the host machine…"

[Goldberg, 1971]

Work supported by SCI-0438246

Architecture of VMs

- How are VMs implemented?

- Core: virtual machine monitor (VMM) software
  - Presents illusion of machine duplicate to other software running on top of it

- Virtualization operates at the (low) level of individual processor instructions
  - Can “boot” existing operating systems, run unmodified applications on top of VM

- Key insight to achieve good performance:
  - Run most instructions directly on hardware
    - Same performance as without a VMM
  - Emulate few instructions that require special handling
    - Virtualization overhead

Work supported by SCI-0438246

Example - performance

- Example quantifying virtualization overhead for typical CPU-intensive scientific applications
- SpecHPC benchmarks
- Experimental setup:
  - Physical: dual Pentium-III, 933MHz, 512MB mem, RedHat 7.1, SMP kernel
  - Virtual: VMware workstation, 128MB mem, RedHat 7.1 uniprocessor kernel

	Native	VM	Overhead
SPECseis	14806s	14946s	1.0%
SPECclimate	9304s	9693s	4.2%

Work supported by SCI-0438246

Overheads can be larger with disk I/O, network activity
Xen VMM achieves near-native performance

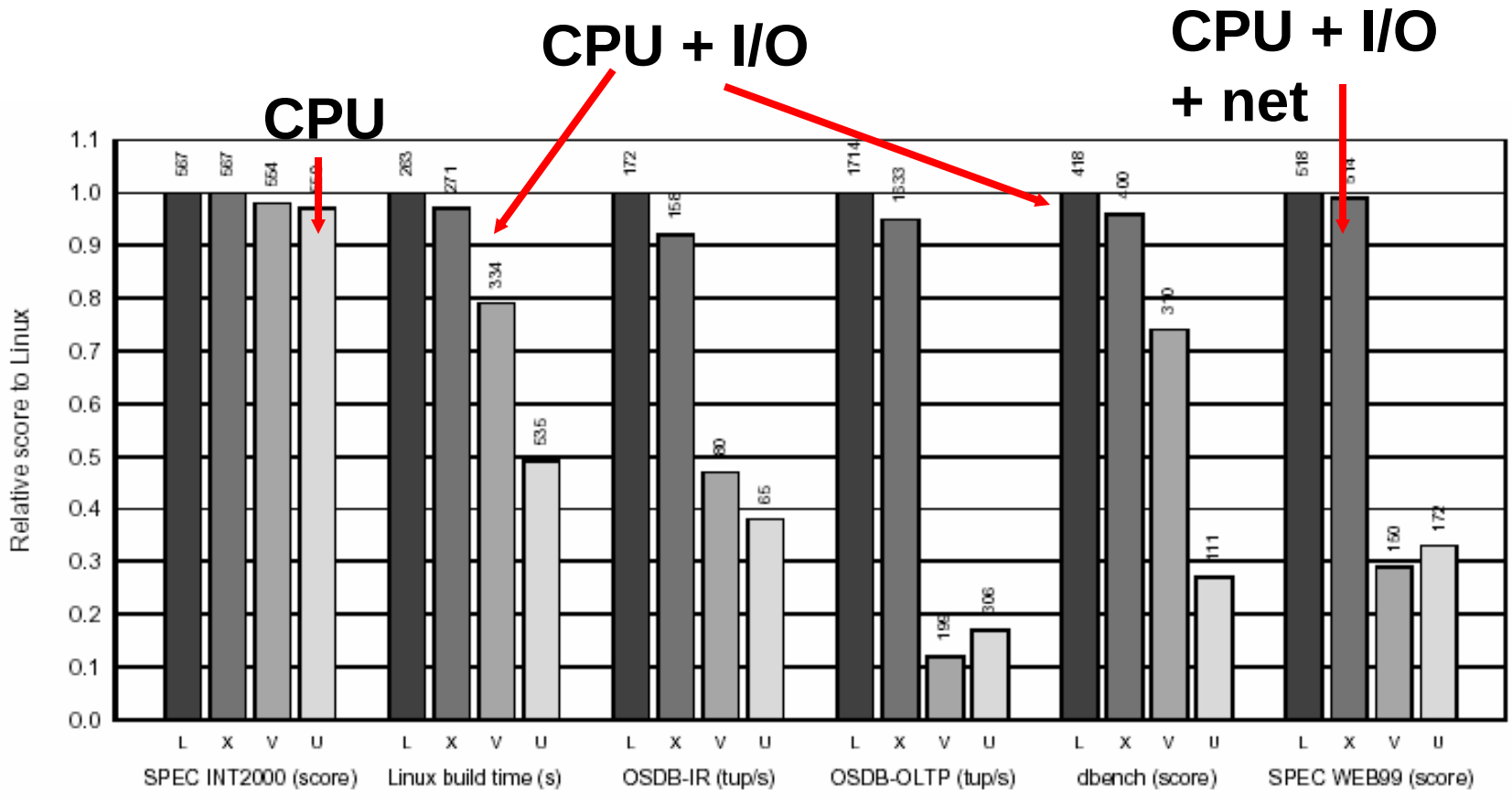


Figure 3: Relative performance of native Linux (L), XenoLinux (X), VMware workstation 3.2 (V) and User-Mode Linux (U).

Barham et al, "Xen and the Art of Virtualization", SOSP 2003

Work supported by SCI-0438246

Virtual machine software

- Several mature virtualization products now available for typical desktops and servers; here is a brief overview

- VMware
  - Pioneer of products bringing VMs to Intel-based desktops and servers
- Products
  - Workstation
    - Easy to install and use; good performance and several features
  - Player
    - Free! Can “play” VMs created by Workstation and ESX
  - GSX
    - “Group” server; more management tools
  - ESX
    - Enterprise server; higher performance, several management features, advanced scheduling
- For more information and downloads: http://www.vmware.com

Work supported by SCI-0438246

Virtual machine software

- Microsoft

- Acquired start-up (Connectix) and added VMs to its line of products

- Plans to have VM support as part of future releases of Windows

- VirtualPC
  - Akin to VMware's workstation. Can run VMware VMs (and vice-versa)
  - Unlike workstation, does not run on top of Linux
  - Can run Intel-based VMs (e.g. Windows) on PowerPC-based Macs
- Virtual Server
  - Akin to VMware's GSX server

For more information:

- http://www.microsoft.com/Windows/virtualpc/default.mspx
- http://www.microsoft.com/windowsserversystem/virtualserver/default.mspx

Work supported by SCI-0438246

Virtual machine software

- Xen
- Open-source VM initially developed at Cambridge
  - Packaging and distribution now handled by start-up company XenSource
- Designed to achieve high performance
- "Para-virtualization" techniques to reduce virtualization overhead
  - But para-virtualization requires operating systems to be modified before they run on Xen
  - Typically, Xen runs Linux, FreeBSD or NetBSD
- With new Intel/AMD processors, Xen is beginning to support unmodified operating systems
- For more information and downloads: http://www.xensource.com

Work supported by SCI-0438246

Virtual machine software

- Parallels

- Workstation
  - Low-cost VM
  - Comparable to VMware workstation, Windows VirtualPC
  - Runs on both Windows and Linux
- Server, Enterprise server: due 2006

- For more information and downloads: http://www.parallels.com

Work supported by SCI-0438246

Hardware support

- Much of the virtualization overhead today are due to processors not being designed with virtualization in mind
- Seminal paper by Popek, Goldberg (1974) provides basic guidelines
  - Efficient VMM can be designed if:
    1. Processor has protection mechanisms, and
    2. Privileged instructions that read/write system status must cause exceptions if run at non-privileged level
- Modern microprocessors support protection (condition 1)
  - However, second condition rarely satisfied
    - E.g. Intel 'x86': 17 problematic instructions
- Today, VMs must use several software 'tricks' to circumvent problematic instructions in processors such as Intel/AMD
- Tomorrow: processors have been/are being redesigned to support more efficient VMs
  - Intel VT in dual-core "Yonah"; AMD "Pacifica"

Work supported by SCI-0438246

Beyond a single host

- One virtual machine
  - Single physical machine (uni/multi-processor); multiple concurrent O/Ss
  - VM monitors (VMware, Xen) virtualize local devices
- Interconnected VMs
  - Emerging applications not considered in early VM days
  - How to virtualize networking and remote storage access?
  - How to schedule, manage VMs beyond local area network (LAN) domain boundaries?

Work supported by SCI-0438246

Applications in Grid computing

- Goal: harness computing power of distributed desktops, servers, clusters
- Computing as utility
- Analogy to power grid
- Computing grid:
  - Outlet is the network jack; pay per processor cycles, storage, bandwidth, application time used
- Applications:
  - Nanoelectronics online simulation - nanoHUB
  - Many others: high-energy physics, hurricane path prediction, seismic modeling, protein sequencing, CAD design/simulation, rendering farms, financial analysis, …
- Virtualization facilitates management and deployment of Grid resources

Work supported by SCI-0438246

Grid computing on virtual resources

- Three virtues of virtualization
  - Manifolding, polymorphism, multiplexing
  - *Multiple*, *independent* virtual resources *share* a physical resource
- VMs:
  - Isolation – improved security
  - Multiple concurrent O/Ss – time-sharing
  - Independent configuration – flexibility
- Approach: apply virtualization machines, disk storage and network
- The nanoHUB and the underlying In-VIGO middleware
  - Application of these techniques result in a unique, flexible cyberinfrastructure

Work supported by SCI-0438246

Summary

- Virtual machines are becoming more and more pervasive in enterprise and desktop environments
- Hardware and software vendors are increasingly supporting virtualization
  - Competition – better products, lower prices
- Virtualization is key to a flexible computational cyberinfrastructure
- nanoHUB and In-VIGO: apply virtualization to bring online simulations to the nano-* community

Work supported by SCI-0438246