Müller, Neumair, Reiser, Dreo Rodosek (Hrsg.): 9. DFN-Forum Kommunikationstechnologien,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2016 11

Schichtung virtueller Maschinen zu Labor- und Lehrinfrastruktur

Vitalian Danciu[1] Tobias Guggemos[1] und Dieter Kranzlmüller[1]

Abstract: Die Nutzung vieler Instanzen einer virtueller Infrastruktur für Laborversuche und Lehrveranstaltungen kann eine große Anzahl virtueller Netze und somit entsprechend hohen Managementaufwand erfordern. Diese Arbeit zeigt einen Ansatz zur Kapselung der virtuellen Komponenten und Netze einer Instanz mittels der Schichtung virtueller Maschinen. Die darin implizite Gruppierung der virtuellen Komponenten und Netze zerlegt die Gesamtmenge virtueller Entitäten in kleinere, leichter handzuhabende und dem Zweck der Instanz entsprechende Managementdomänen geringerer Komplexität. Anwendungen in der Lehre sowie in der Praxis illustrieren den Ansatz.

Keywords: virtualisierte Infrastruktur, Schichtung, Verschachtelung

1 Motivation

Der Einsatz von Virtualisierungstechnik erlaubt die Erzeugung virtueller Infrastruktur. Virtuelle Maschinen (VM) können mit virtuellen Komponenten der Sicherungsschicht verbunden werden und ergeben somit ein Netz, das für wissenschaftliche Versuche oder im Rahmen von Lehrveranstaltungen (Praktika, Demonstrationen) genutzt werden kann. Die Anzahl gleichzeitig einsatzbereiter *virtueller Labore* ist im Prinzip nur durch die verfügbaren physischen Ressourcen (Rechenleistung, Hauptspeicher, Sekundärspeicher) begrenzt, so dass sie als kosteneffiziente Alternative zu traditioneller Laborausstattung immer häufiger zum Einsatz kommen.

Während einzeln genutzte VM hauptsächlich durch ihre Konfiguration charakterisiert sind, ergeben sich die Eigenschaften des virtuellen Labors zusätzlich — und maßgeblich — durch seine Struktur (Topologie). Somit ist eine Laborinstanz enger an die spezifische *Virtualisierungsplattform* gebunden, die seine Komponenten und Verbindungen erzeugt sowie der Ausführung der Laborinstanz zugrundeliegt. Diese bietet in der Regel keine Sicht auf Strukturen, sondern Operationen zum Management einzelner Komponenten und Gruppen einzelner Komponenten. Während die Angaben zu Komponenten (z.B. Maschinenressourcen: RAM, Anzahl CPU) deklarativ vorliegen und somit einfach maschinell analysiert und bearbeitet werden können, wird die Topologie meist prozedural erzeugt, etwa durch Skripten, die die Virtualisierungsplattform ansteuern. Zudem erfordert der Betrieb virtueller Labore häufig eine große Anzahl virtueller Netzkomponenten. Diese Umstände erschweren das Management der virtuellen Labore sowie die Migration zu einer anderen Virtualisierungsplattform oder auch zu einer Instanz derselben Virtualisierungsplattform.

[1] MNM-Team, Ludwig-Maximilians-Universität München, Oettingenstr. 67, 80538 München, {danciu, guggemos, kranzlmueller}@mnm-team.org

Diese Arbeit stellt einen Ansatz zur wiederholten Anwendung von Virtualisierungstechnik vor, um die Struktur virtueller Infrastruktur zu kapseln. Abschnitt 2 analysiert ein Anwendungsszenario und formuliert Anforderungen an die Eigenschaften des Ansatzes, der in Abschnitt 3 vorgestellt wird. Abschnitt 4 beschreibt die konkrete Architektur eines Systems zur Erzeugung gekapselter virtueller Labore, ihre Umsetzung und ihren bisherigen Einsatz in der Praxis. In Abschnitt 6 werden verwandte Arbeiten gezeigt, bevor Abschnitt 7 die Arbeit abschließt und Einsatzmöglichkeiten in der Hochschullehre auf zentralen Rechenzentrumsressourcen sowie praxisrelevante Szenarien im Bereich der Network Function Virtualisation (NFV) diskutiert.

2 Analyse

Wir betrachten den Aufbau virtueller Labore anhand des Praktikums „Rechnernetze“ (RNP) am Lehrstuhl für Kommunikationssysteme und Systemprogrammierung an der Ludwig-Maximilians-Universität München. Das Curriculum beinhaltet Versuche auf der Basis von VM, wobei die einzelnen VM als Koppelkomponenten (Router) des virtuellen Labors oder als Endgeräte eingesetzt werden.

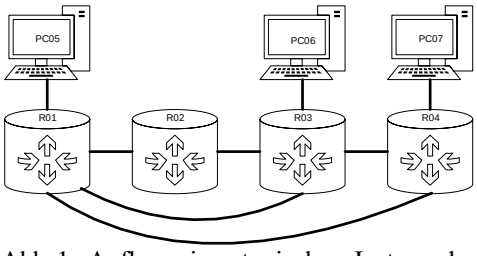


Abb. 1: Aufbau einer typischen Instanz des virtuellen Labors

Abb. 1 zeigt einen typischen Aufbau der Versuchsumgebung, die einer Gruppe von Teilnehmern zugewiesen wird. Sie besteht aus VM, die jeweils über mehrere (virtuelle) Netzschnittstellen verfügen. Jede VM ist mit einer Netzschnittstelle zum Zugang und Management („Managementschnittstelle“) ausgestattet. Die Verbindungen zwischen VM können als Portgruppen eines virtuellen Switch realisiert, oder als virtuelle Brücken/Switches umgesetzt werden. Wir bezeichnen diese Konstruktion im folgenden als *virtuelles Netz*.

Komplexitätsbetrachtung Abstrahiert von der derzeitigen Nutzungstopologie gehen wir von einem einfach vollvermaschten Netz für eine Instanz eines virtuellen Labors aus, also einem vollständigen Graphen mit $\frac{1}{2}(n^2-n)$ Verbindungen zwischen n Netzkomponenten. Hinzu kommen die Managementschnittstellen, die in einem Netz zusammengefasst werden. Für k Instanzen des virtuellen Labors ergibt sich die Gesamtzahl B an virtuellen Netzen bei n VM zu $B = \frac{k}{2}\left(n^2-n\right)+1$

Das polynomielle Wachstum der erforderlichen Anzahl virtueller Netze führt selbst bei bescheidenen Versuchsaufbauten zu unübersichtlichen Strukturen: $k = 20$ Instanzen eines virtuellen Labors mit $n = 4$ vollvermaschten Knoten (Abb. 2) benötigen der Gleichung entsprechend $B = 121$ virtuelle Netze. Bei den sieben Knoten für das Praktikum würde eine Vollvermaschung bereits $B = 421$ virtuelle Netze erfordern. Der Einsatz mehrerer physischer Maschinen zur Lastverteilung muss die Konnektivität zwischen den VM des Labors

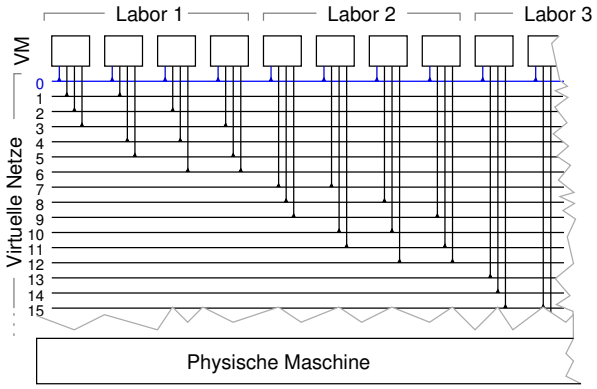


Abb. 2: Aufbau vollvermaschter virtueller Labore

gewährleisten. Gegebenenfalls müssen sich die virtuellen Netze über alle physischen Maschinen im Lastverbund hinweg erstrecken.

Betrieb Architekturbedingt stellen sich im Betrieb einige Herausforderungen. Die virtuellen Labore werden auf einem Rechnerverbund ausgeführt, der auch anderen Anwendungen dient (andere Lehrveranstaltungen, Projekte, etc.) und von einem Rechenzentrum (in diesem Fall dem Leibniz-Rechenzentrum München) betrieben wird. Es muss sichergestellt werden, dass andere parallel betriebene Anwendungen nicht gestört werden und kein unerwünschter Verkehr aus den Versuchen in das Hochschulnetz gelangt. Teilnehmer erhalten administrativen Zugang zu den Komponenten ihrer Laborinstanz, nicht aber zu der Virtualisierungsplattform, die sie erzeugt. Gelegentlich „haverierte“ Instanzen müssen somit von Lehrpersonal wiederhergestellt werden, was einen erhöhten Managementaufwand erzeugt. Die Managementdomäne der virtuellen Netze erstreckt sich über die gesamte Anzahl B dieser Netze. Managementaktivitäten wie etwa Suche und -diagnose von Fehlfunktionen können daher nicht die Partitionierung in Laborinstanzen ausnutzen, da sie sich nicht im Aufbau der virtuellen Infrastruktur widerspiegelt.

Wir formulieren folgende Anforderungen an die Konstruktion der virtuellen Labore. Die diese erfüllenden Konzepte sind in der weiteren Arbeit mit (Anf. <Nr>) gekennzeichnet.

1. *Erzeugung aus Modell.* Eine Laborinstanz soll ohne Betrachtung der Virtualisierungsplattform aus einem Modell bzw. einer Spezifikation (automatisch) erzeugt werden.

2. *Heterogene Modelle.* Die Koexistenz von Instanzen aus mehreren Modellen soll möglich sein.

3. *Wahlfreiheit der Hostplattform.* Der Aufbau soll geringe/keine Anforderungen an die Software der physischen Maschinen (Hosts) stellen.

4. *Archiv* Eine Instanz soll als Ganzes archiviert oder weitergegeben werden können.

5. *Skalierbarkeit* Die Kosten für Laborinstanzen sollen höchstens linear mit der Anzahl Instanzen wachsen. Die Skalierung bezieht sich auf die Nutzung der Ressourcen der darunterliegenden physischen Plattform, der Komplexität des Gesamtaufbaus aus allen Laborinstanzen und die Aufwendungen im Management.
6. *Isolation* Laborinstanzen sollen voneinander isoliert ausgeführt werden, so dass keine Kommunikationsartefakte "von außerhalb" durchdringen, z.B. keine Managementprotokolle der virtuellen Sicherungsschicht.
7. *Unabhängigkeit* Lastkontrolle/-verteilung auf den Hosts sollte auf der Basis der Instanzen erfolgen, statt auf der Basis einzelner Ressourcen.
8. *Beliebige Adressierung* Netze innerhalb der Laborinstanzen sollen unabhängig von der Umgebung mit beliebigen (auch in zwei Instanzen identischen) Adressen der Sicherungs- und Vermittlungsschicht versehen werden können.
9. *Nutzerzugang* Der Zugang zu jedem virtuellen Labor soll per Text- und Graphikschnittstellen möglich sein.

3 Ansatz

Der in dieser Arbeit vorgestellte Ansatz verfolgt die Erfüllung der Anforderungen aus Abschnitt 2 mittels Schichtung von Virtualisierungsumgebungen, also der Ausführung virtueller Komponenten auf der Basis darunterliegender virtueller Komponenten. Die Darstellung dieses Modells in Abb. 3 führt die folgende Notation ein:

0-VM bezeichnet die physische Maschine, die einen Hypervisor ausführt und in der Lage ist, VM und virtuelle Netze bereitzustellen. Wir nutzen die Bezeichnung „VM“ für die Konsistenz der Notation.

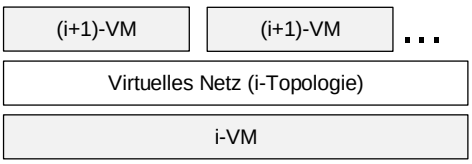


Abb. 3: Architekturprinzip

i-VM bezeichnet eine Maschine der Stufe *i*. Die VM, die direkt auf der 0-VM ausgeführt werden sind also 1-VM. Erzeugt eine 1-VM selbst virtuelle Maschinen, sind dies 2-VM, etc.

(i+1)-VM bezeichnet eine virtuelle Maschine, welche auf der Basis einer i-VM betrieben wird.

i-Topologie bezeichnet virtuelle Netze mit Netzkomponenten, die in einer i-VM erzeugt werden und (i+1)-VM verbinden.

Eigenschaften des Ansatzes Die Beziehung einer VM der Stufe $(i+1)$ zur darunterliegenden VM der Stufe *i* entspricht dabei einer Enthaltenseinrelation, also einer *Verschachtelung* der *inneren* $(i+1)$-VM in der *äußeren i*-VM. Diese Architektur hat die Eigenschaft, dass auch die virtuellen Netze der Labortopologie innerhalb der äußeren VM gekapselt

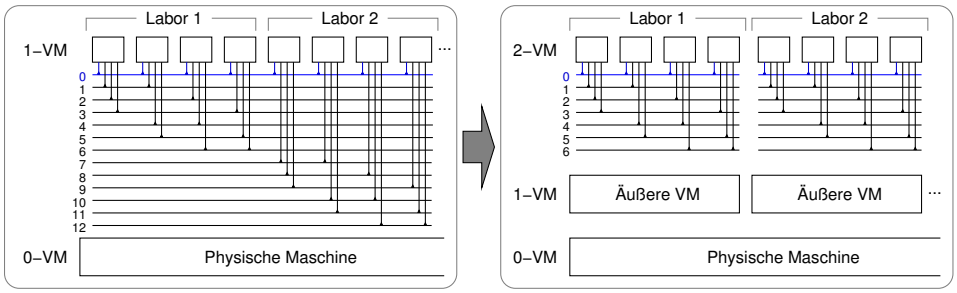


Abb. 4: Ansatz

werden (Anf. 6,7,8). Abb. 4 zeigt als Konsequenz für die Netze von in i-VM gekapselter virtueller Labore, dass die Anzahl in der identischen i-VM betriebenen Netze konstant bleibt, unabhängig von der Anzahl der Laborinstanzen (Anf. 5). Die Gesamtanzahl der Netze steigt durch die Kapselung des Managementnetzes um k, das heißt um eine zusätzliches virtuelles Netz pro Instanz.

Die ursprüngliche, alle Netze umfassende Managementdomäne zerfällt allerdings in jeweils den Instanzen entsprechenden Domänen. Eine Laborinstanz betreffende Managementaktivitäten werden somit konstruktionsbedingt auf diese eine Instanz beschränkt, was unter anderem auch heterogene Modelle ermöglicht (Anf. 2).

4 Entwurf

Wir betrachten nachstehend die prozedurale Erzeugung verschachtelter virtueller Infrastruktur sowie das Speicherkonzept für die Abbilder der sie erbringenden geschichteten virtuellen Maschinen.

Abb. 5 zeigt die Schritte zur Erstellung eines virtuellen Labors und ihre Eingabe: Die **i-Konfiguration** enthält Einstellungen einer i-VM (z.B.: IP-Adressen, Routen, Hostnamen; Konfiguration und Start von Diensten) (Anf. 9). Die **i-Topologie-Spezifikation** beschreibt Erstellung und Konfiguration virtueller Netzkomponenten innerhalb der i-VM sowie die Verbindung der (i+1)-VM mit den virtuellen Netzkomponenten. Die **(i+1)-Maschinen-Spezifikation** beschreibt die virtuelle Hardware für jede von einer i-VM verwalteten (i+1)-VM, z.B. Anzahl virtueller Prozessoren, Arbeitsspeicher, Netzschnittstellen.

1. Der *Startvorgang* beschreibt den Start einer Maschine mit *i-Konfiguration*. Falls eine i-VM keine (i+1)-VMs enthält, ist dies der einzig notwendige Schritt.

2. In der i-VM werden anhand der *i-Topologie-Spezifikation* virtuelle Netzkomponenten (Brücken/Switches) zwischen (i+1)-VMs erstellt.

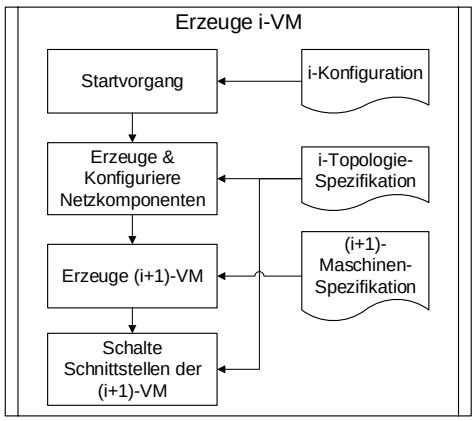
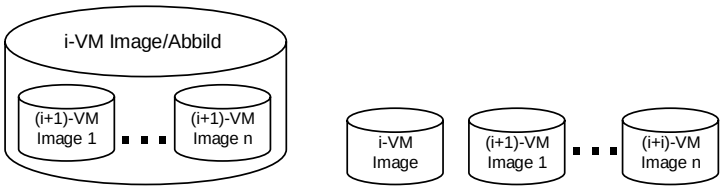


Abb. 5: Erzeugungs- und Konfigurationskonzept für VMs

3. Die in der *i-Topologie-Spezifikation* referenzierten (i+1)-VMs werden entsprechend der *(i+1)-Maschinen-Spezifikation* erzeugt und anschließend gestartet. Hier beginnt der *Startvorgang* der (i+1)-VM.

4. Entsprechend der *i-Topologie-Spezifikation* werden die Netzschnittstellen der (i+1)-VMs mit den Netzkomponenten der i-Topologie verbunden. Dieser Vorgang entspricht dem Stecken von Kabeln oder Schalten von Ports in einem physischen Netz.

Speicherkonzepte Für die Speicherung von Abbildern bzw. *Images* der (i+1)-VM innerhalb der i-VM sind zwei Ansätze denkbar (siehe Abb. 6). Bei der gekapselten Speicherung (Abb. 6a) enthält jedes i-VM-Abbild die Speicherblöcke der Abbilder „ihrer“ (i+1)-VMs. Im Gegensatz dazu sind im linearen Konzept (Abb. 6b) die Abbilder von i-VMs und (i+1) nebeneinander auf der gleichen Abstraktionsebene der Blockgeräte gespeichert.



(a) gekapselt (b) linear

Abb. 6: Speicherkonzept

Die Kapselung sichert die Konsistenz der Abbilder einer Laborinstanz auf Kosten der Einführung einer weiteren Blockabstraktion und dem damit verbundenen Leistungsverlust. Diese zusätzliche Blockabstraktion ist erforderlich, um Abbilder von (i+1)-VM, die als Dateien im Abbild der i-VM gespeichert werden, den (i+1)-VM als Blockgeräte (virtuelle Festplatten) zur Verfügung stellen zu können. Die Vor- und Nachteile des linearen An-

satzes sind invers: die ungekapselten Abbilder aller VM werden ungruppiert nebeneinander gehalten und sie können dadurch durch die gleiche Blockabstraktion genutzt werden; die Konsistenzerhaltung ist dabei nicht mehr durch die Konstruktion unterstützt sondern muss insgesamt als Managementaufgabe wahrgenommen werden. Beide Ansätze erlauben Archivierung (Anf. 4), wobei der gekapselte Ansatz diese intuitiver unterstützt.

5 Umsetzung am Beispiel RNP

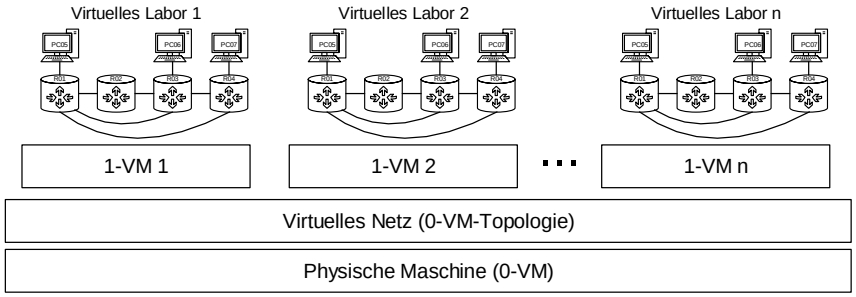


Abb. 7: Aufbau der Laborinfrastruktur für das RNP

Für das in Abschnitt 2 beschriebene RNP-Szenario wurde ein 3-stufiges System implementiert. Die 0-VM enthält für jeden Teilnehmer des Praktikums eine 1-VM, welche wiederum 2-VM mit dem im Praktikum eingesetzten virtuellen Labor enthält (siehe Abb. 7).

```
#How many routers and PCs should be created
NO_OF_ROUTERS=4
NO_OF_PCS=3

#Bridge definitions
BRIDGES="(1,2) (1,3) (1,4) (1,5) (2,3) (3,4) (3,6) (4,7)"
```

Abb. 8: Spezifikation der Verbindungen für ein virtuelles Labor

Die nötigen Topologie- und Maschinenspezifikationen (vgl. Abb. 5) werden aus einer deklarativen Angabe der Topologie generiert (Anf. 1). Aus der Angabe der Verbindungen zwischen VM (vgl. Abb. 8) kann die Anzahl erforderlichen Maschinen und die jeweilige Anzahl der Netzschnittstellen ermittelt werden. Die Maschinen werden unabhängig von ihrer Funktion als Router oder PC fortlaufend durchnummeriert, weshalb im Beispiel mit sieben Geräten Router01-04 und PC05-07 entstehen. Dies verdeutlicht die intendierte Rolle der VM aus didaktischen Gründen, ist jedoch technisch nicht erforderlich.

Die isolierten virtuelle Labore dürfen in jeder 1-VM bezüglich der Konfiguration (z.B. Adressierung) ihrer jeweiligen 2-VM identisch sein. Für jede i-VM werden die erforderlichen Konfigurationsdateien in einem Baum abgelegt, wie in Abb. 9 dargestellt. Sie werden in das Abbild der VM übertragen, bevor diese gestartet wird. Die Abbildung zeigt die

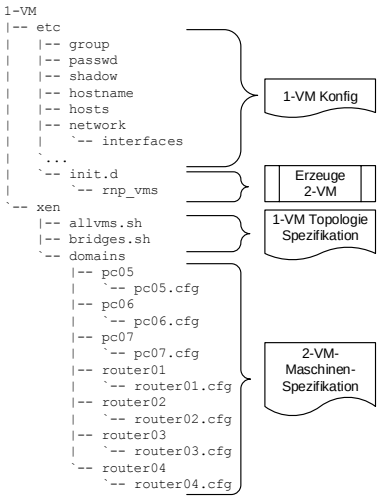


Abb. 9: Ordnerstruktur für die Konfiguration der Maschinen.

nötigen Dateien für eine 1-VM. Dies erlaubt einerseits Unterschiede zwischen VM der gleichen Stufe (z.B. verschiedene MAC- und IP-Adressen der 2-VM einer Laborinstanz) aber auch die Koexistenz verschieden bestückter 1-VM.

Szenariobedingt entschieden wir uns für das gekapselte Speicherkonzept. Im Allgemeinen scheint das gekapselte Konzept für Lehrveranstaltungen geeignet zu sein.

Auf Grund der Anforderung nach einer möglichst unangepassten 0-VM (Anf. 3), entschieden wir uns für den vom Linux-Kernel unterstützten KVM-Hypervisor in der 0-VM. Die 1-VM ist ein für Xen angepasster Linux-Kernel, da Xen einige Konfigurationsvorteile bietet. Beim Betrieb mit 25 Teilnehmern und ebensovielen Laborinfrastrukturen konnten keine für den Betrieb des Praktikums relevante Leistungseinbußen festgestellt werden.

6 Verwandte Arbeiten

Virtualisierung im Lehrbetrieb einzusetzen ist heutzutage keine Seltenheit und oft sogar organisatorisch alternativlos [Ga08]. Vor allem für systemnahe Kurse, in welchen Studenten Netzmanagement, Systemadministration oder Systemprogrammierung erlernen ist ein administrativer Zugriff oft unerlässlich. Virtuelle Maschinen bieten hier den unschätzbaren Vorteil, dass – vorausgesetzt sie sind richtig konfiguriert – viele administrative Aufgaben einfach und zentralisiert durchgeführt werden können. Vor allem ausschalten, zurücksetzen und abschotten bei Sicherheitsproblemen kann schnell und einfach vom Administrator des Hypervisors durchgeführt werden. Zusätzlich sind die Anschaffungskosten niedriger und Platz wird effizienter genutzt. Lindinger et. al [LRg08] zeigen eine nicht-geschichtete virtuelle Infrastruktur für einen praktischen Kurs in IT-Sicherheit und diskutieren die Anforderungen und Schwierigkeiten einer solchen Architektur.

In Forschungsprojekten sind virtuelle Labore wie etwa PlanetLab [Ch03] bzw. dessen europäische Variante PlanetLabEU entstanden. Dabei werden Teile einer international verteilten Infrastruktur (sogenannte *Slices*) einzelnen Forschungsvorhaben zugewiesen. Die darunterliegenden realen Netze erlauben dabei realitätsnahe Ergebnisse.

Die Idee virtuelle Umgebungen geschichtet bzw. verschachtelt (*nested*) zu implementieren ist so alt wie die Virtualisierung selbst und wird teilweise auch als rekursive Virtualisierung bezeichnet. Schon 1973 beschrieben Lauer et. al [LW73] die Möglichkeit, dass jeder Prozess seinen eigenen virtuellen Speicher definieren und verwalten könne.

Neuere Arbeiten beschäftigen sich mit der Optimierung von geschichteten Virtualisierungsumgebungen für verschiedene Plattformen und Anwendungsfälle. So verwenden Pan et al. [Pa11] KVM als Hypervisor wohingegen Ben-Yehuda et al. [Be10] und Zhang et al. [Zh11] den Xen-Hypervisor [Ba03] verwenden, der von Intel mittlerweile mit Hardwareoperationen unterstützt wird [ZD12], um die Performance zu verbessern. KVM [KI07] selbst unterstützt ebenfalls geschichtete Virtualisierung auch ohne Hardwaresupport, wird allerdings von Pan et. al auf Grund der schlechten Performance für ungeeignet im realen Betrieb betrachtet. Vor allem im Bereich des IO-Benchmarking zeigen Ben-Yehuda et. al beeindruckende Ergebnisse, wenn die IO-Zugriffe direkt an den Hypervisor auf Stufe 1 durchgereicht werden können. Alle Arbeiten gehen von einem Verlust der CPU-Performance von 4-6% aus, solange man maximal zweistufig geschichtet und entweder die Hardware oder der Hypervisor der ersten Stufe Hilfsmittel für Stufe 2 zur Verfügung stellt.

7 Zusammenfassung und Ausblick

Die praktische Arbeit im Labor ist ein wichtiger Bestandteil der Lehre und Ausbildung an Hochschulen und Universitäten und generiert eine hohe Anzahl von Einsatzmöglichkeiten. Um diese flexibel und effizient im Lehrbetrieb verwenden zu können, wird wieder einmal auf Virtualisierungstechniken zurückgegriffen. In diesem Papier wurde ein neuer Ansatz vorgestellt, mit dem sowohl die Kapselung, aber auch die Gruppierung von Netzen innerhalb einer virtuellen Infrastruktur ermöglicht und vereinfacht werden. Wir beschreiben sowohl die Anforderungen an eine solche virtuelle Laborumgebung, als auch den Aufbau und die dafür notwendigen Komponenten. Die vorgestellte Lösung wurde in der Praxis evaluiert und optimiert und wird seit einiger Zeit erfolgreich im Lehrbetrieb eingesetzt. Die damit erreichte Flexibilität und Skalierbarkeit hat sich als sehr wertvoll erwiesen, um schwankende Zahlen von Studenten in gleicher Qualität unterrichten zu können. Die Abschottung zur Außenwelt hat bisher keinerlei Beeinträchtigung des Betriebs im Wissenschaftsnetz ergeben.

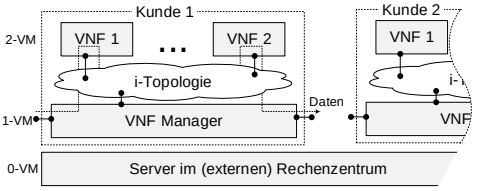


Abb. 10: Beispielhafte Architektur von geschichteter Virtualisierung für NFV.

Szenarien in der Praxis Die inhärente Kapselung der Netze einer geschichteten Infrastruktur erlaubt auch vielfältige industrielle Einsatzmöglichkeiten. Das „European Telecommunications Standards Institute“ (ETSI) beschreibt eine Architektur [ET14], in der mehrere virtuelle Netzfunktionen (VNF) miteinander verbunden und nacheinander ausgeführt und orchestriert werden, um zusammen einen Dienst zu erbringen. Die im Netz erforderlichen Dienste können aus sogenannten *Microservices* (z.B. Paketfilter, Wegewahldienste oder auch E-Mail, Spamfilter und Virenscanner) aufgebaut und von einem VNF-Manager orchestriert werden.

Legt man dieser Architektur eine zweistufig geschichtete Infrastruktur zugrunde, wie in Abb. 10 dargestellt, könnten die VNF in 2-VM instantiiert werden, durch virtuelle Switches verbunden und von einem VNF-Manager innerhalb der 1-VM orchestriert werden. Die Kapselung des so zusammengesetzten Dienstes ermöglicht sein Management als Ganzes (z.B. die Schaltung der Lebenszyklusphasen), seinen Betrieb auch in Rechenzentren dienstagnostischer, externer Betreiber (etwa Cloud/IaaS Anbieter) sowie seine Verschiebung zwischen Standorten.

Aktuell erweitern wir die bereits deklarative Topologiespezifikation um Angaben zur Konfiguration einzelner Komponenten sowie eine automatische Zuweisung von Adressräumen an die erzeugten Infrastrukturinstanzen. Ziel ist eine formale Sprache, mit der geschichtete virtuelle Infrastruktur beschrieben und anschließend erzeugt werden kann.

Literaturverzeichnis

[Ba03] Barham, Paul; Dragovic, Boris; Fraser, Keir; Hand, Steven; Harris, Tim; Ho, Alex; Neugebauer, Rolf; Pratt, Ian; Warfield, Andrew: Xen and the art of virtualization. ACM SIGOPS Operating Systems Review, 37(5):164–177, 2003.

[Be10] Ben-Yehuda, Muli; Day, Michael D. et al.: The Turtles Project: Design and Implementation of Nested Virtualization. In: 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI 10). USENIX Association, Vancouver, BC, Oktober 2010.

[Ch03] Chun, Brent; Culler, David; Roscoe, Timothy; Bavier, Andy; Peterson, Larry; Wawrzoniak, Mike; Bowman, Mic: PlanetLab: an overlay testbed for broad-coverage services. ACM SIGCOMM Computer Communication Review, 33(3):3–12, 2003.

[ET14] ETSI: Network Functions Virtualisation (NFV); Architectural Framework. Group Specification RGS/NFV-002, European Telecommunication Standards Institute (ETSI), Dezember 2014.

[Ga08] Gaspar, Alessio; Langevin, Sarah; Armitage, William; Sekar, R.; Daniels, T.: The role of virtualization in computing education. ACM SIGCSE Bulletin, 40(1):131–132, 2008.

[KI07] KIVITY, Avi: KVM: The Linux Virtual Machine Monitor. In: Proceedings of the Linux Symposium. Ottawa, Ontario, 2007.

[LRg08] Lindinger, Tobias; Reiser, Helmut; gentschen Felde, Nils: Virtualizing an IT-Lab for Higher Education Teaching. In (Gesellschaft für Informatik e.V., Paderborn, Hrsg.): GI/ITG KuVS Fachgespräch "Virtualisierung". S. 97–104, 2008.

[LW73] Lauer, Hugh C.; Wyeth, David: A Recursive Virtual Machine Architecture. In: Proceedings of the Workshop on Virtual Computer Systems. ACM, New York, NY, USA, S. 113–116, 1973.

[Pa11] Pan, Z.; He, Q.; Jiang, W.; Chen, Y.; Dong, Y.: NestCloud: Towards practical nested virtualization. In: 2011 International Conference on Cloud and Service Computing (CSC). S. 321–329, Dec 2011.

[ZD12] Nested Virtualization Update From Intel, `http://docs.huihoo.com/xen/summit/2012/aug28/6a-nested-virtualization-update-from-intel.pdf`.

[Zh11] Zhang, Fengzhe; Chen, Jin; Chen, Haibo; Zang, Binyu: , CloudVisor: retrofitting protection of virtual machines in multi-tenant cloud with nested virtualization, 2011.