Using the Xen Hypervisor to Turbocharge OS Deployment

Mike D. Day
Ryan Harper
Anthony Liguori
Andrew Theurer
Michael Hohnbaum
Real-world Hypervisor Applications

- Make more efficient use of expensive hardware
- Server Consolidation
- Virtual hosting
- Systems Management???
Hypervisors and OS Deployment

- Is it feasible to use a Hypervisor as an OS deployment tool?
- What are the problems?
- What are the benefits?
- Xen as a deployment engine?
Background: Linux Deployment

• Current Methods and tools
  – Manual
  – Shared r/o filesystems
  – Kickstart, YaST Autoinstaller, others
  – Clone and Customize
    • “ghost for linux”

• Lots of experiments in the clustering realm
• Linux has good deployment solutions, but users always need better tools
Improvements to OS Deployment Using Hypervisor Technology

- Hardware normalization
  - Deploy to “virtually identical” machines
- Dynamic hardware control
- Use of Virtual I/O Devices
  - Block
  - Network
  - USB
“Hardware” Normalization

• Regardless of physical platform configuration, normalize the configuration of the virtual machine
  – e.g., all VMs have SCSI devices, 4 CPUs, 2 ethernets.
  – One kernel configuration can replace multiple kconf's
  – Uniform drivers
  – Narrows the range of application configuration settings

• This is normally done today by standardizing on a single hardware platform
Hardware Normalization (cont'd)

• /etc/fstab, /etc/udev/devices, ...
  - Normalized virtual machines allow identical hardware configuration

• Network
  - Use a canonical network configuration for all guest domains. Give them networks within the private address ranges. However, some customization of network configuration still required.
  - Domains hosting network services can use bi-directional nat (via domain 0) to allow public access to hosted services
Dynamic Hardware Control

- “Virtual Hotplug”
  - Add memory, block and network devices, CPU
- Physical computer and hypervisor continue running
  - Other domains continue
- Requires a domain restart today, but equivalent feature without domain restart is on the Xen roadmap
- Full evolution of Linux hotplug will make this feature more powerful
File System Reuse

- Use one file system image as basis for many variants
  - Maintain one file system instead of many
  - Shared r/o filesystems
  - Images mounted via loopback
  - NFS mounts
  - SAN (storage virtualization)

- Requires automated customization
  - COW
File System Reuse (cont.'d)

• Virtually all benefit to be gained from file system reuse is gained by maintaining a single image and having multiple domains use that maintenance automatically
  – Kernel Upgrades
  – Security patches

• This is one reason why OS Containers are attractive
Xen Domain Deployment

• Domain 0
  – Domain 0 is usually a regular linux distribution that runs with extra privileges
  – Virtualizes i/o devices for other domains
  – Xen is not “running” until domain 0 is running
  – Domain 0 used to start “guest” domains
Xen 2.0 Domain Configuration File

# -*- mode: python; -*-

# Python configuration setup for 'xm create'.
# This script sets the parameters used when a domain is created using 'xm create'.
# You use a separate script for each domain you want to create, or
# you can set the parameters for the domain on the xm command line.

# Kernel image file.
#kernel = "/boot/vmlinuz-2.6.10-xenU"

# Optional ramdisk.
#ramdisk = "/boot/initrd.gz"

# The domain build function. Default is 'linux'.
What's Missing for Deployment?

- Image Management
  - Xen VBDs provide a simple mechanism for using images to deploy domains.
  - Virtual storage infrastructure (SAN) is a more complicated, better performing mechanism also available for use with Xen
  - Customization of images

- Integration of image management with Xen domain configuration tools
First Attempt: Xen Container Syntax

• Definition of a Container
  – Existing Xen configuration syntax
  – File System Images
    • One or more images that will be exported by Xen to the new Domain
  – Customization Scripts
    • Syntax to customize images for each new Domain. Need to be repeatable.
  – Init Hooks
    • Further customization to be done by init after Domain is started
Xen Container Syntax

[create /etc/ file:///home/mdday/src/ols/generic_etc.cpio.gz][end]

• [image /home/mdday/src/ols/generic_etc 200mb Generic Etc][end]

• [replace /etc/xinetd.d/echo
  • file:///home/mdday/src/ols/generic_etc.cpio.gz
  • out_archive=file:///home/mdday/src/ols/custom_etc.cpio.gz
  • service echo
  •{
    • type = INTERNAL
    • id = echo-stream
    • socket_type = stream
    • protocol = tcp
    • user = root
    • wait = no
    • disable = no
  }][end]
Composing a Xen Container

- Include “container syntax” within existing domain configuration file
- Pre-process the container file to execute the container syntax and then pipe the existing domain configuration to the Xen domain creation tool
Xen Container Processing

Remote Server

Pre-processor

Image Preparation

Domain Preparation using existing Xen 2.0 tools
Init Processing

echo -n "Mounting devpts: "
mount /dev/pts
check_status
. /etc/rc.d/init_hook1

echo -n "Enabling swap space: "
swapon -a
check_status

echo -n "Setting hostname: "
hostname -F /etc/HOSTNAME
check_status

. /etc/rc.d/init_hook2
Container Tool

- 1400 lines of bash
- Proof-of-concept
- “garbage bag” of image tools plus pre-processor
  - Sparse disk image creation
  - Archive creation
  - Patch generation
  - File copy/replace
  - Retrieve/store images and archives on remote server
Problems

- Modifying binary files
- Customizing large directory trees
  - Works best with discrete file changes
    - e.g., group.diff passwd.diff shadow.diff

--- xen-tty-img/etc/group 2004-08-21 16:03:20.000000000 -0400
@@ -5,3 +5,4 @@
 web:x:300:
 nobody:x:65534:
 guest:x:500:
+mdaday:x:501:
Problems (cont.d)

• Network repository for file system images
  – As number of “container files” increases, complexity of managing container files and system images increases.

• These are “standard” deployment problems
Benefits

• Once Xen is resident on a platform and the container is defined, deploying linux can be simpler and faster than existing methods.

• Container approach encourages defining “canned” systems for specific purposes.
  – DBMS, LAMP, Clusters, etc.

• Re-use of file system images reduces impact of new kernels and other updates.

• Workload management using domains
How to deploy Xen?

• Bootable image
  – Remote boot

• Firmware
  – Lot's of examples of this hypervisor format in larger platforms
  – Would open up new uses of hypervisor as a systems management tool
Xen 3.x

- Management and control architecture will be much improved
- Will work to incorporate image management and improved “container” into Xen tool-set.