Servervirtualisierung

Christian Baun · Marcel Kunze
Thomas Ludwig

Der Begriff der Virtualisierung bezieht sich in der Informatik auf Konzepte und Technologien, mit denen eine einheitliche, abstrakte Sicht auf eine heterogene physische Ressourcenlandschaft erzeugt wird. Die Virtualisierung erhebt den Anspruch, dass Ressourcen effektiver und flexibler genutzt werden können. Bei der Servervirtualisierung werden die Ressourcen eines Rechnersystems auf mehrere Klienten verteilt und können von mehreren unabhängigen Betriebssysteminstanzen gleichzeitig genutzt werden. Dabei können unterschiedliche Anwendungsklassen auf einer geringeren Anzahl physischer Rechner konsolidiert und damit die verfügbaren Ressourcen gleichmäßig ausgelastet werden.

Prinzip der Virtualisierung

Eine virtuelle Maschine ist ein nachgebildeter Rechner, der in einer isolierten Umgebung auf einem realen System läuft. Jede virtuelle Maschine verhält sich dabei wie ein vollwertiger Computer mit eigenen Komponenten wie CPU, Hauptspeicher, Festplatten, Grafikkarte, Netzwerkkarten, usw. Auf einige Hardwarekomponenten, darunter CPU und der Hauptspeicher des Computers, kann eine virtuelle Maschine direkt zugreifen. Andere Hardwarekomponenten (u. a. Netzwerkkarten) werden dagegen meist komplett emuliert. In einer virtuellen Maschine kann im Idealfall ein Betriebssystem mit Anwendungen genauso wie auf einem realen Computer installiert und betrieben werden. Die Software merkt nicht, dass sie sich in Wirklichkeit in einer virtuellen Maschine befindet. Anforderungen des Gastbetriebssystems werden unbemerkt von der Virtualisierungssoftware abgefangen und auf die reale Hardware umgesetzt. Die Virtualisierung wurde bei Großrechnern bereits vor über 40 Jahren zur Optimierung des Betriebs eingeführt. IBM stellte In den 1960er-Jahren die Virtual Machine Facility/370, kurz VM/370 vor. Auf dieser Plattform wurde Mehrbenutzerbetrieb gefahren, indem mehrere Einzelbenutzerbetriebinstanzen in virtuellen Maschinen ausführt wurden. Jede virtuelle Maschine stellte eine vollständige Nachbildung der darunter liegenden, physischen Hardware dar [7, 12].

Auf der Basis aktueller Technologien kann eine virtuelle Infrastruktur heutzutage auch in Mikroprozessorumgebungen ähnlich zuverlässig und skalierbar wie ein Großrechner betrieben werden. Die virtuelle Infrastruktur umfasst dabei nahezu alle Aspekte eines modernen Rechenzentrums wie Server, Netze, Datenspeicher, Software, Anwendungen und macht selbst vor Desktopsystemen nicht halt. Dieser Beitrag behandelt als spezielles Thema die Virtualisierung von x86-Servern.

DOI 10.1007/s00287-008-0321-6

Christian Baun, Marcel Kunze
Karlsruher Institut für Technologie,
Steinbuch Centre for Computing,
Hermann-von-Helmholtz-Platz 1,
76344 Eggenstein-Leopoldshafen
E-Mail: {baun, marcel.kunze}@iwr.fzk.de

Thomas Ludwig
Ruprecht-Karls-Universität Heidelberg,
Institut für Informatik,
Im Neuenheimer Feld 348/018,
69120 Heidelberg
E-Mail: t.ludwig@computer.org

Vorteile der Servervirtualisierung

Es gibt mehrere Gründe, die für die Servervirtualisierung sprechen [12]. Durch Serverkonsolidierung ist eine bessere Auslastung der Hardware möglich. Das Zusammenführen mehrerer Server auf einem Rechner führt zur Kostensenkung bei Hardware, Verbrauchskosten (Strom, Kühlung), Stellplätzen, Administration, usw. Durch die Reduktion der Anzahl der physischen Rechner und den Einsatz ausgereifter Managementwerkzeuge ist eine vereinfachte Administration möglich und Routineaufgaben können leicht automatisiert werden. Die Bereitstellung von neuen Infrastrukturen und Servern wird beschleunigt: Innerhalb weniger Minuten können diese manuell oder automatisch aus einem Pool von Ressourcen erzeugt werden.

Servervirtualisierung führt auch zu einer vereinfachten Wartung, denn virtuelle Maschinen können im laufenden Betrieb zwischen verschiedenen Rechnern verschoben werden. Diese, als Live Migration bekannte Fähigkeit, geschieht aus Sicht der Benutzer unterbrechungsfrei. Dadurch entfällt die Notwendigkeit zur Vereinbarung von Zeitfenstern bei Hardwarewartungen, da laufende Serverdienste auf eine andere reale Maschine übertragen werden können. Auch Technologiewechsel können dadurch ohne Betriebsunterbrechung vollzogen werden. Die Dimensionierung der Ressourcen (Sizing) wird vereinfacht. Der Ressourcenbedarf neuer Anwendungen muss zuvor nur grob abgeschätzt werden. Die genaue Einstellung kann später dynamisch im laufenden Betrieb erfolgen. Aktuell wird davon ausgegangen, dass durch den Einsatz von Servervirtualisierung die Investitionen in neue Hard- und Software um bis zu 70% sinken können und im Rechenzentrum sind Kosteneinsparungen von bis zu 50% erreichbar [14].

Die Servervirtualisierung ermöglicht maximale Flexibilität, da die gesamte virtuelle Maschine in wenigen Dateien gespeichert wird. Dadurch können virtuelle Maschinen problemlos vervielfältigt und gesichert werden. Abbilder (Snapshots) vom aktuellen Zustand einer virtuellen Maschine können erzeugt und zu einem späteren Zeitpunkt in wenigen Sekunden wieder hergestellt werden. Die benötigte Zeit ist abgängig von der Größe der virtuellen Maschine und der Leistungsfähigkeit der eingesetzten Hardware. Auch die Sicherheit wird erhöht, da virtuelle Maschinen gegenüber anderen virtuellen Maschinen und dem Host-System isoliert sind. Unternehmenskritische Anwendungen können in einer virtuellen Maschine gekapselt werden und damit in einer sicheren Umgebung laufen.

Es gibt erweiterte Möglichkeiten zur Vereinbarung garantierter Güte und Verfügbarkeit, sogenannte Service Level Agreements (SLA), für Ressourcen oder Dienste. Beim Ausfall einer virtuellen Maschine bleiben die übrigen virtuellen Maschinen und der Host davon unberührt. Die betroffene Maschine kann gegebenenfalls auf einer anderen Ressource automatisch wieder hochgefahren werden. Softwaretests und Softwareentwicklung werden optimiert, denn durch den gleichzeitigen Betrieb mehrerer Betriebssysteme auf einem Rechnersystem können zusätzliche Testumgebungen ohne Bereitstellung zusätzlicher Hardware schnell aufgesetzt werden. Ein weiterer Vorteil ist die Unterstützung historischer Anwendungen. Sogenannte Legacy-Betriebssysteme oder Legacy-Anwendungen, für die keine Hardwareunterstützung mehr zu bekommen ist, können durch Virtualisierung am Leben gehalten werden.

Nachteile der Servervirtualisierung

Durch die Umsetzung der Zugriffe haben virtuelle Maschinen immer eine geringere Leistung als reale Hardware. Die aktuellen Virtualisierungstechnologien sind allerdings so ausgereift, dass sich dieser Nachteil mit 5–10% [8] Leistungsverlust nicht sonderlich auswirkt. Da die aktuellen Mehrkernprozessorsysteme mit Virtualisierung besonders effektiv genutzt werden können, spielt dieser Leistungsverlust eine zunehmend untergeordnete Rolle. Nicht geeignet für Virtualisierung sind dagegen Systeme, die spezielle Hardwareanforderungen wie z. B. Kopierschutzstecker (Hardwaredongles) haben. Ferner ist zu beachten, dass bei der Serverkonsolidierung beim Ausfall eines Hosts mehrere virtuelle Server gleichzeitig ausfallen würden. Gefragt sind daher ausgeklügelte Ausfallkonzepte und redundante Installationen, was in einem Rechenzentrum zusätzliches Fachwissen und hochwertigere Infrastruktur erfordert.

Virtualisierungskonzepte

Virtualisierung ist in der Informatik nur ein Überbegriff für eine Gruppe von unterschiedlichen Konzepten und Technologien. Speziell im Zusammenhang mit Servervirtualisierung spielen die folgenden Themen eine Rolle:

- Hardware-Emulation,
- Partitionierung,
- vollständige Virtualisierung,
- Paravirtualisierung,
- Hardwarevirtualisierung und
- Betriebssystemvirtualisierung bzw. Container.

Diese Konzepte unterscheiden sich nicht nur hinsichtlich Ihrer Implementierung, sondern auch bezüglich ihrer Praxisrelevanz und Einsatzhäufigkeit auf dem Gebiet der Servervirtualisierung.

Hardware-Emulation

Bei der Hardware-Emulation wird in den meisten Fällen versucht, die komplette Hardware eines Rechnersystems funktionell nachzubilden, um so einem unveränderten Betriebssystem, das für eine andere Hardwarearchitektur ausgelegt ist, den Betrieb zu ermöglichen. Vorteilhaft ist, dass bei der Emulation einer Architektur bzw. eines Betriebssystems keinerlei Anpassungen am Betriebssystem bzw. den Anwendungen nötig sind. Es ist durch Emulation auch möglich, andere Architekturen als die hardwaretechnisch vorhandene zu verwenden und neue, noch nicht in Hardware existierende Architekturen zu studieren. Der große Nachteil der Emulation besteht darin, dass die Entwicklung von Emulationsumgebungen sehr aufwändig ist und die Ausführungsgeschwindigkeit in der Regel im Vergleich zu anderen Virtualisierungslösungen deutlich geringer ist. Beispiele für Emulatoren in der PC-Welt sind Bochs, QEMU, PearPC, Wabi, DOSBox, DOSEMU und Microsoft Virtual PC (in der Version für MacOS-X). Speziell in der Servervirtualisierung spielt die Hardware-Emulation nur eine untergeordnete Rolle.

Laufzeitumgebungen wie z. B. WINE für Windows-Anwendungen unter Linux und anderen Betriebssystemen sind keine Emulatoren, da sie nur eine Kompatibilitätsschicht darstellen, die nicht zum Ziel hat, ein komplettes Betriebssystemsystem oder eine Architektur nachzubilden, sondern sich auf die Emulation von Schnittstellen beschränkt.

Partitionierung

Bei der Partitionierung können auf den Gesamtressourcen eines Computersystems Teilsysteme definiert werden, in denen eine lauffähige Betriebssysteminstanz installiert werden kann, um diese als eigenständigen Server zu verwenden. Diese Form der Virtualisierung verwaltet Ressourcen wie Prozessor, Hauptspeicher, Datenspeicher und I/O-Kanäle über die Firmware des Rechners und teilt diese einer virtuellen Maschine zu.

Die Partitionierungstechnik kommt z. B. bei IBM in Großrechnern der zSerie oder Midrange-Systemen der pSerie mit den Prozessorfamilien Power5 und Power6 zum Einsatz. Ohne Neustart ist es möglich, im laufenden Betrieb eine Ressourcenzuteilung nahezu beliebig zu verändern. Auf einem Großrechner aktueller Bauart können auf diese Weise problemlos mehrere hundert bis tausend Linux-Instanzen gleichzeitig laufen [1].

Bei anderen Mikroprozessoren aktueller Bauart gibt es diese Form der Partitionierung des Gesamtsystems nicht, sondern es wird lediglich die Partitionierung der CPU unterstützt, wie bei Intel Vanderpool oder AMD Pacifica.

Virtualisierungsgrundlagen in der x86-Architektur

Die Virtualisierung in der x86-Architektur lässt sich leicht verstehen, wenn man das Schutzkonzept der Ringe betrachtet. Aktuelle Prozessoren enthalten zur Erhöhung des Schutzes und der Stabilität vier Privilegienstufen für die Ausführung von Befehlen. Ein Prozess kann immer nur in einem einzelnen Ring ausgeführt werden und ist nicht in der Lage, sich selbständig aus diesem zu befreien. Nur Ring 0 hat vollen Zugriff auf die Hardware. Die Zugriffsprivilegien auf den Befehlssatz des Prozessors werden von Ring 0 zu Ring 3 immer weiter eingeschränkt. Das Konzept der vier Privilegienstufen ist in Abb. 1 zu sehen. In Ring 0 läuft der Betriebssystemkern (Kernel) und zumindest die zum Start des Betriebssystems notwendigen Hardwaretreiber. Dieser darf den vollständigen Befehlsatz des Prozessors nutzen. In Ring 3 laufen in der Regel die Anwendungen. Weil die meisten anderen Prozessorarchitekturen nur 2 Ringe implementieren, nutzen die verbreiteten Betriebssysteme wie z. B. Linux aus Gründen der Portabilität ausschließlich Ring 0 und 3. Eine Ausnahme bildet OS/2, das Ring 2 für Anwendungen, die auf Hardware und Eingabe-/Ausgabeschnittstellen zugreifen dürfen [13]. Im Fall von OS/2 gehörten die Grafiktreiber zu dieser Gruppe. Die Ringe 0 und 3 werden im Bereich der Betriebssysteme häufig auch einfach als Kernel-Bereich (Kernel-Space) und Benutzerbereich (User-Space) bezeichnet.

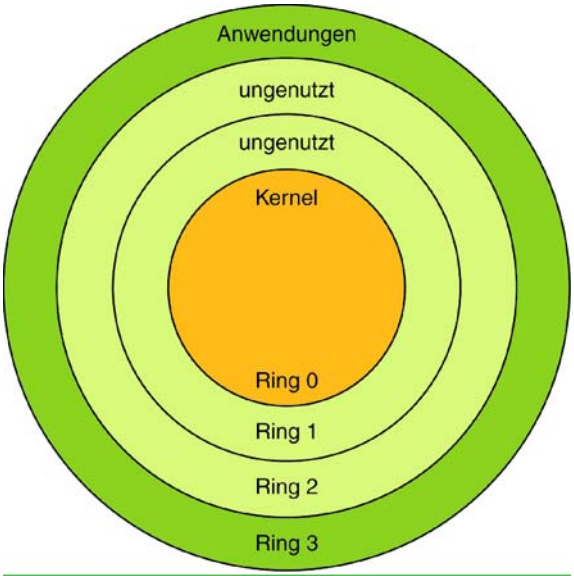

Abb. 1 Die vier Privilegienstufen zum Speicherschutz bei x86-Prozessoren

Sobald ein Prozess in einem weniger privilegierten Ring eine privilegierte Operation aufruft, erzeugt der Prozessor eine Ausnahme (Exception), die im benachbarten privilegierteren Ring abgefangen und dort behandelt wird. Ausnahmen, die nicht abgefangen werden können, verursachen eine allgemeine Schutzverletzung (General Protection Fault) und bringen den aufrufenden Prozess zum Absturz. Handelt es sich bei diesem Prozess um den Betriebssystemkern, stürzt das gesamte System ab [5].

Vollständige Virtualisierung

Vollständige Virtualisierungslösungen bieten einer virtuellen Maschine eine vollständige Umgebung inklusive eigenem BIOS: Jedes Gastbetriebssystem hat einen eigenen virtuellen Rechner mit virtuellen Ressourcen wie CPU, Hauptspeicher, Laufwerken, Netzwerkkarten, usw. zur Verfügung. Es kommt ein sogenannter Virtueller Maschinenmonitor (VMM) zum Einsatz, der die Hardwareressourcen des Rechners durch Hardware-Emulation oder Hardwarevirtualisierung an die virtuellen Maschinen intelligent verteilt.

Vollständige Virtualisierungslösungen nutzen die Tatsache, dass bei x86-Systemen nur zwei von vier möglichen Privilegienstufen verwendet werden. Der VMM befindet sich in Ring 0 auf der Ebene des Betriebssystemkerns des Host-Betriebssystems und

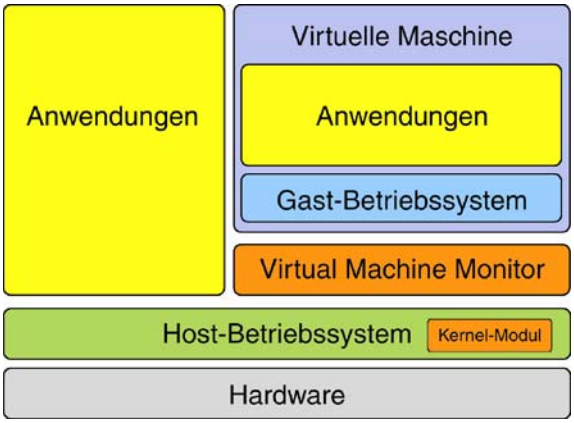

Abb. 2 Der virtuelle Maschinenmonitor (VMM)

hat vollen Zugriff auf die Hardware des Systems. Die virtuellen Maschinen (Gastbetriebssysteme) befinden sich in einem der weniger privilegierten Ringe. Der VMM stellt für jede denkbare Ausnahme eine Behandlung zur Verfügung, welche die privilegierten Operationen der Gastbetriebssysteme abfängt, interpretiert und ausführt. Durch die Existenz des VMM ist sichergestellt, dass die virtualisierten Systeme, wie in Abb. 2 zu sehen ist, nur über den Umweg des VMM Zugriff auf die Hardware des Systems erhalten und ein kontrollierter Zugriff auf die gemeinsam genutzten Systemressourcen gewährleistet ist.

Die Vorteile der Virtualisierung mittels VMM sind, dass kaum Änderungen an Host- und gar keine Änderungen an Gastbetriebssystemen erforderlich sind und dass eine fast native Verarbeitungsgeschwindigkeit der Gastbetriebssysteme erreicht wird, da der Zugriff auf die wichtigsten Ressourcen nur durchgereicht wird. Ein weiterer Vorteil ist die hohe Flexibilität, da jedes Gastbetriebssystem seinen eigenen Betriebssystemkern verwendet.

Die bekanntesten Beispiele für Virtualisierungslösungen, die auf dem Konzept des Virtual Machine Monitor basieren, sind VMware ESX, VMware Server, VMware Workstation und VMware Fusion, Microsoft Virtual PC (in der Version für x86), Microsoft Hyper-V, Parallels Desktop und Parallels Workstation, VirtualBox, Kernel Virtual Machine (KVM) und Mac-on-Linux (MoL).

Paravirtualisierung

Auch beim Konzept der Paravirtualisierung, wie es das populäre Xen und die kommerziellen Versionen Citrix Xenserver und Virtual Iron implementiert,

laufen die Gastbetriebssysteme nicht im privilegierten Ring 0, sondern im weniger privilegierten Ring 1. Wie in Abb. 3 zu sehen ist, läuft in Ring 0 der sogenannte Hypervisor, das Äquivalent zum VMM.

Der Hypervisor ist bei der Paravirtualisierung eine abstrakte Verwaltungsschicht (Abb. 4), über den die Gastbetriebssysteme auf die physischen Ressourcen wie Speicher, Ein-/Ausgabegeräte und Netzwerkschnittstellen zugreifen. Der Hypervisor ist quasi ein auf ein Minimum reduziertes Metabetriebssystem, das die Hardwareressourcen unter den Gastsystemen verteilt, so wie ein Betriebssystem dieses unter den laufenden Prozessen tut. Ein Meta-Betriebssystem ermöglicht den unabhängigen Betrieb unterschiedlicher Anwendungen und Betriebssysteme auf einem einzigen Prozessor. Bei der Paravirtualisierung steht den Gastbetriebssystemen keine emulierte Hardwareebene zur Verfügung, sondern lediglich eine Anwendungsschnittstelle.

Weil die Betriebssysteme (der Betriebssystemkern) nicht mehr in Ring 0, sondern in Ring 1 laufen, können diese keine privilegierten Anweisungen mehr ausführen. Aus diesem Grund stellt der Hypervisor Ersatzfunktionen, die sogenannten Hypercalls, zur Verfügung. Das Konzept des Hypercalls ist in Abb. 5 zu sehen.

Hypercalls sind vergleichbar mit Systemaufrufen (System Calls), jedoch ist die Interrupt-Nummer verschieden. Wenn eine Anwendung die Ausführung

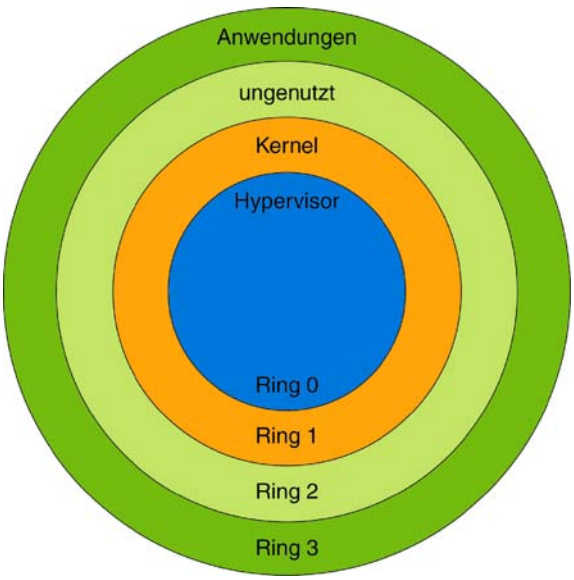

Abb. 3 Position des Hypervisors bei Paravirtualisierung

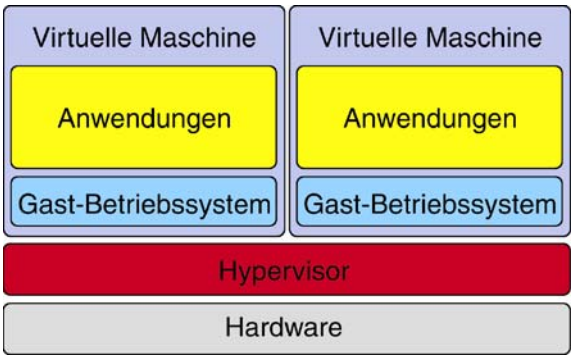

Abb. 4 Konzept der Paravirtualisierung

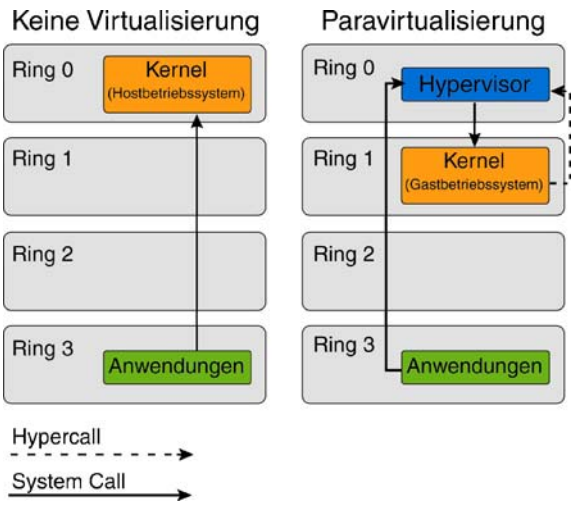

Abb. 5 Systemaufrufe und Hypercalls der Paravirtualisierung

eines Systemaufrufs anfordert, wird eine Ersatzfunktion im Hypervisor aufgerufen [6, 13]. Der Hypervisor weist die Ausführung des Systemaufrufs über die Kernel-API des Host-Betriebssystems an. Das Erweitern des Betriebssystemskerns um die Hypercall-Funktionalität macht somit eine Modifikation sowohl der Host- als auch der Gastbetriebssysteme notwendig. Das Abfangen und Prüfen aller Systemaufrufe durch den Hypervisor führt zu lediglich geringen Geschwindigkeitseinbußen bei der Paravirtualisierung.

Auf einem Xen-Host werden alle Systeme als Domänen bezeichnet. Die privilegierte Domäne (Domo) hat über den Hypervisor als einzige den vollen Zugriff auf alle Ressourcen. Der Hypervisor organisiert auch die Zugriffe der Gastsysteme, die in den sogenannten unprivilegierten Domänen (DomU) liegen. Er legt dabei für alle I/O-Anfragen der Domänen eine Ablaufreihenfolge (Scheduling) fest und kann diese unterschiedlich priorisieren. So-

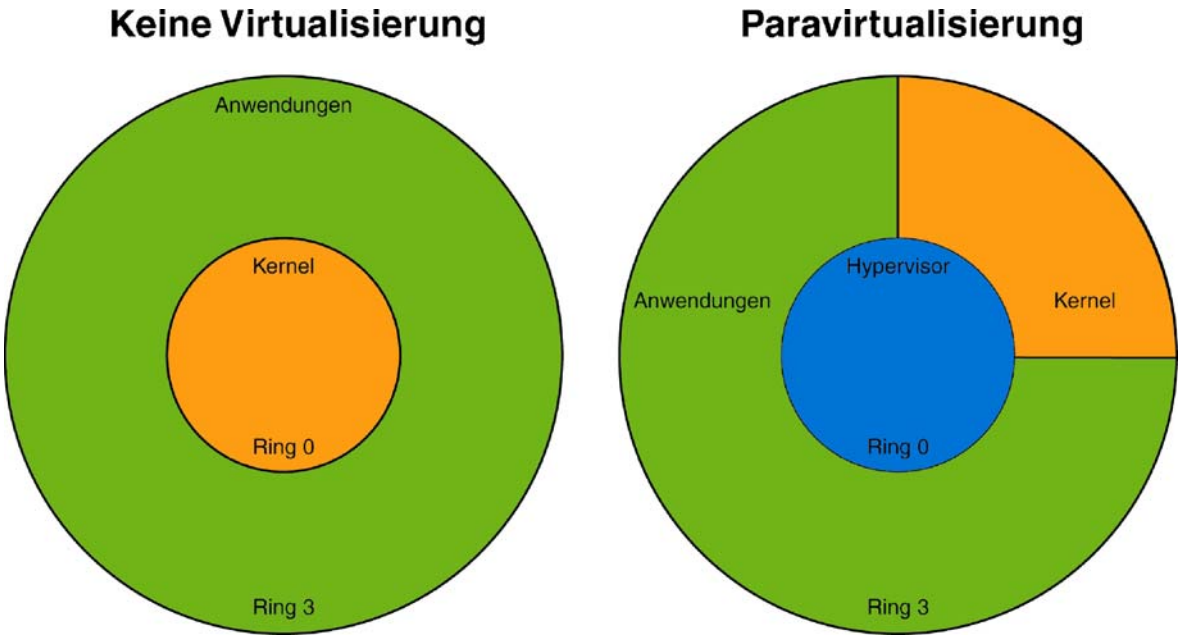

Abb. 6 Die Privilegienstufen zum Speicherschutz bei x86-64-Prozessoren

mit kann bestimmten Domänen mehr oder weniger CPU-Zeit oder I/O-Leistung zugewiesen werden [5].

Bei der Entwicklung der x86-64-Architektur wurde auf die als unnötig erachteten Ringe 1 und 2 verzichtet [4]. Dadurch haben x86-64-Architekturen wie IA64 nur zwei Privilegienstufen zum Speicherschutz (Abb. 6). Der Xen-Hypervisor befindet sich auf der x86-64-Architektur von AMD und Intel genau wie bei der x86-32-Architektur in Ring 0. Das Betriebssystem aber mit seinem Betriebssystemkern wird bei der der x86-64-Architektur in Ring 3 zu den Anwendungen verschoben.

Um Datensicherheit zu gewährleisten, haben die unprivilegierten Domänen keinen direkten Hardwarezugriff. Es existieren zwei unterschiedliche Konzepte, um den Zugriff auf die physikalischen Rechnerressourcen zu realisieren. Die eine Möglichkeit ist der Einsatz virtueller Gerätetreiber, die auch als Virtual Split Driver bezeichnet werden. Die andere Möglichkeit sind emulierte Gerätetreiber.

Mithilfe virtueller Gerätetreiber werden von Xen I/O-Anforderungen aus den unprivilegierten Domänen über den Hypervisor direkt an den physischen Gerätetreiber innerhalb der privilegierten Domäne weitergeleitet. Dadurch werden alle I/O-Anforderungen gesammelt und die Hardware- und Datenzugriffe kanalisiert. Beim Einsatz virtueller Gerätetreiber müssen die I/O-Anforderungen nur einmal innerhalb der privilegierten Domäne interpretiert und verarbeitet werden. Der Treiber innerhalb der unprivilegierten Domäne reicht die Dateien einfach weiter.

Beim Einsatz emulierter Gerätetreiber werden die I/O-Anforderungen über emulierte Geräte vom Hypervisor auf die realen Geräte (z. B. eine Netzwerkkarte) weitergeleitet. Bei diesem Ansatz kommen, wie in Abb. 7 zu sehen ist, zwei Treiberebenen zum Einsatz, da die I/O-Anforderungen innerhalb der unprivilegierten Domäne und noch einmal innerhalb der privilegierten Domain inter-

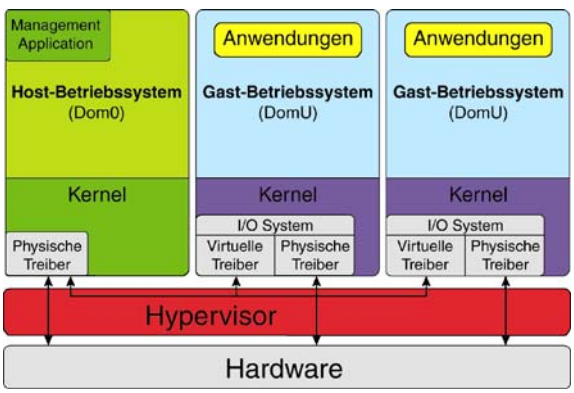

Abb. 7 I/O-Pfade in Xen mit virtuellen Gerätetreibern und dem Weg eines Pakets von der DomU zum Netwerkgerät

pretiert und verarbeitet werden müssen. Aus diesem Grund erzeugt der Einsatz virtueller Gerätetreiber eine geringere Zusatzlast und eine höhere Leistung als der Einsatz emulierter Gerätetreiber.

Beispiele für Virtualisierungslösungen, die Paravirtualisierung unterstützen, sind Xen und VMware ESX.

Hardwarevirtualisierung

Sollen unmodifizierte Betriebssysteme (z. B. Windows) mit Xen virtuell betrieben werden, muss der Prozessor selbst die Virtualisierungsfunktionen direkt unterstützen. Aktuelle CPU-Generationen von Intel und AMD implementieren daher Virtualisierungserweiterungen, die unter dem Begriff Hardwarevirtualisierung zusammengefasst werden. Intel und AMD gehen bei der konkreten Realisierung der Hardwarevirtualisierung ähnliche aber inkompatible Wege.

AMD erweitert seit Juni 2006 seine AMD64 CPUs um den sogenannten Secure-Virtual-Machine-Befehlssatz (SVM). Diese Lösung trägt den Namen AMD-V und war vorher unter dem Stichwort Pacifica bekannt. Die konkurrierende Lösung von Intel mit der Bezeichnung VT-x für IA32-CPUs und VT-i für Itanium implementieren dagegen den sogenannten Virtual-Machine-Extensions-Befehlssatz (VMX). Intels Lösung lief vormals unter dem Stichwort Vanderpool. Kern der Neuerung ist bei beiden Lösungen eine Überarbeitung der Privilegienstruktur in der Prozessarchitektur. Die neuen Befehle bei AMD und Intel bieten virtuellen Maschinen Prozessor-Level-Support, indem sie eine Erweiterung zu den bereits beschriebenen Privilegienstufen Ring 0 und Ring 3 definieren. Die Ringstruktur wurde durch eine Erweiterung von Ring 0 um eine Ebene, die neue Hypervisor-Schicht, ergänzt [4]. Diese Ebene wird als Root-Betriebsmodus, gelegentlich auch als Ring – 1 bezeichnet. Der Hypervisor bzw. virtuelle Maschinenmonitor läuft im Root-Betriebsmodus und besitzt jederzeit die volle Kontrolle über den Prozessor und die Ressourcen, da damit ein höheres Privileg als Ring 0 implementiert ist. Dieses hat den großen Vorteil, dass die Gastbetriebssysteme nicht angepasst werden müssen und der Kernel nicht wie bei der Paravirtualisierung mit den Anwendungen auf einer Privilegienstufe läuft. Die virtuellen Maschinen arbeiten auf ihren virtuellen Prozessoren im Non-Root-Betriebsmodus, wobei auch privilegierte Befehle erlaubt sind. Es ist dabei für virtuelle Betriebssysteminstanzen nicht erkennbar, dass sie unter der Kontrolle eines VMM laufen, da ihnen der gewohnte Zugriff auf die Befehle von Ring 0 zur Verfügung steht. Ein virtueller Prozessor verhält sich exakt wie ein nicht-virtualisierter Prozessor.

AMDs Virtualisierungstechnologie Pacifica ist mit Vanderpool von Intel nicht kompatibel. AMD64-Prozessoren unterscheiden sich darüber hinaus auch durch ihren integrierten Speicher-Controller von Intels x86-CPUs darin, dass Pacifica den Speicher-Controller ebenfalls virtualisiert. Eine weitere Eigenschaft von Pacifica stellt der Device Exclusion Vector (DEV) dar, der in virtuellen Maschinen Geräte behandelt, die ohne Hilfe des Prozessors direkt auf den Speicher des Systems zugreifen können [1].

Das aktuelle Xen Version 3 unterstützt neben der Paravirtualisierung auch die Hardwarevirtualisierung. Hier können neben den verschiedenen Linux-Derivaten auch mehrere, unabhängige und unmodifizierte Instanzen von Windows 2000, 2003 Server, XP und Vista parallel zueinander betrieben werden. Auch Windows Server 2008 nutzt mit Hyper-V die Hardwarevirtualisierung. Es ist aber im Vergleich festzustellen, dass bei den aktuellen Implementierungen der Hardwarevirtualisierung die Leistung meist noch etwas hinter der Paravirtualisierung zurückbleibt.

Betriebssystemvirtualisierung bzw. Container

Bei der Betriebssystemvirtualisierung, die auch als Container oder Jails bezeichnet wird, spielt das Host-Betriebssystem eine entscheidende Rolle. Hier laufen unter ein und demselben Betriebssystemkern mehrere voneinander abgeschottete identische Systemumgebungen. Es wird kein zusätzliches Betriebssystem, sondern eine isolierte Laufzeitumgebung virtuell in einem geschlossenen Container erzeugt. Diese Container werden unter BSD häufig als Jails bezeichnet. Nach außen treten die virtuellen Umgebungen wie eigenständige Systeme auf. Alle laufenden Anwendungen verwenden aber den denselben Betriebssystemkern. Die Anwendungen sehen nur Prozesse, mit denen sie sich gemeinsam in einer virtuellen Umgebung befinden. Vorteile der Betriebssystemvirtualisierung sind ein geringer Ressourcenbedarf und eine hohe Leistung, da der Betriebssystemkern

in gewohnter Weise die Hardware des Systems verwaltet.

Nachteilig ist, dass alle virtuellen Umgebungen denselben Betriebssystemkern und dasselbe Betriebssystem in einer einheitlichen Version fahren müssen. So können nur mehrere unabhängige Instanzen desselben Betriebssystems gestartet werden und es ist nicht möglich, verschiedene Betriebssysteme gleichzeitig zu verwenden. Diese Art der Virtualisierung wird hauptsächlich genutzt, um Anwendungen in isolierten Umgebungen zu betreiben und so eine höhere Sicherheit zu gewährleisten. Interessant ist der Ansatz auch zur Skalierung von gehosteten Servern oder Webdiensten (Web-Services) auf Mehrkernprozessorarchitekturen.

Abbildung 8 zeigt das Konzept der Betriebssystemvirtualisierung. Beispiele für die Betriebssystemvirtualisierung sind die Containertechnologie von Sun Solaris, OpenVZ für Linux, Linux-VServer, FreeBSD Jails, Virtuozzo (kommerzielle Variante von OpenVZ) und FreeVPS. Besonders Internet Service Provider, die (virtuelle) Root-Server anbieten, nutzen diese Art der Servervirtualisierung, da nur sehr wenig Leistung verloren geht und ein hoher Grad an Sicherheit garantiert werden kann.

Cloud-Computing, das aktuelle Schlagwort in der IT, ist eng verwand mit dem Grid-Computing [11]. Während beim Grid-Computing spezielle Angebote für das Hochleistungsrechnen geschaffen werden, ist die Zielsetzung des Cloud-Computing die Bereitstellung skalierbarer IT-Dienste über das Internet für eine potenziell große Zahl externer

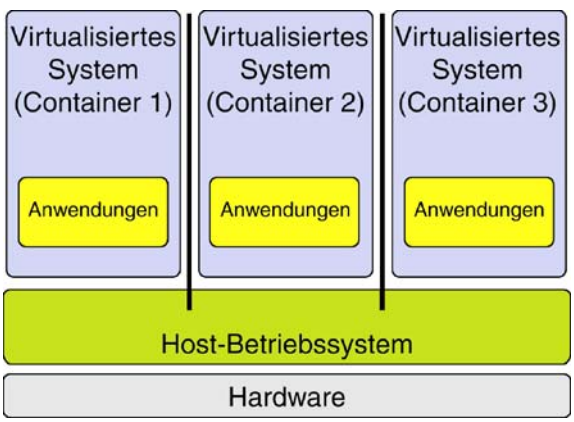

Abb. 8 Betriebssystemvirtualisierung (Container)

Kunden mit sehr heterogenen Anwendungen. Die Cloud-Provider betreiben sehr große Zentren für Rechen- und Speicherkapazitäten und können über die Masse sehr günstige Preise für ihre Leistungen realisieren. Dadurch, dass in einer Cloud die Dienste als virtuelle Maschinen in einem verteilten Rechnernetz laufen, sind sowohl Ausfallsicherheit als auch Skalierbarkeit gewährleistet. Die Kunden des Cloud-Computing sind häufig ihrerseits ebenfalls Anbieter von Internetdiensten, die aber ihre IT-Infrastruktur nicht mehr selbst betreiben, sondern auslagern. Da man als Kunde in der Regel Administrator-Privilegien und Zugriff auf die Einstellung der Firewall bekommt, kann man ebenso bequem arbeiten wie bei einer lokalen Installation und nach Bedarf skalieren. Einfach gesagt, handelt es sich beim Cloud-Computing um verteiltes Rechnen mit virtuellen Maschinen [2, 10].

Cloud-Computing ist sehr industrierelevant: Sehr viele namhafte IT-Firmen investieren zurzeit in Cloud-Computing-Infrastruktur oder haben entsprechende Pläne. Populäre Coud-Dienste sind der Computing-Dienst „Elastic Compute Cloud“ (EC2) und der Speicherdienst „Simple Storage Service“ (S3), die beide von Amazon angeboten werden. Bei beiden Diensten bezahlen die Nutzer per Kreditkarte nur die Ressourcen (Rechenzeit, Speicher, Datentransfer und Dienstleistungen), die sie verbraucht haben. Virtuelle Instanzen können einfach und schnell erzeugt und kontrolliert werden. Beschränkungen hinsichtlich der verwendeten Betriebssysteme existieren nur wenige. Unterstützt werden Linux und Open Solaris. Mithilfe eines Emulators wie QEMU kann auch Windows betrieben werden.

Das Cloud-Computing hat das Potenzial, die Versprechen einzulösen, an denen das Grid-Computing bislang scheitert. Als Vorteil sind insbesondere zu nennen: Die einfache Benutzbarkeit, unkomplizierter Zugang zu Ressourcen und geringere Kosten [3]. Ein fundamentaler Unterschied besteht darin, dass aktuelle Grid-Systeme die Anpassung der Anwendungen an die Infrastruktur erfordern, während sich beim Cloud-Computing die Infrastruktur dynamisch an die Anforderungen der Anwendungen anpassen kann.

Fazit zur Virtualisierung

Der Einsatz von Virtualisierungstechnologien bietet viele Vorteile und kaum Risiken oder Nachteile. Bei

Verwendung einer geeigneten Managementumgebung für die virtuelle Infrastruktur kann der Betrieb von Rechenzentren sehr stark optimiert werden: Die Kapazitätsplanung wird in Zukunft einen größeren Raum einnehmen als die Diskussion um Hardwarebeschaffungen und Installationen. Virtualisierung bietet auf der einen Seite ein großes Einsparpotenzial, eröffnet aber auch neue Sicherheitslücken und Angriffspunkte, insbesondere auf der Ebene des Hypervisors. Aus diesem Grund wird von VMware die Entwicklung einer schlanken Virtualisierungs-Appliance vorangetrieben, bei der ein spezieller 32 MB großer Virtualisierungs-Kernel die Virtualisierungsfunktionen ohne Betriebssystem direkt auf der Hardware realisiert (VMware ESXi). Durch dieses Vorgehen wird eine potenzielle Angriffsfläche dramatisch minimiert.

Während zurzeit die Verwendung eines virtuellen Maschinenmonitors (VMM) der Standard ist, nimmt die Verwendung von Hardwarevirtualisierung durch die neuen Prozessorgenerationen zu. Aktuelle Linux-Distributionen mit Xen und neue Betriebssysteme wie z. B. Windows Server 2008 unterstützen diesen Trend.

Virtualisierung ist aktuell eines der großen Schlagworte in der Informatik und wird in den nächsten Jahren auch wegen der deutlich besseren Energieeffizienz und der unkomplizierten Nutzung von Mehrkernprozessoren eine stetig wachsende Rolle spielen. Virtualisierung ist inzwischen weniger eine technische, sondern vielmehr eine strategische Frage. Laut Gartner zählt Virtualisierung zu den wichtigsten Technologien bis zum Jahr 2010 und Nach einer Studie [9] der International Data Group zum Thema Servervirtualisierung von Juni 2008 rechnen Unternehmen, die Servervirtualisierung nutzen, bis 2009 mit einem Zuwachs von 52 Prozent an neu zu virtualisierenden Servern. Von den befragten Unternehmen, die Servervirtualisierung gegenwärtig noch nicht nutzen, wollen 54 Prozent innerhalb der nächsten zwölf Monate Virtualisierungstechnologie einsetzen, um von den bekannten Vorteilen der Servervirtualisierung zu profitieren. 59 Prozent der Implementierungen erzielten durch die Servervirtualisierung ein Konsolidierungsverhältnis von bis zu 4 : 1 bei den Servern. 45 Prozent der befragten Organisationen nutzen Virtualisierung entweder bereits als Standardplattform bei der Bereitstellung neuer Applikationen oder erwarten in Kürze eine Etablierung dieser Praxis.

Literatur

1. Bengel G, Baun C, Kunze M, Stucky K-U (2008) Masterkurs Parallele und Verteilte Systeme: Grundlagen und Programmierung von Multicoreprozessoren, Multiprozessoren, Cluster und Grid. Vieweg und Teubner, Wiesbaden
2. An EGEE Comparative Study (2008) Grids and Clouds – Evolution or Revolution? https://edms.cern.ch/file/925013/3/EGEE-Grid-Cloud.pdf (Zugriff: 18.01.2009)
3. Opitz A, König H, Szamlewska S (2008) What Does Grid Computing Cost? J Grid Comput 6(4):385–397
4. Chisnall D (2008) The Definitive Guide to the Xen Hypervisor. Prentice Hall, USA
5. Sprang H, Benk T, Zdrzalek J, Dehner R (2007) Xen. Virtualisierung unter Linux. Open Source Press, München
6. Hagen W von (2008) Professional Xen Virtualization. Wiley Publishing, Indianapolis, IN, USA
7. Creasy RJ (1981) The origin of the VM/370 time-sharing system. IBM J Res Dev 25(5):483–490
8. Hardt M (2005) Virtualisation for Grid-Computing. Cracow Grid Workshop, 20.–23. November 2005, Krakau, Polen
9. http://www.idc.com/getdoc.jsp?containerId=GE56Q (Zugriff 18.01.2009)
10. Hülsenbusch R (2008) Cloud Computing auf altbekannten Wegen. iX 12:131–134
11. Baun C (2008) Grid-, Cloud-, Cluster- und Meta-Computing. c't 21:132–133
12. Amit Singh A (2005) An Introduction to Virtualization. http://www.kernelthread.com/publications/virtualization (Zugriff: 18.01.2009)
13. Barham P, Dragovic B, Fraser K, Hand S, Harris T, Ho A, Neugebauer R, Pratt I, Warfield A (2003) Xen and the Art of Virtualization. http://www.cl.cam.ac.uk/research/srg/netos/papers/2003-xensosp.pdf (Zugriff: 18.01.2009)
14. Hantelmann F (2008) Mit Virtualisierung RZ-Kosten halbieren. iX 12:88–91