42 Hosts in 1U

Using Virtual Machines

Ewen McNeill
<ewen@naos.co.nz>

Naos Ltd

2007/01/31 — Sysadmin Miniconf, NZNOG 2007

Outline

1. Introduction
   - Administriva
   - Overview
   - Motivation
2. Virtual Machine Technology
3. Virtualisation Example
   - Xen Overview
   - Xen Installation
   - Xen domU configuration
4. 42 Hosts in 1U
5. Suitability and Limitations
6. Summary

Administrivia

- About the speaker
  - Runs Naos Ltd
  - A Wellington based Linux/Unix, Networking and VoIP consultancy
- Questions Policy
  - If it is about the current slide, raise your hand.
  - Please ask more general questions at the end.
- Slides: http://www.naos.co.nz/talks/42-hosts-in-1u/

What is a host?

- Some processing power — CPU(s)
- Some working memory — RAM
- Some storage space — Disk(s)
- Network connection(s) — NIC(s)
- Power supply, console, ...

What is a virtual machine?

- Some processing power — a portion of a CPU
- Some working memory — an allocation of RAM
- Some storage space — an allocation of disk space
- Network connection(s) — virtual NICs

- All “emulated” inside a physical host
- Resources shared with other virtual machines

Why virtual machines – Part 1

- Consider 42 hosts from 2U servers (like HP DL380)
- Physical space:
  - 42U rack/2U servers = 21 — need 2 racks
- Power:
  - 42 servers * 250W = 10.5kW
- Heat:
  - 10.5kW — enough said
- Network connections:
  - 42 network cables, 2 * 24-port switches
- Cost:
  - 42 * $4k+ = $lots

Why virtual machines – Part 1

- Consider 42 hosts from 2U servers (like HP DL380)
- Physical space:
  - 42U rack/2U servers = 21 — need 2 racks
- Power:
  - 42 servers * 250W = 10.5kW
- Heat:
  - 10.5kW — enough said
- Network connections:
  - 42 network cables, 2 * 24-port switches
- Cost:
  - 42 * $4k+ = $lots

Why virtual machines – Part 1

- Consider 42 hosts from 2U servers (like HP DL380)
- Physical space:
  - 42U rack/2U servers = 21 — need 2 racks
- Power:
  - 42 servers * 250W = 10.5kW
- Heat:
  - 10.5kW — enough said
- Network connections:
  - 42 network cables, 2 * 24-port switches
- Cost:
  - 42 * $4k+ = $lots

Why virtual machines – Part 1

- Consider 42 hosts from 2U servers (like HP DL380)
- Physical space:
  - 42U rack/2U servers = 21 — need 2 racks
- Power:
  - 42 servers * 250W = 10.5kW
- Heat:
  - 10.5kW — enough said
- Network connections:
  - 42 network cables, 2 * 24-port switches
- Cost:
  - 42 * $4k+ = $lots

Why virtual machines – Part 1

- Consider 42 hosts from 2U servers (like HP DL380)
- Physical space:
  - 42U rack/2U servers = 21 — need 2 racks
- Power:
  - 42 servers * 250W = 10.5kW
- Heat:
  - 10.5kW — enough said
- Network connections:
  - 42 network cables, 2 * 24-port switches
- Cost:
  - 42 * $4k+ = $lots

Why virtual machines – Part 1

- Consider 42 hosts from 2U servers (like HP DL380)
- Physical space:
  - 42U rack/2U servers = 21 — need 2 racks
- Power:
  - 42 servers * 250W = 10.5kW
- Heat:
  - 10.5kW — enough said
- Network connections:
  - 42 network cables, 2 * 24-port switches
- Cost:
  - 42 * $4k+ = $lots

Why virtual machines – Part 2

- Perhaps 1U servers (like HP DL320)
  - Physical Space: 1 (42U) rack
  - Power: 42 servers * 175W = 7.4kW
  - Heat: 7.4kW — over 3 heaters worth
  - Network connections: still 42
  - Cost: 42 * $2.5k+ = still $lots
- Or blades (like HP BladeSystem p-Class)
  - Physical Space: 8 Blades/6U => 36U (plus 6 spare slots)
  - Power: 42 servers * 125W? = 5.25kW
  - Heat: 5.25kW — still over 2 heaters worth
  - Network connections: 1-2 switches per in blade enclosure
  - Cost: 6 * $2k+ (enclosures) + 42 * $2.5k+ = $lots

Why virtual machines – Part 2

- Perhaps 1U servers (like HP DL320)
  - Physical Space: 1 (42U) rack
  - Power: 42 servers * 175W = 7.4kW
  - Heat: 7.4kW — over 3 heaters worth
  - Network connections: still 42
  - Cost: 42 * $2.5k+ = still $lots
- Or blades (like HP BladeSystem p-Class)
  - Physical Space: 8 Blades/6U => 36U (plus 6 spare slots)
  - Power: 42 servers * 125W? = 5.25kW
  - Heat: 5.25kW — still over 2 heaters worth
  - Network connections: 1-2 switches per in blade enclosure
  - Cost: 6 * $2k+ (enclosures) + 42 * $2.5k+ = $lots

Why virtual machines – Part 3

- Where virtual machines are applicable they offer:
  - Substantial space savings
  - Substantial power and heat savings
  - Much less cabling
  - Substantial cost savings
- Not suitable for every situation
  - We'll consider suitable and unsuitable situations later

Why virtual machines – Part 3

- Where virtual machines are applicable they offer:
  - Substantial space savings
  - Substantial power and heat savings
  - Much less cabling
  - Substantial cost savings
- Not suitable for every situation
  - We'll consider suitable and unsuitable situations later

Virtualisation Overview

- There are several different types of virtualisation:
  - Emulation
  - Native Virtualisation
  - Paravirtualisation
  - Operating System level virtualisation
- The principle differences are:
  - Efficiency
  - Emulation of hardware
  - Ability to run unmodified operating systems

Virtualisation Overview

- There are several different types of virtualisation:
  - Emulation
  - Native Virtualisation
  - Paravirtualisation
  - Operating System level virtualisation
- The principle differences are:
  - Efficiency
  - Emulation of hardware
  - Ability to run unmodified operating systems

Emulation

- Simulates everything, including CPU, in software
- Often simulates real, legacy, hardware
  - Eg, MAME (http://www.mame.net/)
- Other examples:
  - QEMu (http://fabrice.bellard.free.fr/qemu/)
  - Virtual PC on PowerPC based Macs — simulates PC
- Advantages:
  - Run unmodified operating system and applications
  - Run programs for different CPU architecture
  - Accurate environment (eg, per-clock-cycle simulation)
- Disadvantages:
  - Slow! (eg, 10% of native CPU speed)

Native Virtualisation

- Simulates some real hardware, uses native CPU
- Requires CPU support for traps to virtualisation
- Examples:
  - VMWare Workstation, VMWare Server
  - Mac on Linux (PowerPC)
- Advantages:
  - Run unmodified operating system and applications
  - Fairly fast (eg, 80-90% of native speed)
- Disadvantages:
  - Only programs for same CPU architecture
  - Only some hardware emulated
  - Often only simple (slower) legacy hardware

Paravirtualisation

- Uses a hypervisor to access real hardware
- Uses native CPU
- Simulates virtualisation-efficient disk, network, etc devices
- Guest OS use custom device drivers
- Examples:
  - Xen (http://www.cl.cam.ac.uk/research/srg/netos/xen/)
  - L4 microkernel (http://www.l4hq.org/)
- Advantages:
  - Fast (eg, 90-95% of native speed)
- Disadvantages:
  - OS needs porting to virtualisation technology
  - Only programs for same CPU architecture

Operating system level virtualisation

- Partitions operating system into isolated areas
- OS kernel manages separation between virtual “machines”
- Examples:
  - Linux VServer (http://linux-vserver.org/)
  - Solaris Zones (http://www.sun.com/bigadmin/content/zones/)
  - FreeBSD Jails (http://en.wikipedia.org/wiki/FreeBSD_Jail)
- Advantages:
  - Fast (essentially native speed)
- Disadvantages:
  - Only one OS/kernel type and version
  - Generally less isolation of virtual “machines”

Xen Overview

- http://www.cl.cam.ac.uk/research/srg/netos/xen/
- Current release: version 3.0.3
- Licensed under GPL (GNU Public License)
- Runs on x86 and x86/64 architectures (more coming)
- Linux (2.4 and 2.6), NetBSD, FreeBSD (domU only) supported
- Paravirtualisation approach, using a hypervisor

Xen Architecture

- Paravirtualisation: small hypervisor manages
  - Resource allocation (eg, CPU scheduling)
  - Communication between virtual machines
  - Virtual device access (routed via dom0)
- Privileged “dom0” virtual machine (one only)
  - Responsible for all real hardware I/O
  - Manages startup/shutdown of virtual machines
- Unprivileged “domU” virtual machines
  - Allocated some RAM, virtual disk, virtual network interface(s)
  - Get (partial) use of one (or more) CPUs

Xen Architecture

- Paravirtualisation: small hypervisor manages
  - Resource allocation (eg, CPU scheduling)
  - Communication between virtual machines
  - Virtual device access (routed via dom0)
- Privileged “dom0” virtual machine (one only)
  - Responsible for all real hardware I/O
  - Manages startup/shutdown of virtual machines
- Unprivileged “domU” virtual machines
  - Allocated some RAM, virtual disk, virtual network interface(s)
  - Get (partial) use of one (or more) CPUs

Xen Architecture

- Paravirtualisation: small hypervisor manages
  - Resource allocation (eg, CPU scheduling)
  - Communication between virtual machines
  - Virtual device access (routed via dom0)
- Privileged “dom0” virtual machine (one only)
  - Responsible for all real hardware I/O
  - Manages startup/shutdown of virtual machines
- Unprivileged “domU” virtual machines
  - Allocated some RAM, virtual disk, virtual network interface(s)
  - Get (partial) use of one (or more) CPUs

Xen Installation

- Xen is not (yet) integrated into the Linux kernel mainline
- But included in many distributions
  - Debian 4 (Etch — release due soon), Fedora 6, OpenSuse 10, etc

- Need:
  - Working Linux installation
  - The Xen Hypervisor
  - Linux kernel with Xen dom0 support and drivers for real hardware
  - Grub (boot loader)

Xen Setup in Grub: /boot/grub/menu.lst

```
title Xen 3.0 / XenLinux 2.6.16.26
root (hd0,0)
kernel /xen-3.0-i386-pae.gz
module /xen0-linux-2.6.16.26-naos.xen0 root=/dev/cciss/c0d0p5 ro console=tty0
module /xen0-modules-2.6.16.26-naos.xen0
```

Xen Management

- dom0 machine runs much like “real” machine
- Essentially full hardware access
  - Should be able to reboot into Xen or back to native machine
- Xen Utilities (xen-utils) to manage the Xen environment
  - xm front end (domU start/stop/console, etc)
  - xentop
  - Network management scripts
- Once dom0 is running you can configure and start domU
- Can configure (some) domU to start on boot of dom0

Configuring domU — Part 1

- Need Linux (etc) kernel with Xen DomU device drivers
- Configuration required for:
  - Processing power — virtual CPUs
  - Working memory — RAM allocation
  - Storage space — virtual disk
  - Network connection(s) — virtual NICs
- Specified by configuration file, typically in /etc/xen
- Need Linux (etc) install on virtual disk

Configuring domU — Part 2

- Processing power:
  - cpus = "LIST" (physical CPUs to let domU use)
  - vcpus = N (number of virtual CPUs for domU)
  - Default is to let Xen pick one CPU to share with domU
- Working memory:
  - memory = N (megabytes of memory for domU)

Configuring domU — Part 2

- Processing power:
  - cpus = "LIST" (physical CPUs to let domU use)
  - vcpus = N (number of virtual CPUs for domU)
  - Default is to let Xen pick one CPU to share with domU
- Working memory:
  - memory = N (megabytes of memory for domU)

Configuring domU — Part 3

- Storage space:
  - Backing for storage can be physical disk partition
  - Or logical volume (eg, LVM)
  - Or file (less efficient)
  - On local disk (faster) or network file server (riskier)
- Storage configuration (alternatives):
  - disk = ['phy:vg/domu_root,hda1,w']
  - disk = ['phy:sda5,hda1,w']
  - disk = ['file://path/to/file,hda1,2']
- Can define multiple disks in one disk line (see documentation)

Configuring domU — Part 4

- Network interfaces:
  - By default bridged to physical network interfaces
  - Can set up additional (inside physical machine) bridges
    - Then route traffic to virtual machines
    - Through a firewall running on dom0 (or in another domU)
- Network configuration
  - vif = [’bridge=xenbr0’]
  - domU can be multihomed if desired (see documentation)

Starting domU

- Create Xen configuration file
- Install base operating system for domU onto disk area
- xm start debian_unstable

/etc/xen/debian_unstable

```
name = "debian_unstable"
builder=’linux’
kernel = "/boot/xenu-linux-2.6-standard"

ncpus = 1
memory = 128
disk = [ ’phy:r5/debian_unstable,sda1,w’ ]
vif = [ ’bridge=br-sv’ ]

root = "/dev/sda1 ro"
```

Managing domUs

- xm list
- xm console DOMU (eg, debian_unstable)
- xm pause
- xm unpause
- Log into domU and shut it down
- domU considers network interface, disk, etc to be “real”
- Much like managing any other host

42 Hosts in 1U

- Can it be done? Need:
  - Processors: 1-2 physical CPU(s), dual/quad core
  - Memory: 42 * 128MB = 5376MB RAM
  - Storage: 42 * 4GB = 168GB disk space
  - Network: 1-4 gigE NICs
- Even doubling memory and disk requirements is not impractical
- Possible hardware:
  - HP DL360 G5 (1U)
    - 2 * dual-core Intel CPUs, 8GB RAM, 4 * 146GB SAS, dual GigE
  - Sun Sunfire X4100 (1U)
    - 2 * dual-core AMD CPUs, 8GB RAM, 4 * 146GB SAS, quad GigE
- Both those servers can take more RAM (max 32GB)
- DL360 G5 will take quad-core CPUs

42 Hosts in 1U

- Can it be done? Need:
  - Processors: 1-2 physical CPU(s), dual/quad core
  - Memory: 42 * 128MB = 5376MB RAM
  - Storage: 42 * 4GB = 168GB disk space
  - Network: 1-4 gigE NICs
- Even doubling memory and disk requirements is not impractical
- Possible hardware:
  - HP DL360 G5 (1U)
    - 2 * dual-core Intel CPUs, 8GB RAM, 4 * 146GB SAS, dual GigE
  - Sun Sunfire X4100 (1U)
    - 2 * dual-core AMD CPUs, 8GB RAM, 4 * 146GB SAS, quad GigE
- Both those servers can take more RAM (max 32GB)
- DL360 G5 will take quad-core CPUs

Suitability — Part 1

- Works best for “mostly idle” VMs sharing a machine
- Most useful if hardware is under utilised
  - Possibly you can expand your hardware to achieve this
- CPU bound tasks particularly problematic
  - Xen has no CPU usage limitation
  - One CPU-bound VM will dominate
  - Multi-CPU/multi-core setups help
- I/0 bound tasks also problematic
  - Perhaps you can add more disk paths or NICs
  - Or maybe you need a cluster

Suitability — Part 1

- Works best for “mostly idle” VMs sharing a machine
- Most useful if hardware is under utilised
  - Possibly you can expand your hardware to achieve this
- CPU bound tasks particularly problematic
  - Xen has no CPU usage limitation
  - One CPU-bound VM will dominate
  - Multi-CPU/multi-core setups help
- I/0 bound tasks also problematic
  - Perhaps you can add more disk paths or NICs
  - Or maybe you need a cluster

Suitability — Part 1

- Works best for “mostly idle” VMs sharing a machine
- Most useful if hardware is under utilised
  - Possibly you can expand your hardware to achieve this
- CPU bound tasks particularly problematic
  - Xen has no CPU usage limitation
  - One CPU-bound VM will dominate
  - Multi-CPU/multi-core setups help
- I/0 bound tasks also problematic
  - Perhaps you can add more disk paths or NICs
  - Or maybe you need a cluster

Suitability — Part 2

- Common uses:
  - Network glue: Recursive DNS, Authoritive DNS, Tacacs, ...
  - Lightly-used Web applications
  - Cold-standby for production machines
  - Development and test environments
  - Per-customer hosts (web, PBX, ...)
- Many of these could be installed on one native host
  - But only at the expense of more complicated management
  - Virtual machines gives you “one task: one host” with less cost
  - Virtual machines let you use most appropriate OS for each task
- Xen supports virtual machine migration
  - Move running VM to different physical hardware
  - Some others support this too

Issues and Risk Management

- Concentration at single point of failure
  - But service probably should have been clustered anyway
  - Bring up “identical” VM for service at each POP
  - Or load balanced cluster across two boxes hosting VMs
- More (virtual) hosts to manage
  - You need automated management tools
  - Pay attention to other talks today!
- Virtual machine overhead
  - Multiple copies of kernel and libraries in RAM
  - Multiple OS installations on disk
  - No worse than physical hosts
    - ★ But temptation to create more virtual machines

That’s All Folks!

- Virtual machine technology is being widely deployed now
- Used properly gives you better utilisation and reliability
- Xen commonly used for Linux hosts

- Questions?

- Slides: http://www.naos.co.nz/talks/42-hosts-in-1u/