# Sun™ xVM Hypervisor

Gary Pennington
Solaris Kernel Engineer
April 24, 2008

# Agenda

- Hypervisors 101
- Introduction to Sun™ xVM Hypervisor
- Use Cases
- Using the hypervisor
  - Control domain: booting, services, tools
  - Guest domains: creation, booting
  - Debugging
- Futures

# Hypervisors 101

- Provides a “Virtual Machine”
- Not new – VM/370 over 30 years ago
- Controls hardware – memory/cpu/io devices
- Schedules cpus/memory/io rate
- May emulate real devices
- For x86/x64 multiple choices available:
  - Xen
  - VMWare
  - MSFT Virtual Server
  - Others

# Para vs. Full Virtualization

- Full Virtualization (HVM):
  - Runs binary image of “metal” OS
  - Must emulate i/o devices
  - Can be slow
  - Need help from hardware
  - May use trap and emulate or rewriting
- Para-virtualization
  - Runs OS ported to special arch
  - Uses generic “virtual” device drivers
  - Can be more efficient since it is hypervisor-aware

# Full Virtualization (HVM)

- Some operating systems have not been paravirtualized
  - Microsoft, older Solaris, older Linux, OS/2 (!), ...
- New processor features to enable full virtualization
  - Intel VT and AMD-V
    - Needs to be enabled by the BIOS, so having the right CPU may not be enough
  - Trap to the hypervisor for “unsafe” instructions, memory access, etc.
    - Hypervisor emulates some effects, uses device emulation for others

# Agenda

- Hypervisors 101
- Introduction to Sun™ xVM Hypervisor
- Use Cases
- Using the hypervisor
  - Control domain: booting, services, tools
  - Guest domains: creation, booting
  - Debugging
- Futures

# What is Sun™ xVM hypervisor?

- An open source hypervisor
- A port of Solaris to run on the hypervisor
- A set of control tools for the hypervisor
- A set of support tools for running other operating systems on the hypervisor under the direction of Solaris

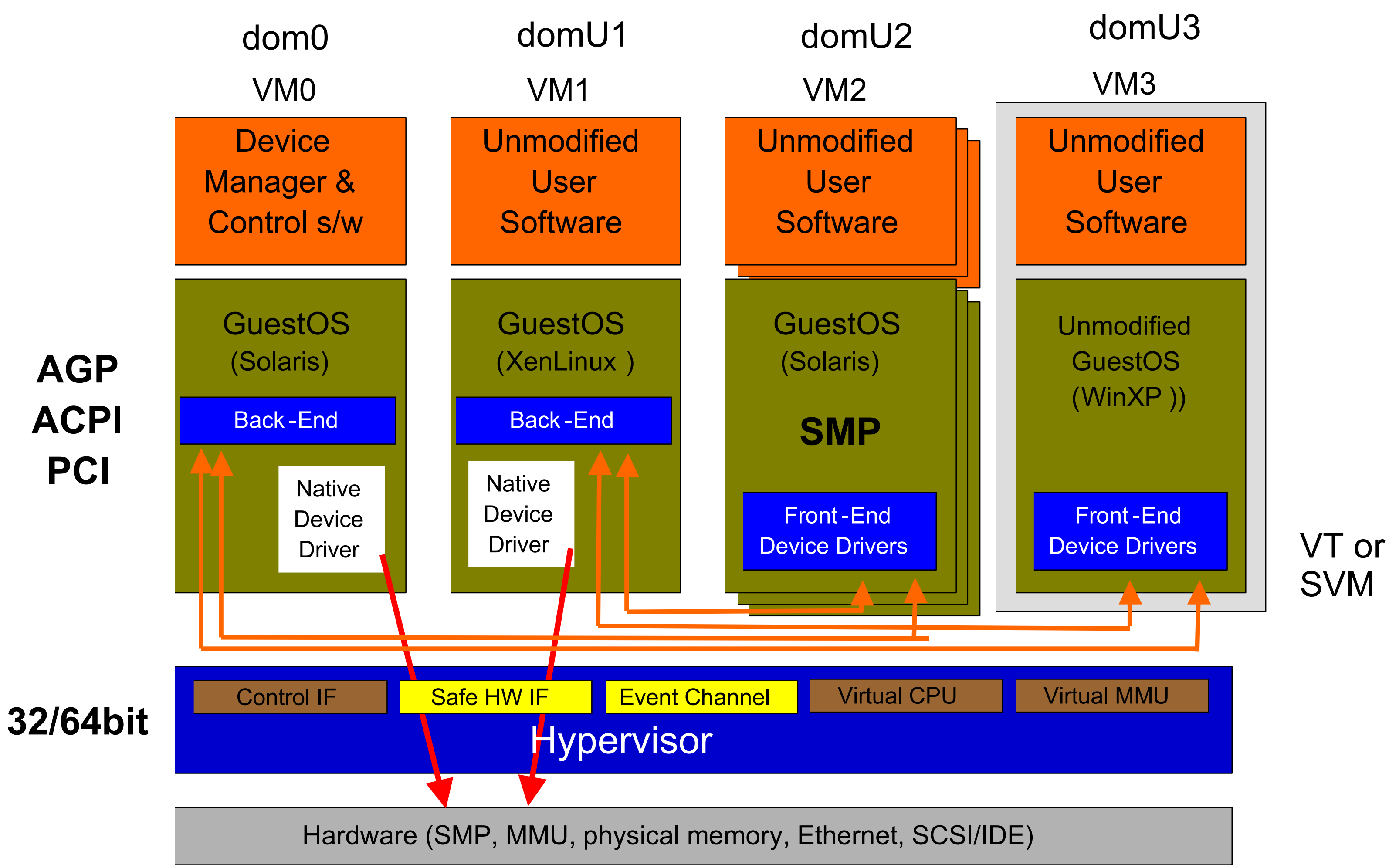
# Open source hypervisor technology

- Originally developed at the University of Cambridge, England
  - Licensed under the GPLv2 and LGPL
  - XenSource (now Citrix): a start-up created by the original developers of the project to commercialize the results
- Significant contributions from Intel, AMD, IBM, HP, Fujitsu, *and more*
- Mostly x86, but also available on PPC and Itanium
- Now at version 3.1.3 (3.1.4-rc8)

# Hypervisor Design Principles and Goals

- Existing applications and binaries must run unmodified
- Support for multi-process, multi-application application environments
  - Permit complex server configurations to be virtualized within a single guest OS instance
- Paravirtualization (PV) enables high performance and strong isolation between domains
  - Particularly on uncooperative architectures (x86)
- Support up to 100 active VM instances on modern servers
- Live migration of VM instances between servers

# Sun™ xVM Architecture



# Key Capabilities

- Checkpoint/restart and live migration
  - Managed provisioning
  - Grid operations: virtual platform
- Multiple OSes running simultaneously
  - Solaris, Linux, Windows
  - No longer a boot-time decision
- Special purpose kernels
  - JVM, drivers, filesystems, ...

# Agenda

- Hypervisors 101
- Introduction to Sun™ xVM Hypervisor
- Use Cases
- Using hypervisor
  - Control domain: booting, services, tools
  - Guest domains: creation, booting
  - Debugging
- Futures

# Use Cases (Enterprise)

- Single node consolidation/test system
- Multi (many) node virtual infrastructure
  - Windows, Linux, Solaris Consolidation
  - Application Grids
  - e.g. Oracle's datacenters
- Utility Computing
  - Amazon EC2
- Virtual Desktop environments
  - Call centers (DT)
- Quick roll out/re-provision/disaster recovery
- Virtual appliance deployment

# Use Cases (Developers)

- Good for:
  - Develop and test:
    - Fast turn-around time (shutdown and reboot)
    - User-level code
    - Installation
    - General kernel components
  - Older Solaris, Microsoft, Linux, ...
  - “Network in a box”
  - Sharing canned system configurations
- Clone and snapshot of zvols
  - Quickly produce multiple identical guest domains
  - Quickly return to a known stable state

# Agenda

- Hypervisors 101
- Introduction to Sun™ xVM Hypervisor
- Use Cases
- Using hypervisor
  - Control domain: booting, services, tools
  - Guest domains: creation, booting
  - Debugging
- Futures

# Using xVM: Booting the control domain

- Grub loads the hypervisor, kernel and boot archive:

```
title Solaris xVM
kernel$ /boot/$ISADIR/xen.gz
module$ /platform/i86xpv/kernel/$ISADIR/unix
  /platform/i86xpv/kernel/$ISADIR/unix
module$ /platform/i86pc/$ISADIR/boot_archive
```

- Hypervisor:
  - Initializes, probes hardware, etc.
  - Creates dom0 environment around the kernel and boot archive
  - Jumps to dom0 kernel
- Note:
  - Extended Grub syntax to allow expansion of environment specific tokens (`kernel$`, `module$`, `$ISADIR`)
  - Boot archive is separated into 32 bit and 64 bit

# Using xVM: Serial Consoles

- If you want to see hypervisor output over a serial line, edit the kernel$ line:

```
title Solaris xVM
kernel$ /boot/$ISADIR/xen.gz console=com1 com1=9600,8n1
module$ /platform/i86xpv/kernel/$ISADIR/unix
  /platform/i86xpv/kernel/$ISADIR/unix -B console=hypervisor
module$ /platform/i86pc/$ISADIR/boot_archive
```

# Using xVM: dom0 services

- `svc:/system/xvm/store:default`
  - File-based database used to store configuration of known domains
- `svc:/system/xvm/xend:default`
  - Long running daemon used by administrative tools to communicate with the hypervisor
  - Performs much of the work of creating guest domains, migration, etc.
- `svc:/system/xvm/console:default`
  - Mediates access to guest domain consoles (badly)
- `svc:/system/xvm/domains:default`
  - Automatically creates and destroys guest domains at service start/stop time (typically system boot/shutdown)

# Using xVM: dom0 tools (1)

- `xm`
  - Low-level xVM specific command to query the state of the hypervisor, create domains, manipulate configuration, etc.

```
shocks# xm start x1
shocks# xm list
Name                  ID   Mem VCPUs      State   Time(s)
Domain-0               0   984     2     r-----    810.3
x1                     2  1023     1     r-----      9.1
shocks# xm console x1
...
x1 console login: root
Password:
Last login: Sat Sep  8 02:02:28 on console
Sep  8 18:00:13 x1 login: ROOT LOGIN /dev/console
Sun Microsystems Inc.   SunOS 5.11      matrix-build-2007-08-21 October 2007
#
```

# Using xVM: dom0 tools (2)

- `virsh`
  - hypervisor agnostic command to query the state of the hypervisor, create domains, manipulate configuration, etc.
    - Only xVM support for now, but Logical Domains coming
  - Built on `libvirt`

```
: shocks#; virsh dominfo x1
Id:             2
Name:           x1
UUID:           b0bece06-8bee-085b-b657-dd642da0daa0
OS Type:        linux
State:          blocked
CPU(s):         1
CPU time:       98.7s
Max memory:     1048576 kB
Used memory:    1047540 kB
: shocks#;
```

# Using xVM: dom0 tools (3)

- `virt-install`
  - Facilitate the installation of para-virtual and HVM guests
  - Interactive or command line arguments
  - Install off media (DVD), from an ISO, or over NFS
  - Built on `libvirt`

**Solaris PV Guest**

```
virt-install -n solarisPV --paravirt -r 1024 \
  --nographics -f /export/solarisPV/root.img -s 16 \
  -l /ws/matrix-gate/public/isos/72-0910/solarisdvd.iso
```

**Solaris HVM Guest**

```
virt-install -n solarisHVM --hvm -r 1024 --vnc \
  -f /export/solarisHVM/root.img -s 16 \
  -c /ws/matrix-gate/public/isos/72-0910/solarisdvd.iso
```

# Using xVM: dom0 tools (3) cont'd

- `virt-install`

  **WinXP HVM Guest**

```
# virt-install -n winxp --hvm -r 1024 --vnc \
  -f /export/winxp/root.img -s 16 -c /windows/media.iso
```

- Set the VNC password property in xend's SMF configuration before starting a HVM domain which uses VNC

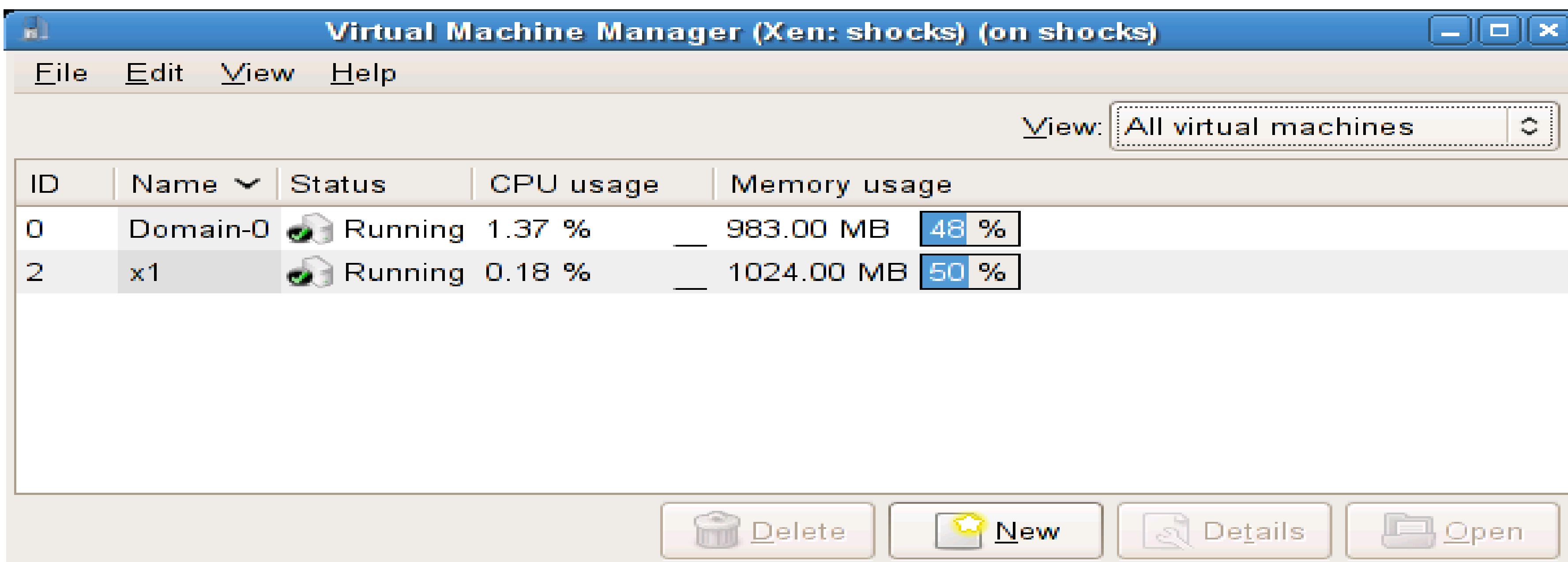
```
# svccfg -s xvm/xend setprop \
        config/vncpasswd = astring: \"somepwd\"
# svcadm refresh xvm/xend; svcadm restart xvm/xend
```

- If remotely displaying the VNC session, you must also set the `vnc-listen` property

```
# svccfg -s xvm/xend setprop \
        config/vnc-listen = astring: \"0.0.0.0\"
# svcadm refresh xvm/xend; svcadm restart xvm/xend
```

# Using xVM: dom0 tools (4)

- `virt-manager` (not yet integrated)
  - Gnome desktop application for managing virtual machines
  - Single physical system focus
  - Built on `libvirt`
  - `http://opensolaris.org/os/project/jds/`



# Beyond dom0

- xVM Ops Center
  - Combining virtualization and management
  - See http://www.sun.com/software/products/xvmopscenter/ir
- OpenxVM
  - See https://openxvm.dev.java.net/

# Using xVM: Guest domain creation

- Create new guest domains using virt-install
  - Normal Solaris install for the guest domain, including jumpstart, etc.
  - Linux and HVM (e.g. Windows) install still something of a work in progress
- Acquire guest domain disk images and configuration from others
  - Save the need for everyone to run through the installation
  - Guest domains have relatively small configuration matrix
  - Clone and snapshot of ZFS volumes a powerful management tool

# Using xVM: Debugging the hypervisor

- printf() is your friend (or not)
- If the hypervisor panics, Solaris can usually take a dump
  - Includes the hypervisor image, which looks like a kernel module in the dump

# Using xVM: Debugging dom0

- Typical OpenSolaris tools work well
  - `mdb`, `kmdb`, `dtrace`
- The hypervisor console can be used to send a 'break' signal to domains
  - Type '`^A^A^A`' at the hypervisor console to start
  - Particularly useful for dom0
- Dom0 tools
  - Many are written in python
  - /usr/lib/python2.4/vendor-packages/xen/
  - Edit and restart xend smf service

# Using xVM: Debugging domU

- Dom0 tools can be used to:
  - Send a 'break' signal to guest domains:
    - `xm sysrq b <domain>`
  - Dump the image of a guest domain, for use with `mdb`:
    - `xm dump-core <domain> <dump-file>`
    - `mdb <dump-file>`

# When things go wrong

- Log files in `/var/log/xen`:
  - `xend.log` – logging and backtraces from the long running daemon
  - `xpvd-event.log` – logs from backend device creation, removal, etc.

# Agenda

- Hypervisors 101
- Introduction to Sun™ xVM Hypervisor
- Use Cases
- Using hypervisor
  - Control domain: booting, services, tools
  - Guest domains: creation, booting
  - Debugging
- Futures

# Past Solaris Work

- snv_75
  - Xen 3.0.4
  - Libvirt 0.2.3
  - Virt-install 0.103.0
- snv_81
  - PV net drivers
- snv_85
  - Xen 3.1.2
  - Libvirt 0.4.0
- snv_87
  - PV disk drivers

# PV drivers for Solaris 10

- No PV version of Solaris 10
  - IO performance using emulated hardware (IDE and RTL8139) is poor
- Provide PV disk and network drivers for older Solaris releases
- Bundled in a future Solaris 10 update
- Performance of PV drivers in HVM domain looks similar to that of a fully PV guest domain

# Windows PV drivers

- Planned for 2008

# Future Solaris work

Projects that are still in early development/ porting phase

- blktap
- virt-install 0.300
- FMA for xVM
- Security for xVM
- Crossbow
- Live CD and Image Packaging System (IPS)

# Finding out more

- OpenSolaris community
  - xen-discuss@opensolaris.org
  - http://opensolaris.org/os/community/xen
  - irc://irc.oftc.net/solaris-xen
- OpenxVM Community
  - http://www.openxvm.org/

開
放
的
열린
مفتوح
libre
मुक्त
ಮುಕ್ತ
livre
libero
ముక్త
开放的
açık
open
nyílt
פתוח
オープン
livre
ανοικτό
offen
otevřený
öppen
открытый
வெளிப்படை

# open

USE IMPROVE EVANGELIZE

## Thank you!

Gary Pennington
Solaris Kernel Engineer
http://blogs.sun.com/garypen

"open" artwork and icons by chandan:
http://blogs.sun.com/chandan