

Xenologie oder wie man einen Plastikmainframe baut

Alexander Schreiber <als@thangorodrim.de>

<http://www.thangorodrim.de/>

Chemnitzer Linux-Tage 2006

*I think there is a world market for maybe five computers.
– Thomas Watson, chairman of IBM, 1943.*

Übersicht

- 1 Einführung
- 2 Xen
- 3 Erfahrungen
- 4 Ende

Virtualisierung – Einführung

- mehrere virtuelle Maschine auf einer physischen Maschine
- 60er Jahre: IBM Mainframes
- VMWare: Virtualisierung auf x86
- Xen: Open Source Virtualisierung

Virtualisierung – Einführung

- mehrere virtuelle Maschine auf einer physischen Maschine
- 60er Jahre: IBM Mainframes
- VMWare: Virtualisierung auf x86
- Xen: Open Source Virtualisierung

Virtualisierung – Einführung

- mehrere virtuelle Maschine auf einer physischen Maschine
- 60er Jahre: IBM Mainframes
- VMWare: Virtualisierung auf x86
- Xen: Open Source Virtualisierung

Virtualisierung – Einführung

- mehrere virtuelle Maschine auf einer physischen Maschine
- 60er Jahre: IBM Mainframes
- VMWare: Virtualisierung auf x86
- Xen: Open Source Virtualisierung

Virtualisierung – Einführung

- mehrere virtuelle Maschine auf einer physischen Maschine
- 60er Jahre: IBM Mainframes
- VMWare: Virtualisierung auf x86
- Xen: Open Source Virtualisierung

Virtualisierung – Warum?

- isolierte Testsysteme ohne extra Hardware
- transparente Updates
- Konsolidierung
- dynamische Lastverteilung

Virtualisierung – Warum?

- isolierte Testsysteme ohne extra Hardware
- transparente Updates
- Konsolidierung
- dynamische Lastverteilung

Virtualisierung – Warum?

- isolierte Testsysteme ohne extra Hardware
- transparente Updates
- Konsolidierung
- dynamische Lastverteilung

Virtualisierung – Warum?

- isolierte Testsysteme ohne extra Hardware
- transparente Updates
- Konsolidierung
- dynamische Lastverteilung

Virtualisierung – Kostenvorteile

- typischer Serverraum: viele Maschinen, wenig Auslastung
- mit Virtualisierung: wenige große Maschinen, gute Auslastung
- Einsparungspotential:
 - deutlich weniger Server
 - weniger Serverraumkapazität notwendig
 - weniger Energieverbrauch (Serverbetrieb & Kühlung)
 - weniger Serviceverträge

Virtualisierung – Kostenvorteile

- typischer Serverraum: viele Maschinen, wenig Auslastung
- mit Virtualisierung: wenige große Maschinen, gute Auslastung
- Einsparungspotential:
 - deutlich weniger Server
 - weniger Serverraumkapazität notwendig
 - weniger Energieverbrauch (Serverbetrieb & Kühlung)
 - weniger Serviceverträge

Virtualisierung – Kostenvorteile

- typischer Serverraum: viele Maschinen, wenig Auslastung
- mit Virtualisierung: wenige große Maschinen, gute Auslastung
- Einsparungspotential:
 - deutlich weniger Server
 - weniger Serverraumkapazität notwendig
 - weniger Energieverbrauch (Serverbetrieb & Kühlung)
 - weniger Serviceverträge

Virtualisierung – Kostenvorteile

- typischer Serverraum: viele Maschinen, wenig Auslastung
- mit Virtualisierung: wenige große Maschinen, gute Auslastung
- Einsparungspotential:
 - deutlich weniger Server
 - weniger Serverraumkapazität notwendig
 - weniger Energieverbrauch (Serverbetrieb & Kühlung)
 - weniger Serviceverträge

Virtualisierung – Kostenvorteile

- typischer Serverraum: viele Maschinen, wenig Auslastung
- mit Virtualisierung: wenige große Maschinen, gute Auslastung
- Einsparungspotential:
 - deutlich weniger Server
 - weniger Serverraumkapazität notwendig
 - weniger Energieverbrauch (Serverbetrieb & Kühlung)
 - weniger Serviceverträge

Virtualisierung – Kostenvorteile

- typischer Serverraum: viele Maschinen, wenig Auslastung
- mit Virtualisierung: wenige große Maschinen, gute Auslastung
- Einsparungspotential:
 - deutlich weniger Server
 - weniger Serverraumkapazität notwendig
 - weniger Energieverbrauch (Serverbetrieb & Kühlung)
 - weniger Serviceverträge

Virtualisierung – Kostenvorteile

- typischer Serverraum: viele Maschinen, wenig Auslastung
- mit Virtualisierung: wenige große Maschinen, gute Auslastung
- Einsparungspotential:
 - deutlich weniger Server
 - weniger Serverraumkapazität notwendig
 - weniger Energieverbrauch (Serverbetrieb & Kühlung)
 - weniger Serviceverträge

Virtualisierung – Kostenvorteile

- typischer Serverraum: viele Maschinen, wenig Auslastung
- mit Virtualisierung: wenige große Maschinen, gute Auslastung
- Einsparungspotential:
 - deutlich weniger Server
 - weniger Serverraumkapazität notwendig
 - weniger Energieverbrauch (Serverbetrieb & Kühlung)
 - weniger Serviceverträge

Xen – Übersicht

- Hypervisor-Technologie, sichere Isolation
- minimale Anpassungen an Guest-OS-Kernel
- mehrere HW-Plattformen, mehrere Betriebssysteme
- nahezu native Performance
- live Migration
- breite Hardware-Unterstützung
- unterstützt von Intel, AMD, IBM, HP, ...

Xen – Übersicht

- Hypervisor-Technologie, sichere Isolation
- minimale Anpassungen an Guest-OS-Kernel
- mehrere HW-Plattformen, mehrere Betriebssysteme
- nahezu native Performance
- live Migration
- breite Hardware-Unterstützung
- unterstützt von Intel, AMD, IBM, HP, ...

Xen – Übersicht

- Hypervisor-Technologie, sichere Isolation
- minimale Anpassungen an Guest-OS-Kernel
- mehrere HW-Plattformen, mehrere Betriebssysteme
- nahezu native Performance
- live Migration
- breite Hardware-Unterstützung
- unterstützt von Intel, AMD, IBM, HP, ...

Xen – Übersicht

- Hypervisor-Technologie, sichere Isolation
- minimale Anpassungen an Guest-OS-Kernel
- mehrere HW-Plattformen, mehrere Betriebssysteme
- nahezu native Performance
- live Migration
- breite Hardware-Unterstützung
- unterstützt von Intel, AMD, IBM, HP, ...

Xen – Übersicht

- Hypervisor-Technologie, sichere Isolation
- minimale Anpassungen an Guest-OS-Kernel
- mehrere HW-Plattformen, mehrere Betriebssysteme
- nahezu native Performance
- live Migration
- breite Hardware-Unterstützung
- unterstützt von Intel, AMD, IBM, HP, ...

Xen – Übersicht

- Hypervisor-Technologie, sichere Isolation
- minimale Anpassungen an Guest-OS-Kernel
- mehrere HW-Plattformen, mehrere Betriebssysteme
- nahezu native Performance
- live Migration
- breite Hardware-Unterstützung
- unterstützt von Intel, AMD, IBM, HP, ...

Xen – Übersicht

- Hypervisor-Technologie, sichere Isolation
- minimale Anpassungen an Guest-OS-Kernel
- mehrere HW-Plattformen, mehrere Betriebssysteme
- nahezu native Performance
- live Migration
- breite Hardware-Unterstützung
- unterstützt von Intel, AMD, IBM, HP, ...

Xen – Übersicht

- Hypervisor-Technologie, sichere Isolation
- minimale Anpassungen an Guest-OS-Kernel
- mehrere HW-Plattformen, mehrere Betriebssysteme
- nahezu native Performance
- live Migration
- breite Hardware-Unterstützung
- unterstützt von Intel, AMD, IBM, HP, ...

Xen - Technologie 1

- Vollvirtualisierung: unmodifiziertes Guest-OS
- auf x86 extrem schwer, einige privilegierte Befehle versagen stillschweigend ohne Trap
- aufwendige Emulation privilegierter Operation nötig, insbesondere bei I/O → Performanceverluste

Xen - Technologie 1

- **Vollvirtualisierung: unmodifiziertes Guest-OS**
- auf x86 extrem schwer, einige privilegierte Befehle versagen stillschweigend ohne Trap
- aufwendige Emulation privilegierter Operation nötig, insbesondere bei I/O → Performanceverluste

Xen - Technologie 1

- Vollvirtualisierung: unmodifiziertes Guest-OS
- auf x86 extrem schwer, einige privilegierte Befehle versagen stillschweigend ohne Trap
- aufwendige Emulation privilegierter Operation nötig, insbesondere bei I/O → Performanceverluste

Xen - Technologie 1

- Vollvirtualisierung: unmodifiziertes Guest-OS
- auf x86 extrem schwer, einige privilegierte Befehle versagen stillschweigend ohne Trap
- aufwendige Emulation privilegierter Operation nötig, insbesondere bei I/O → Performanceverluste

Xen - Technologie 2

- Paravirtualisierung: Guest-OS wird minimal angepaßt
- aber: Sourcecode des Guest-OS Kernels notwendig ...
- Guest-OS läuft auf virtueller Umgebung wo nötig (privilegierte Operationen → Hypervisor-Calls)
- Guest-OS sieht weitgehend echte Hardware
→ Performancegewinn
- Xen verwendet Segmentierung zur Isolation → kollidiert mit NPT TLS → Emulation & binary rewriting

Xen - Technologie 2

- Paravirtualisierung: Guest-OS wird minimal angepaßt
- aber: Sourcecode des Guest-OS Kernels notwendig ...
- Guest-OS läuft auf virtueller Umgebung wo nötig (privilegierte Operationen → Hypervisor-Calls)
- Guest-OS sieht weitgehend echte Hardware
→ Performancegewinn
- Xen verwendet Segmentierung zur Isolation → kollidiert mit NPT TLS → Emulation & binary rewriting

Xen - Technologie 2

- Paravirtualisierung: Guest-OS wird minimal angepaßt
- aber: Sourcecode des Guest-OS Kernels notwendig ...
- Guest-OS läuft auf virtueller Umgebung wo nötig (privilegierte Operationen → Hypervisor-Calls)
- Guest-OS sieht weitgehend echte Hardware
→ Performancegewinn
- Xen verwendet Segmentierung zur Isolation → kollidiert mit NPT TLS → Emulation & binary rewriting

Xen - Technologie 2

- Paravirtualisierung: Guest-OS wird minimal angepaßt
- aber: Sourcecode des Guest-OS Kernels notwendig ...
- Guest-OS läuft auf virtueller Umgebung wo nötig (privilegierte Operationen → Hypervisor-Calls)
- Guest-OS sieht weitgehend echte Hardware
→ Performancegewinn
- Xen verwendet Segmentierung zur Isolation → kollidiert mit NPT TLS → Emulation & binary rewriting

Xen - Technologie 2

- Paravirtualisierung: Guest-OS wird minimal angepaßt
- aber: Sourcecode des Guest-OS Kernels notwendig ...
- Guest-OS läuft auf virtueller Umgebung wo nötig (privilegierte Operationen → Hypervisor-Calls)
- Guest-OS sieht weitgehend echte Hardware
→ Performancegewinn
- Xen verwendet Segmentierung zur Isolation → kollidiert mit NPT TLS → Emulation & binary rewriting

Xen - Technologie 2

- Paravirtualisierung: Guest-OS wird minimal angepaßt
- aber: Sourcecode des Guest-OS Kernels notwendig ...
- Guest-OS läuft auf virtueller Umgebung wo nötig (privilegierte Operationen → Hypervisor-Calls)
- Guest-OS sieht weitgehend echte Hardware
→ Performancegewinn
- Xen verwendet Segmentierung zur Isolation → kollidiert mit NPT TLS → Emulation & binary rewriting

Xen – Plattformen

- Hardware: x86 (komplett), ppc, ia64 (beide in Arbeit)
- OS für dom0 (Managementdomain):
 - Linux 2.4/2.6 (Xen 3: nur 2.6)
 - NetBSD 3
- OS für DomU (Guest):
 - Linux 2.4/2.6 (Xen3: nur 2.6)
 - NetBSD 2/3 (noch nicht Xen 3)
 - FreeBSD 5 (noch nicht Xen 3)
 - Plan9 (noch nicht Xen 3)

Xen – Plattformen

- Hardware: x86 (komplett), ppc, ia64 (beide in Arbeit)
- OS für dom0 (Managementdomain):
 - Linux 2.4/2.6 (Xen 3: nur 2.6)
 - NetBSD 3
- OS für DomU (Guest):
 - Linux 2.4/2.6 (Xen3: nur 2.6)
 - NetBSD 2/3 (noch nicht Xen 3)
 - FreeBSD 5 (noch nicht Xen 3)
 - Plan9 (noch nicht Xen 3)

Xen – Plattformen

- Hardware: x86 (komplett), ppc, ia64 (beide in Arbeit)
- OS für dom0 (Managementdomain):
 - Linux 2.4/2.6 (Xen 3: nur 2.6)
 - NetBSD 3
- OS für DomU (Guest):
 - Linux 2.4/2.6 (Xen3: nur 2.6)
 - NetBSD 2/3 (noch nicht Xen 3)
 - FreeBSD 5 (noch nicht Xen 3)
 - Plan9 (noch nicht Xen 3)

Xen – Plattformen

- Hardware: x86 (komplett), ppc, ia64 (beide in Arbeit)
- OS für dom0 (Managementdomain):
 - Linux 2.4/2.6 (Xen 3: nur 2.6)
 - NetBSD 3
- OS für DomU (Guest):
 - Linux 2.4/2.6 (Xen3: nur 2.6)
 - NetBSD 2/3 (noch nicht Xen 3)
 - FreeBSD 5 (noch nicht Xen 3)
 - Plan9 (noch nicht Xen 3)

Xen – Plattformen

- Hardware: x86 (komplett), ppc, ia64 (beide in Arbeit)
- OS für dom0 (Managementdomain):
 - Linux 2.4/2.6 (Xen 3: nur 2.6)
 - NetBSD 3
- OS für DomU (Guest):
 - Linux 2.4/2.6 (Xen3: nur 2.6)
 - NetBSD 2/3 (noch nicht Xen 3)
 - FreeBSD 5 (noch nicht Xen 3)
 - Plan9 (noch nicht Xen 3)

Xen – Plattformen

- Hardware: x86 (komplett), ppc, ia64 (beide in Arbeit)
- OS für dom0 (Managementdomain):
 - Linux 2.4/2.6 (Xen 3: nur 2.6)
 - NetBSD 3
- OS für DomU (Guest):
 - Linux 2.4/2.6 (Xen3: nur 2.6)
 - NetBSD 2/3 (noch nicht Xen 3)
 - FreeBSD 5 (noch nicht Xen 3)
 - Plan9 (noch nicht Xen 3)

Xen – Plattformen

- Hardware: x86 (komplett), ppc, ia64 (beide in Arbeit)
- OS für dom0 (Managementdomain):
 - Linux 2.4/2.6 (Xen 3: nur 2.6)
 - NetBSD 3
- OS für DomU (Guest):
 - Linux 2.4/2.6 (Xen3: nur 2.6)
 - NetBSD 2/3 (noch nicht Xen 3)
 - FreeBSD 5 (noch nicht Xen 3)
 - Plan9 (noch nicht Xen 3)

Xen – Plattformen

- Hardware: x86 (komplett), ppc, ia64 (beide in Arbeit)
- OS für dom0 (Managementdomain):
 - Linux 2.4/2.6 (Xen 3: nur 2.6)
 - NetBSD 3
- OS für DomU (Guest):
 - Linux 2.4/2.6 (Xen3: nur 2.6)
 - NetBSD 2/3 (noch nicht Xen 3)
 - FreeBSD 5 (noch nicht Xen 3)
 - Plan9 (noch nicht Xen 3)

Xen – Plattformen

- Hardware: x86 (komplett), ppc, ia64 (beide in Arbeit)
- OS für dom0 (Managementdomain):
 - Linux 2.4/2.6 (Xen 3: nur 2.6)
 - NetBSD 3
- OS für DomU (Guest):
 - Linux 2.4/2.6 (Xen3: nur 2.6)
 - NetBSD 2/3 (noch nicht Xen 3)
 - FreeBSD 5 (noch nicht Xen 3)
 - Plan9 (noch nicht Xen 3)

Xen – Plattformen

- Hardware: x86 (komplett), ppc, ia64 (beide in Arbeit)
- OS für dom0 (Managementdomain):
 - Linux 2.4/2.6 (Xen 3: nur 2.6)
 - NetBSD 3
- OS für DomU (Guest):
 - Linux 2.4/2.6 (Xen3: nur 2.6)
 - NetBSD 2/3 (noch nicht Xen 3)
 - FreeBSD 5 (noch nicht Xen 3)
 - Plan9 (noch nicht Xen 3)

Xen - Struktur

- Xen Hypervisor als dünner Layer auf Hardware
- Hypervisor bootet nach Start dom0
- dom0:
 - erste privilegierte Domain, verwaltet Hardware
 - steuert Hypervisor (create, destroy, ... Domains)
 - exportiert Devices (Block & Netz als virtuelle HW)
- Kommunikation via shared memory → schnell
- direkter Zugriff auf PCI-Geräte wenn erlaubt (Xen 2.0)

Xen - Struktur

- Xen Hypervisor als dünner Layer auf Hardware
- Hypervisor bootet nach Start dom0
- dom0:
 - erste privilegierte Domain, verwaltet Hardware
 - steuert Hypervisor (create, destroy, ... Domains)
 - exportiert Devices (Block & Netz als virtuelle HW)
- Kommunikation via shared memory → schnell
- direkter Zugriff auf PCI-Geräte wenn erlaubt (Xen 2.0)

Xen - Struktur

- Xen Hypervisor als dünner Layer auf Hardware
- Hypervisor bootet nach Start dom0
- dom0:
 - erste privilegierte Domain, verwaltet Hardware
 - steuert Hypervisor (create, destroy, ... Domains)
 - exportiert Devices (Block & Netz als virtuelle HW)
- Kommunikation via shared memory → schnell
- direkter Zugriff auf PCI-Geräte wenn erlaubt (Xen 2.0)

Xen - Struktur

- Xen Hypervisor als dünner Layer auf Hardware
- Hypervisor bootet nach Start dom0
- dom0:
 - erste privilegierte Domain, verwaltet Hardware
 - steuert Hypervisor (create, destroy, ... Domains)
 - exportiert Devices (Block & Netz als virtuelle HW)
- Kommunikation via shared memory → schnell
- direkter Zugriff auf PCI-Geräte wenn erlaubt (Xen 2.0)

Xen - Struktur

- Xen Hypervisor als dünner Layer auf Hardware
- Hypervisor bootet nach Start dom0
- dom0:
 - erste privilegierte Domain, verwaltet Hardware
 - steuert Hypervisor (create, destroy, ... Domains)
 - exportiert Devices (Block & Netz als virtuelle HW)
- Kommunikation via shared memory → schnell
- direkter Zugriff auf PCI-Geräte wenn erlaubt (Xen 2.0)

Xen - Struktur

- Xen Hypervisor als dünner Layer auf Hardware
- Hypervisor bootet nach Start dom0
- dom0:
 - erste privilegierte Domain, verwaltet Hardware
 - steuert Hypervisor (create, destroy, ... Domains)
 - exportiert Devices (Block & Netz als virtuelle HW)
- Kommunikation via shared memory → schnell
- direkter Zugriff auf PCI-Geräte wenn erlaubt (Xen 2.0)

Xen - Struktur

- Xen Hypervisor als dünner Layer auf Hardware
- Hypervisor bootet nach Start dom0
- dom0:
 - erste privilegierte Domain, verwaltet Hardware
 - steuert Hypervisor (create, destroy, ... Domains)
 - exportiert Devices (Block & Netz als virtuelle HW)
- Kommunikation via shared memory → schnell
- direkter Zugriff auf PCI-Geräte wenn erlaubt (Xen 2.0)

Xen - Struktur

- Xen Hypervisor als dünner Layer auf Hardware
- Hypervisor bootet nach Start dom0
- dom0:
 - erste privilegierte Domain, verwaltet Hardware
 - steuert Hypervisor (create, destroy, ... Domains)
 - exportiert Devices (Block & Netz als virtuelle HW)
- Kommunikation via shared memory → schnell
- direkter Zugriff auf PCI-Geräte wenn erlaubt (Xen 2.0)

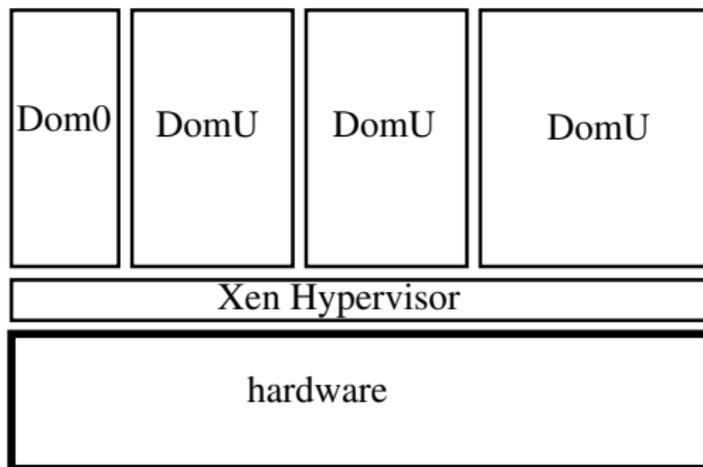
Xen - Struktur

- Xen Hypervisor als dünner Layer auf Hardware
- Hypervisor bootet nach Start dom0
- dom0:
 - erste privilegierte Domain, verwaltet Hardware
 - steuert Hypervisor (create, destroy, ... Domains)
 - exportiert Devices (Block & Netz als virtuelle HW)
- Kommunikation via shared memory → schnell
- direkter Zugriff auf PCI-Geräte wenn erlaubt (Xen 2.0)

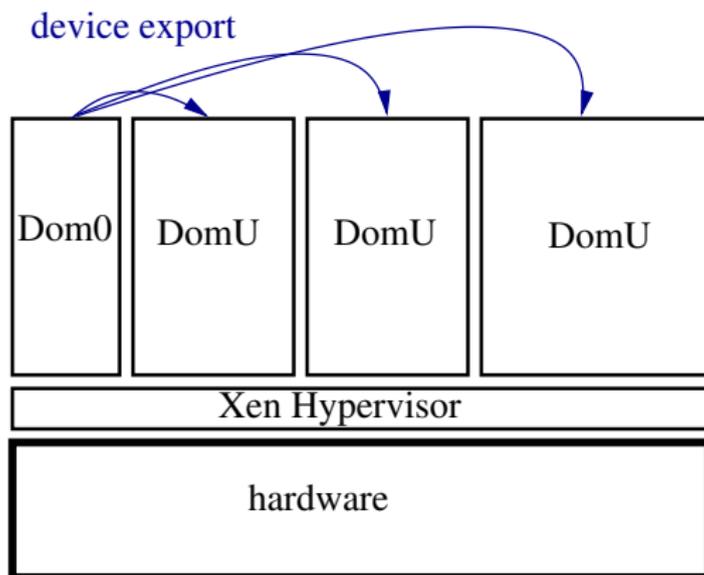
Xen - Struktur



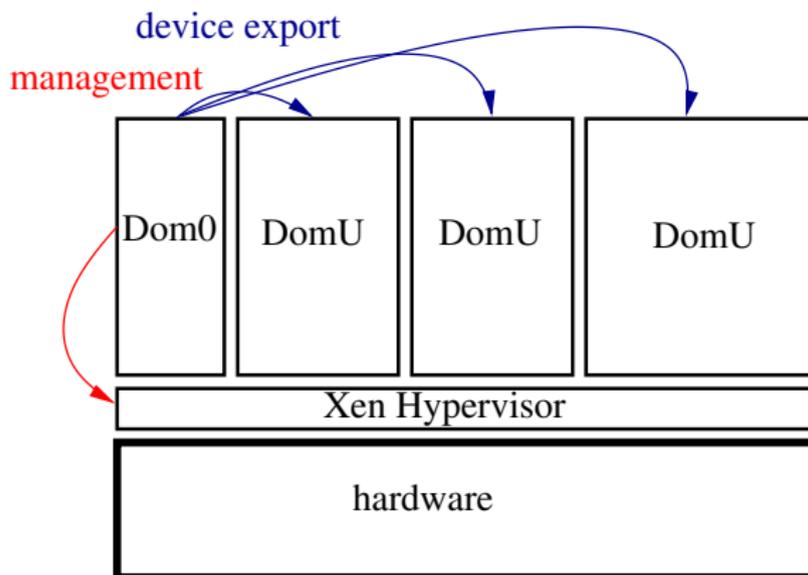
Xen - Struktur



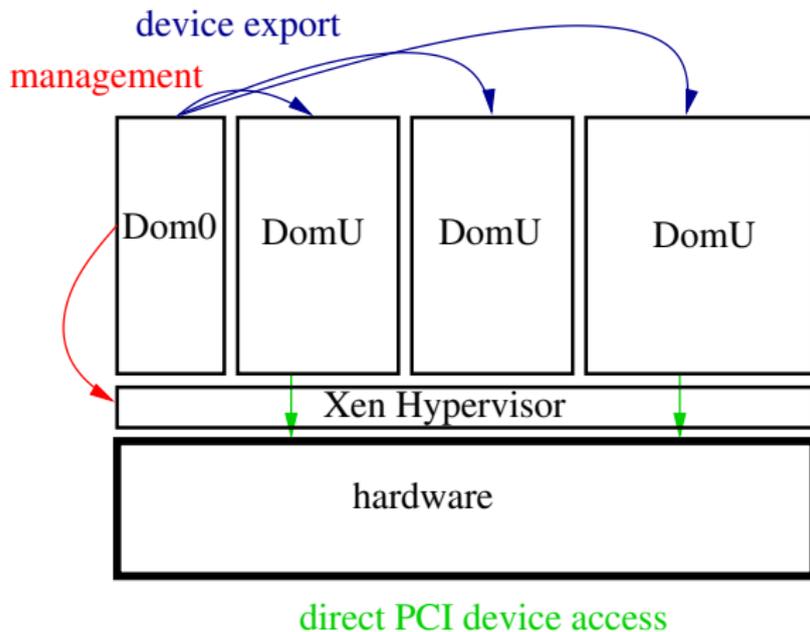
Xen - Struktur



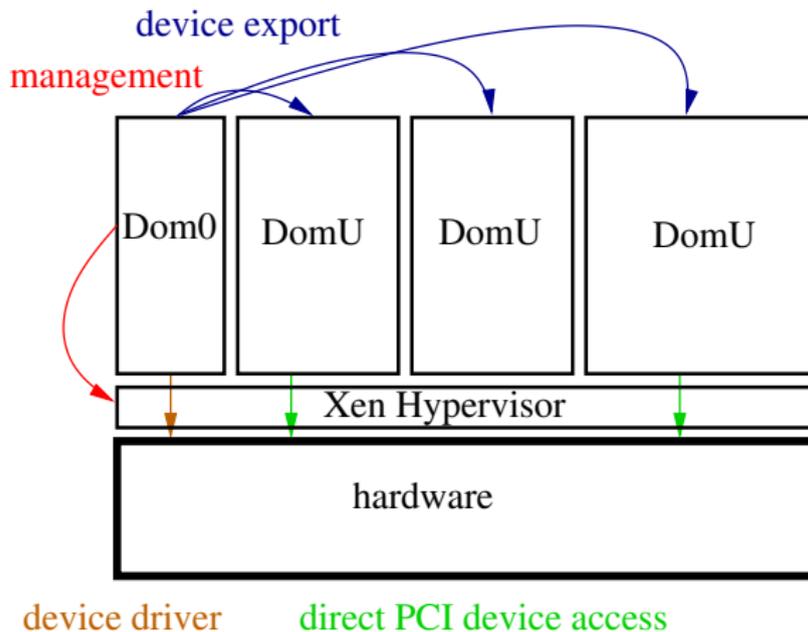
Xen - Struktur



Xen - Struktur



Xen - Struktur



Xen & Vollvirtualisierung

- neue x86 CPUs unterstützen Vollvirtualisierung direkt:
 - Intel: Vanderpool Technology (VT)
 - AMD: Pacifica
- Xen 3 unterstützt bereits VT, Hardware verfügbar
- Pacifica-Unterstützung in Arbeit, noch keine Hardware
- xenusers: Windows XP läuft auf Xen 3.0.0 mit VT-Hardware

Xen & Vollvirtualisierung

- neue x86 CPUs unterstützen Vollvirtualisierung direkt:
 - Intel: Vanderpool Technology (VT)
 - AMD: Pacifica
- Xen 3 unterstützt bereits VT, Hardware verfügbar
- Pacifica-Unterstützung in Arbeit, noch keine Hardware
- xenusers: Windows XP läuft auf Xen 3.0.0 mit VT-Hardware

Xen & Vollvirtualisierung

- neue x86 CPUs unterstützen Vollvirtualisierung direkt:
 - Intel: Vanderpool Technology (VT)
 - AMD: Pacifica
- Xen 3 unterstützt bereits VT, Hardware verfügbar
- Pacifica-Unterstützung in Arbeit, noch keine Hardware
- xenusers: Windows XP läuft auf Xen 3.0.0 mit VT-Hardware

Xen & Vollvirtualisierung

- neue x86 CPUs unterstützen Vollvirtualisierung direkt:
 - Intel: Vanderpool Technology (VT)
 - AMD: Pacifica
- Xen 3 unterstützt bereits VT, Hardware verfügbar
- Pacifica-Unterstützung in Arbeit, noch keine Hardware
- xenusers: Windows XP läuft auf Xen 3.0.0 mit VT-Hardware

Xen & Vollvirtualisierung

- neue x86 CPUs unterstützen Vollvirtualisierung direkt:
 - Intel: Vanderpool Technology (VT)
 - AMD: Pacifica
- Xen 3 unterstützt bereits VT, Hardware verfügbar
- Pacifica-Unterstützung in Arbeit, noch keine Hardware
- xenusers: Windows XP läuft auf Xen 3.0.0 mit VT-Hardware

Xen & Vollvirtualisierung

- neue x86 CPUs unterstützen Vollvirtualisierung direkt:
 - Intel: Vanderpool Technology (VT)
 - AMD: Pacifica
- Xen 3 unterstützt bereits VT, Hardware verfügbar
- Pacifica-Unterstützung in Arbeit, noch keine Hardware
- xenusers: Windows XP läuft auf Xen 3.0.0 mit VT-Hardware

Xen & Vollvirtualisierung

- neue x86 CPUs unterstützen Vollvirtualisierung direkt:
 - Intel: Vanderpool Technology (VT)
 - AMD: Pacifica
- Xen 3 unterstützt bereits VT, Hardware verfügbar
- Pacifica-Unterstützung in Arbeit, noch keine Hardware
- xenusers: Windows XP läuft auf Xen 3.0.0 mit VT-Hardware

Plastik-Mainframe

- Projekt eines Bekannten (VAXpower)
- vorher: Herde von einzelnen Maschinen
- nachher:
 - eine große Maschine: 2x 2.8 GHz Xeon, 4 GB RAM, viel Disk
 - nach Bedarf bis zu 14 virtuelle Maschinen auf NetBSD/Xen
 - komplettes Heimrechenzentrum in einer Maschine
- mehrere VMs auf einer Maschine: ursprünglich Mainframetechnik, jetzt aber auf PCs (== Plastik)
→ „Plastik-Mainframe“

Plastik-Mainframe

- Projekt eines Bekannten (VAXpower)
- vorher: Herde von einzelnen Maschinen
- nachher:
 - eine große Maschine: 2x 2.8 GHz Xeon, 4 GB RAM, viel Disk
 - nach Bedarf bis zu 14 virtuelle Maschinen auf NetBSD/Xen
 - komplettes Heimrechenzentrum in einer Maschine
- mehrere VMs auf einer Maschine: ursprünglich Mainframetechnik, jetzt aber auf PCs (== Plastik)
→ „Plastik-Mainframe“

Plastik-Mainframe

- Projekt eines Bekannten (VAXpower)
- vorher: Herde von einzelnen Maschinen
- nachher:
 - eine große Maschine: 2x 2.8 GHz Xeon, 4 GB RAM, viel Disk
 - nach Bedarf bis zu 14 virtuelle Maschinen auf NetBSD/Xen
 - komplettes Heimrechenzentrum in einer Maschine
- mehrere VMs auf einer Maschine: ursprünglich Mainframetechnik, jetzt aber auf PCs (== Plastik)
→ „Plastik-Mainframe“

Plastik-Mainframe

- Projekt eines Bekannten (VAXpower)
- vorher: Herde von einzelnen Maschinen
- nachher:
 - eine große Maschine: 2x 2.8 GHz Xeon, 4 GB RAM, viel Disk
 - nach Bedarf bis zu 14 virtuelle Maschinen auf NetBSD/Xen
 - komplettes Heimrechenzentrum in einer Maschine
- mehrere VMs auf einer Maschine: ursprünglich Mainframetechnik, jetzt aber auf PCs (== Plastik)
→ „Plastik-Mainframe“

Plastik-Mainframe

- Projekt eines Bekannten (VAXpower)
- vorher: Herde von einzelnen Maschinen
- nachher:
 - eine große Maschine: 2x 2.8 GHz Xeon, 4 GB RAM, viel Disk
 - nach Bedarf bis zu 14 virtuelle Maschinen auf NetBSD/Xen
 - komplettes Heimrechenzentrum in einer Maschine
 - mehrere VMs auf einer Maschine: ursprünglich Mainframetechnik, jetzt aber auf PCs (== Plastik)
→ „Plastik-Mainframe“

Plastik-Mainframe

- Projekt eines Bekannten (VAXpower)
- vorher: Herde von einzelnen Maschinen
- nachher:
 - eine große Maschine: 2x 2.8 GHz Xeon, 4 GB RAM, viel Disk
 - nach Bedarf bis zu 14 virtuelle Maschinen auf NetBSD/Xen
 - komplettes Heimrechenzentrum in einer Maschine
- mehrere VMs auf einer Maschine: ursprünglich Mainframetechnik, jetzt aber auf PCs (== Plastik)
→ „Plastik-Mainframe“

Plastik-Mainframe

- Projekt eines Bekannten (VAXpower)
- vorher: Herde von einzelnen Maschinen
- nachher:
 - eine große Maschine: 2x 2.8 GHz Xeon, 4 GB RAM, viel Disk
 - nach Bedarf bis zu 14 virtuelle Maschinen auf NetBSD/Xen
 - komplettes Heimrechenzentrum in einer Maschine
- mehrere VMs auf einer Maschine: ursprünglich Mainframetechnik, jetzt aber auf PCs (== Plastik)
→ „Plastik-Mainframe“

Plastik-Mainframe

- Projekt eines Bekannten (VAXpower)
- vorher: Herde von einzelnen Maschinen
- nachher:
 - eine große Maschine: 2x 2.8 GHz Xeon, 4 GB RAM, viel Disk
 - nach Bedarf bis zu 14 virtuelle Maschinen auf NetBSD/Xen
 - komplettes Heimrechenzentrum in einer Maschine
- mehrere VMs auf einer Maschine: ursprünglich Mainframetechnik, jetzt aber auf PCs (== Plastik)
→ „Plastik-Mainframe“

Nagios-Herde – Übersicht

- mehrere Nagios-Server (10 .. 14)
- Trennung aus organisatorisch/technischen Gründen
- Last pro Server: für aktuelle HW geringfügig
- → virtualisieren, auf eine Maschine zusammenfassen
- → eingesparte Ressourcen ermöglichen Redundanz

Nagios-Herde – Übersicht

- mehrere Nagios-Server (10 .. 14)
- Trennung aus organisatorisch/technischen Gründen
- Last pro Server: für aktuelle HW geringfügig
- → virtualisieren, auf eine Maschine zusammenfassen
- → eingesparte Ressourcen ermöglichen Redundanz

Nagios-Herde – Übersicht

- mehrere Nagios-Server (10 .. 14)
- Trennung aus organisatorisch/technischen Gründen
- Last pro Server: für aktuelle HW geringfügig
- → virtualisieren, auf eine Maschine zusammenfassen
- → eingesparte Ressourcen ermöglichen Redundanz

Nagios-Herde – Übersicht

- mehrere Nagios-Server (10 .. 14)
- Trennung aus organisatorisch/technischen Gründen
- Last pro Server: für aktuelle HW geringfügig
- → virtualisieren, auf eine Maschine zusammenfassen
- → eingesparte Ressourcen ermöglichen Redundanz

Nagios-Herde – Übersicht

- mehrere Nagios-Server (10 .. 14)
- Trennung aus organisatorisch/technischen Gründen
- Last pro Server: für aktuelle HW geringfügig
- → virtualisieren, auf eine Maschine zusammenfassen
- → eingesparte Ressourcen ermöglichen Redundanz

Nagios-Herde – Übersicht

- mehrere Nagios-Server (10 .. 14)
- Trennung aus organisatorisch/technischen Gründen
- Last pro Server: für aktuelle HW geringfügig
- → virtualisieren, auf eine Maschine zusammenfassen
- → eingesparte Ressourcen ermöglichen Redundanz

Nagios-Herde – Technik

- 2 Maschinen, Primärsystem & Backup
- derzeit: je 1x 2.66 GHz Xeon, 1 GB RAM, 150 GB Disk
- geplant: 2x 2.66 GHz Xeon, 4 GB RAM
- 256 MB dom0
- Nagios-DomU: je 64 .. 256 MB RAM, 3 GB Disk (root+swap)
- „Disk“ für DomU: LVM-Volumes
- root-Volumes via drbd zw. Primär- & Backupsystem gespiegelt → Wiederanlauf auf Backupsystem ohne Datenverluste
- derzeit 4 Nagios-DomU, Endausbau 10 .. 14

Nagios-Herde – Technik

- 2 Maschinen, Primärsystem & Backup
- derzeit: je 1x 2.66 GHz Xeon, 1 GB RAM, 150 GB Disk
- geplant: 2x 2.66 GHz Xeon, 4 GB RAM
- 256 MB dom0
- Nagios-DomU: je 64 .. 256 MB RAM, 3 GB Disk (root+swap)
- „Disk“ für DomU: LVM-Volumes
- root-Volumes via drbd zw. Primär- & Backupsystem gespiegelt → Wiederanlauf auf Backupsystem ohne Datenverluste
- derzeit 4 Nagios-DomU, Endausbau 10 .. 14

Nagios-Herde – Technik

- 2 Maschinen, Primärsystem & Backup
- derzeit: je 1x 2.66 GHz Xeon, 1 GB RAM, 150 GB Disk
- geplant: 2x 2.66 GHz Xeon, 4 GB RAM
- 256 MB dom0
- Nagios-DomU: je 64 .. 256 MB RAM, 3 GB Disk (root+swap)
- „Disk“ für DomU: LVM-Volumes
- root-Volumes via drbd zw. Primär- & Backupsystem gespiegelt → Wiederanlauf auf Backupsystem ohne Datenverluste
- derzeit 4 Nagios-DomU, Endausbau 10 .. 14

Nagios-Herde – Technik

- 2 Maschinen, Primärsystem & Backup
- derzeit: je 1x 2.66 GHz Xeon, 1 GB RAM, 150 GB Disk
- geplant: 2x 2.66 GHz Xeon, 4 GB RAM
- 256 MB dom0
- Nagios-DomU: je 64 .. 256 MB RAM, 3 GB Disk (root+swap)
- „Disk“ für DomU: LVM-Volumes
- root-Volumes via drbd zw. Primär- & Backupsystem gespiegelt → Wiederanlauf auf Backupsystem ohne Datenverluste
- derzeit 4 Nagios-DomU, Endausbau 10 .. 14

Nagios-Herde – Technik

- 2 Maschinen, Primärsystem & Backup
- derzeit: je 1x 2.66 GHz Xeon, 1 GB RAM, 150 GB Disk
- geplant: 2x 2.66 GHz Xeon, 4 GB RAM
- 256 MB dom0
- Nagios-DomU: je 64 .. 256 MB RAM, 3 GB Disk (root+swap)
- „Disk“ für DomU: LVM-Volumes
- root-Volumes via drbd zw. Primär- & Backupsystem gespiegelt → Wiederanlauf auf Backupsystem ohne Datenverluste
- derzeit 4 Nagios-DomU, Endausbau 10 .. 14

Nagios-Herde – Technik

- 2 Maschinen, Primärsystem & Backup
- derzeit: je 1x 2.66 GHz Xeon, 1 GB RAM, 150 GB Disk
- geplant: 2x 2.66 GHz Xeon, 4 GB RAM
- 256 MB dom0
- Nagios-DomU: je 64 .. 256 MB RAM, 3 GB Disk (root+swap)
- „Disk“ für DomU: LVM-Volumes
- root-Volumes via drbd zw. Primär- & Backupsystem gespiegelt → Wiederanlauf auf Backupsystem ohne Datenverluste
- derzeit 4 Nagios-DomU, Endausbau 10 .. 14

Nagios-Herde – Technik

- 2 Maschinen, Primärsystem & Backup
- derzeit: je 1x 2.66 GHz Xeon, 1 GB RAM, 150 GB Disk
- geplant: 2x 2.66 GHz Xeon, 4 GB RAM
- 256 MB dom0
- Nagios-DomU: je 64 .. 256 MB RAM, 3 GB Disk (root+swap)
- „Disk“ für DomU: LVM-Volumes
- root-Volumes via drbd zw. Primär- & Backupsystem gespiegelt → Wiederanlauf auf Backupsystem ohne Datenverluste
- derzeit 4 Nagios-DomU, Endausbau 10 .. 14

Nagios-Herde – Technik

- 2 Maschinen, Primärsystem & Backup
- derzeit: je 1x 2.66 GHz Xeon, 1 GB RAM, 150 GB Disk
- geplant: 2x 2.66 GHz Xeon, 4 GB RAM
- 256 MB dom0
- Nagios-DomU: je 64 .. 256 MB RAM, 3 GB Disk (root+swap)
- „Disk“ für DomU: LVM-Volumes
- root-Volumes via drbd zw. Primär- & Backupsystem gespiegelt → Wiederanlauf auf Backupsystem ohne Datenverluste
- derzeit 4 Nagios-DomU, Endausbau 10 .. 14

Nagios-Herde – Technik

- 2 Maschinen, Primärsystem & Backup
- derzeit: je 1x 2.66 GHz Xeon, 1 GB RAM, 150 GB Disk
- geplant: 2x 2.66 GHz Xeon, 4 GB RAM
- 256 MB dom0
- Nagios-DomU: je 64 .. 256 MB RAM, 3 GB Disk (root+swap)
- „Disk“ für DomU: LVM-Volumes
- root-Volumes via drbd zw. Primär- & Backupsystem gespiegelt → Wiederanlauf auf Backupsystem ohne Datenverluste
- derzeit 4 Nagios-DomU, Endausbau 10 .. 14

Nagios-Herde – Erfahrung

- Basissystem SLES9
- Installation Xen nicht ganz reibungslos
- aber: in einigen aktuellen Distributionen bereits integriert
- Xen 2.0.7 im Einsatz, stabil & zuverlässig
- Spiegelung via drbd zuverlässig & performant
- Performance virtuelle Maschinen: gut

Nagios-Herde – Erfahrung

- Basissystem SLES9
- Installation Xen nicht ganz reibungslos
- aber: in einigen aktuellen Distributionen bereits integriert
- Xen 2.0.7 im Einsatz, stabil & zuverlässig
- Spiegelung via drbd zuverlässig & performant
- Performance virtuelle Maschinen: gut

Nagios-Herde – Erfahrung

- Basissystem SLES9
- Installation Xen nicht ganz reibungslos
- aber: in einigen aktuellen Distributionen bereits integriert
- Xen 2.0.7 im Einsatz, stabil & zuverlässig
- Spiegelung via drbd zuverlässig & performant
- Performance virtuelle Maschinen: gut

Nagios-Herde – Erfahrung

- Basissystem SLES9
- Installation Xen nicht ganz reibungslos
- aber: in einigen aktuellen Distributionen bereits integriert
- Xen 2.0.7 im Einsatz, stabil & zuverlässig
- Spiegelung via drbd zuverlässig & performant
- Performance virtuelle Maschinen: gut

Nagios-Herde – Erfahrung

- Basissystem SLES9
- Installation Xen nicht ganz reibungslos
- aber: in einigen aktuellen Distributionen bereits integriert
- Xen 2.0.7 im Einsatz, stabil & zuverlässig
- Spiegelung via drbd zuverlässig & performant
- Performance virtuelle Maschinen: gut

Nagios-Herde – Erfahrung

- Basissystem SLES9
- Installation Xen nicht ganz reibungslos
- aber: in einigen aktuellen Distributionen bereits integriert
- Xen 2.0.7 im Einsatz, stabil & zuverlässig
- Spiegelung via drbd zuverlässig & performant
- Performance virtuelle Maschinen: gut

Nagios-Herde – Erfahrung

- Basissystem SLES9
- Installation Xen nicht ganz reibungslos
- aber: in einigen aktuellen Distributionen bereits integriert
- Xen 2.0.7 im Einsatz, stabil & zuverlässig
- Spiegelung via drbd zuverlässig & performant
- Performance virtuelle Maschinen: gut

Erfahrungen – Zusammenfassung

- zuverlässige, stabile Technologie
- gut geeignet für Serverkonsolidierung
- sehr aktive Community
- sehr aktive Weiterentwicklung
- sehr gute Hardwareunterstützung

Erfahrungen – Zusammenfassung

- zuverlässige, stabile Technologie
- gut geeignet für Serverkonsolidierung
- sehr aktive Community
- sehr aktive Weiterentwicklung
- sehr gute Hardwareunterstützung

Erfahrungen – Zusammenfassung

- zuverlässige, stabile Technologie
- gut geeignet für Serverkonsolidierung
- sehr aktive Community
- sehr aktive Weiterentwicklung
- sehr gute Hardwareunterstützung

Erfahrungen – Zusammenfassung

- zuverlässige, stabile Technologie
- gut geeignet für Serverkonsolidierung
- sehr aktive Community
- sehr aktive Weiterentwicklung
- sehr gute Hardwareunterstützung

Erfahrungen – Zusammenfassung

- zuverlässige, stabile Technologie
- gut geeignet für Serverkonsolidierung
- sehr aktive Community
- sehr aktive Weiterentwicklung
- sehr gute Hardwareunterstützung

Erfahrungen – Zusammenfassung

- zuverlässige, stabile Technologie
- gut geeignet für Serverkonsolidierung
- sehr aktive Community
- sehr aktive Weiterentwicklung
- sehr gute Hardwareunterstützung

URLs

- <http://www.cl.cam.ac.uk/Research/SRG/netos/xen/>
- <http://www.xensource.com/>
- <http://wiki.xensource.com/xenwiki/>
- <http://www.google.de/search?q=xen+hypervisor>

URLs

- <http://www.cl.cam.ac.uk/Research/SRG/netos/xen/>
- <http://www.xensource.com/>
- <http://wiki.xensource.com/xenwiki/>
- <http://www.google.de/search?q=xen+hypervisor>

Ende

Vielen Dank für Ihr Interesse!