



Virtual systems: current practices and future possibilities

Or: Smoke and mirrors for doing less with less, and achieving more



Virtual systems: current practices and future possibilities

- What are virtual systems?
- What is the current technology?
- What can we do with virtual systems?
- Demonstration: Akimbi Slingshot (now VMWare Lab Manager)
- What's the next step?
- But before we get started: what about this "do less with less"?



Administrivia

- Trevor Grove
CSCF Research Support Group
David R. Cheriton School of Computer Science

trg@cs.*the-usual-place*
Davis Centre 2562
x34679

- Versions of the presentation will be available at:
<http://www.cs.uwaterloo.ca/~trg/public/toc.php>



What are virtual systems?

- The usual definition goes something like:
 - “A way to run some software on a computer that makes it look like there’s more than one computer there.”
 - A way to run multiple OS instances simultaneously on one physical computer (as opposed to multiple-boot computers which can only run one OS at a time)
 - Saves the expense of buying extra computers, saves power and heat, etc.
- But, that’s not really doing justice to the concept!

- Some history: first there was virtual memory:
 - When a computer uses its disk drives to fake real RAM. Very slow.
www.math.okstate.edu/system/terms.html
 - Well sort of, but kind of misses the point
 - Virtual (or logical) memory is a concept that, [...] use a very large range of memory or storage addresses for stored data. [...]
www.cheap-computers-guide.com/computer-related-glossary.html
- Right! It's a concept, or abstraction



What are virtual systems?

3

- Then there were virtual machines:
 - A functional simulation of a computer and its associated devices. appl.nasa.gov/resources/lexicon/terms_v.html
 - Not bad; not sure about “simulation”, though
 - An abstract specification for a computing device that can be implemented in different ways, in software or hardware. [...]
software.allindiansite.com/java/uvjava.html
- Right. And there’s that “abstraction” word again...

- And generally, there is virtualization:
 - In computing, virtualization involves the process of presenting computing resources in ways that users and applications can easily get value out of them, rather than presenting them in a way dictated by their implementation, geographic location, or physical packaging. In other words, it provides a logical rather than physical view of data, computing power, storage capacity, and other resources.
en.wikipedia.org/wiki/Virtualization



So really: what are virtual systems?

■ My definition:

- A virtual system is a software—hardware hybrid system that enables users to define idealized hardware and software platforms without regard to the actual underlying hardware and software
- We can create an abstraction that (usually) simplifies the details of the hardware, and thus lets us build software for a truly standard platform
- We become isolated from the vagaries of hardware manufacturer
- N.B. my use of *virtual system* vs. *virtual machine*



What is the current technology?

- Historical note: virtualization isn't new:
 - IBM's first commercial products were in use in the late 1960s (IBM 360/67)
 - UW was an early adopter of IBM's VM products (1970s)
 - Used in administrative, research and teaching through the 1980s and into the early 1990s
- Types of virtualization technology:
 - Hardware
 - Software that is the real hardware's OS
 - Software that runs on (or beside) a traditional OS
 - Application virtualization



What is the current technology?

2

■ Hardware virtualization:

- For example: IBM zSeries mainframes
- Successor to the 390 mainframes, which were successors to the 370 (e.g. UW's 370/158 and 43xx mainframes)
- Implemented in hardware/firmware: partition a single hardware platform into many independent pieces, each of which is equivalent to the original, but independent (called LPARs – logical partitions)
- Also available in IBM's pSeries (PowerPC)
- Hardware typically has special support for virtualization (x86 in ~2005, IBM zSeries for 30+ years)
- Typically less overhead cost than software solutions



What is the current technology?

3

- Software that is the real hardware's OS:
 - Install like an os
 - Principal job is to implement virtualization *hypervisor*
 - Provides little or no end-user interaction
- VMware's ESX product is an example
 - For Intel x86/AMD platforms
 - Installs on the "bare metal" and implements the virtualization hypervisor
 - Intended to run current server-class OSes unmodified
 - Intended for enterprise deployments; bells&whistles+ +

- Another example: IBM's z/VM and its ancestors (VM/ESA, VM/CMS)
 - Two components: the hypervisor and the guest OS
 - Hypervisor installs in an LPAR and creates a platform to run many virtual machines (called CP)
 - Virtual machine is equivalent to the underlying hardware
 - Can run any guest OS capable of running on the hardware, including a nested z/VM
 - Linux for zSeries hardware is widely used as a guest
 - Virtual machines can interact with CP via an API or interactive command interface



What is the current technology?

5

- Virtualization under an existing OS:
 - VMware GSX (now [VMware Server](#)) and derivatives
 - [Microsoft Virtual PC 2004](#) and [Virtual Server 2005](#)
 - And many others (list growing daily)
- Virtualization software installs as an application under the host OS
 - Defines the hardware for the virtual machine
 - Standard video, sound, disk, RAM, networking etc are defined in terms of host OS resources
 - Guest operating systems must support the virtual hardware



What is the current technology?

6

- Xen is a variation of “virtualization under OS”:
 - Open-source project from Cambridge, now commercial
 - A set of kernel modifications for the host OS: “para-virtualization”
 - Blends the hypervisor into the modified kernel
 - Several Linux/Unix host OSes are supported
 - Guest OSes generally require modification
 - But: as of Xen v3.0, x86 hardware virtualization assist allows unmodified Windows guests



What is the current technology?

7

- Solaris containers/zones are another variation:
 - A modified kernel partitions the single OS system
 - Software version of LPARs
 - Similar techniques include “chroot jails”, FreeBSD jails
 - Essential idea is to isolate resources and make accidental interaction impossible

■ Application virtualization:

- Application programs that define and implement some kind of virtual environment
- Java virtual machine
- Microsoft's .Net
- DOSBox – runs DOS programs by complete emulation of a DOS-based computer
- Could consider that Windows Remote Desktop or XDMCP are a kind of virtualization



What is the current technology?

9

■ Key points to remember:

- Virtualization represents an abstraction of hardware
- The abstraction is derived from (sometimes independent of) the underlying real hardware
- “Virtualization under an OS”, such as VMware Server, MS Virtual PC/Virtual Server and Xen, and “bare-metal virtualization” like VMWare ESX are the leading players
- The host OS and the guest OS are independent
- This independence is what makes virtualization so attractive from the perspective of IT management
- Virtual systems might lower hardware costs, but OS costs are the same



What can we do with virtual systems?

- So what good is all this technology?
- Several “classic” uses:
 - Reduction of real hardware (consolidation)
 - Testing and development environments
 - Infrastructure management
 - Backup and recovery

- Reduction of real hardware (consolidation):
 - Run two, three or more virtual systems on one real platform
 - Reduce hardware costs
 - Reduce power (and UPS) requirement
 - Reduce cooling requirements
 - Reduce floor/rack space usage
 - Most servers are under-utilized and have excess capacity, especially newer multi-core, multi-cpu servers

- Testing and development environments:
 - Production systems are never used for testing or development ☺
 - Testing environments can be built and torn down as required
 - Multi-version and multi-platform environments can be hosted conveniently
 - Virtual networks (of virtual systems) can be created to simulate workloads, interactions
 - Current products support (to varying degrees) snapshots and storing of VM states for later re-activation

■ Infrastructure management:

- Providing service redundancy without hardware costs
- E.g. DNS, Active Directory, email and web servers – allows server restarts without interrupting services
- Isolation of critical services for reliability or security
- Guest OS images can be moved, cloned and deployed easily
- Workloads can be moved easily between virtual system hosts as required

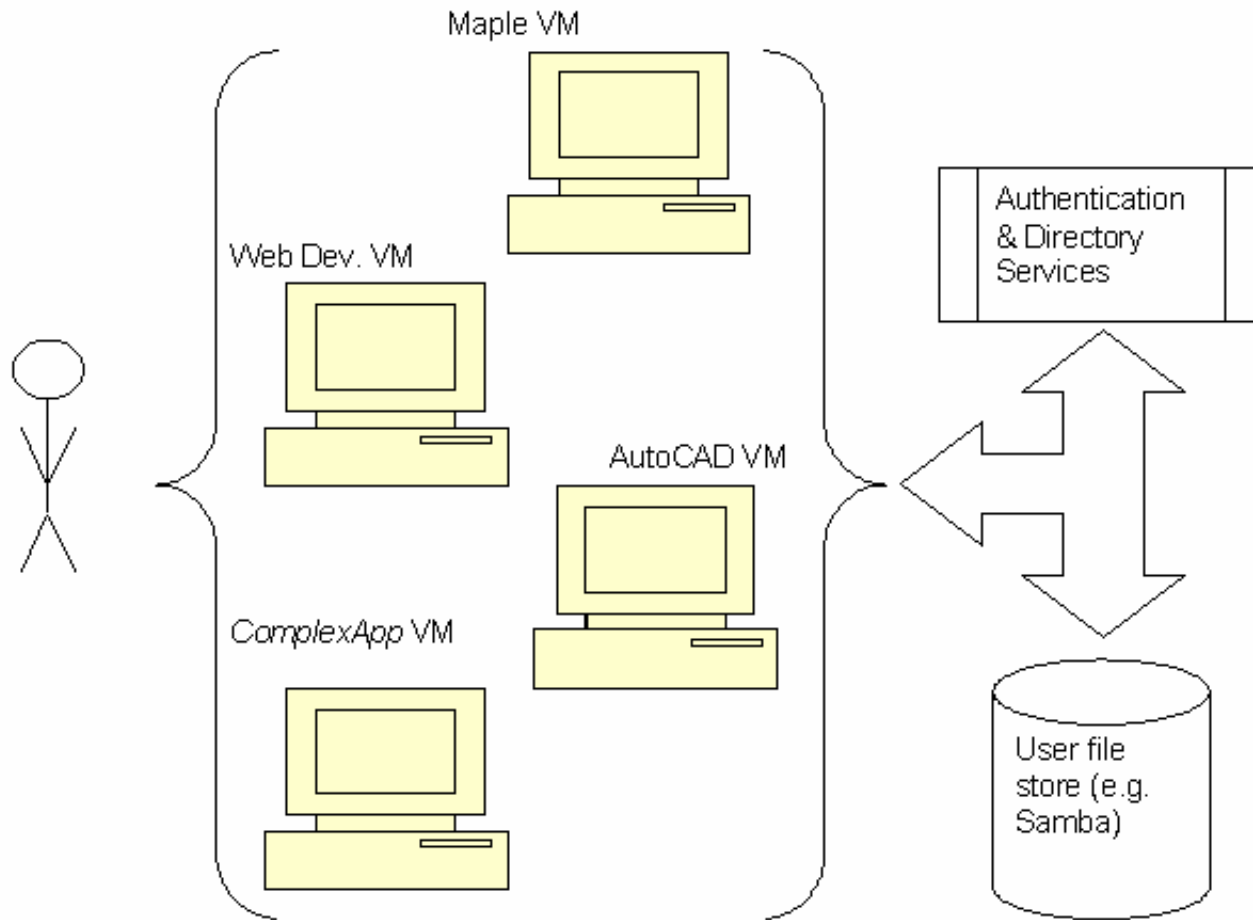
■ Backup and recovery:

- Even with the best backup and recovery strategies, a failed disk or external storage takes time to recover
- Most current products store virtual disks as files within the host OS file system
- The VM definitions themselves are also files
- Recovering a system and bringing up a replacement is simply a matter of copying files and starting a new VM
- Disaster recovery \equiv virtual machine migration

- Some less-common uses for virtual systems
- Software distributions:
 - For complex software, instead of receiving a set installation media, receive an image of a ready-to-run VM
 - For guest OSes and “big” applications
- Teaching labs & standard computing platforms
 - Install a completely stock host OS, then run a carefully-crafted guest OS
 - Users never interact with the host OS, only the guest
 - User files stored outside the virtual system (e.g. network-based)

- More less-common uses for virtual systems
- Platforms for special-purpose applications
 - Instead of installing software on the user's platform, give the user the complete platform with the application pre-installed
 - Create a different VM for each application
 - Users choose the VM that is appropriate to the task
 - User authentication and file storage is elsewhere

What ... with virtual systems? 8



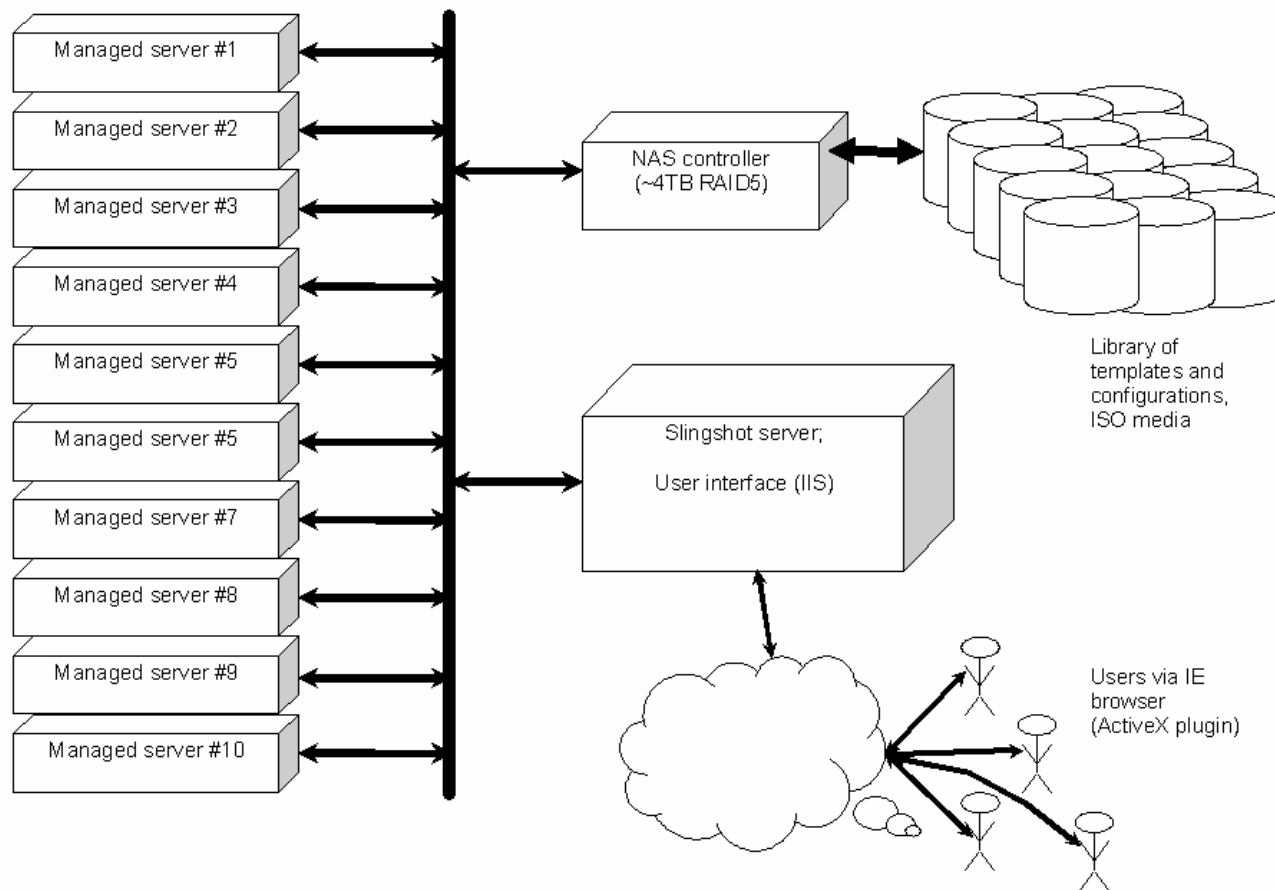


Demonstration

- Akimbi Slingshot (now [VMware Lab Manager](#)):
 - SCS Tetherless Computing Lab; for network simulation
 - Management system for controlling sets of virtual systems
 - presents a high-level abstraction called *configurations*
 - Exploits the programming API for VMware ESX (prior to VMware acquisition, also worked with Microsoft Virtual Server – demo will be of the Akimbi MS VS version)
 - Supports multiple users who can share virtual systems
 - Consists of a management server and a set of worker systems that host virtual machine (*managed servers*)



Demonstration





Demonstration

- Akimbi Slingshot management server:
 - Create virtual system templates and configurations and store them in a shareable library
 - Deploy configurations across managed systems completely transparently to the end user
 - Manager handles IP addressing, NATting, firewalling
 - Running configurations can be “snapshotted”, suspended, stored, restarted, shared
 - Users interact only with the management server, which brokers all interactions with the virtual guests
 - Currently requires IE (uses ActiveX controls to display VM consoles)



Demonstration

- Script:
 - Log into system
 - Create a VM template
 - Deploy to bare machine
 - Install an OS (start – won't complete this)
 - Create a single-machine configuration and deploy
 - Add another VM to the configuration
 - Suspend configuration
 - Remote desktop
 - Log in as administrator to look at overall system



What's next: a vision for computing

- Disclaimer
- This is a personal opinion and extrapolation and in **no way whatsoever** represents anything even vaguely or remotely associated with CSCF, the DRCSCS or any official policies thereof
- This discussion is given from the perspective of requirements of Computer Science teaching and research, but I believe many of the ideas are applicable in other disciplines and environments



A vision for computing

- CS teaching facilities require packaging applications to fit into the available OS environment.
 - The environment varies, from single-user Macs to multi-user Solaris & Linux
- Instead of packaging applications for an OS and bringing students to the OS, package the OS around the application and bring it to the student
- Requires personal workstations conforming to a minimal standard – capable of running a VM



A vision for computing

■ Examples:

- First-year CS students need a Java environment that has a specific set of tools and underlying OS environment
 - Upper-year students requires specific tools on a course-by-course basis – some work better in Windows, or Linux, or Solaris or a customized OS kernel
 - Grad students use one OS environment for their own work, but have TA responsibilities for other environments
- The only standard required is the virtual system platform technology



A vision for computing

■ So imagine:

- Students have personal systems running any host OS they want as long as it can run the VM
- Enrolling in a course automatically grants access to any required virtual system images defined for the course
- OS images are acquired and used, and can be refreshed and replaced trivially
- Personal data is not stored permanently in the VM images, so there is requirement for infrastructure to provide a globally-accessible file-store – off-campus access will be required



A vision for computing

■ User advantages:

- Use any platform for general computing – the need to conform to a mandated platform is mostly eliminated
- No requirements to acquire, install, configure or maintain application packages
- Ultimate portability – like taking the “lab computer” with you
- Coursework is independent of personal work
- Potential to increase diversity and exposure to different computing environments



A vision for computing

■ Infrastructure advantages:

- Software licencing costs reduced – user community size is constrained to enrolled students
- Reduces or eliminates the need for general-purpose labs and multi-use systems
- Focus on infrastructure: networking, file-stores, email and directory services
- VM images can be created by faculty or staff, tested, tuned and adjusted independently – no more “version conflicts” or forced updates of packages to meet pre-requisites

- Virtualization is an old idea that presents new and interesting opportunities
- There are many virtualization products available of differing technologies
- Virtualized environments can reduce costs, improve reliability and resource management, simplify management
- A new way to use virtualization: visionary or hallucinogenic? 😊



The End