Xen and the Art of Virtualization (2003)

Gabriele Genovese, Erik Koci

January 17, 2025

Introduction

- High-performance and resource-managed Virtual Machine Monitor (VMM).
- Commercial type-1 bare-metal hypervisor.
- Initially developed by researchers at the University of Cambridge, it has now been acquired by Citrix Systems.

Goals of Xen

- Isolation between VMs to prevent performance interference.
- Support for multiple commodity OSes (e.g., Linux, Windows).
- Minimal performance overhead compared to bare-metal systems.
- Scalability to host up to 100 VMs on modern hardware.

OS subsection	**# lines**	
	Linux	**XP**
Architecture-independent	78	1299
Virtual network driver	484	–
Virtual block-device driver	1070	–
Xen-specific (non-driver)	1363	3321
Total	**2995**	**4620**
(**Portion of total x86 code base**	**1.36%**	**0.04%**)

Figure: The simplicity of porting commodity OSes to Xen.

Paravirtualization

Memory Management	
Segmentation	Cannot install fully-privileged segment descriptors and cannot overlap with the top end of the linear address space.
Paging	Guest OS has direct read access to hardware page tables, but updates are batched and validated by the hypervisor. A domain may be allocated discontiguous machine pages.
CPU	
Protection	Guest OS must run at a lower privilege level than Xen.
Exceptions	Guest OS must register a descriptor table for exception handlers with Xen. Aside from page faults, the handlers remain the same.
System Calls	Guest OS may install a 'fast' handler for system calls, allowing direct calls from an application into its guest OS and avoiding indirecting through Xen on every call.
Interrupts	Hardware interrupts are replaced with a lightweight event system.
Time	Each guest OS has a timer interface and is aware of both 'real' and 'virtual' time.
Device I/O	
Network, Disk, etc.	Virtual devices are elegant and simple to access. Data is transferred using asynchronous I/O rings. An event mechanism replaces hardware interrupts for notifications.

Figure: The paravirtualized x86 interface

- Requires modifications to the guest OS for better performance.
- Improved performance compared to full virtualization.

Architecture of Xen

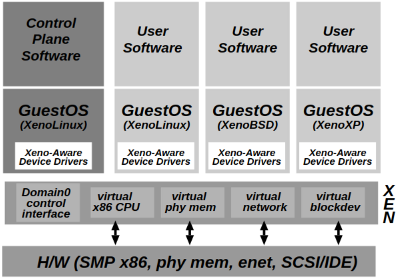

Figure: The structure of the Xen hypervisor

- **Domain0**: domain responsible for managing hardware and other VMs.
- **GuestOS**: domains running guest OSes.
- **Communication**: hypercalls and events are used between domains.

Performance Evaluation

- Comparison with stand-alone Linux, VMware Workstation and User-Mode Linux (UML).
- The results presented are the median of seven trials.
- Superior network and I/O performance compared to other VMMs.
- Scales effectively with multiple VMs.

Performance of computationally-intensive applications

Measurement made with one VM and one core:

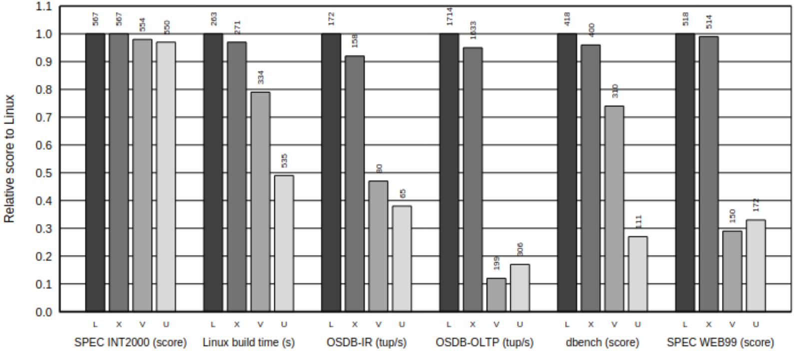

Figure: Relative performance of native Linux (L), XenoLinux (X), VMware workstation 3.2 (V) and User-Mode Linux (U).

Microbenchmarks for targeting particular subsystems

In 24/37 microbenchmarks, XenoLinux performs similarly to native Linux.

Config	null call	null I/O	stat	open close	slct TCP	sig inst	sig hndl	fork proc	exec proc	sh proc
L-SMP	0.53	0.81	2.10	3.51	23.2	0.83	2.94	143	601	4k2
L-UP	0.45	0.50	1.28	1.92	5.70	0.68	2.49	110	530	4k0
Xen	0.46	0.50	1.22	1.88	5.69	0.69	1.75	**198**	**768**	**4k8**
VMW	0.73	0.83	1.88	2.99	11.1	1.02	4.63	874	2k3	10k
UML	24.7	25.1	36.1	62.8	39.9	26.0	46.0	21k	33k	58k

Figure: Processes - times in μs

Config	0K File create	0K File delete	10K File create	10K File delete	Mmap lat	Prot fault	Page fault
L-SMP	44.9	24.2	123	45.2	99.0	1.33	1.88
L-UP	32.1	6.08	66.0	12.5	68.0	1.06	1.42
Xen	32.5	5.86	68.2	13.6	**139**	1.40	**2.73**
VMW	35.3	9.3	85.6	21.4	620	7.53	12.4
UML	130	65.7	250	113	1k4	21.8	26.3

Figure: File & VM system latencies in μs

Config	2p 0K	2p 16K	2p 64K	8p 16K	8p 64K	16p 16K	16p 64K
L-SMP	1.69	1.88	2.03	2.36	26.8	4.79	38.4
L-UP	0.77	0.91	1.06	1.03	24.3	3.61	37.6
Xen	**1.97**	**2.22**	**2.67**	**3.07**	**28.7**	**7.08**	39.4
VMW	18.1	17.6	21.3	22.4	51.6	41.7	72.2
UML	15.5	14.6	14.4	16.3	36.8	23.6	52.0

Figure: Context switching - times in μs

	TCP MTU 1500 TX	TCP MTU 1500 RX	TCP MTU 500 TX	TCP MTU 500 RX
Linux	897	897	602	544
Xen	897 (-0%)	897 (-0%)	516 (-14%)	467 (-14%)
VMW	291 (-68%)	615 (-31%)	101 (-83%)	137 (-75%)
UML	165 (-82%)	203 (-77%)	61.1(-90%)	91.4(-83%)

Figure: Bandwidth in Mb/s

Scalability

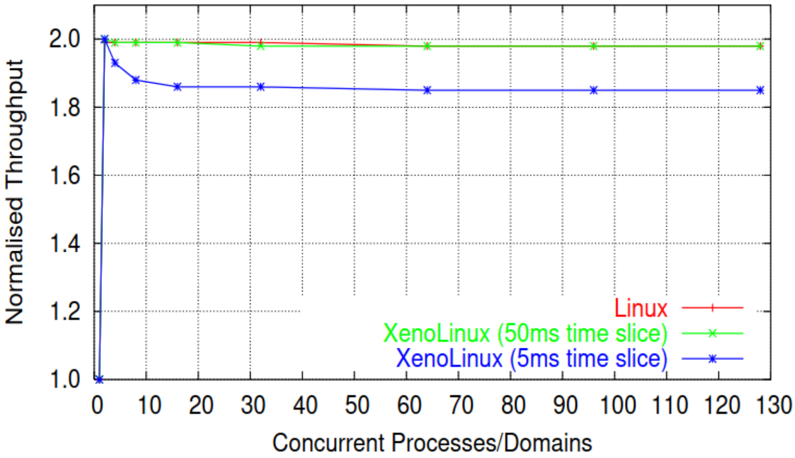

Figure: Performance of running concurrently on 1-128 domains

Future Work and Conclusion

- Planned features:
  - Universal buffer caching for improved data sharing.
  - Copy-on-write for virtual block devices.
- Xen provides an efficient platform for hosting secure and isolated virtual environments.
- Its paravirtualization approach balances performance and isolation.

A recent comparison of Xen

#	Performance Parameter	Benchmark	Benchmark Operation	Xen	KVM	Proxmox
1	Response Efficiency	Apache	Serving static pages	*LB	**BS***	
2	CPU Throughput	John the Ripper	Blowfish	LB	**BS**	
			Traditional DES		LB	**BS**
			MD5	LB	**BS**	
3	Memory	RAMspeed	Integer (Average)	LB	**BS**	
			Float (Average)		**BS**	LB
4	Cache Performance	CacheBench	Read	LB	**BS**	
			Write	LB	**BS**	
			Read/modify/write	LB	**BS**	
5	File System Performance	IOzone	Disk read	**BS**	LB	
			Disk write	**BS**	LB	
6	Application Performance	LZMA	File compression	**BS**	LB	
		Sqlite	Database inserts	**BS**		LB

* LB represents Least Performance and BS represents Best Performance

Figure: Taken from Algarni, Sultan Abdullah, et al. "Performance evaluation of Xen, KVM, and Proxmox hypervisors" (2018)

Thank you for your attention!