September 2011 - oi_151a
Status Update + KVM Demo

Alasdair Lumsden (al@everycity.co.uk)
OpenIndiana Project Lead
& Founder of EveryCity Managed Hosting

openindiana

everycity

Happy Anniversary OpenIndiana!

- oi_151a released on Sept 14th 2011, 1 year after our official announcement

- This is our first release based on Illumos, the open source fork of OpenSolaris' OS/Net (Operating System + Networking - the kernel and core userland components)

Major Feature Enhancements

- Single biggest new feature is KVM (Kernel Mode Virtual Machine) support for Intel chips - this allows running high performance guest VMs such as Windows and Linux
- Illumos brings many enhancements, including:
  - Many open replacements for closed components such as libc internationalisation, locales, and many userland tools
  - Many bug fixes including for ZFS
  - Updated Terminal definitions
  - New whois tool
  - Grub support for large sector disks
  - Zone administration enhancements
  - ZFS aclmode property added
  - iostat -E now shows serial number for non-Sun branded disks
  - SCSI UNMAP support in COMSTAR
  - Plus over 300 commits over the past 12 months

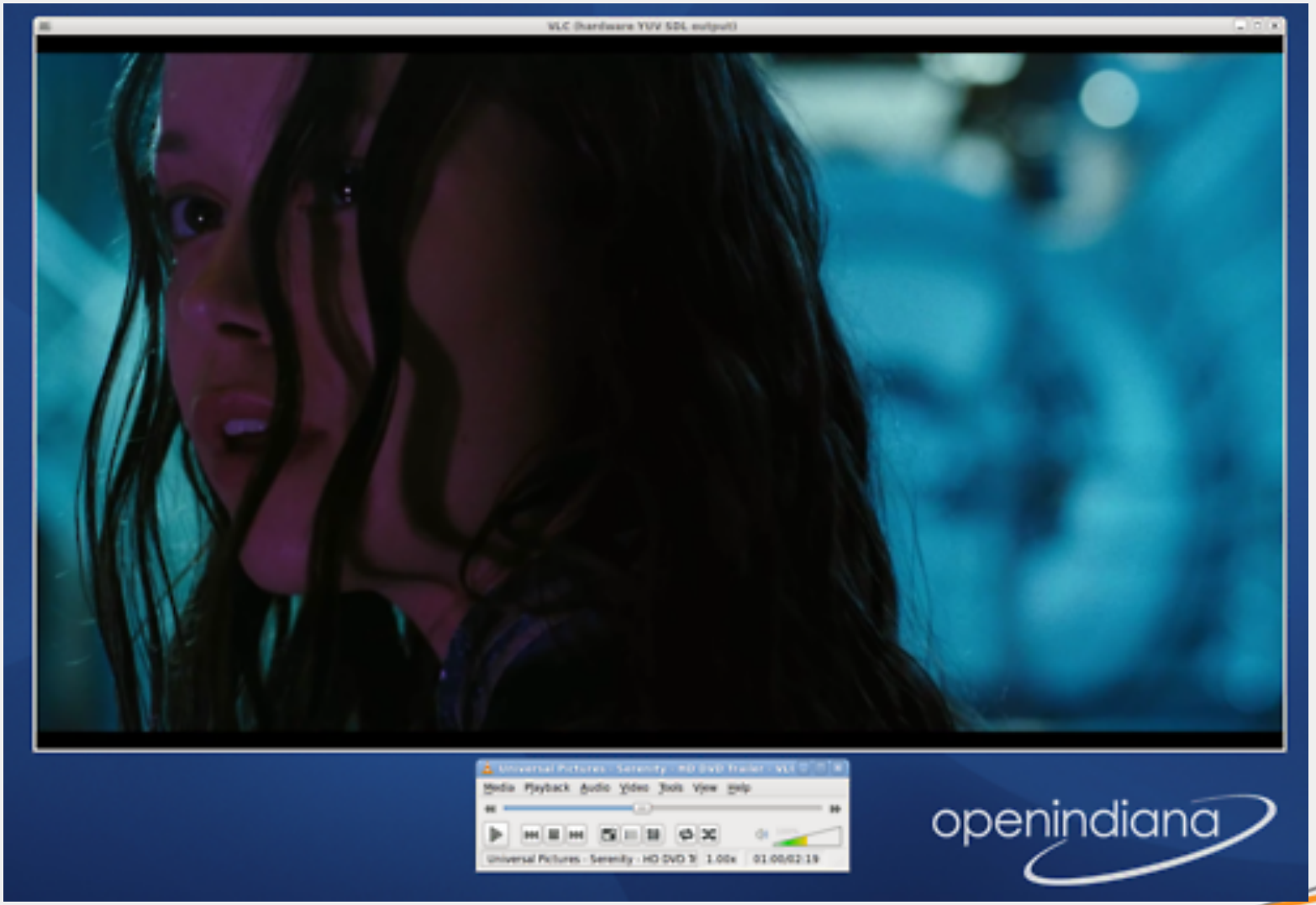
Spec Files Extra

- A new software repository called Spec Files Extra is available
- Two repos, pkg.openindiana.org/sfe and /sfe-encumbered
- /sfe-encumbered contains multimedia software that may be covered by software patents in the USA (thankfully the UK & Europe doesn't have "pure" software patents), including goodies such as:
  - vlc (VideoLan multimedia player)
  - ffmpeg (media transcoder)
  - mplayer (multimedia player)
  - mpd (music player daemon)
  - lame, xvid, x264, faac, faad, mpg123 and many other multimedia codec libraries

Spec Files Extra - cont...

- pkg.openindiana.org/sfe contains:
  - Arora Web Browser (webkit based)
  - AbiWord
  - Scribus
  - Postgresql 9
  - Blender
  - Inkscape
  - Wine
  - Graphviz
  - Qt
  - Stellarium
  - ImageMagick
  - Plus over 200 packages

VLC in Action



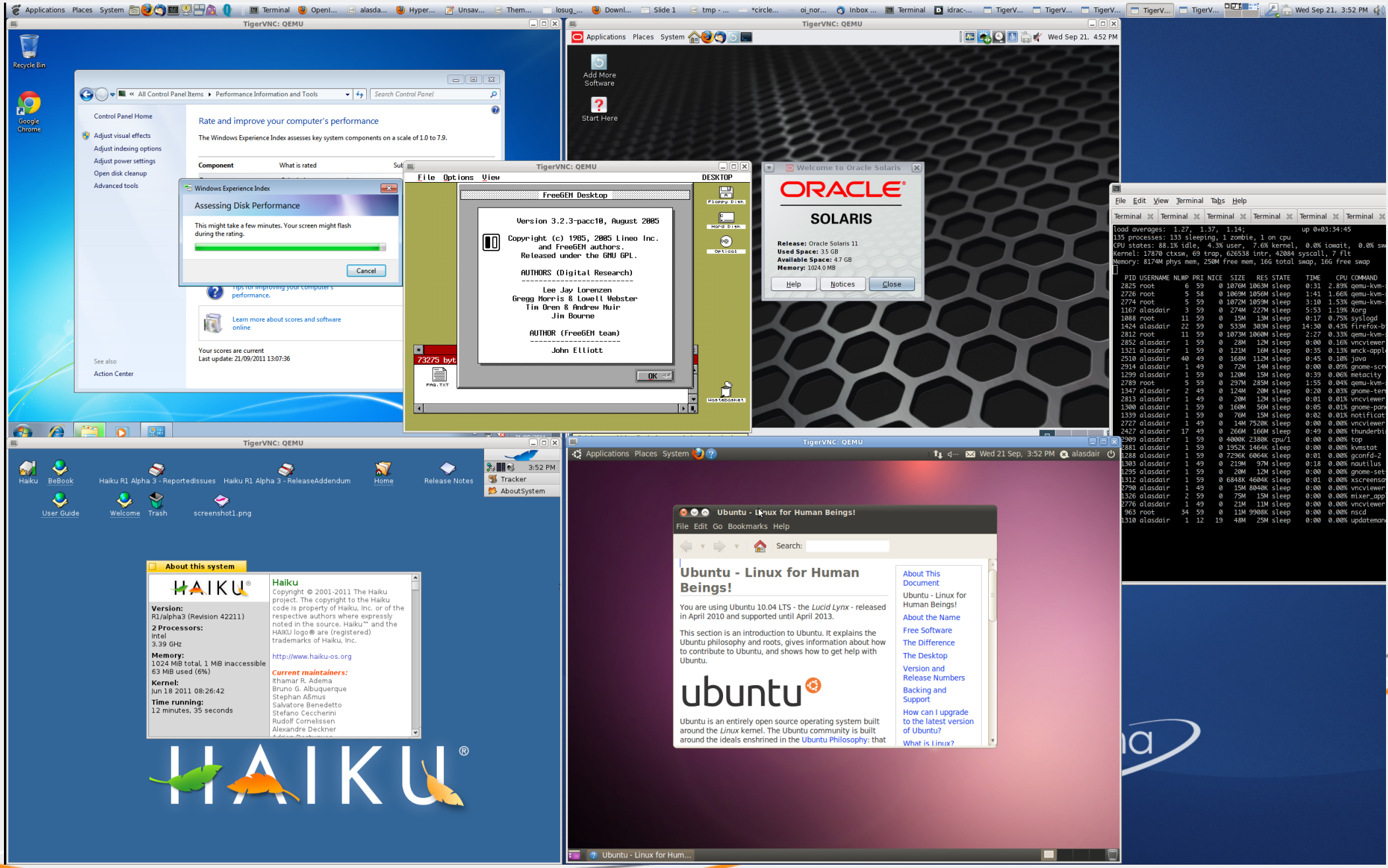
Stable Release

- oi_151a is the basis of our stable release, which we hope to launch by the end of the year.
- To produce the stable release, we are taking oi_151a, and applying security and bug fixes for a wide range of software on the operating system.
- Once the stable branch is released, it will receive regular security updates and bug fixes, free of charge, making OpenIndiana sutable for use on production servers facing the internet.

Future Dev Releases

- In parallel to the stable release which focuses on security and bug fixes only, we are working on our next dev release
- Our aim is to update as much software as we can in the operating system, bringing much of the software up to date and in-line with equivalent Linux/BSD distributions
- We are also transitioning from the Sun Studio compiler to GCC 4.6.1 and we are also looking carefully at LLVM/Clang

KVM - Kernel-based Virtual Machine



KVM - Kernel-based Virtual Machine

- KVM is a Linux kernel extension that allows user-space programs to utilise the hardware virtualisation features of modern CPUs.
- QEMU, a PC emulator, can use KVM to provide a high performance virtualisation platform similar to Xen or VMWare
- Joyent, a US based Cloud Computing company famous for their use of OpenSolaris, have ported KVM to Illumos for use in their cloud operating system, SmartOS
- As OpenIndiana uses Illumos, we have added support for KVM, giving OpenIndiana it's first major unique feature not present in OpenSolaris/Solaris 11

KVM - Usage Scenarios

- KVM's main strengths are performance and manageability, making it ideally suited to server virtualisation workloads
- KVM supports "virtio" high performance drivers for Disk and Networking - signed Windows drivers are available from the Fedora project, and Joyent in the near future
- For graphics, KVM can use SDL or VNC - this is functional but VirtualBox can provide a better desktop experience

KVM - Benefits on Illumos vs Linux

- Illumos has key benefits for running KVM
- Managing disk images via ZFS provides exceptional management - snapshots, rollbacks, clones, backups
- ZFS also provides unparalleled data integrity through checksums and fancy RAID options (RAIDZ2, RAIDZ3, etc)
- The Crossbow virtual network stack gives each VM its own virtualised network adapter, which includes the security benefits crossbow offers
- DTrace can provide insight into the inner workings of the guests in a way that just isn't possible with any other virtualisation platform

KVM in Zones, a jail within a jail

- There have been security holes found in QEMU in the past, which when exploited have allowed a guest access to the host system
- By combining QEMU/KVM with Solaris Zones, you get an additional security layer - an attacker that successfully exploits QEMU will simply find themselves in an empty Zone!
- This severely limits the risks of running multi-tenanted KVM servers
- Joyent with SmartOS have implemented QEMU/KVM management as a branded Zone - giving them the same management interface for Windows/Linux guests as they have with SmartOS zones

Installing QEMU & KVM on oi_151a

To use KVM, you will need a recent Intel CPU supporting the vmx extension - you can check this with isainfo -v:

```
64-bit amd64 applications
    vmx xsave pclmulqdq aes sse4.2 sse4.1 ssse3 popcnt tscp cx16 sse3 sse2
    sse fxsr mmx cmov amd_sysc cx8 tsc fpu
```

To Install QEMU & KVM on oi_151a, simply:

```
pfexec pkg install system/qemu \
                   system/qemu/kvm \
                   driver/i86pc/kvm
```

Preparing a filesystem

To create a ZFS filesystem for your first guest, simply:

```
pfexec zfs create -p -V 10G -s \
    rpool/kvm/guest/disk0
```

"-p" = create parent systems (eg kvm/guest)
"-V" = create a block volume
"-s" = make volume sparse (don't pre-allocate space)

The above command creates a volume which is then accessible via /dev/zvol/dsk/rpool/kvm/guest/disk0

Creating a vnic for use with KVM

If you'd like your guest to have networking support, you can create a vnic for it, by simply doing:

pfexec dladm create-vnic -l e1000g0 guest0

(You'll want to adjust e1000g0 to match your primary network interface name)

Start qemu-kvm

Now we should be ready to start QEMU:

```
VNIC=guest0
MAC=`dladm show-vnic $VNIC | grep ^$VNIC | awk '{print $4}'`
ISO=/export/home/alasdair/ubuntu.iso
pfexec qemu-kvm -enable-kvm \
    -vnc 0.0.0.0:1 \
    -smp 2 \
    -m 1024 \
    -no-hpet \
    -localtime \
    -boot order=d \
    -drive file=/dev/zvol/rdsk/rpool/kvm/guest/disk0,if=ide,index=0 \
    -drive file=$ISO,media=cdrom,if=ide,index=2 \
    -net nic,vlan=0,name=net0,model=e1000,macaddr=$MAC \
    -net vnic,vlan=0,name=net0,ifname=$VNIC,macaddr=$MAC
```

What's next for KVM?

- AMD support is being actively worked on by Illumos developers within the community and progress has been made - guests are able to boot and run on AMD processors
- Once this work is complete, we aim to integrate it into OpenIndiana
- We are evaluating importing Joyent's KVM branded zones support to allow management of KVM guests via the zoneadm/zonecfg commands
- We are also evaluating adding libvirt support

Questions?

Obtaining OpenIndiana: www.openindiana.org
Community Support: #openindiana on irc.freenode.net
Community Mailing List: http://openindiana.org/mailman