

Virtualisierung und Containerisierung

Seminar „Neueste Trends im Hochleistungsrechnen“

Thomas Berreis

15. März 2016

Betreuer: Dr. Michael Kuhn

Abstract

Die einstmals nur für Großrechner bekannte Virtualisierung hat in den letzten Jahren durch die Weiterentwicklung und den Hype um Containerlösungen einen starken Aufschwung erlebt. In vielen Bereichen lässt sich die Performance beeinflussende Virtualisierung mittels Hypervisor durch leichtgewichtige Container ablösen und so ganze Landschaften viel flexibler gestalten. Trotz diverser Techniken zur Optimierung konnten die zahlreichen Studien durch eigene Tests bewiesen werden, in denen in Containern isolierte Anwendungen deutlich bessere Performance erreichten, als in virtuellen Maschinen. Mit großem Einfluss vor allem in den Bereichen der agilen Softwareentwicklung und DevOps haben Containerlösungen bereits heute einen festen Platz eingenommen und sind aus modernen IT-Abteilungen nicht mehr wegzudenken.

Inhaltsverzeichnis

| | |
|--|-----------|
| 1 Motivation | 4 |
| 2 Techniken der Virtualisierung | 6 |
| 2.1 Virtualisierung mittels Hypervisor | 6 |
| 2.2 „Leichtgewichtige“ Virtualisierung mittels Container | 8 |
| 3 Techniken zur Optimierung | 10 |
| 3.1 DirectPath I/O | 10 |
| 3.2 Write Combining | 10 |
| 3.3 DMA | 11 |
| 3.4 SR-IOV | 13 |
| 3.5 NUMA Aware Scheduling | 14 |
| 4 Pro und Contra Virtualisierung | 15 |
| 5 Performance | 17 |
| 6 Fazit | 20 |

1 Motivation

Obwohl die Anfänge der Virtualisierung schon in den 1960er Jahren liegen und dessen Grundstein damals von IBM gelegt wurde,¹ findet in der IT-Branche seit wenigen Jahren wieder ein Umdenken statt. Durch Produkte bekannter Großunternehmen, wie der Amazon Elastic Compute Cloud² oder VMware vSphere³ und vor allem auch quelloffener Virtualisierungs- und Cloudlösungen wie Docker⁴ und OpenStack⁵ ist der Umstieg von echter Hardware auf energie- und platzsparende Virtualisierung viel einfacher und zudem auch erschwinglicher geworden. Das im Bereich der Informationstechnologie führende Forschungs- und Beratungsunternehmen Gartner stellte in diesem Zusammenhang im Jahr 2012 bereits ein starkes Wachstum fest. Wie in Abbildung 1 erkennbar, würde laut der Studie die Anzahl installierter virtueller Maschinen weiter ansteigen und sich so von 2009 bis 2013 erneut verdreifachen.

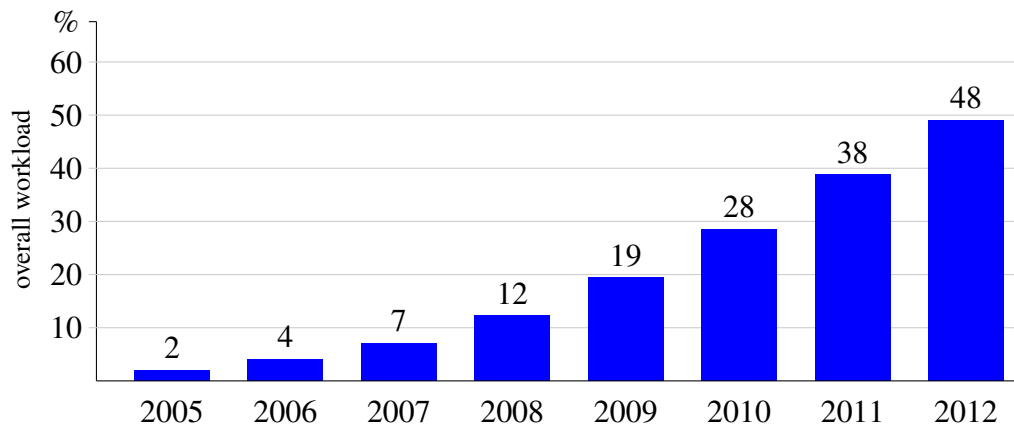


Abbildung 1: auf virtuellen Maschinen erzeugter Workload

Virtualization-Facts and Trends.

<http://de.slideshare.net/HPESoftwareSolutions/hp-software-universe-managing-virtualized-environments-agentless-and-agentbased-solutions-for-managing-vmware-environments-using-hp-sitescope-operations-manager-agents-and-virtualization-spi>. Gartner, Abruf: 23.02.2016.

In der Wirtschaft und auch im Bereich der Forschung ist es unumgänglich, die eigene IT so effizient und skalierbar wie möglich zu betreiben. Soll weiter die Portierbarkeit der eigenen Daten gewährleistet und möglichst plattformunabhängig gearbeitet werden, ist die Virtualisierung

¹Vgl. ebenda, [BS12], S. 1 ff.

²[AM16]

³[VM16a]

⁴[DO16]

⁵[OS16]

der eigenen Systeme häufig das Mittel der Wahl. Als bewiesen gilt, dass die Virtualisierung die jährlichen Energiekosten eines Unternehmens enorm senken und die Umwelt schonen kann. Das Unternehmen Green House Data spricht bei der Virtualisierung von 100 Systemen von einer effizienteren CPU-Nutzung von 40 bis 60 Prozent und vergleicht dies mit dem Pflanzen von 1569 Bäumen.⁶ Bei der Virtualisierung von zehn Systemen mit Windows Server 2008 gibt Microsoft eine jährliche Energieeinsparung von circa 11.000 kWh nach vier und etwa 40.000 kWh nach 10 Betriebsjahren an,⁷ was einer Kosteneinsparung von mehreren Tausend Euro entspricht.

Ein weiteres Berechnungsbeispiel zeigt, dass Virtualisierung auch zur Platzeinsparung genutzt werden kann. Geht man von einer durchschnittlichen Höhe pro Server mit zwei Höheneinheiten und einem herkömmlichen Rack mit 42 Höheneinheiten aus, lassen sich in diesem maximal 21 Server betreiben. In der Realität sind diese Systeme häufig nur zu einem Bruchteil ausgelastet – meist spricht man von 10 bis 50 Prozent.⁸ Bei einer Auslastung von 50 Prozent ließen sich theoretisch zwei virtuelle Maschinen auf einem Server betreiben und diesen so zu 100 Prozent auslasten. Dadurch ließe sich die Kapazität des eigenen Rechenzentrums um 100 Prozent, also auf 42 Systeme erhöhen und die Applikationen blieben voneinander isoliert. Hierbei wird jedoch auch eines sehr schnell deutlich: Tritt auf der Hardware eine Störung auf, fallen gleichzeitig zwei Systeme aus. Welche Vor- und Nachteile beim Einsatz von Virtualisierungslösungen entstehen und ob die in den letzten Jahren entstandenen Lösungen auch sinnvoll im Bereich des High Performance Computing eingesetzt werden können, gilt es zu überprüfen.

⁶Vgl. ebenda, [GH16]

⁷Vgl. ebenda, [FA16]

⁸Vgl. ebenda, [AB13]

2 Techniken der Virtualisierung

Virtualisierung ist im Allgemeinen eine Technologie, mit welcher man eine physikalische Umgebung in logische Einheiten aufteilen kann.⁹ Neben der Virtualisierung im Softwarebereich – beispielsweise der Aufteilung von Domains durch Virtual Hosts oder auch ganzer Netzwerke durch VLAN und VPN – spielt die Ressourcenteilung auch im Hardwareumfeld eine große Rolle. Am ehesten trifft man im privaten Bereich beim Einsatz virtueller Festplatten zur Verschlüsselung der eigenen Daten auf diese Technik. Im professionellen Bereich wird die Virtualisierung jedoch auf vielfältigere Art und Weise eingesetzt.

2.1 Virtualisierung mittels Hypervisor

Eine Form der Virtualisierung stellt der Einsatz einer zusätzlichen Abstraktionsschicht zwischen der vorhandenen Hardware und der darauf zu installierenden Betriebssysteme dar. Der Layer – Hypervisor oder auch Virtual Machine Monitor genannt – ermöglicht einen simultanen Betrieb verschiedener Gastsysteme auf einem Hostsystem und verwaltet dabei die Zuteilung der Ressourcen für die einzelnen virtuellen Maschinen. Der Hypervisor ermöglicht also eine Teilung der tatsächlich verfügbaren Ressourcen und schafft so Transparenz für die Gastsysteme.

Auf dem Markt existieren derzeit zwei Arten der Hypervisor-Architektur. Während der Hypervisor des Typs 1 direkt auf der vorhandenen Hardware betrieben und daher auch Bare-Metal-Hypervisor genannt wird, kann der des Typs 2 als Applikation direkt auf einem herkömmlichen Betriebssystem eingesetzt werden.^{10,11} Prominente Beispiele für den Bare-Metal-Hypervisor sind unter anderem IBM z/VM zur Virtualisierung der Mainframe-Plattform¹², oder der laut einer Studie von Gartner in der Branche am häufigsten eingesetzte VMware vSphere¹³ (Vgl. Abbildung 2).

⁹Vgl. ebenda, [WI16a]

¹⁰Vgl. ebenda, [WI16b]

¹¹Vgl. ebenda. [IT16a]

¹²[IB16]

¹³[VM16a]

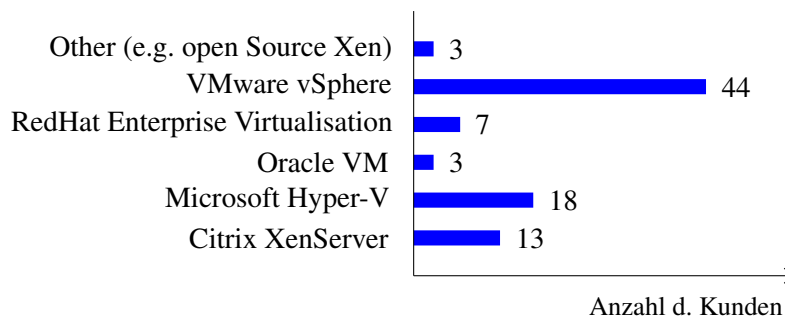


Abbildung 2: Verteilung eingesetzter Hypervisoren

Chris Wolf: *Heterogeneous Virtualization Trends at Gartner Data Center*.
<http://blogs.gartner.com/chris-wolf/2012/12/11/heterogeneous-virtualization-trends-at-gartner-data-center/>.
 Gartner, Abruf: 24.02.2016.

Der häufig auch im privaten Bereich eingesetzte Hypervisor Typ 2 nutzt die Gerätetreiber des darunter liegenden Betriebssystems und verwaltet auch die Ressourcen direkt darüber. Dieser Typ wird daher auch Hosted Hypervisor genannt¹⁴ und findet unter anderem Einsatz bei den Virtualisierungslösungen Oracle VirtualBox¹⁵ und der VMware Workstation¹⁶. Eine Stärke dieser Technik ist die Unabhängigkeit zum Betriebssystem des Hosts, wodurch nahezu jede Art Gast-Betriebssystem eingesetzt werden kann. Wie in Abbildung 3 zu sehen ist, werden die Unterschiede der beiden Typen auch im Schichtenmodell schnell deutlich.

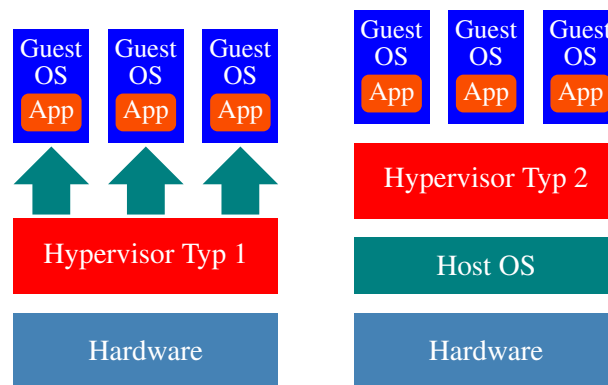


Abbildung 3: Schichtenmodell der Hypervisor-Architektur

Hypervisor :: ITWissen.info
<http://www.itwissen.info/bilder/hypervisor-architekturen.png>.
 Abruf: 24.02.2016.

¹⁴Vgl. ebenda, [MR14]

¹⁵[VI16], S. 13

¹⁶[VM16b]

2.2 „Leichtgewichtige“ Virtualisierung mittels Container

Neben den auf einem Hypervisor basierenden Virtualisierungslösungen existiert auch eine Virtualisierung auf Betriebssystem-Ebene. Bei dieser Methode werden die Bibliotheken und der Kernel des Hostsystems mitbenutzt, wodurch im Vergleich zur virtuellen Maschine wesentlich weniger Speicher benötigt wird, jedoch die Wählbarkeit des Gast-Betriebssystems entfällt.¹⁷ Die meist unter dem Begriff „Container“ geführten Umgebungen gelten als leichtgewichtige Alternative zur herkömmlichen VM, jedoch werden zusätzliche Techniken benötigt, um dieselbe Isolation zu erhalten. Erste Ansätze für diese Abgrenzung wurden bereits 1979 entwickelt und erstmals als Chroot 1982 in BSD implementiert.¹⁸ Chroot steht dabei für „Change root“ und bietet eine Möglichkeit, das Stammverzeichnis für einen Prozess und dessen Forks zu ändern. Außerhalb dieses so festgelegten Stammverzeichnisses können dann keine Kommandos mehr ausgeführt und auf keine Dateien mehr zugegriffen werden. Eine so geschaffene Umgebung wird auch „chroot jail“ genannt.¹⁹ Nach einigen Jahren der Weiterentwicklung folgten 2001 das Projekt „Linux VServer“ und drei Jahre später die Einführung von Zonen unter Solaris. Mithilfe dieser Zonen war es erstmals möglich, Snapshots anzulegen und Container zu klonen. Die Möglichkeiten zur Virtualisierung und Partitionierung von Systemdiensten, wie von Dateisystemen, CPU und Arbeitsspeicher reichten dabei schon weit über die eines chroot jails hinaus.²⁰ 2006 folgte dann durch Google die Einführung der ursprünglich als „process container“ bezeichneten Control-Groups. Diese CGroups sind neben Namespaces bis heute Bestandteil der meisten Container-Distributionen und dienen der Zuordnung und Limitierung von Ressourcen wie CPU, Arbeitsspeicher und Netzwerk-Bandbreite, sowie der Priorisierung und Isolation.²¹ Mithilfe der Namespaces lassen sich Container zusätzlich voneinander abgrenzen, was eine Wiederverwendung von Ressourcen ermöglicht. Innerhalb eines Linux-Systems wäre es beispielsweise nicht möglich, Benutzer mit derselben ID anzulegen – mithilfe der Namespaces spielt dies jedoch keine Rolle und alle auf demselben Host betriebenen Container könnten dieselbe Benutzer-ID verwenden.^{22,23} Durch zahlreiche Verfahren, wie dem Einsatz von Beschreibungssprachen oder durch Build Automation lassen sich Container sehr viel schneller als

¹⁷Vgl. ebenda, [WI16c]

¹⁸Vgl. ebenda, [PI16]

¹⁹Vgl. ebenda, [AR16]

²⁰Vgl. ebenda, [SP16]

²¹Vgl. ebenda, [RS11]

²²Vgl. ebenda, [WI16d]

²³Vgl. ebenda, [WI16e]

virtuelle Maschinen erstellen, da unter anderem das Installieren des Betriebssystems entfällt. Dies hat enorme Vorteile, führt jedoch zu einer gewissen Kurzlebigkeit der Container, was diverse Studien und Umfragen, wie die des auf Softwareanalyse spezialisierten Unternehmens New Relic belegen und in der Abbildung 4 veranschaulicht wurden.

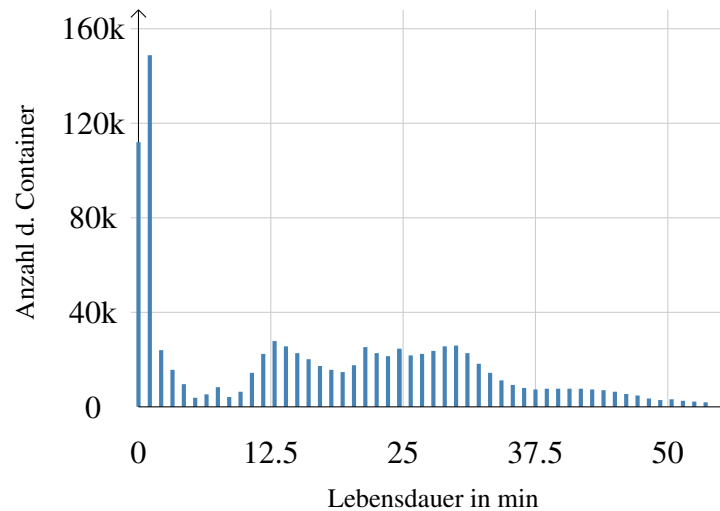


Abbildung 4: Lebensdauer eines Containers

Timothy Prickett Morgan: *Monitoring Is Different For Ephemeral Infrastructure*

<http://www.nextplatform.com/2015/10/07/monitoring-is-different-for-ephemeral-infrastructure/>. Abruf: 24.02.2016.

So wurden Container in letzter Zeit häufig als Zukunft der Anwendungsentwicklung gesehen und bieten beispielsweise den Entwicklern eine der Zielumgebung identische und gut skalierbare Landschaft. Durch Continuous Delivery lassen sich Container automatisiert bereitstellen, aktualisieren und auch wieder zerstören. Mithilfe einer so geschaffenen, konsistenten Umgebung lassen sich Konflikte mit anderen Anwendungen vermeiden und Applikationen einfacher portieren. Die Kosten im eigenen Haus können durch agile Softwareentwicklung gesenkt und die Qualität maßgeblich gesteigert werden.^{24,25,26}

²⁴Vgl. ebenda, [RB14]

²⁵Vgl. ebenda, [NE16]

²⁶Vgl. ebenda, [FP15]

3 Techniken zur Optimierung

Da bei den zuvor genannten Möglichkeiten der Virtualisierung stets ein zusätzlicher Layer zwischen Hard- und Software implementiert wird, welcher selbst Ressourcen benötigt und gegebenenfalls auch Datenraten beeinflusst, werden in diesem Zusammenhang häufig auch weitere Techniken genutzt, die der Optimierung dienen. Abhängig von der Art der Virtualisierung und der vom Hersteller implementierten Features lassen sich meist nur ein Teil dieser Techniken nutzen.

3.1 DirectPath I/O

DirectPath I/O – auch VMDirectPath oder PCI Passthrough genannt – ist ein Feature, welches einem Gastsystem einen direkten Zugriff auf I/O-Geräte erlaubt. Dadurch erhält die VM die Möglichkeit, den Virtualisierungslayer zu umgehen und direkt auf physische PCI-Funktionen zuzugreifen. Da bei Nutzung dieser Technik etliche Nachteile entstehen können, wie beispielsweise dem Wegfall der Hochverfügbarkeit oder von Snapshots, sollte diese nur bei absoluter Notwendigkeit eingesetzt werden. Muss eine virtuelle Maschine beispielsweise direkt auf einen USB-Port zugreifen, so ist dieser Port auf anderer Hardware gegebenenfalls nicht vorhanden und die VM kann so nicht mehr oder nur mit administrativem Aufwand auf andere Hardware migriert werden.^{27,28,29}

3.2 Write Combining

Mittels Write Combining können Daten temporär in einem Buffer – dem Write Combine Buffer (WCB) – gespeichert und später gebündelt in den Arbeitsspeicher geschrieben werden. Dadurch wird die Anzahl der Zugriffe auf den Arbeitsspeicher verringert und der Anwendung ein Zeitfenster zur Verfügung gestellt, in welchem weitere Daten geliefert werden können. Sendet die Applikation beispielsweise vier Pakete mit je vier Kilobyte, so werden diese im WCB gesammelt und als ein Paket mit 16 Kilobyte an den Arbeitsspeicher weitergeleitet. In modernen CPUs gibt es neben L1- und L2-Caches mehrere separate Write Combine Buffer.^{30,31}

²⁷Vgl. ebenda, [WF16b]

²⁸Vgl. ebenda, [VM16c]

²⁹Vgl. ebenda, [SS11]

³⁰Vgl. ebenda, [IN98], S. 5ff

³¹Vgl. ebenda, [IN09], S. 158ff

3.3 DMA

Direct Memory Access (DMA) dient der Verbesserung der Performance und der Minimierung des Overheads, indem die Kommunikation zwischen einer Komponente und dem Arbeitsspeicher auf direktem Weg hergestellt wird. In der Regel muss der Prozessor den Schreibzugriff der betroffenen Komponente in den Arbeitsspeicher veranlassen, wobei die Daten erst in das interne Register gelesen und dann durch die CPU in den Arbeitsspeicher geschrieben werden. Diese Verarbeitungsschritte verringern unter Umständen die Ausführungsgeschwindigkeit laufender Programme und können durch den direkten Speicherzugriff vom Prozessor ferngehalten werden. Über spezielle Leitungen auf dem Mainboard wird eine direkte Verbindung zwischen dem Peripheriegerät und dem Arbeitsspeicher hergestellt und neben der Entlastung der CPU gleichzeitig eine schnellere Datenübertragung erreicht.^{32,33} Diese Technik erlaubt auch eine transparente Übertragung über das Netzwerk (remote DMA), wobei von einem System ein direkter Zugriff auf den Arbeitsspeicher eines anderen Systems ermöglicht wird. Ein mit RDMA häufig in Verbindung gebrachtes Modell – das in Abbildung 5 dargestellte Kernel Bypass Modell – erlaubt der Anwendung mittels standardisierter Protokolle zusätzlich direkten Zugriff auf die Hardware. Zum Einsatz kommen dabei das Sockets Direct Protocol (SDP) und das Upper Layer Protocol (ULP).³⁴

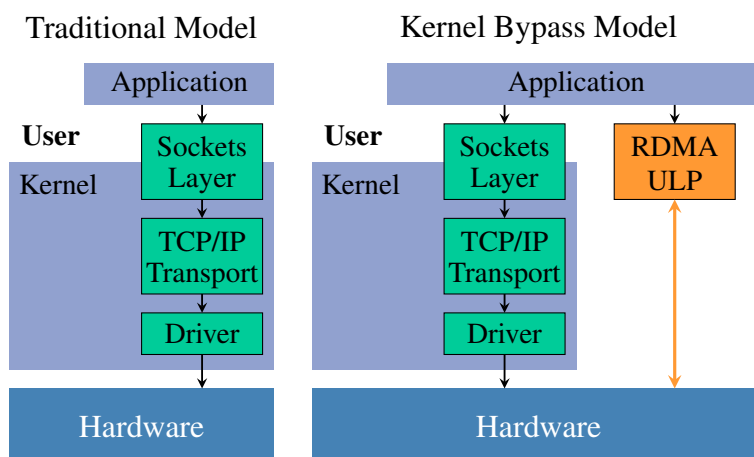


Abbildung 5: RDMA und ULP im Kernel Bypass Modell

M. Risca, D. Malik, A. Kessler: *Trading Floor Architecture*

http://www.cisco.com/c/en/us/td/docs/solutions/Verticals/Trading_Floor_Architecture-E.html, Abruf: 26.02.2016.

³²Vgl. ebenda, [WI16f]

³³Vgl. ebenda, [EK16]

³⁴Vgl. ebenda, [WI16g]

Greift man aus einer solchen Umgebung direkt auf Komponenten wie dem Arbeitsspeicher zu, scheint die Isolation der virtuellen Umgebung auf den ersten Blick deutlich eingeschränkt oder gar verloren. Hierfür wurden jedoch Vorkehrungen getroffen, welche durch das sogenannte DMA Remapping umgesetzt werden. Das Remapping bietet eine Möglichkeit, den Speicherzugriff selbst zu isolieren und mittels voneinander unabhängiger I/O Page Tables jedem Gerät eine bestimmte Domain zuzuweisen. Eine Domain stellt eine isolierte Umgebung dar, welche eine Teilmenge des real verfügbaren Arbeitsspeichers beinhaltet. Der Zugriff auf den Speicher wird abgefangen, die Anfrage überprüft und anschließend an die korrekte physikalische Speicheradresse weitergeleitet. (Vgl. Abbildung 6) Zuvor wird sichergestellt, dass die Berechtigungen zum Zugriff auf die physikalische Adresse vorhanden sind.^{35,36} Abhängig vom Ressourcenverbrauch lassen sich also eine bestimmte Anzahl virtueller Umgebungen mit DMA betreiben, ohne dass die Isolation darunter leidet.

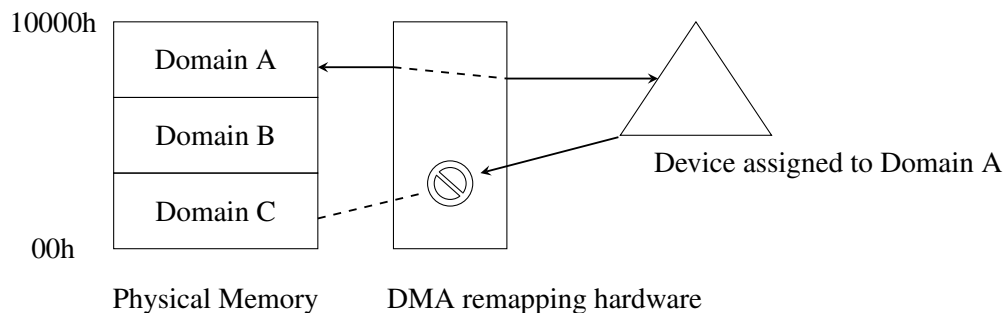


Abbildung 6: DMA Remapping

Mit Änderungen entnommen aus: Teresa Burger: *Intel Virtualization Technology for Directed I/O (VT-d)*
<https://software.intel.com/en-us/articles/intel-virtualization-technology-for-directed-io-vt-d-enhancing-intel-platforms-for-efficient-virtualization-of-io-devices>.
 Abruf: 25.03.2016.

³⁵Vgl. ebenda, [TB12]

³⁶Vgl. ebenda, [QS16]

3.4 SR-IOV

Die Technik Single-Root I/O Virtualization (SR-IOV) ermöglicht mehreren virtuellen Maschinen den nativen Zugriff auf ein gemeinsam genutztes I/O-Gerät, wobei dieses für jede VM als separates Gerät bereitgestellt wird und dabei jeweils eigene Adressregister und eigene Queues besitzt.^{37,38} Mithilfe dieser Technik sollen vor allem I/O-Engpässe beseitigt und höhere Datenraten realisiert werden,³⁹ wobei die in der Abbildung 7 veranschaulichte Virtualisierung von Netzwerkkarten eines der häufigsten Einsatzgebiete darstellt. Mittels der Multi-Root I/O Virtualization (MR-IOV) lassen sich I/O-Geräte auch über mehrere physikalische Systeme verteilt nutzen, was vor allem bei Blade-Systemen zum Einsatz kommt.⁴⁰

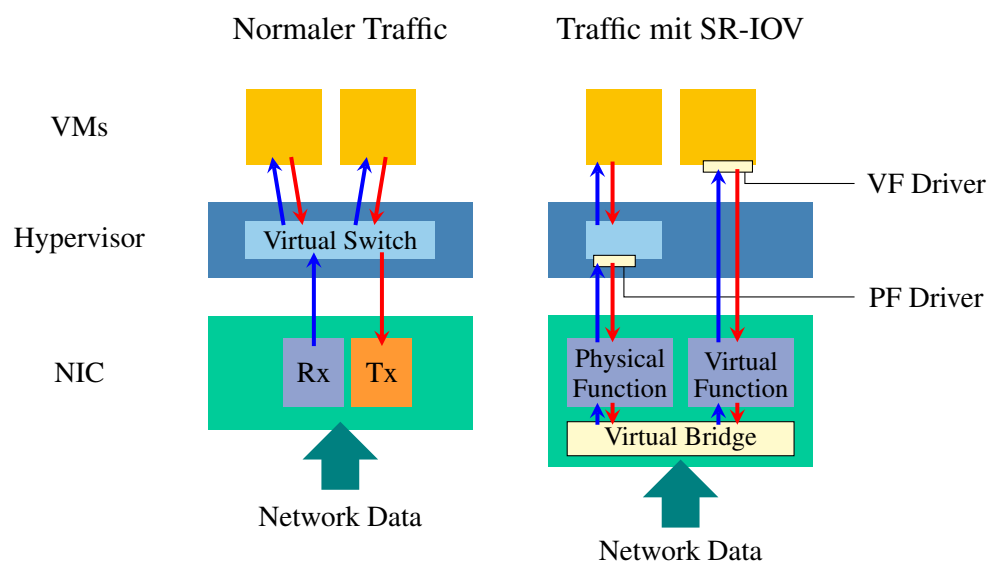


Abbildung 7: Netzwerkfluss in virtuellen Umgebungen

John Savill: *Traffic flow with traditional Hyper-V networking and with SR-IOV*

<http://windowsitpro.com/virtualization/understanding-windows-server-2012-hyper-v-networking-changes>.

Abruf: 26.02.2016.

³⁷Vgl. ebenda, [WF16a]

³⁸Vgl. ebenda, [RH16]

³⁹Vgl. ebenda, [IT16b]

⁴⁰Vgl. ebenda, [MR13]

3.5 NUMA Aware Scheduling

NUMA steht für Non-Uniform Memory Access und kommt in speziellen Architekturen zum Einsatz. In diesen Mehrprozessor-Architekturen befinden sich mehrere Systembusse, von denen jeder eine Gruppe der verfügbaren Prozessoren bedient, welche wiederum über ihren eigenen Arbeitsspeicher verfügen. (Vgl. Abbildung 8) Jede CPU kann dabei auf kohärente Weise auf den Arbeitsspeicher der anderen Gruppen zugreifen. Durch den lokal verfügbaren Speicher werden deutlich schnellere Zugriffszeiten ermöglicht. Ein NUMA-Scheduler managed in einer solchen Umgebung die virtuellen Maschinen und sorgt so für einen dynamischen Ausgleich von Prozessorlast und Arbeitsspeicherbelegung.^{41,42,43}

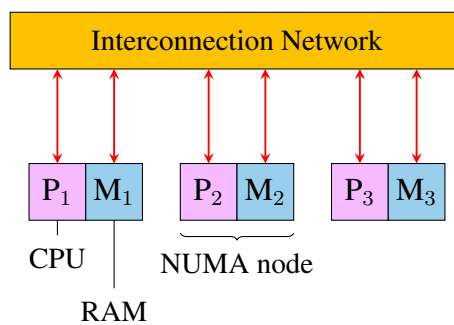


Abbildung 8: Schematische Darstellung einer NUMA-Architektur

M. Doubek, J. Wells: *An Overview of MPP Systems*
<http://www.blight.com/ja-wells/311/report.html>.
Abruf: 03.03.2016.

⁴¹Vgl. ebenda, [VM16d]

⁴²Vgl. ebenda, [MS16]

⁴³Vgl. ebenda, [XE16]

4 Pro und Contra Virtualisierung

Neben den bereits erwähnten Einsparungen der Rechenzentrumsfläche und der Energiekosten bietet der Einsatz von Virtualisierungslösungen viele weitere Vorteile. Während im Jahr 2010 noch nahezu die Hälfte der Unternehmen Zweifel an den Techniken äußerte und die Komplexität, Mehrkosten und den erhöhten Personalaufwand als Risiken der Virtualisierung nannte, findet man heute kaum noch Gegner.⁴⁴ Dies liegt auch daran, dass die meisten Argumente im Laufe der Zeit entkräftet werden konnten und man durch Vergleichswerte, etablierte Lizenzmodelle und vor allem auch durch die Erfahrungen anderer Unternehmen den eigenen Aufwand und die Kosten heutzutage viel besser planen kann, als noch vor einigen Jahren. So führte eine Befragung von 268 Unternehmen aus Deutschland, Österreich und der Schweiz zu einem in der Abbildung 9 wiedergegebenen, deutlichen Ergebnis.

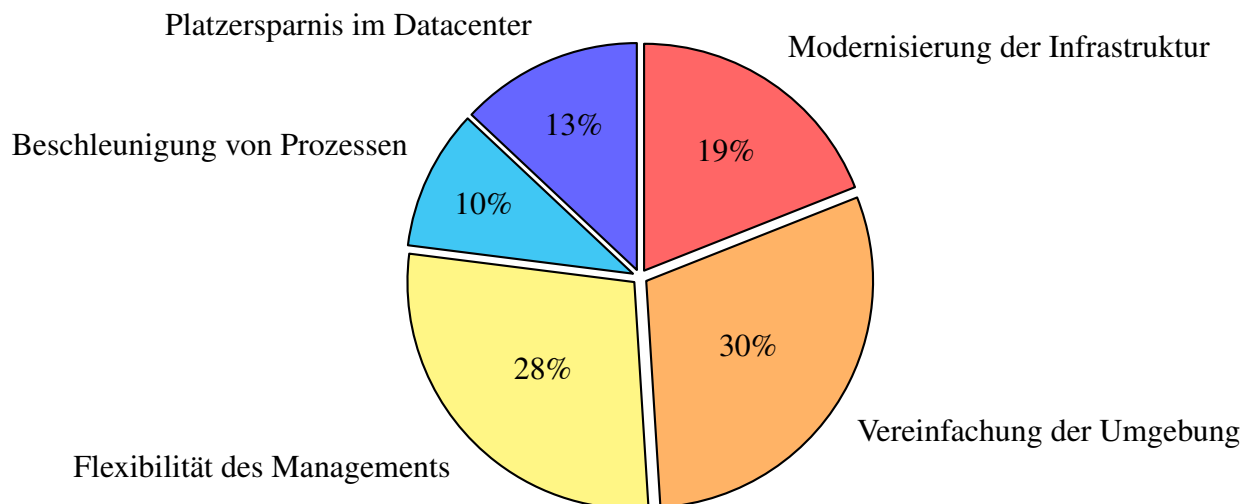


Abbildung 9: Hauptvorteile einer virtualisierten Umgebung

NetApp Deutschland GmbH: *Pro und Contra bei Virtualisierung*

<https://www.all-about-security.de/security-artikel/applikations-host-sicherheit/virtualisierung/artikel/10899-netapp-befragt-unternehmen-zu-pro-und-contra-bei-virtualisie/>

Abruf: 27.02.2016.

⁴⁴Vgl. ebenda, [HW10]

Ein Vorteil, welchen auch 30 Prozent der Befragten als Hauptvorteil aufführten, ist die breite Hardwarekompatibilität der Virtualisierungslösungen (Vgl. Abbildung 10), die damit geringere Bindung an einen Hersteller und damit auch eine gute Basis bei zukünftigen Preisverhandlungen.^{45,46} Mit einem intelligenten Ressourcenmanagement und der Möglichkeit, Prozesse wie das Deployment virtueller Maschinen oder auch der Container zu automatisieren, lassen sich interne Abläufe verbessern und gegebenenfalls sogar Personalkosten einsparen. Mit modernen Verfahren, wie der Notfallwiederherstellung (Disaster Recovery) oder einem automatischem Schwenk bei Störungen der Hardware sind virtuelle Maschinen selbst im Katastrophenfall gut abgesichert.⁴⁷ Container bieten ähnliche Techniken und lassen sich mit einfachen Mitteln oder auch mit zusätzlicher Software gegen Datenverluste absichern.^{48,49} Da sich zudem die Anzahl des Arbeitsspeichers oder auch die Größe der virtuellen Festplatten im laufenden Betrieb anpassen, sowie Netzwerkadapter jederzeit hinzufügen oder entfernen lassen, werden die Ausfallzeiten der Systeme und der administrative Aufwand minimal gehalten. Dass die Virtualisierung einen festen Platz in der IT eingenommen hat, zeigt sich auch daran, dass immer mehr geschäftskritische Anwendungen wie Datenbanken und SAP-Systeme in virtuelle Umgebungen migriert werden.⁵⁰ Welche Art dabei zum Einsatz kommt, hängt jedoch von vielen Faktoren ab.

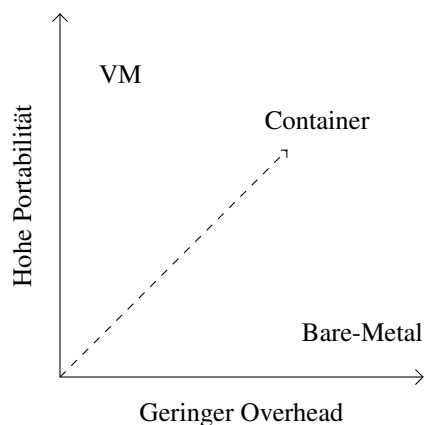


Abbildung 10: Schwerpunkte der Virtualisierungslösungen

Cedric Hurst: *10 Things to Know About Docker*

<http://www.spantree.net/blog/2015/04/29/10-things-to-know-about-docker.html>. Abruf: 05.03.2016.

⁴⁵Vgl. ebenda, [VM16e]

⁴⁶Vgl. ebenda, [VZ16]

⁴⁷Vgl. ebenda, [WI16h]

⁴⁸Vgl. ebenda, [AS16]

⁴⁹Vgl. ebenda, [LC16]

⁵⁰Vgl. ebenda, [RD15], S. 6

5 Performance

Trotz der vielen Vorteile hält sich ein Gegenargument seit Jahren hartnäckig: Die Performance soll in Gastsystemen schlechter und der Unterschied sogar spürbar sein.^{51,52} Die möglichen Ursachen für Performance-Engpässe sind jedoch vielfältig und eine komplexe Landschaft mit unzähligen Komponenten lässt sich nur mühsam analysieren. In den letzten Jahren haben sich etliche namhafte Unternehmen der Aufgabe gewidmet, die Performance verschiedener Virtualisierungslösungen zu analysieren, zu vergleichen und durch die Weiterentwicklung von Treibern, Betriebssystemen und anderen Komponenten zu verbessern. Ein nennenswertes Beispiel stellt VMware unter der Verwendung von InfiniBand und RDMA dar und zeigt damit, dass die Unterschiede zwischen virtuellen Maschinen und Bare Metal doch sehr gering sein können. (Vgl. Abbildung 11)

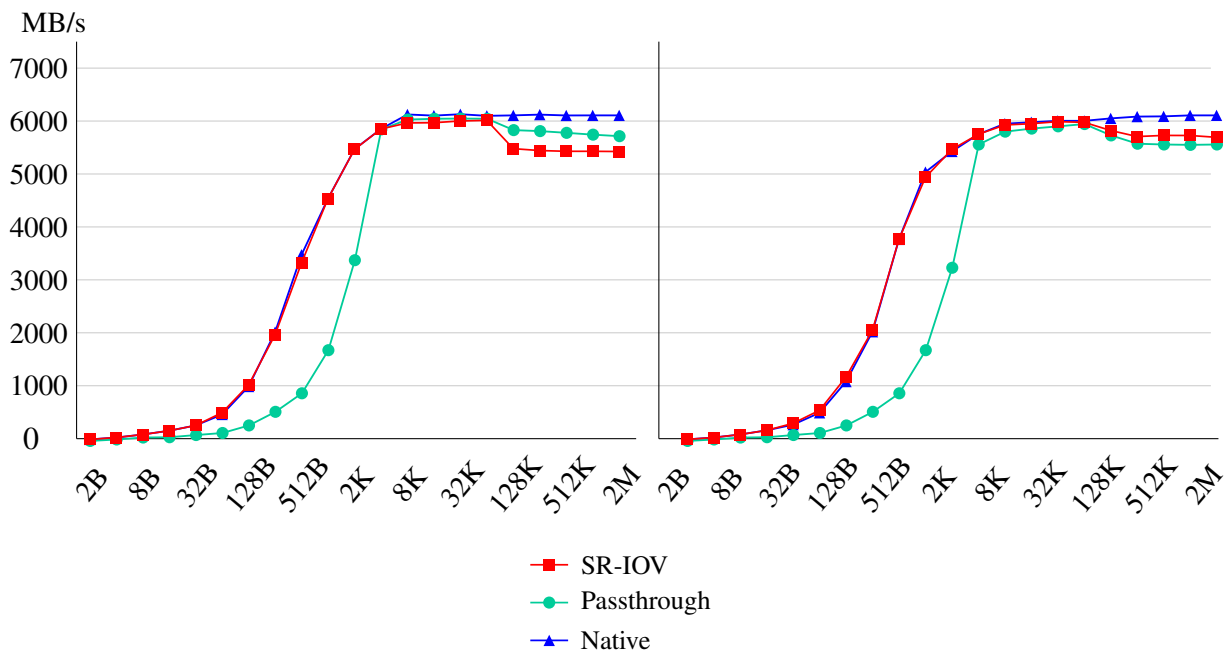


Abbildung 11: Bandbreite READ (links) und WRITE (rechts)

Josh Simons: *Running HPC Applications on vSphere using InfiniBand RDMA*
<https://blogs.vmware.com/cto/running-hpc-applications-vsphere-using-infiniband/>
Abruf: 01.03.2016.

Mittels Open Fabrics Enterprise Distribution (OFED) Micro-Benchmark, welcher aus meh-

⁵¹Vgl. ebenda, [JT16]

⁵²Vgl. ebenda, [DD16]

renen Tests zur Bestimmung der Performance eines Clusters besteht, konnte gezeigt werden, dass beim Einsatz von VMware ESX nahezu 100 Prozent der Leistung erreicht werden kann, welche man ohne die Virtualisierung hätte. Trotz der in der Abbildung schlechteren Ergebnisse beim Einsatz von Passthrough soll sich dieser Overhead mittels Write Combining vollständig eliminieren lassen. Lediglich im Bereich sehr großer Pakete ist der Overhead bei allen unterstützten Techniken deutlich.^{53,54} Die verwendete Teststellung aus vier, in einem Clusterverbund betriebenen Systemen ähnelt der Hardware, welche man bei mittelständischen IT-Providern findet und stellt keinesfalls eine Besonderheit dar. Um ein genaueres Bild für den Bereich des High Performance Computing zu erhalten, hat das texanische Unternehmen DELL im Jahr 2014 einen Test angestellt und die Containerlösung Docker, den Hypervisor QEMU KVM und OpenStack mit Bare-Metal-Systemen verglichen. Dabei konnte mit Docker im Durchschnitt 99 und mit VMs lediglich 87 Prozent der Performance eines Bare-Metal-Systems erreicht werden, wie in Abbildung 12 deutlich zu erkennen ist.⁵⁵

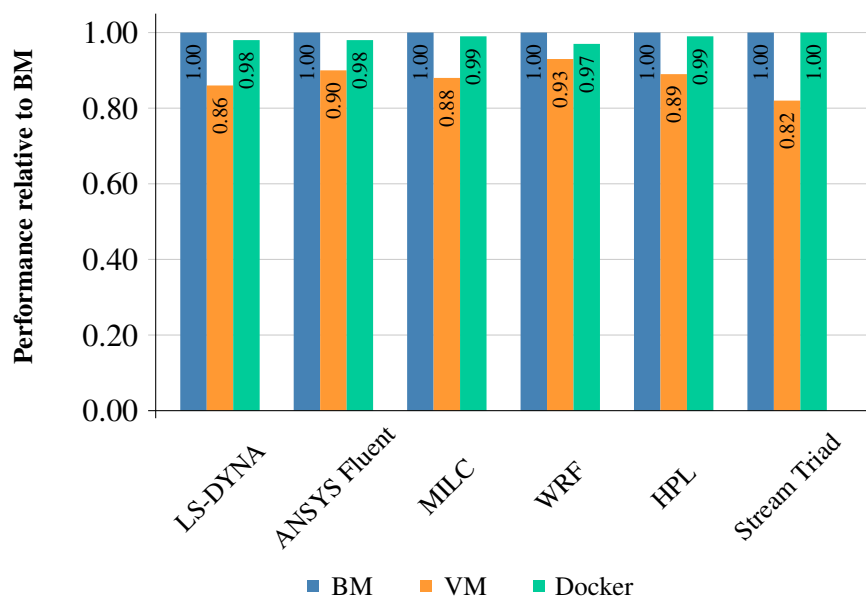


Abbildung 12: Vergleich der Performance von Containern, VMs und Bare-Metal-Systemen

N. Dandapanthula, J. Stanfield: *Application performance of Containers and VMs relative to BMs*
http://en.community.dell.com/techcenter/high-performance-computing/b/general_hpc/archive/2014/11/04/containers-docker-virtual-machines-and-hpc. Abruf: 03.03.2016.

Obwohl QEMU KVM in der Branche kaum genutzt wird und selbst im Vergleich zwischen

⁵³Vgl. ebenda, [JS14]

⁵⁴Vgl. ebenda, [OF08]

⁵⁵Vgl. ebenda, [ND14]

verschiedenen Hypervisoren nur durchschnittliche Performance liefert, lässt sich aufgrund der in dieser Versuchsreihe angewandten und auf das High Performance Computing abgestimmten Tests ein klares Ergebnis ablesen.^{56,57} Wie auch durch viele weitere Vergleiche belegt wird, ist beim Einsatz eines Hypervisoren mit einem Overhead von circa fünf Prozent zu rechnen, während man bei Containerlösungen nahezu ohne Overhead auskommt.^{58,59,60}

In der in Abbildung 13 dargestellten, selbst durchgeführten Versuchsreihe wurde die Performance eines Bare-Metal-Systems mit den aktuellen Versionen von Docker und VMware verglichen. Aufgrund der durch den Intel Optimized LINPACK Benchmark⁶¹ ermittelten Werte ließen sich die bisher veröffentlichten Tests und die einschlägig bekannten Performanceeinbußen durch den Einsatz eines Hypervisoren bestätigen. Überraschend war jedoch, dass die Messergebnisse innerhalb des Docker Containers vor allem bei großen Problemgrößen und Dimensionen denen des Bare-Metal-Systems sehr nahe waren und teils sogar übertrafen. Um kurzzeitige Peaks auszuschließen, wurde für die Auswertung der Durchschnitt aus je 40 Messungen pro Problem Size und Plattform gebildet.

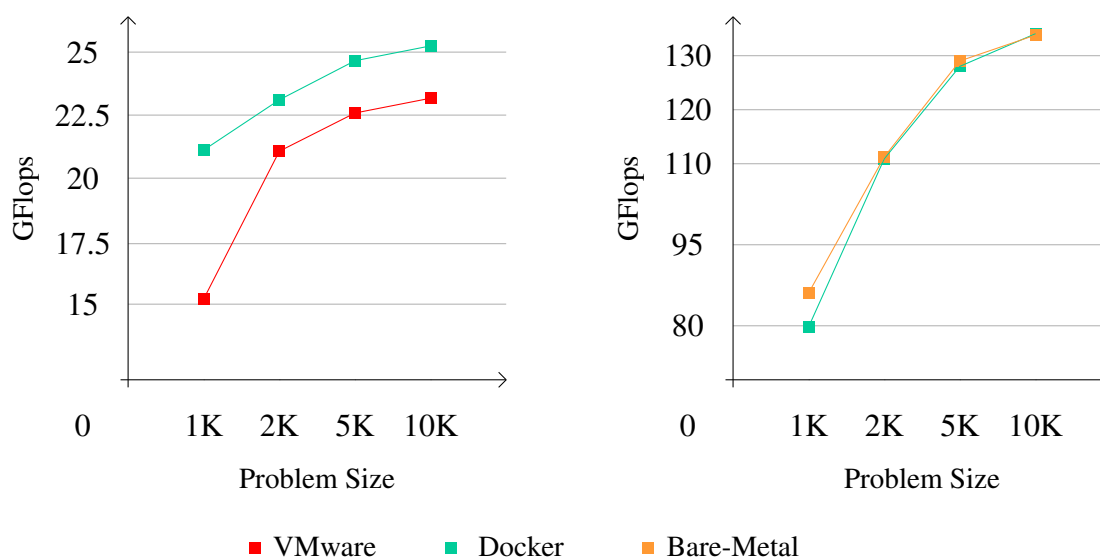


Abbildung 13: Vergleich der Performance von Containern, VMs und Bare-Metal-Systemen
(vom Autor erstellt)

⁵⁶Vgl. ebenda, [BG13]

⁵⁷Vgl. ebenda, [MJ14]

⁵⁸Vgl. ebenda, [VM16f]

⁵⁹Vgl. ebenda, [BA12]

⁶⁰Vgl. ebenda, [MF14]

⁶¹[ZZ15]

6 Fazit

Deutlich hervorzuheben ist, dass jede Art der Virtualisierung Vor- und Nachteile besitzt und es im direkten Vergleich keinen klaren Sieger geben kann. So muss vor einer jeden Entscheidung geprüft werden, unter welchen Voraussetzungen die Virtualisierung eingesetzt werden soll, ob die Performance oder die Sicherheit und Isolation höher gewichtet werden und ob der Einsatz von quelloffenen Produkten infrage kommt. Meist spielen dabei das vorhandene Budget und auch politische Entscheidungen eine große Rolle. Im Rahmen der Recherchen ist deutlich geworden, dass es sich bei Containern nicht nur um einen kurzweiligen Trend handelt, sondern um eine erwachsen gewordene Technik, die etliche dieser Entscheidungen beeinflussen wird. Deutlich wurde auch, dass die Performance einen der größten Unterschiede darstellt, durch die Weiterentwicklung der Hypervisoren jedoch davon auszugehen ist, dass die Performance in der nächsten Zeit deutlich bessere Werte erzielen wird und so eventuell mit Containerlösungen gleichauf sein könnte. Ein solcher Fortschritt lässt sich beispielsweise bei den Produkten von VMware erkennen. So konnte durch neue Features, wie der Verwendung von NUMA I/O bis zu 15 Prozent Performancegewinn erzielt werden.⁶² Computing as a Service wird in der schnelllebigen Gesellschaft weiter Einfluss ausüben und es bleibt abzuwarten, wie sich der Markt entwickelt und welche neuen Techniken dabei entstehen.^{63,64} In aktuellen Beiträgen zur Trendforschung ist die effizientere Virtualisierung stets ein Thema und die Anzahl der Unternehmen, die auf diesen Zug aufspringen, wird wachsen. Beispiele lassen sich bereits heute etliche finden, wie Kubernetes by Google⁶⁵, Apache CloudStack⁶⁶ und auch UberCloud⁶⁷ zeigen. Bei all diesen Themen, Trends und Entwicklungen sollte stets klar sein, welche Ziele man mit dem Einsatz der Virtualisierung verfolgt und was einem dabei wichtiger ist – minimalen Overhead oder maximale Flexibilität. Geeignet im Bereich des High Performance Computing sind diese Techniken jedoch auf jeden Fall.

⁶²Vgl. ebenda. [RM15]

⁶³Vgl. ebenda, [RW16a]

⁶⁴Vgl. ebenda, [RW16b]

⁶⁵[GO16]

⁶⁶[AP16]

⁶⁷[UC16]

Literatur

- [BS12] B. Bitner and S. Greenlee: *z/VM - A Brief Review of Its 40 Years History*.
<http://www.vm.ibm.com/vm40hist.pdf>. Abruf: 23.02.2016.
- [AM16] *Amazon Elastic Compute Cloud (EC2) - Dokumentation*.
<http://aws.amazon.com/de/documentation/ec2/>. Abruf: 26.02.2016.
- [VM16a] *vSphere Hypervisor*.
<https://www.vmware.com/de/products/vsphere-hypervisor/>. Abruf: 24.02.2016.
- [DO16] *Docker Docs*.
<https://docs.docker.com/>. Abruf: 26.02.2016.
- [OS16] *OpenStack Docs*.
<http://docs.openstack.org/>. Abruf: 26.02.2016.
- [GH16] *The Impact of Green Data Centers*.
<https://www.greenhousedata.com/green-data-centers>. Abruf: 26.02.2016.
- [FA16] Francois Ajenstat: *Get VIRTUAL Now - Virtualization and „Green IT“*.
<http://blogs.msdn.com/b/microsoft-green/archive/2008/09/09/get-virtual-now-virtualization-and-green-it.aspx>. Abruf: 26.02.2016.
- [AB13] Alex Benik: *The sorry state of server utilization and the impending post-hypervisor era*.
<https://gigaom.com/2013/11/30/the-sorry-state-of-server-utilization-and-the-impending-post-hypervisor-era/>. Abruf: 26.02.2016.
- [WI16a] *Virtualisierung (Informatik)*.
[https://de.wikipedia.org/wiki/Virtualisierung_\(Informatik\)](https://de.wikipedia.org/wiki/Virtualisierung_(Informatik)). Abruf: 24.02.2016.
- [WI16b] *Hypervisor*.
<https://de.wikipedia.org/wiki/Hypervisor>. Abruf: 24.02.2016.
- [IT16a] *Hypervisor :: ITWissen.info*.
<http://www.itwissen.info/definition/lexikon/virtual-machine-monitor-VMM.html>.
Abruf: 24.02.2016.

- [IB16] *IBM: z/VM Hypervisor.*
<http://www.vm.ibm.com/>. Abruf: 24.02.2016.
- [MR14] Margaret Rouse: *Typ-2-Hypervisor (Hosted Hypervisor).*
<http://www.searchdatacenter.de/definition/Typ-2-Hypervisor-Hosted-Hypervisor>.
Abruf: 24.02.2016.
- [VI16] *Oracle VM VirtualBox User Manual.*
<http://download.virtualbox.org/virtualbox/5.0.14/UserManual.pdf>.
Abruf: 24.02.2016.
- [VM16b] *VMware Workstation Features.*
<https://www.vmware.com/products/workstation/features.html>. Abruf: 24.02.2016.
- [WI16c] *Operating-system-level virtualization.*
https://en.wikipedia.org/wiki/Operating-system-level_virtualization.
Abruf: 24.02.2016.
- [PI16] *Moments In Container History.*
<http://pivotal.io/platform/infographic/moments-in-container-history>.
Abruf: 24.02.2016.
- [AR16] *Change root - ArchWiki.*
https://wiki.archlinux.org/index.php/change_root. Abruf: 24.02.2016.
- [SP16] S. Pasewaldt, M. Koeleman, R. Reicherdt: *Solaris 10 Zones.*
<http://www.dcl.hpi.uni-potsdam.de/teaching/osAdminSem05/Solaris%20Zones.pdf>.
Abruf: 25.03.2016.
- [RS11] Ralf Spenneberg: *Cgroups zur Ressourcenkontrolle in Linux.*
<http://www.admin-magazin.de/Das-Heft/2011/06/Cgroups-zur-Ressourcenkontrolle-in-Linux>. Abruf: 24.02.2016.
- [WI16d] *cgroups.*
<https://en.wikipedia.org/wiki/Cgroups>. Abruf: 24.02.2016.
- [WI16e] *Namespace.*
<https://en.wikipedia.org/wiki/Namespace>. Abruf: 24.02.2016.

- [RB14] René Büst: *Docker-Container: Die Zukunft moderner Applikationen und Multi-Cloud Deployments?*.
<http://www.computerwoche.de/a/docker-container-die-zukunft-moderner-applikationen-und-multi-cloud-deployments,3065454>. Abruf: 29.02.2016.
- [NE16] *Jetzt mit Docker durchstarten*.
<https://www.neofonie.de/docker/>. Abruf: 29.02.2016.
- [FP15] Frank Pientka: *Wie DevOps die IT beschleunigen*.
<http://www.computerwoche.de/a/wie-devops-die-it-beschleunigen,3071433>.
Abruf: 29.02.2016.
- [WF16b] Werner Fischer: *VMware VMDirectPath zum Durchreichen von PCI Karten*.
https://www.thomas-krenn.com/de/wiki/VMware_VMDirectPath_zum_Durchreichen_von_PCI_Karten. Abruf: 26.02.2016.
- [VM16c] *DirectPath I/O*.
<https://pubs.vmware.com/vsphere-51/index.jsp?topic=%2Fcom.vmware.vsphere.networking.doc%2FGUID-BF2770C3-39ED-4BC5-A8EF-77D55EFE924C.html>. Abruf: 26.02.2016.
- [SS11] Sebastian Solbach: *Oracle VM - Konfiguration von PCI Passthrough*.
http://www.oracle.com/webfolder/technetwork/de/community/dbadmin/tipps/ovm_pcipassthrough/index.html. Abruf: 26.02.2016.
- [WI16f] *Speicherdirektzugriff*.
<https://de.wikipedia.org/wiki/Speicherdirektzugriff>. Abruf: 26.02.2016.
- [EK16] *DMA - Direct Memory Access*.
<http://www.elektronik-kompodium.de/sites/com/0808131.htm>.
Abruf: 26.02.2016.
- [WI16g] *Sockets Direct Protocol*.
https://de.wikipedia.org/wiki/Sockets_Direct_Protocol. Abruf: 26.02.2016.

- [TB12] Teresa Burger: *Intel Virtualization Technology for Directed I/O (VT-d)*.
<https://software.intel.com/en-us/articles/intel-virtualization-technology-for-directed-io-vt-d-enhancing-intel-platforms-for-efficient-virtualization-of-io-devices>. Abruf: 26.02.2016.
- [QS16] Dr. Qingni Shen *Hardware virtualization technology and its security*.
<http://www.ss.pku.edu.cn/vs/style/resources/Hardware%20virtualization%20technology%20and%20its%20security.pdf>. Abruf: 25.03.2016.
- [WF16a] Werner Fischer: *Virtualisierungsfunktion SR-IOV aktivieren*.
https://www.thomas-krenn.com/de/wiki/Virtualisierungsfunktion_SR-IOV_aktivieren. Abruf: 26.02.2016.
- [RH16] *Chapter 13. SR-IOV*.
https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Virtualization_Host_Configuration_and_Guest_Installation_Guide/chap-Virtualization_Host_Configuration_and_Guest_Installation_Guide-SR_IOV.html.
Abruf: 26.02.2016.
- [IT16b] *SR-IOV (single root I/O virtualization)*.
<http://www.itwissen.info/definition/lexikon/SR-IOV-single-root-I-O-virtualization.html>. Abruf: 26.02.2016.
- [MR13] Margaret Rouse: *Multi-Root IOV (MR-IOV)*.
<http://whatis.techtarget.com/definition/Multi-Root-IOV-MR-IOV>.
Abruf: 26.02.2016.
- [VM16d] *So funktioniert die ESXi-NUMA-Planung*.
<https://pubs.vmware.com/vsphere-51/index.jsp?topic=%2Fcom.vmware.vsphere.resmgmt.doc%2FGUID-BD4A462D-5CDC-4483-968B-1DCF103C4208.html>. Abruf: 01.03.2016.
- [MS16] *Grundlegendes zu NUMA (Non-Uniform Memory Access)*.
[https://technet.microsoft.com/de-de/library/ms178144\(v=sql.105\).aspx](https://technet.microsoft.com/de-de/library/ms178144(v=sql.105).aspx).
Abruf: 01.03.2016.

- [XE16] *Xen 4.3 NUMA Aware Scheduling.*
http://wiki.xenproject.org/wiki/Xen_4.3_NUMA_Aware_Scheduling.
Abruf: 01.03.2016.
- [IN98] *Write Combining Memory Implementation Guidelines.*
<ftp://download.intel.com/design/PentiumII/applnots/24442201.pdf>.
Abruf: 03.03.2016.
- [IN09] *Intel 64 and IA-32 Architectures Optimization Reference Manual.*
<http://www.naic.edu/~phil/software/intel/248966.pdf>. Abruf: 03.03.2016.
- [HW10] Hartmut Wiehr: *Pro und Contra: Die befürchteten Risiken bei Virtualisierung.*
<http://www.cio.de/a/die-befuerchteten-risiken-bei-virtualisierung,2232059>.
Abruf: 27.02.2016.
- [RD15] Ralf Damschek: *Positionierung von vsom über ein Assessment beim Kunden.*
<http://docplayer.org/678555-Positionierung-von-vsom-ueber-ein-assessment-beim-kunden.html>. Abruf: 27.02.2016.
- [VM16e] *VMware Compatibility Guide.*
<http://www.vmware.com/resources/compatibility/search.php>. Abruf: 27.02.2016.
- [VZ16] *Requirements - Virtuozzo.*
<http://www.virtuozzo.com/products/virtuozzo/requirements/>. Abruf: 27.02.2016.
- [WI16h] *Disaster Recovery.*
https://de.wikipedia.org/wiki/Disaster_Recovery. Abruf: 27.02.2016.
- [AS16] *Docker Container Backup and Recovery.*
<http://www.asigra.com/solutions/docker-container-environments>.
Abruf: 27.02.2016.
- [LC16] *Docker container: Backup and Recovery.*
<https://linuxconfig.org/docker-container-backup-and-recovery>. Abruf: 27.02.2016.
- [JT16] Jom Toigo: *The struggle between virtual machine performance and storage.*
<http://searchvirtualstorage.techtarget.com/opinion/The-struggle-between-virtual-machine-performance-and-storage>. Abruf: 27.02.2016.

- [DD16] David Davis: *10 ways to improve virtual server storage*.
<http://searchvirtualstorage.techtarget.com/feature/10-ways-to-improve-virtual-server-storage>. Abruf: 27.02.2016.
- [JS14] Josh Simons: *Running HPC Applications on vSphere using InfiniBand RDMA*.
<https://blogs.vmware.com/cto/running-hpc-applications-vsphere-using-infiniband/>.
Abruf: 03.03.2016.
- [OF08] *Open Fabrics Enterprise Distribution (OFED) Performance Tests README for OFED 1.4*.
http://downloads.openfabrics.org/OFED/archive/ofed-1.4/OFED-1.4-docs/PERF_TEST_README.txt. Abruf: 03.03.2016.
- [ND14] Nishanth Dandapanthula: *Containers, Docker, Virtual Machines and HPC*.
http://en.community.dell.com/techcenter/high-performance-computing/b/general_hpc/archive/2014/11/04/containers-docker-virtual-machines-and-hpc. Abruf: 03.03.2016.
- [BG13] Brendan Gregg: *Virtualization Performance: Zones, KVM, Xen*.
<http://dtrace.org/blogs/brendan/2013/01/11/virtualization-performance-zones-kvm-xen/>. Abruf: 03.03.2016.
- [MJ14] Major Hayden: *Performance benchmarks: KVM vs. Xen*.
<https://major.io/2014/06/22/performance-benchmarks-kvm-vs-xen/>.
Abruf: 03.03.2016.
- [VM16f] *A Performance Comparison of Hypervisors*.
https://www.vmware.com/pdf/hypervisor_performance.pdf. Abruf: 03.03.2016.
- [BA12] Baptiste Assmann: *Hypervisors virtual network performance comparison from a virtualized load-balancer point of view*.
<http://blog.haproxy.com/2012/04/24/hypervisors-virtual-network-performance-comparison-from-a-virtualized-load-balancer-point-of-view/>. Abruf: 03.03.2016.

- [MF14] Markus Feilner: *IBM-Performance-Studie sieht Docker-Container vor KVM-Virtualisierung*.
<http://www.linux-magazin.de/NEWS/IBM-Performance-Studie-sieht-Docker-Container-vor-KVM-Virtualisierung>. Abruf: 03.03.2016.
- [ZZ15] Zhang Z.: *Intel Math Kernel Library Benchmarks (Intel MKL Benchmarks)*.
<https://software.intel.com/en-us/articles/intel-mkl-benchmarks-suite>.
Abruf: 04.03.2016.
- [RM15] Rishi Mehta: *Network Improvements in vSphere 6 Boost Performance for 40G NICs*.
<https://blogs.vmware.com/performance/2015/04/network-improvements-vsphere-6-boost-performance-40g-nics.html>. Abruf: 05.03.2016.
- [RW16a] Reimund Willig: *IT-Trends für 2016: Container, Clouds und Fast Big Data*.
<http://ap-verlag.de/it-trends-fuer-2016-container-clouds-und-fast-big-data/17187/>.
Abruf: 05.03.2016.
- [RW16b] Reimund Willig: *Die wichtigsten IT-Trends im neuen Jahr - ein kleiner Nachtrag*.
<https://blog.emc2.de/cloud/die-wichtigsten-it-trends-im-neuen-jahr-ein-kleiner-nachtrag>. Abruf: 05.03.2016.
- [GO16] *Kubernetes by Google*.
<http://kubernetes.io/>. Abruf: 05.03.2016.
- [AP16] *Apache CloudStack: Open Source Cloud Computing*.
<https://cloudstack.apache.org/>. Abruf: 05.03.2016.
- [UC16] *About UberCloud*.
<https://www.theubercloud.com/about-ubercloud/>. Abruf: 05.03.2016.

Abbildungsverzeichnis

| | | |
|---|---|----|
| 1 | auf virtuellen Maschinen erzeugter Workload | |
| | <i>Virtualization-Facts and Trends.</i> | |
| | http://de.slideshare.net/HPESoftwareSolutions/hp-software-universe-managing-virtualized-environments-agentless-and-agentbased-solutions-for-managing-vmware-environments-using-hp-sitescope-operations-manager-agents-and-virtualization-spi . Gartner, Abruf: 23.02.2016. | 4 |
| 2 | Verteilung eingesetzter Hypervisoren | |
| | Chris Wolf: <i>Heterogeneous Virtualization Trends at Gartner Data Center.</i> | |
| | http://blogs.gartner.com/chris-wolf/2012/12/11/heterogeneous-virtualization-trends-at-gartner-data-center/ . | |
| | Gartner, Abruf: 24.02.2016. | 7 |
| 3 | Schichtenmodell der Hypervisor-Architektur | |
| | <i>Hypervisor :: ITWissen.info</i> | |
| | http://www.itwissen.info/bilder/hypervisor-architekturen.png . | |
| | Abruf: 24.02.2016. | 7 |
| 4 | Lebensdauer eines Containers | |
| | Timothy Prickett Morgan: <i>Monitoring Is Different For Ephemeral Infrastructure</i> | |
| | http://www.nextplatform.com/2015/10/07/monitoring-is-different-for-ephemeral-infrastructure/ . Abruf: 24.02.2016. | 9 |
| 5 | RDMA und ULP im Kernel Bypass Model | |
| | M. Risca, D. Malik, A. Kessler: <i>Trading Floor Architecture</i> | |
| | http://www.cisco.com/c/en/us/td/docs/solutions/Verticals/Trading_Floor_Architecture-E.html , Abruf: 26.02.2016. | 11 |
| 6 | DMA Remapping | |
| | Mit Änderungen entnommen aus: Teresa Burger: <i>Intel Virtualization Technology for Directed I/O (VT-d)</i> | |
| | https://software.intel.com/en-us/articles/intel-virtualization-technology-for-directed-io-vt-d-enhancing-intel-platforms-for-efficient-virtualization-of-io-devices . | |
| | Abruf: 25.03.2016. | 12 |
| 7 | Netzwerkfluss in virtuellen Umgebungen | |
| | John Savill: <i>Traffic flow with traditional Hyper-V networking and with SR-IOV</i> | |
| | http://windowsitpro.com/virtualization/understanding-windows-server-2012-hyper-v-networking-changes . | |
| | Abruf: 26.02.2016. | 13 |

| | | |
|-----------|---|----|
| 8 | Schematische Darstellung einer NUMA-Architektur | |
| | M. Doubek, J. Wells: <i>An Overview of MPP Systems</i> | |
| | http://www.blight.com/ja-wells/311/report.html . | |
| | Abruf: 03.03.2016. | 14 |
| 9 | Hauptvorteile einer virtualisierten Umgebung | |
| | NetApp Deutschland GmbH: <i>Pro und Contra bei Virtualisierung</i> | |
| | https://www.all-about-security.de/security-artikel/applikations-host-sicherheit/virtualisierung/artikel/10899-netapp-befragt-unternehmen-zu-pro-und-contra-bei-virtualisier/ . | |
| | Abruf: 27.02.2016. | 15 |
| 10 | Schwerpunkte der Virtualisierungslösungen | |
| | Cedric Hurst: <i>10 Things to Know About Docker</i> | |
| | http://www.spantree.net/blog/2015/04/29/10-things-to-know-about-docker.html . Abruf: 05.03.2016. | 16 |
| 11 | Bandbreite READ (links) und WRITE (rechts) | |
| | Josh Simons: <i>Running HPC Applications on vSphere using InfiniBand RDMA</i> | |
| | https://blogs.vmware.com/cto/running-hpc-applications-vsphere-using-infiniband/ . | |
| | Abruf: 01.03.2016. | 17 |
| 12 | Vergleich der Performance von Containern, VMs und Bare-Metal-Systemen | |
| | N. Dandapanthula, J. Stanfield: <i>Application performance of Containers and VMs relative to BMs</i> | |
| | http://en.community.dell.com/techcenter/high-performance-computing/b/general_hpc/archive/2014/11/04/containers-docker-virtual-machines-and-hpc . Abruf: 03.03.2016. | 18 |
| 13 | Vergleich der Performance von Containern, VMs und Bare-Metal-Systemen | |
| | (vom Autor erstellt) | 19 |