

Virtualisierungstechnologien in Grid Rechenzentren

Stefan Freitag

stefan.freitag@tu-dortmund.de

Abstract: Kommerzielle und auch akademische Rechenzentren stehen vor der Einführung der Virtualisierungstechnologie oder nutzen diese bereits auf die ein oder andere Weise. Die Hauptmotivation liegt zumeist in der verbesserten Ausnutzung der vorhandenen Kapazitäten durch z.B. Serverkonsolidierung. Diese Arbeit beschreibt den Einfluß von Virtualisierung auf Grid Rechenzentren und weiterhin Integrationsmöglichkeiten auf zwei Ebenen des Grid Middleware Stacks.

Weiterhin wird ein Ansatz zur Virtualisierungsplattform-übergreifenden Erzeugung von Disk Images skizziert. Im Kontext des Übergangs von der Einreichung einfacher Jobs hin zur Einreichung virtueller Maschinen stellt dies eine wichtige Notwendigkeit dar.

1 Einleitung

Die Idee zur Virtualisierung von Hardware stammt aus den Tagen des Mainframe Computing [Stra59] und erlebte in den letzten Jahren einen enormen Zuwachs an Popularität. Insbesondere im Umfeld akademischer und kommerzieller Rechenzentren sind die Vorteile, die sich durch den Einsatz von Virtualisierung ergeben, von besonderem Interesse. Das Hauptziel liegt zumeist in der Steigerung der Gesamtauslastung der vorhandenen Rechen- und Speicherkapazitäten.

Langfristig werden auch Nutzer durch die Kombination von Grid Computing und Virtualisierung profitieren. So sind Nutzer nicht mehr gezwungen ihre Anwendungen an die heterogenen Ausführungsumgebungen auf den Ressourcen anzupassen, sondern liefern ihre Anwendung in einer virtuellen Maschine an die Ressourcenbetreiber aus.

Diese Arbeit richtet ihren Fokus auf den Nutzen von Plattformvirtualisierung für Grid Rechenzentren. Die hier vorgestellten Betrachtungen sind jedoch vollständig auf Cloud Computing übertragbar, welches sich in den letzten Jahren neben Grid Computing etabliert hat. Das Konzept des Cloud Computings besitzt Verbindungen zum Grid Computing und zu anderen Technologien wie dem Cluster Computing oder allgemein, den verteilten Systemen. Jedoch fehlt noch eine allgemein akzeptierte Definition [Fos08]. Insgesamt stehen Grids und Clouds vor ähnlichen Problemstellungen bei der effizienten und automatisierten Ressourcennutzung.

Zunächst ist festzustellen, dass Virtualisierung oftmals als Synonym für Plattformvirtualisierung verstanden wird. Der Begriff der Virtualisierung ist jedoch viel allgemeiner aufzufassen und gliedert sich in Plattformvirtualisierung [Ram04] und Ressourcenvirtualisierung.

Abschnitt 2 stellt die Virtualisierungsformen vor, danach folgt in Abschnitt 3 eine kurze Einführung in Grid Computing. Bisherige und andauernde Arbeiten auf dem Gebiet der Virtualisierung werden in Abschnitt 4 beleuchtet. Abschnitt 5 geht auf Integrationsmöglichkeiten von Virtualisierung in den Grid Middleware Stack ein. Die Arbeit schließt mit einer Zusammenfassung in Abschnitt 6.

2 Virtualisierung

Die Virtualisierungstechnologie fügt eine Abstraktionsschicht in den Hard- und Softwarestack ein. Innerhalb dieser neuen Schicht findet entweder eine Ressourcen-Dekomposition oder -Aggregation statt. Für Nutzer sowie Dienste oberhalb der Virtualisierungsschicht sind dies vollkommen transparente Prozesse. Aktuelle Beispiele für Ressourcenvirtualisierung sind etwa Virtual LANs und Storage Array Networks.

Netzwerkvirtualisierung Auf der Netzwerkschicht spricht man von interner und externer Virtualisierung. Interne Virtualisierung bietet Software-Containern wie virtuellen Maschinen auf einer Ressource eine Netzwerk-ähnliche Funktionalität an. Xen nutzt dieses Konzept um Gästen den Netzwerkzugriff über das Host-System zu ermöglichen ¹. Externe Virtualisierung verschmilzt mehrere Netzwerkpartitionen zu einer einzigen logischen. Bekannte Beispiele für externe Virtualisierung sind VPNs (Virtual Private Networks) [Vpn98] und VLANs (Virtual Local Area Networks) [Vla06]. Jede an einem VPN teilhabende Netzwerkpartition verfügt über einen sog. Gateway. Die Kommunikation zwischen Rechnern in den verschiedenen Netzwerkpartitionen wird über die Gateways getunnelt. Insgesamt erscheinen so die verschiedenen Partitionen für Nutzer innerhalb des VPNs als ein zusammenhängendes, lokales Netzwerk.

Speichervirtualisierung Die Speichervirtualisierung befaßt sich zumeist mit der Aggregation von physikalischem Speicher in größere, logische Einheiten. Hierbei findet Virtualisierung auf unterschiedlichen Ebenen statt: Server Ebene, Fabric Ebene, Storage Subsystem Ebene und Dateisystem Ebene.

Hinsichtlich des Flußes von Kontrollinformation und Daten gliedern sich hier die Virtualisierungsansätze in in-band, out-of-band und split-path [Tat03]. Im Gegensatz zum in-band Ansatz, fließen Kontrollinformation und Daten beim out-of-band (split-path) Ansatz über (teilweise) getrennte Kanäle (vgl. Abbildung 1). Der wesentliche Unterschied zwischen split-path und out-of-band liegt in der Anordnung der Kontroll-Einheit, welche aus intelligenten SAN-Switches oder speziellen Virtualisierungsappliances besteht.

Plattformvirtualisierung Plattformvirtualisierung reduziert für viele Ressourcenbetreiber langfristig gesehen Kosten wie Wartungs- oder Stromkosten bei gleichzeitiger Steigerung von Redundanz und Dienstgüte. So werden etwa vor anstehenden Wartungsarbeiten

¹<http://www.cl.cam.ac.uk/Research/SRG/netos/xen/>

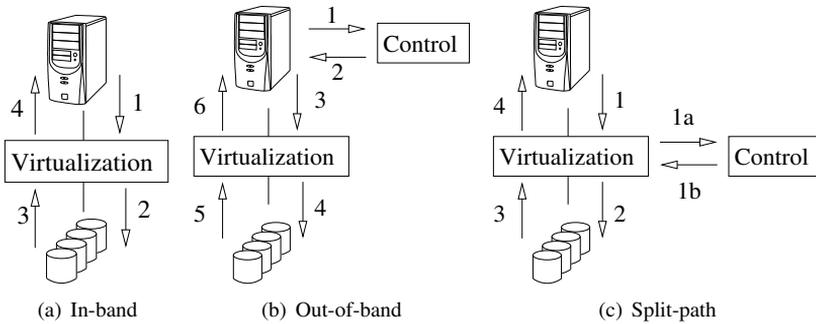


Abbildung 1: Kontroll- und Datenfluß bei Speichervirtualisierung

an physikalischen Hosts darauf laufende virtuelle Maschinen verschoben. Aus Sicht des Nutzers, dessen Anwendung in einer virtuellen Maschine ausgeführt wird, ist die Verschiebung ein transparenter Vorgang [Bha08], eine merkliche Nicht-Verfügbarkeit seiner Anwendung entsteht nicht.

Die Nutzung der Plattformvirtualisierung birgt auch Nachteile. Die gesamte Kommunikation der virtuellen Maschine mit der Außenwelt durchläuft die Virtualisierungsschicht, wodurch sich Performanz-Einbußen von bis zu 20% ergeben können [Tat06].

Standards in der Plattformvirtualisierung Die Bündelung von virtueller Maschine, Betriebssystem und Anwendung prägt den Begriff der Virtual Appliance [Vmw07a]. Virtual Appliances kapseln zudem Meta-Daten wie eine hersteller-neutrale Beschreibung und plattformabhängige Informationen zur Installation und Konfiguration. 2008 wurde der Open Virtual Machine Format (OVF) Standard [Ovf08a, Ovf08b] veröffentlicht. Dieser ermöglicht eine Hypervisor-neutrale Beschreibung von Mengen virtueller Maschinen, die als eine logische Einheit aufgefasst und aufgesetzt werden.

3 Grid Computing

Der Fokus beim Grid Computing liegt auf dem sicheren Zugriff auf entfernte Ressourcen (Rechenkraft, Software und Daten) in einer dynamischen, heterogenen Umgebung. Nach Foster et al. [Fos02] lassen sich Grids durch drei Eigenschaften charakterisieren: 1) Bereitstellung nicht-trivialer Dienstgüte, 2) Einsatz von offenen und standardisierten Protokollen und Schnittstellen, 3) Koordination von Ressourcen ohne zentralen Kontrolle.

An der Umsetzung einiger Charakteristika in den aktuellen Grid Middlewares wird noch gearbeitet, derzeit rückt der Einsatz von Standards mehr und mehr in den Vordergrund. So orientieren sich UNICORE [Rie05] und gLite hin zur Nutzung standardisierter Schnittstellen wie OGSA-BES [Bes07]. Langfristiges Ziel ist die Steigerung der Grid-Interoperabilität und somit der Job-Austausch über die Grenzen einer Grid Middleware hinaus.

Das Erreichen dieses Ziel scheint realistisch, da sich viele Grid Middlewares auf ein ge-

meinsames Schichtenmodell abbilden lassen. Die einzelnen Schichten sind in Abbildung 2 dargestellt. Die Infrastrukturebene enthält neben Rechen- und Speicher- ebenso Netzwer-

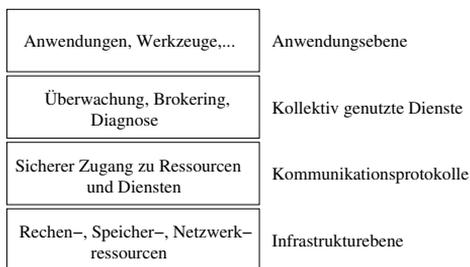


Abbildung 2: Mehr-Schichten Grid-Architektur (in Anlehnung an [Fos03], Seite 47)

kressourcen. Auch Software-Lizenzen können im weiteren Sinne unter dem Ressourcen-Begriff aufgefasst werden. Der Zugriff auf diese lokalen Ressourcen erfolgt über standardisierte Grid-Protokolle wie etwa SRM oder GSI-ssh.

Zudem muß es Grid-Nutzern über Mechanismen erlaubt sein, Aktionen auf der Ressource auszuführen. Im Falle von Rechenressourcen ist eine Job-Kontrolle (Starten/ Stoppen/ Anhalten) sowie -Überwachung sinnvoll. Des Weiteren muß der Transfer von Eingabe- und Ausgabedateien erlaubt sein. Bei Speicherressourcen sind Mechanismen zum Ablegen und Abholen von Dateien notwendig, diese können um Verfahren zur Speicherverwaltung und Advanced Reservation ergänzt werden.

4 Existierende Arbeiten

Plattformvirtualisierung wird im Bereich des Grid Computing vornehmlich zur Server-Konsolidierung und Bereitstellung von HA Diensten verwendet [Car07].

Die Globus Alliance entwickelte den Virtual Workspace Service [Kea05, Fos06], der über eine WSRF-Schnittstelle den Zugriff auf Managementfunktionen virtueller Maschinen erlaubt. Die Funktionalität des Dienstes wurde in [Aga07] gezeigt. Es wird jedoch nur Xen unterstützt.

Die anwendungsabhängige, dynamische Cluster Re-Konfiguration ist in [Eme06] untersucht. Hierbei wurde das LRMS (Local Resource Management System) modifiziert, so dass der Batchsystem Server bei Bedarf Xen-basierte virtuelle Maschinen zu der Liste der verfügbaren Ressourcen hinzufügen bzw. von ihr entfernen kann.

5 Einsatz von Virtualisierung im Grid Computing Umfeld

Die Einbringung der Virtualisierung in Grid Computing ist auf verschiedenen Ebenen möglich. Im Folgenden werden Ansätze auf der Grid Middleware und der LRMS Ebene vorgestellt. Der darauf folgende Abschnitt skizziert eine Lösung zur flexiblen Erstellung

von Virtual Appliances. Dies ist insofern wichtig, da das LRMS bedarfsweise virtuelle Maschinen mit unterschiedlichsten Kombinationen von Betriebssystem und Anwendung bereitstellen muss.

Unabhängig von der untersuchten Ebene ist der Begriff des (Batch-)Jobs zu erweitern. Bisher enthalten Jobs auszuführende Shell-Skripte oder sind in einer wohl-definierten Beschreibungssprache wie JDL [Jdl06] oder JSDL [Jsd07] gehalten. Durch die Integration der Plattformvirtualisierung auf LRMS Ebene werden auch virtuelle Maschinen bzw. Virtual Appliances als Jobs ausführbar. Der Unterschied zwischen Job und virtueller Maschine scheint in diesem Kontext zu verschwinden.

5.1 Grid Middleware Ebene

Grid Middleware besitzt, wie jede andere Software auch, Abhängigkeiten, die spätestens während der Installation aufzulösen sind. Diese Abhängigkeiten können sich auf das Betriebssystem oder auf andere Software (Bibliotheken, ...) beziehen. So hat die gLite Middleware starke Abhängigkeiten zu Scientific Linux 4, die UNICORE Middleware ist durch die Ausführung in einer Java VM betriebssystemunabhängig, setzt aber ein installiertes JDK voraus.

Insbesondere im Fall der Betriebssystemabhängigkeit zeigen Virtual Appliances ihren Charme. Die Anwendungen, hier Grid Middleware Dienste, werden mit dem empfohlenen Betriebssystem zu einer Virtual Appliance gebündelt. Das Softwaremanagement wie etwa das Auflösen der Anwendungsanforderungen liegt im Verantwortungsbereich des Virtual Appliance Erzeugers. Ressourcenbetreiber stellen nur noch die Virtualisierungsplattform zur Ausführung der Appliances bereit und übernehmen Managementaufgaben.

Eine Erweiterung des Konzepts stellt die Erzeugung von (Grid Middleware) Diensten durch Nutzer dar. In diesem dynamischen Szenario werden die Managementaufgaben bezüglich der Virtual Appliances vom LRMS übernommen. Dies erhöht den Grad der Automation im Cluster und ermöglicht z. B. den parallelen Betrieb von zwei oder mehr Grid Middlewares auf einer Ressource - ohne gegenseitige Beeinflussung und ohne Eingriff des Ressourcenbetreibers.

Unabhängig davon, ob das Management der Virtual Appliances durch das LRMS oder händisch erfolgt, müssen erforderliche Virtual Appliances vor der Ausführung lokal auf der Ressource vorliegen. Es ist jedoch nicht unbedingt zwingend diese alle lokal vorzuhalten. Storage Elemente, wie sie in den meisten Grid Middlewares existieren, z.B. dCache² in der gLite Middleware, bieten sich als Repository an. Der Zugriff auf die Storage Elemente erfolgt über standardisierte Protokolle wie OGSA-DAI in der Globus Toolkit Middleware³ oder SRM in der gLite Middleware.

Im Bereich des Grid Computing spielt neben der Plattformvirtualisierung ebenso Speichervirtualisierung eine große Rolle. Die Kosten pro Megabyte Speicher fielen in den letzten Jahren kontinuierlich, so dass Ressourcenbetreiber ihre Speicherkapazität drastisch erhöhen konnten. Der Verwaltungsaufwand stieg jedoch überproportional zur hin-

²<http://www.dcache.org>

³<http://www.globus.org/toolkit/>

zugewonnenen Kapazität an. Die Virtualisierung von Speicher reduziert die Komplexität durch Übernahme von Managementaufgaben. Neben Betreibern profitieren auch Nutzer durch das vereinfachte Datenmanagement. Die Entwicklungen in diesem Bereich sind noch nicht weit fortgeschritten. Langfristig lässt sich erkennen, dass schnellere lokale und Wide-Area-Netzwerke den tatsächlichen Datenstandort zu einer vernachlässigbaren Größe werden lassen.

5.2 LRMS Level

Die Integration der Plattformvirtualisierung inklusive zugehöriger Vorteile wie Checkpointing, Snapshotting und Migration von virtuellen Maschinen in existierende LRMS ist ein aktives Forschungsfeld im Bereich des Grid Computing. Üblicherweise unterstützen LRMS Job Suspension und Checkpointing. Plattformvirtualisierung bietet gleiches im Kontext virtueller Maschinen, tatsächlich bietet sie mit Migration (live oder stop-n-copy) mehr.

Die Kombination der LRMS Eigenschaften mit der Migration virtueller Maschinen erlaubt die dynamische Änderung der Ressourcenallokation. LRMS schauen nicht in die Zukunft, sondern streben für die aktuelle Situation eine optimale Lösung an. Die Auslastung der Ressource ändert sich jedoch dynamisch; einmal getroffene LRMS Entscheidungen können sich zu einem späteren Zeitpunkt als nicht mehr effizient erweisen. Mittels Checkpointing und Suspendierung virtueller Maschinen lässt sich die getroffene, aktuell nicht mehr optimale Ressourcenallokation aufheben.

Die freigewordenen physikalischen Ressourcen werden der Liste der verfügbaren Ressourcen hinzugefügt und die Jobs (virtuelle Maschinen) bis zur weiteren Abarbeitung in einen hold (suspended) Status versetzt. Das LRMS verteilt die Jobs neu auf die verfügbaren Ressourcen und erzielt eine effizientere Lösung. Während der Suspendierung als auch im laufenden Betrieb virtueller Maschinen sind Parameter wie die Anzahl zugewiesener CPUs oder RAM re-konfigurierbar. Somit lässt sich ebenso die Dienstgüte dynamisch anpassen.

5.3 Flexible Erzeugung von Virtual Appliances

Wie in Abschnitt 2 beschrieben handelt es sich bei Virtual Appliances um die Bündelung einer virtuellen Maschine mit einem Betriebssystem und einer spezifischen Anwendung. Einem Rechenzentrum reicht die Existenz einer Virtual Appliance für die Virtualisierungsplattform, die vom Rechenzentrum unterstützt wird, aus. Nutzer, die ihre Virtual Appliances auf mehreren Rechenzentren (mit unterschiedlichen Virtualisierungsplattformen) ausführen wollen, wie es im Grid Computing der Fall ist, stehen vor einem Problem. Sie brauchen ein Werkzeug, welches ein von ihnen erstelltes Festplattenabbild inklusive Betriebssystem und Anwendung in die verschiedenen Formate der Virtualisierungsplattformen transformiert.

Für den Compute Cluster des DGRZR (D-Grid Ressourcen Zentrum Ruhr) ist ein Mischbetrieb von Xen (Workernodes) und VMware (Grid Middleware Dienste) geplant. Der Einsatz von VMware soll die bisher Xen-basierten virtuellen Maschinen für die Grid Middleware Dienste ablösen, um eine Hochverfügbarkeit dieser Dienste zu ermöglichen. Zudem liegt - bedingt durch die Umsetzung der D-Grid Referenz-Installation⁴ - mit Scientific Linux 4.x und SUSE Enterprise Linux Server ein Mix an Betriebssystemen vor. Vor diesem Hintergrund wurde Software evaluiert, die Virtual Appliances mit mehreren Betriebssystemen erzeugen kann und mindestens Xen und VMware als Virtualisierungstechnologien unterstützt. Eine Lösung bietet das KIWI Image System [Sch08].

KIWI gestaltet den Erzeugungsprozess eines virtualisierungsplattformabhängigen Abbildes zweistufig. In der ersten Phase wird für das Abbild ein neues Wurzelverzeichnis angelegt. Darin werden aus zuvor definierten Quellen Betriebssystempakete und optional Anwendungspakete eingespielt. Das so gefüllte Wurzelverzeichnis wird als *physical extend* definiert. Am Ende der ersten Phase sind über einen Hook eigene Änderungen wie das De-/Aktivieren von Diensten vornehmbar.

Im zweiten Schritt wird aus dem *physical extend* ein *logical extend* erzeugt. Dieser *logical extend* ist plattformspezifisch und besteht z.B. für Xen aus einer Konfigurationsdatei, Disk Image(s) sowie einem *initrd* Image und einem Kernel. Unabhängig von der als Ausgabeformat gewählten Virtualisierungsplattform wird immer das gleiche Wurzelverzeichnis verwendet. Für Nutzer liegt hierin der wesentliche Vorteil: einmaliges Vorbereiten des Wurzelverzeichnisses und Portabilität auf viele der heute existierenden Virtualisierungsplattformen. Neben den Plattformen Xen und VMware sind auch Live CD/ DVD/ USB Stick als weitere Ausgabeformat möglich. Im Folgenden wird der bisherige Stand der Evaluation kurz beschrieben. KIWI ist zunächst beschränkt auf die Erstellung von Virtual Appliances mit den Betriebssystemen openSUSE, SUSE Linux Enterprise Desktop (SLED) bzw. Server (SLES). Erste Adaptionsschritte zur Unterstützung von Scientific Linux 4, wie es auf dem Compute Cluster zum Einsatz kommt, wurden erarbeitet.

- *smart*⁵, einer der beiden von KIWI unterstützten Paketmanager, wurde auf Scientific Linux 4 sowie 5 übersetzt. *smart* verarbeitet mehr Paketformate als dessen Alternative *zypper*⁶ und wurde daher präferiert.
- Nach der Erstellung des *physical* und des *logical extends* werden über Hooks Funktionen ausgeführt, die SUSE spezifische Anpassungen vornehmen. Für Scientific Linux sind entsprechende Funktionen zu erstellen. Eine Integration neuer Funktionen in KIWI ist aufgrund des verwendeten Namenschemas ohne weiteres möglich: der Name einer Funktion beginnt mit einem Kürzel, das die Linux Distribution beschreibt.
- Für Scientific Linux 4.5, 4.7 und 5.2 konnten bereits *physical extends* in 32 Bit und 64 Bit Varianten erstellt werden.

⁴<http://dgieref.d-grid.de/wiki/Introduction>

⁵<http://labix.org/smart>

⁶<http://en.opensuse.org/Zypper>

Nach Abschluss der Evaluation soll ein Prozess, zur Integration weiterer Betriebssysteme in KIWI, realisiert sein.

6 Zusammenfassung und Ausblick

Die Arbeit beschreibt Vorteile des Zusammenspiels von Plattformvirtualisierung und Grid Computing aus Sicht von Grid Rechenzentren. Der Trend in den Rechenzentren zur Virtualisierung (sowohl Plattform- als auch Speicher- und Netzwerkvirtualisierung) überzugehen, wurde nicht durch Grid Communities, sondern durch das Bestreben der effizienten Nutzung vorhandener Kapazitäten vorangetrieben. Somit ist Virtualisierung in diesem Kontext keine kurzfristige Erscheinung, ihr Verbreitungsgrad in Rechenzentren wird zunehmen. Virtualisierungstechnologien sind auf verschiedenen Ebenen der Grid Middleware integrierbar, auf zwei von ihnen wurde näher eingegangen. Kommerzielle Anbieter wie Cluster Resources bieten auf Ebene des lokalen Ressourcenmanagements bereits Lösungen für die on-Demand Bereitstellung von Virtual Appliances, jedoch sind diese Lösungen virtualisierungsplattformabhängig.

Ein Vorgehen diese Abhängigkeit zu umgehen wurde ebenso vorgestellt. Das zugehörige Framework KIWI befindet sich derzeit in der Evaluationsphase am DGRZR. Nach Abschluss der Evaluationsphase und der Integration der Unterstützung von Scientific Linux in KIWI, soll über die bereitgestellten Hooks die automatisierte Installation von Grid Middleware Diensten erfolgen. Hierzu wurden Skripte erstellt, die die Installation und auch Konfiguration vieler Dienste der Grid Middlewares gLite3, Globus Toolkit4 und UNICORE5 ermöglichen. Die Kombination der Skripte mit KIWI läßt die Verfügbarkeit von virtualisierungsplattformunabhängigen Grid Middleware Virtual Appliances bedeutend näher rücken.

Literatur

- [Aga07] Deploying HEP Applications Using Xen and Globus Virtual Workspaces. A. Agarwal, A. Charbonneau, R. Desmarais, R. Enge, I. Gable, D. Grundy, A. Norton, D. Penfold-Brown, R. Seuster, R.J. Sobie, D.C. Vanderster. In Proceedings of Computing in High Energy and Nuclear Physics. September 2007.
- [Bes07] OGSA Basic Execution Service Version 1.0. I. Foster, A. Grimshaw, P. Lane, W. Lee, M. Morgan, S. Newhouse, S. Pickles, D. Pulsipher, C. Smith, M. Theimer. <http://forge.gridforum.org/projects/ogsa-bes-wg>. Letzter Zugriff: 15. Juli 2008.
- [Bha08] Virtual Cluster Management with Xen. N. Bhatia, J. S. Vetter. In Lecture Notes in Computer Science, Volume 4854/2008, Seiten 185-194. 2008.
- [Car07] Management of a Grid Infrastructure in GLITE with Virtualization. M. Cardenas, J. Perez-Griffo, M. Rubio, R. Ramos. 1st Iberian Grid Infrastructure Conference, Mai 2007.

- [Eme06] Dynamic Virtual Clustering with Xen and Moab. W. Emenecker, D. Jackson, J. Butikof, D. Stanzione. International Symposium on Parallel and Distributed Processing and Applications (ISPA) Workshops 2006. September 2006.
- [Fos02] What is the Grid? A Three Point Checklist. I. Foster. 2002
- [Fos03] The Grid 2: Blueprint for a New Computing Infrastructure (The Morgan Kaufmann Series in Computer Architecture and Design). I. Foster, C. Kesselman. November 2003.
- [Fos06] Virtual Clusters for Grid Communities. I. Foster., T. Freeman, K. Keahey, D. Scheftner, B. Sotomayor, X. Zhang. In Proceedings of the 6th International Symposium on Cluster Computing and Grid (CCGRID). 2006.
- [Fos08] Cloud Computing and Grid Computing 360-Degree Compared. I. Foster, I. Yong Zhao Raicu, S. Lu. In Grid Computing Environments Workshop, 2008. Seiten 1 - 10.
- [Jdl06] Job Description Language Attributes Specification for the gLite Middleware Version 0.8. F. Pacini. <https://edms.cern.ch/file/590869/1/>. Letzter Zugriff: 15. Juli 2008.
- [Jsd07] Job Submission Description Language (JSDL) Specification, Version 1.0. A. Anjoms-hoaa, M. Drescher, D. Fellows, A. Ly, S. McGough, D. Pulsipher, A. Savva. <http://www.gridforum.org/documents/GFD.56.pdf>. Letzter Zugriff: 15 Juli 2008.
- [Kea05] Virtual Workspaces: Achieving Quality of Service and Quality of Life in the Grid. K. Keahey, I. Foster, T. Freeman, X. Zhang. Scientific Programming Journal - Special Issue: Dynamic Grids and Worldwide Computing, Volume 13, No. 4, Seiten 265-276. 2005.
- [Ram04] Virtualization - Bringing Flexibility and New Capabilities to Computing Platforms. R. Ramanathan, F. Bruening. Technical Paper. Intel Corporation. 2004
- [Ovf08a] The Open Virtual Machine - Whitepaper for OVF Specification Version 0.9. VMware, XenSource. 2007.
- [Ovf08b] OVF - Open Virtual Machine Specification Version 0.9. VMware, XenSource. 2007.
- [Rie05] Standardization Processes of the UNICORE Grid System. M. Riedel, D. Mallmann. In Proceedings of 1st Austrian Grid Symposium 2005, Seiten 191-203. 2005.
- [Sch08] openSUSE - KIWI Image System Cookbook. M. Schäfer. Version 3.01. 24. November 2008
- [Stra59] Time sharing in large fast computers. C. Strachey. In Proceedings of the International Conference on Information Processing, UNESCO, Seiten 336-341. 1959.
- [Tat03] Virtualization in a SAN. J. Tate. RedBooks Paper, IBM. <http://www.redbooks.ibm.com/redpapers/pdfs/redp3633.pdf>. Letzter Zugriff: 15. Juli 2008.
- [Tat06] Making Wide-Area, Multi-Site MPI Feasible Using Xen VM. M. Tatzono, N. Maruyama, S. Matsuoka. International Symposium on Parallel and Distributed Processing and Applications (ISPA) Workshops 2006. September 2006.
- [Vla06] IEEE 802.1: 802.1Q - Virtual LANs. IEEE Computer Society. <http://www.ieee802.org/1/pages/\802.1Q.html>. Letzter Zugriff: 15. Juli 2008.
- [Vpn98] A Comprehensive Guide to Virtual Private Networks, Volume I: IBM Firewall, Server and Client Solutions. T. Bourne, T. Gaidosch, C. Kunzinger, M. Murhammer, L. Rademacher, A. Weinfurter RedBooks Paper, IBM. <http://www.redbooks.ibm.com/redbooks/pdfs/sg245201.pdf>. Letzter Zugriff: 09. März 2009.
- [Vmw07a] Best Practices for Building Virtual Appliances - Whitepaper. VMware. 2007

