



# **Xen™ Core Roadmap**

Keir Fraser, XenSource

# Current status



- Feature freeze for 3.0.3
  - Fixing bugs, measuring performance
- 3.0.3 will branch in next few weeks
- Development will continue in -unstable for next release (3.0.4)
- Likely timeline:
  - 3.0.4 in December/January
  - 3.0.5 in April/May

# Feature roadmap



- Lots of work on HVM, I/O, upstreaming...
- **NUMA**
- **Kexec/kdump**
- **32-bit guests on 64-bit hypervisor**
- Super-page support (hugetlbfs)
- Power management
- Improvements for live relocation
- Performance analysis and tuning

# HVM



- Save/restore, live relocation
  - Working with Intel
- Improve I/O emulation performance and integration
  - Move qemu from dom0 -> stub domain
- Paravirtualized I/O
  - Already in the tree for 3.0.3

# I/O



- Driver domains are back
  - Work on better tools integration
- Driver isolation
  - Support IOMMUs from IBM, Intel, AMD
- Performance analysis & improvements
  - E.g., changes to netfront/back protocol
- Smart I/O devices
  - Virtualization-aware NICs

# Guest OSes



- Ongoing work to port Solaris
- Working to merge Xen support upstream into kernel.org Linux
  - Progress made at OLS / kernel summit
  - Working with IBM to integrate 'paravirt\_ops' generic virtualization API into the kernel
- Perhaps time to move away from the Linux sparse tree?
  - Kernel interfaces are stable -- no need for tight binding between hypervisor and kernel versions and builds

# Super-page support (x86)

- Currently all guest page mappings are 4kB
- This leads to TLB pressure for some application workloads (e.g., databases)
- Want to make larger contiguous memory chunks selectively available to guests
- ...and allow them to map those contiguous chunks as super pages
- Requires OS integration (Linux hugetlbfs)

# Power management



- Currently no access to frequency/voltage scaling, or deep-sleep states
- Linux has a library of routines for interfacing to ACPI and various CPU families
  - We'd like to leverage this as far as possible
- Most of the Linux code can be run as-is in domain0, with a few caveats/exceptions:
  - Require a 1:1 mapping from VCPUs to physical CPUs
  - Xen needs to be notified of CPU frequency changes (unless the CPU features constant-rate TSC)
  - Xen must be involved in suspending CPUs

# Live relocation



- Now fairly stable with new shadow2 code
- But more to be done:
  - What happens if a live relo fails 'in the middle'?
  - Need to be able to restart the suspended original domain (not currently possible in stop-and-copy phase)
- Memory requirements for ballooned guests:
  - Currently allocate maximum-possible memory at target machine, and free up unused memory later
  - Consider a 4GB domain ballooned down to 1GB
  - Better: allocate memory pages as we discover that they are in use

# Performance



- Performance and scalability work
  - Time is right for some close attention
  - 1-4 socket systems the priority
  - Optimizations for bigger systems must not hurt smaller ones (they often help)
    - Onus is on submitter to demonstrate
    - (Patches that clearly hurt larger systems should be rejected too)
- Good performance tools now available
  - s/w perf counters, xen oprofile, tracebuf etc