Xen and the Art of Virtualization

Ian Pratt

University of Cambridge and Founder of XenSource Inc.

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

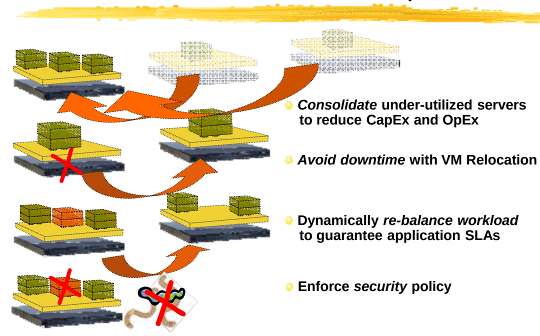
Outline

- Virtualization Overview
- Xen Today : Xen 2.0 Overview
- Architecture
- Performance
- Live VM Relocation
- Xen 3.0 features (Q3 2005)
- Research Roadmap

Virtualization Overview

- Single OS image: Virtuozo, Vservers, Zones
  - Group user processes into resource containers
  - Hard to get strong isolation
- Full virtualization: VMware, VirtualPC, QEMU
  - Run multiple unmodified guest OSes
  - Hard to efficiently virtualize x86
- Para-virtualization: UML, Xen
  - Run multiple guest OSes ported to special arch
  - Arch Xen/x86 is very close to normal x86

Virtualization in the Enterprise



Xen Today : 2.0 Features

- Secure isolation between VMs
- Resource control and QoS
- Only guest kernel needs to be ported
  - All user-level apps and libraries run unmodified
  - Linux 2.4/2.6, NetBSD, FreeBSD, Plan9
- Execution performance is close to native
- Supports the same hardware as Linux x86
- Live Relocation of VMs between Xen nodes

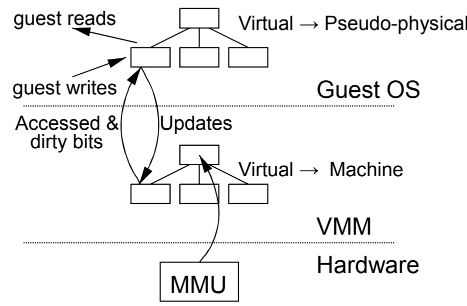
Para-Virtualization in Xen

- Arch xen_x86 : like x86, but Xen hypercalls required for privileged operations
  - Avoids binary rewriting
  - Minimize number of privilege transitions into Xen
  - Modifications relatively simple and self-contained
- Modify kernel to understand virtualised env.
  - Wall-clock time vs. virtual processor time
    - Xen provides both types of alarm timer
  - Expose real resource availability
    - Enables OS to optimise behaviour

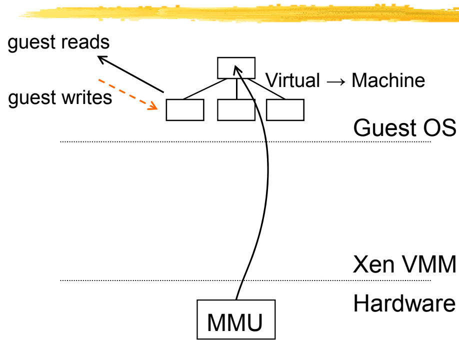
x86 CPU virtualization

- Xen runs in ring 0 (most privileged)
- Ring 1/2 for guest OS, 3 for user-space
  - GPF if guest attempts to use privileged instr
- Xen lives in top 64MB of linear addr space
  - Segmentation used to protect Xen as switching page tables too slow on standard x86
- Hypercalls jump to Xen in ring 0
- Guest OS may install ‘fast trap’ handler
  - Direct user-space to guest OS system calls
- MMU virtualisation: shadow vs. direct-mode

MMU Virtualizion : Shadow-Mode



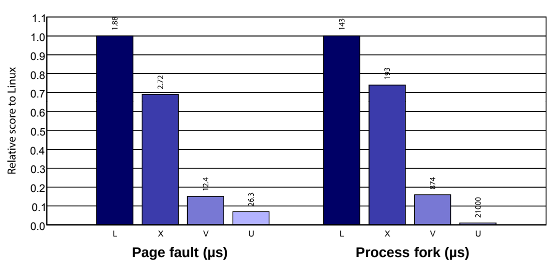
MMU Virtualization : Direct-Mode



Para-Virtualizing the MMU

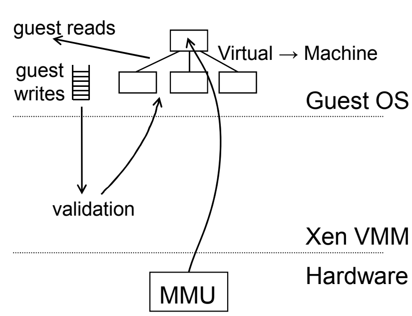
- Guest OSes allocate and manage own PTs
  - Hypercall to change PT base
- Xen must validate PT updates before use
  - Allows incremental updates, avoids revalidation
- Validation rules applied to each PTE:
  1. Guest may only map pages it owns*
  2. Pagetable pages may only be mapped RO
- Xen traps PTE updates and emulates, or ‘unhooks’ PTE page for bulk updates

MMU Micro-Benchmarks

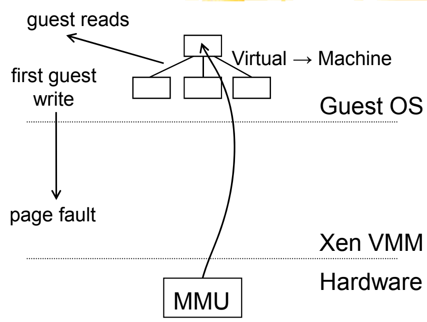


lmbench results on Linux (L), Xen (X), VMWare Workstation (V), and UML (U)

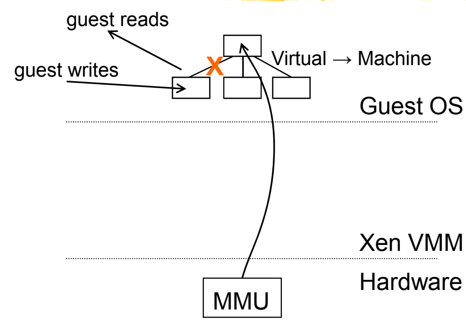
Queued Update Interface (Xen 1.2)



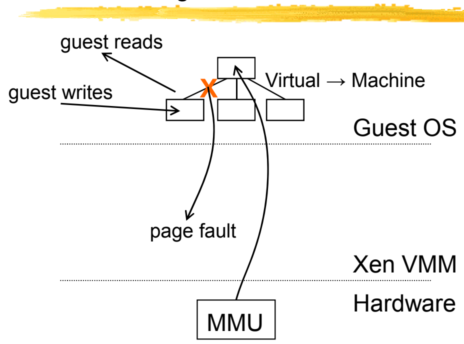
Writeable Page Tables : 1 – write fault



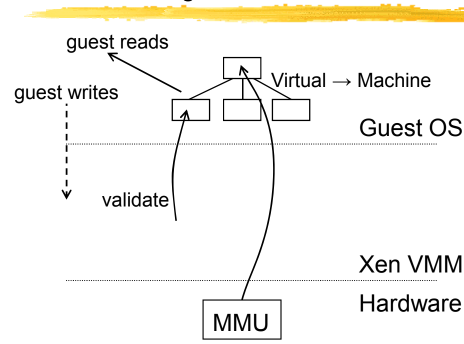
Writeable Page Tables : 2 - Unhook



Writeable Page Tables : 3 - First Use



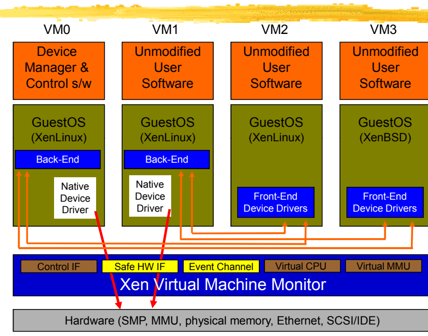
Writeable Page Tables : 4 – Re-hook



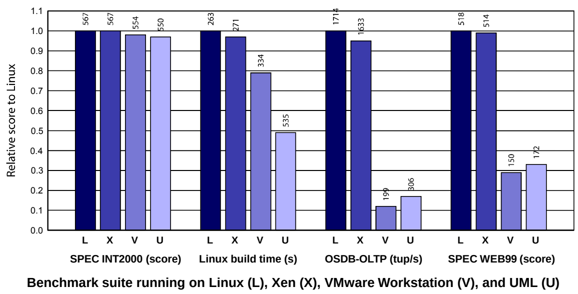
I/O Architecture

- Xen *IO-Spaces* delegate guest OSes protected access to specified h/w devices
  - Virtual PCI configuration space
  - Virtual interrupts
- Devices are virtualised and exported to other VMs via *Device Channels*
  - Safe asynchronous shared memory transport
  - ‘Backend’ drivers export to ‘frontend’ drivers
  - Net: use normal bridging, routing, iptables
  - Block: export any blk dev e.g. sda4,loop0,vg3

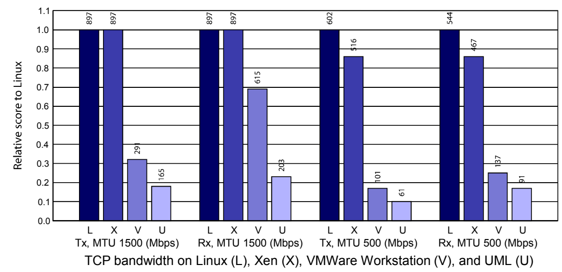
Xen 2.0 Architecture



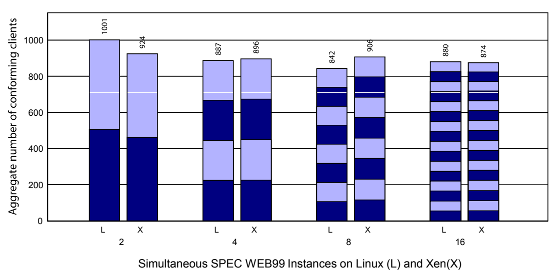
System Performance



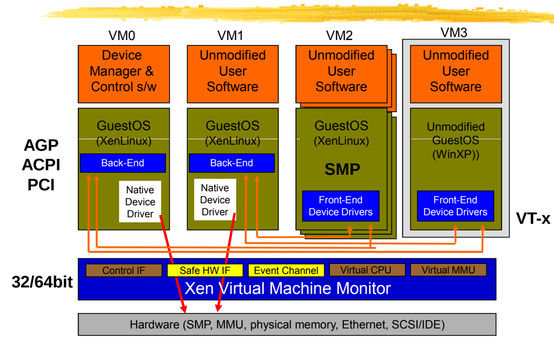
TCP results



Scalability



Xen 3.0 Architecture



3.0 Headline Features

- AGP/DRM graphics support
- Improved ACPI platform support
- Support for SMP guests
- x86_64 support
- Intel VT-x support for unmodified guests
- Enhanced control and management tools
- IA64 and Power support, PAE

x86_64

- Intel EM64T and AMD Opteron
- Requires different approach to x86 32 bit:
  - Can't use segmentation to protect Xen from guest OS kernels as no segment limits
  - Switch page tables between kernel and user
    - Not too painful thanks to Opteron TLB flush filter
  - Large VA space offers other optimisations
- Current design supports up to 8TB mem

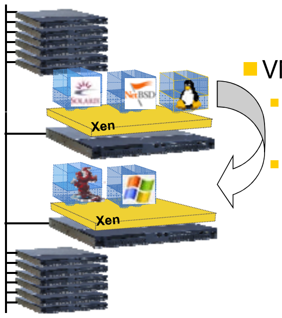
SMP Guest OSes

- Takes great care to get good performance while remaining secure
- Paravirtualized approach yields many important benefits
  - Avoids many virtual IPIs
  - Enables 'bad preemption' avoidance
  - Auto hot plug/unplug of CPUs
- SMP scheduling is a tricky problem
  - Strict gang scheduling leads to wasted cycles

VT-x / Pacifica

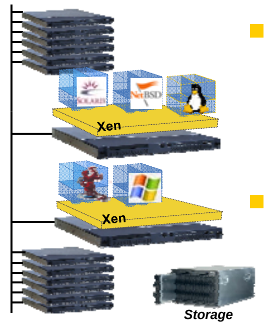
- Will enable Guest OSes to be run without paravirtualization modifications
  - E.g. Windows XP/2003
- CPU provides traps for certain privileged instrs
- Shadow page tables used to provide MMU virtualization
- Xen provides simple platform emulation
  - BIOS, Ethernet (e100), IDE and SCSI emulation
- Install paravirtualized drivers after booting for high-performance IO

VM Relocation : Motivation



- VM relocation enables:
  - High-availability
    - Machine maintenance
  - Load balancing
    - Statistical multiplexing gain

Assumptions

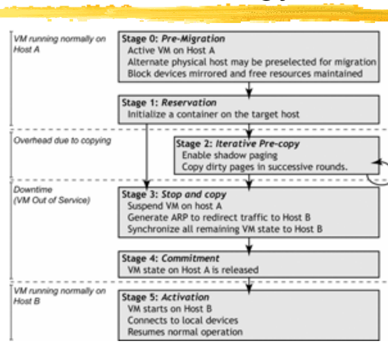


- Networked storage
  - NAS: NFS, CIFS
  - SAN: Fibre Channel
  - iSCSI, network block dev
  - drdb network RAID
- Good connectivity
  - common L2 network
  - L3 re-routeing

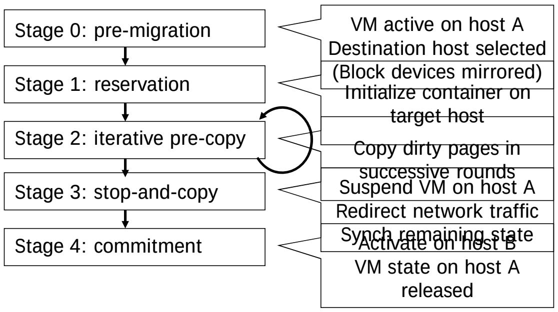
Challenges

- VMs have lots of state in memory
- Some VMs have soft real-time requirements
  - E.g. web servers, databases, game servers
  - May be members of a cluster quorum
  - ➔ **Minimize down-time**
- Performing relocation requires resources
  - ➔ **Bound and control resources used**

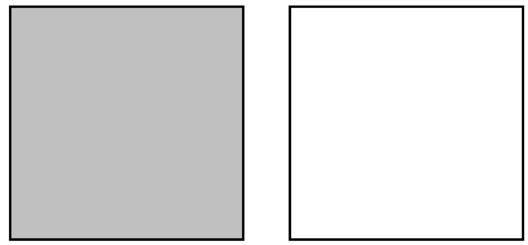
Relocation Strategy



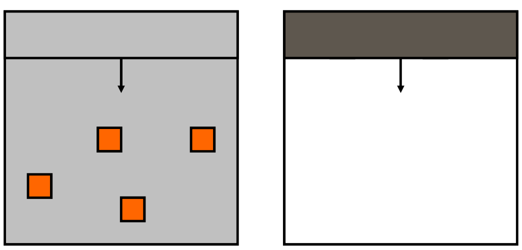
Relocation Strategy



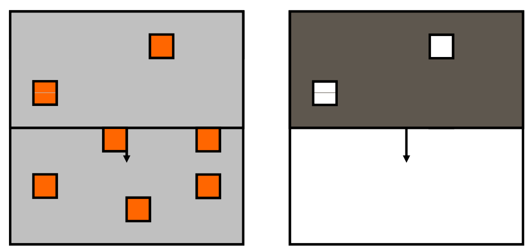
Pre-Copy Migration: Round 1



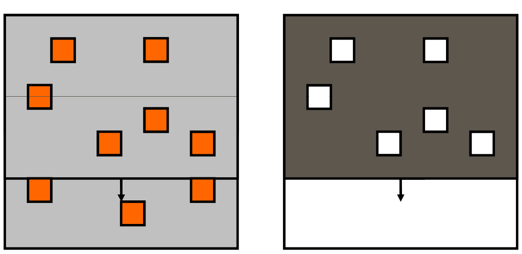
Pre-Copy Migration: Round 1



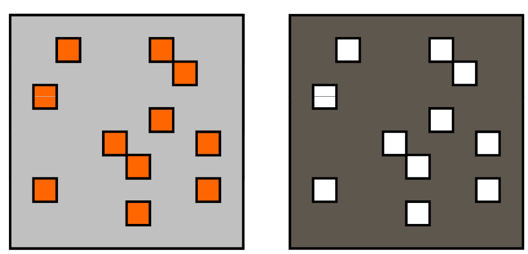
Pre-Copy Migration: Round 1



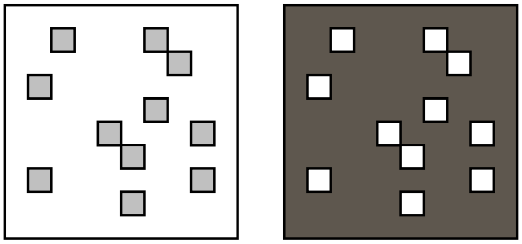
Pre-Copy Migration: Round 1



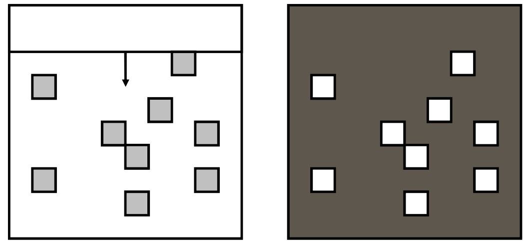
Pre-Copy Migration: Round 1



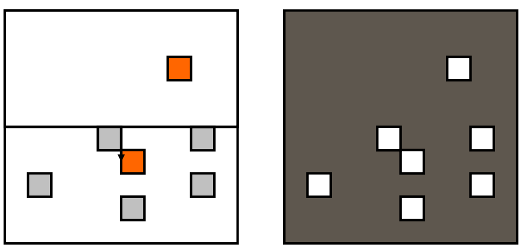
Pre-Copy Migration: Round 2



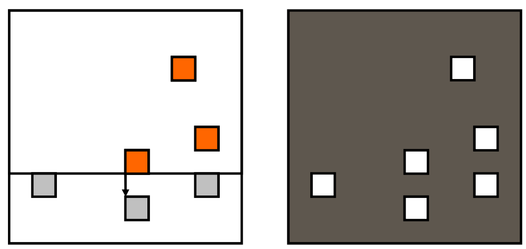
Pre-Copy Migration: Round 2



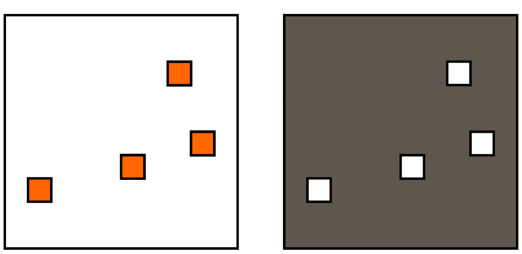
Pre-Copy Migration: Round 2



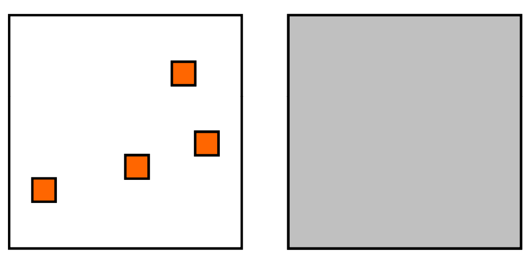
Pre-Copy Migration: Round 2



Pre-Copy Migration: Round 2



Pre-Copy Migration: Final



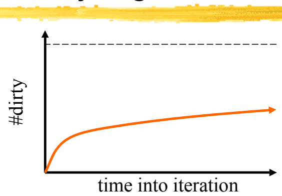
Writable Working Set

- Pages that are dirtied must be re-sent
  - Super hot pages
    - e.g. process stacks; top of page free list
  - Buffer cache
  - Network receive / disk buffers
- Dirtying rate determines VM down-time
  - Shorter iterations → less dirtying → ...
- App. ‘phase changes’ may knock us back

Writable Working Set

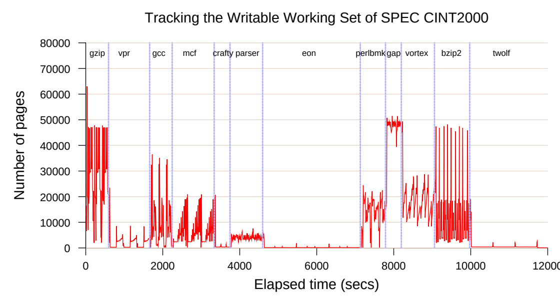
- Set of pages written to by OS/application
- Pages that are dirtied must be re-sent
  - Hot pages
    - E.g. process stacks
    - Top of free page list (works like a stack)
  - Buffer cache
  - Network receive / disk buffers

Page Dirtying Rate

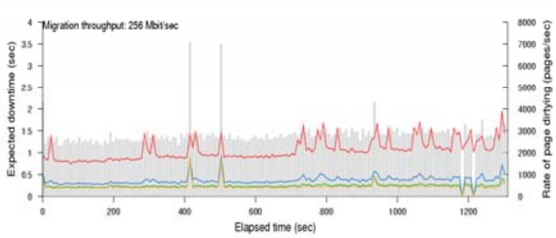


- Dirtying rate determines VM down-time
  - Shorter iters → less dirtying → shorter iters
  - Stop and copy final pages
- Application ‘phase changes’ create spikes

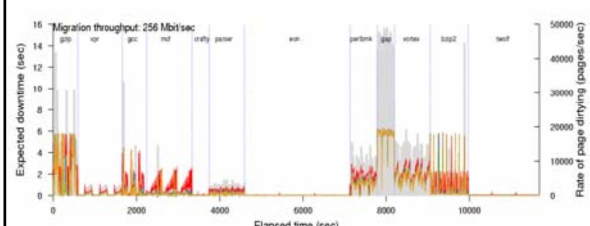
Writable Working Set



PostgreSQL/OLTP down-time



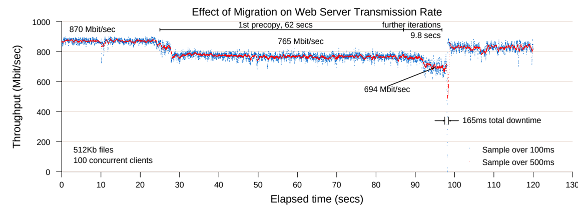
CINT2000 down-time



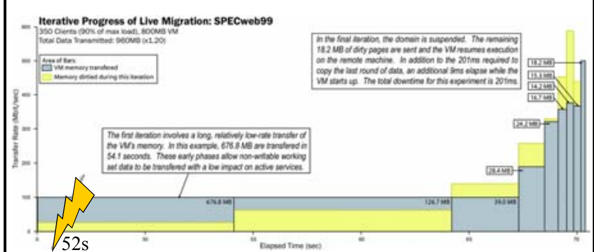
Rate Limited Relocation

- Dynamically adjust resources committed to performing page transfer
  - Dirty logging costs VM ~2-3%
  - CPU and network usage closely linked
- E.g. first copy iteration at 100Mb/s, then increase based on observed dirtying rate
  - Minimize impact of relocation on server while minimizing down-time

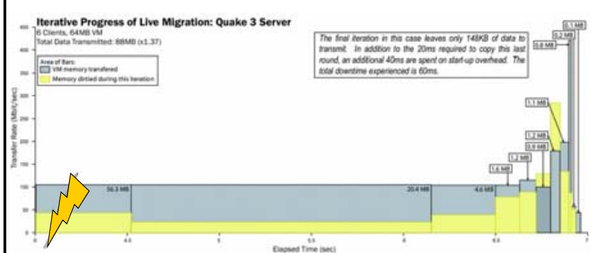
Web Server Relocation



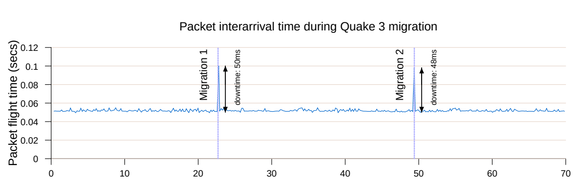
Iterative Progress: SPECWeb



Iterative Progress: Quake3



Quake 3 Server relocation



Relocation Transparency

	Requires VMM support	Changes to OS	Able to adapt	QoS
Transparent	yes	none	no	harder
Assisted	yes	minor	yes	harder
Self	no	significant	Yes	easier

Relocation Notification

- Opportunity to be more co-operative
  - Quiesce background tasks to avoid dirtying
- Doesn't help if the foreground task is the cause of the problem…
- Self-relocation allows the kernel fine-grained control over trade-off
  - Decrease priority of difficult processes

Extensions

- Cluster load balancing
  - Pre-migration analysis phase
  - Optimization over coarse timescales
- Evacuating nodes for maintenance
  - Move easy to migrate VMs first
- Storage-system support for VM clusters
  - Decentralized, data replication, copy-on-write
- Wide-area relocation
  - IPSec tunnels and CoW network mirroring

Research Roadmap

- Software fault tolerance
  - Exploit deterministic replay
- System debugging
  - Lightweight checkpointing and replay
- VM forking
  - Lightweight service replication, isolation
- Secure virtualization
  - Multi-level secure Xen

Xen Supporters

Operating System and Systems Management

Hardware Systems

Acquired by

Platforms & I/O

* Logos are registered trademarks of their owners

Conclusions

- Xen is a complete and robust GPL VMM
- Outstanding performance and scalability
- Excellent resource control and protection
- Live relocation makes seamless migration possible for many real-time workloads
- http://xensource.com

Thanks!

- The Xen project is hiring, both in Cambridge, Palo Alto and New York

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

- ian@xensource.com

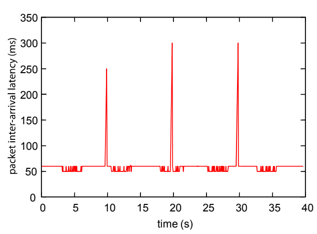
Backup slides

Research Roadmap

- Whole distributed system emulation
  - I/O interposition and emulation
  - Distributed watchpoints, replay
- VM forking
  - Service replication, isolation
- Secure virtualization
  - Multi-level secure Xen
- XenBIOS
  - Closer integration with the platform / BMC
- Device Virtualization

Isolated Driver VMs

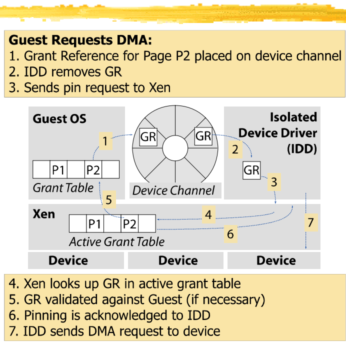
- Run device drivers in separate domains
- Detect failure e.g.
  - Illegal access
  - Timeout
- Kill domain, restart
- E.g. 275ms outage from failed Ethernet driver



Segmentation Support

- Segmentation req'd by thread libraries
  - Xen supports virtualised GDT and LDT
  - Segment must not overlap Xen 64MB area
  - NPT TLS library uses 4GB segs with –ve offset!
    - Emulation plus binary rewriting required ☹
- x86_64 has no support for segment limits
  - Forced to use paging, but only have 2 prot levels
  - Xen ring 0; OS and user in ring 3 w/ PT switch
    - Opteron's TLB flush filter CAM makes this fast

Device Channel Interface



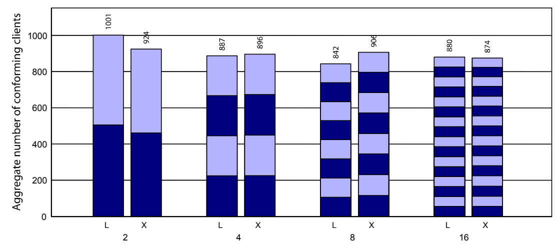
Live migration for clusters

- Pre-copy approach: VM continues to run
- 'lift' domain on to shadow page tables
  - Bitmap of dirtied pages; scan; transmit dirtied
  - Atomic 'zero bitmap & make PTEs read-only'
- Iterate until no forward progress, then stop VM and transfer remainder
- Rewrite page tables for new MFNs; Restart
- Migrate MAC or send unsolicited ARP-Reply
- Downtime typically 10's of milliseconds
  - (though very application dependent)

Scalability

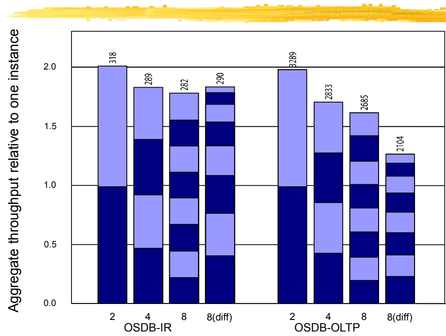
- Scalability principally limited by Application resource requirements
  - several 10’s of VMs on server-class machines
- Balloon driver used to control domain memory usage by returning pages to Xen
  - Normal OS paging mechanisms can deflate quiescent domains to <4MB
  - Xen per-guest memory usage <32KB
- Additional multiplexing overhead negligible

Scalability



Simultaneous SPEC WEB99 Instances on Linux (L) and Xen(X)

Resource Differentation



Simultaneous OSDB-IR and OSDB-OLTP Instances on Xen