

# Hypervisor

---

Stefan Laser

Digitales ist virtuell unterlaufen. Nein, das Virtuelle liegt sogar über dem Digitalen. Das illustrieren Rechenzentren, die Entwicklungen wie die Cloud, Big Data oder AI ermöglichen (vgl. Edwards/Cooper/Hogan 2024; Hu 2015; Pasek et al. 2023; Pickren 2018). Denn der Erfolg heutiger Rechenzentren ist direkt verbunden mit einem scheinbar allsehenden Supervisor: dem Hypervisor. Einzelne Operationen, Programme und Betriebssysteme laufen in modernen Rechenzentren oft nicht auf einzelnen physischen Maschinen. Es gilt: Software ist über Rechner verteilt und Operationen können dynamisch verschoben werden; Aufgaben und Daten sind logisch verbunden (vgl. Heintz 1993). Der Hypervisor ist damit eine logische Schicht und eine klar abgrenzbare Einheit, in der Operationen gesammelt, aufgeteilt, voneinander abgetrennt und wieder verknüpft werden, sodass User:innen, Programme oder Netzwerke nebeneinander liegen und ihre Ressourcen teilen können, ohne, dass die dadurch konstruierten Elemente voneinander mitbekommen oder sich stören können.

Der Hypervisor ist eine zentrale Technologie der Virtualisierung von Servern, und über die praktische Pflege des Hypervisors lässt sich eine eigene Ökologie der Virtualität entfalten, ein Lexikon im Lexikon. Ein Hypervisor lässt sich nicht ohne ein grundlegendes Verständnis von Chip-Technologie verstehen, er ist eng verwoben mit der Verwaltung und Verdrahtung von Speicher und Speicherplatz, mit Administration, und im Lauf der Geschichte hing der Hypervisor mit weiteren kleinen informatischen Innovationen zu-

sammen, die mit Begriffen der Virtualität hantierten und durchaus kontrovers diskutiert wurden. Ich wage mich deswegen an eine umsichtige Grundbegriffsarbeit, bei der der Hypervisor im Zentrum steht und mich zu einigen Meilensteinen der Informatik führt.

Der Hintergrund für diesen Beitrag sind Gespräche mit Systemadministratoren und historisch-konzeptionelle Probleme der Arbeit am und im Rechenzentrum (siehe den Schwesterbeitrag zu → Multiplizität). Im Fokus stehen Hinweise aus Handbüchern und wissenschaftliche Artikel der Informatik, die Virtualisierungstechnologien geprägt haben. Generell folge ich Methodologien der *Science and Technology Studies*, blicke also auf Technik und Wissenschaft im Vollzug, widme mich aus einer symmetrischen Perspektive computerwissenschaftlichen Konzepten und ihren historischen und kulturellen Kontexten, Übersetzungen und kontroversen Entscheidungen (vgl. Beck/Niewöhner/Sørensen 2012; Laser/Ochs 2018; Latour 2018).

## Eine falsche Fährte

Wir sind in empirischen Forschungsinterviews auf die Technologie des Hypervisors gestoßen, mit der wir ein universitäres Rechenzentrum historisch und materiell rekonstruieren. Besser gesagt: Wir sind auf die Virtualisierungstechnologie *gestoßen worden*. Und mit *wir* meine ich: Meine Forschungskollegin Estrid Sørensen und ich sprachen mit zwei Mitarbeitern aus der IT-Betreuung unserer Universität (→ Universität Bochum, Ruhr-). Wir führten das Thema der Verteilung von Ressourcen im Rechenzentrum als Problemfall ein (vgl. Sørensen/Laser 2023), die Schlagworte Bedarf und Nachhaltigkeit, und das Team des

Rechenzentrums verwies in ihrer Reaktion auf unseren Trigger fast im Vorbeigehen auf die unübertreffliche Effizienz ihrer virtuellen Maschinen. Ganz die Hobbybastler:innen hatten wir im Gespräch als Beispiel für eine *leichte* Rechenumgebung einen Raspberry Pi-Kleinstcomputer genannt. Dazu hatten wir im Vorhinein eine Mastodon-Instanz getestet, die Twitter-Alternative im langen Sommer dezentraler sozialer Medien (vgl. Laser et al. 2022) (→ Media, social). Wir wollten Interesse signalisieren, mit aktuellen Ideen und einem eigenen technischem Setup, einem scheinbar effizienten zumal.

»Ja, interessant«, so Thomas Kluge vom IT-Management (alle Namen pseudonymisiert).

Aber die IT war nicht begeistert. Nach kurzen Ausführungen zu aktuellen Cloud-Entwicklungen sagte Thomas: »Also ich glaube, was ein Riesenspunkt ist, und das beobachten wir selber, ein Riesengewinn an Effizienz ist, ist Virtualisierung.« Thomas begann zu rechnen:

»Also wir haben ja eine große VMware-Farm [eine Virtualisierungs-Technologie, auf die es zurückzukommen gilt; S.L.] an zwei Standorten, parallel gespiegelt, synchron gespiegelt aus Verfügbarkeitsgründen. Und das sind insgesamt (zögert) 24 Server, 12 auf jeder Seite, da laufen ungefähr etwas über 1.000 virtuelle Maschinen drauf. Und wir hatten mal geguckt, pro Seite ein Stromverbrauch von 6 Kilowatt, für 500 Maschinen. Also effizienter geht es, glaube ich, nicht.«

Er fügte hinzu: »Da kommt selbst ein Raspberry nicht mit«. Die konkrete Berechnung ist kompliziert und von diversen Unsicherheiten bestimmt, aber im Vergleich Virtuelle Maschine versus Pi benötigt die virtuelle Maschine maximal ein Drittel des Stromverbrauchs eines Pis.

Interessant, ja.

Wir merken, dass Virtualisierung den Kern des Rechenzentrums trifft: Mehrere digitale Systeme laufen flexibel und gleichzeitig nebeneinander, bei geringem Energieverbrauch (womit sich die Industrie zunehmend auseinandersetzen muss (vgl. Brodie 2020; Masanet et al. 2020; Pasek 2023)) und hoher Sicherheit (was ein klassisches Marketing-Instrument des Sektors ist (vgl. Dommann/Rickli/Stadler 2020)). Alles gut skalierbar. Aber: Was heißt das genau, wer handhabt eine solche virtuelle Infrastruktur mit welchen Tools? Daten sind für User:innen stets verfügbar, die Cloud weist den Weg (vgl. Velkova/Plantin 2023) und strukturiert Routinen (→ File, empty, → INF, → Lab of Unfinished Thoughts, → Text, plain). Aber wo siedeln wir die Wolke an, in einem imaginierten schwerelosen Raum, hoch im Himmel (vgl. Mackenzie 2003: 367)? Das ist wohl kaum mit virtuellen Maschinen gemeint. Welche Anforderungen und Kompetenzen benötigt eine virtuelle Umgebung, welchen Unterschied macht sie in welchen Situationen (→ Situationsanalyse, situierte)? Ja, was für eine Form der Virtualisierung ist es eigentlich, von der wir hier im Interview hören?

Diese Fragen waren der Aufhänger für ein zweites Treffen mit dem Virtualisierungs-Team, bei dem uns ihr eigenes System demonstriert wurde, um den Hypervisor und seine informatische Komplexität besser zu verstehen.

## Zur Geschichte und Aktualisierung der Kerntechnologie

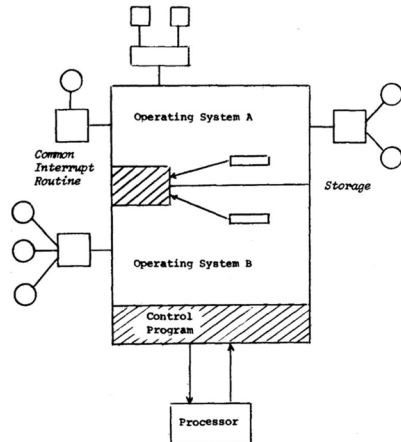
Die Technologie des Hypervisors wurde bereits in den 1970er Jahren entwickelt. Sie sollte das praktische Problem lösen, Instruktionen, die für einen bestimmten Computer geschrieben wurde, auf ein

neues System zu *übersetzen*. Die neuen Großmaschinen, die alle paar Jahre auf den Markt kamen, waren oft gänzlich anders verdrahtet und kodiert. Zur Handhabung der unterschiedlichen Logiken wurde als Hilfe ein Hypervisor in Form eines vorgeschalteten Kontrollprogramms konzipiert, um gleichzeitig auf einer Maschine einen Emulator und ein genuines Betriebssystem laufen zu lassen. Die Abbildung 1 macht dabei deutlich, wie erst das Kontrollprogramm die Trennung von Systemen ermöglichte. Emulator meinte dabei eine Mischung aus Hardware- und Software, ausgerichtet auf eine andere Architektur als die des eigentlichen Computers. Ein Hybrid. Er erlaubte reibungslosen Fortschritt in den Maschinenhallen. Der Begriff Emulator hat sich mittlerweile weiterentwickelt und nahezu von Hardware entkoppelt. Er ist auch vielen gewöhnlichen User:innen bekannt: Will ich etwa auf meinem PC ein Atari-Spiel aufsetzen, starte ich einen Emulator. Es ist auch ein Feature von neuen kommerziellen Cloud-Systemen und ferngewarteten Rechenzentren: User:innen können etwa emulierte Grafikkartenleistung auf lokale Systeme buchen, um sich sozusagen die eigenen Rechner mit Hilfe aus der Ferne aufzuwerten. Das sind kleine Errungenschaften, die jedoch auf eine größere Entwicklung verweisen. Der zusammen mit der Emulation entwickelte *Hypervisor* ist dank seiner Verankerung mit virtuellem Speicher, Partitionierung und durch die Erfahrungen mit Timesharing zur zentralen Schaltstelle im Rechenzentrum geworden (und auf diese Begriffe komme ich gleich zurück).

Mit dem zentralen Fachbegriff des Hypervisors begrüßten uns nun die IT-Admins bei unserer zweiten Software-Demonstration – mit der *Virtualisierungsschicht*, wie es unsere Gesprächspartner:innen auch formulierten. Dabei verwendet

die Universität ein kommerzielles Hypervisor-System.

Abb. 1: *Hypervisor Multiprogramming nach Katzan 1970: 111*



Die hier untersuchte Universität betreibt seit ihren Anfängen einen Maschinenraum, später ausgebaut und umfassend integriert in ein Rechenzentrum. Man begann mit einem systematischen Ausbau einer virtuellen Umgebung für das Rechenzentrum in den ausgehenden 2000er Jahren, mit einer Investition in VMWare – die damals und heute meistverbreitete kommerzielle Lösung virtueller Server. Schon vorher liefen einzelne virtuelle Systeme auf den universitären Rechnern. Es gab also Vorläufer, die allerdings Einschränkungen unterlagen. Wie es im Interview hieß: »Das Schöne war, man konnte dann nicht nur einen Dienst oder einen Server betreiben, sondern fünf, sechs, sieben, je nach Größe der Ausstattung. Aber das war es dann auch. Man konnte nicht wechseln, man hatte keine hohe Verfügbarkeit gehabt und gar nichts.« VMWare erlaubte die Multiplizier-

rung und Öffnung, mit einem Hypervisor als Herzstück.

VMWare ist eine Software-Umgebung, die Virtualisierung auf »bare« oder »close to the metal« realisiert (Rella 2023): mit direktem logischen Zugriff auf die physische Hardware. Dieses virtuelle System steuert mehrere Betriebssysteme und wird als ein »Hypervisor Typ 1« bezeichnet, wie es etwa im Glossar von VMWare definiert ist. In der Informatik ist das Prinzip auch als »Virtual Machine Monitor« bekannt (Popek/Goldberg 1974). Wenn ein Betriebssystem ein *Supervisor* mit Überblick über alle Anwendungen ist, ist ein *Hypervisor Typ 1* der Supervisor unter den Supervisoren. So ist es möglich, auf vielen separaten virtuellen Maschinen und über mehrere Geräte hinweg individuelle User mit eigenen Betriebssystemen zu starten.

Der Hypervisor überblickt dabei alles; er ist für die korrekte Zuweisung und Verarbeitung von digitalen Operationen auf der physikalischen Ebene zuständig, jongliert also CPU- und Grafikkarten-Leistung sowie Arbeitsspeicher-Adressen, und weist diesen Programmen und Betriebssystemen zu. Ein Hypervisor nutzt eine komplexe Partionierungs-Technologie, die es ermöglicht, auf einer Festplatte gleichzeitig mehrere Betriebssysteme bzw. unterschiedlichste Programme auszuführen, die unabhängig voneinander laufen, ja, die so arrangiert sind, dass die Anwender:innen nicht einmal wissen, dass *neben ihnen* auf der physischen Maschine andere Programme und Daten Arbeit verrichten. Nur der Hypervisor kennt die logische Verteilung von Isolation und Partitionierung der einzelnen Elemente. Beim *Hypervisor Typ 2* hingegen ist das anders, den startet man auf einem Desktop-Betriebssystem als ein ordinäres Programm, um darin dann ein Betriebssystem oder einzelne Programme zu starten. Das kann hilfreich

sein, um etwa ein Apple-System auf einem Windows-Rechner zu emulieren und Software in geschützten Umgebungen zu starten. Der *Hypervisor Typ 2* hat keinen vollen Zugriff auf Systemressourcen, ist langsam und für ein Rechenzentrum nicht in demselben Maße von Bedeutung.

Ein Hypervisor muss die Verschachtelungen und Logiken von Computerchips nutzen. Dabei sind Unterschiede zwischen den Chips wegweisend. Die Schaltkreise und Transistoren auf Mikroprozessoren werden nicht zufällig für jede Prozessorgeneration festgelegt, sondern folgen standardisierten Design-Richtlinien. Es ist zwar eine informatische Binsenweisheit, dass die Industrie nur eine Art des Computers kennt und bauen kann, eine universelle Maschine, die prinzipiell alle Programme lesen kann – wie von Alan Turing erdacht und von John von Neumann in einer Architektur skizziert: digital, binär, elektronisch, sequenziell (vgl. Heintz 1993: 213; Laser 2020: 238; Rammert 2013). Aber die Binse ist irreführend. Virtualisierung auf Servern mit Intel oder AMD-CPU (mit x86-Architektur) läuft zum Beispiel anders ab als auf SoC-basierten Smartphones (sog. simplifizierte *System-on-a-Chip*-Designs). Die Virtualisierungstechnik lässt sich nicht übertragen. (Auf SoC-basierten Smartphones ist Virtualisierung einfacher und sicherer; das als *Sandboxing* bekannte Vorgehen lässt insbesondere tradierte Desktop-Systeme alt, langsam und vulnerabel aussehen.) Dabei wird Ingenieur:innen von VMWare zugeschrieben, zur Jahrtausendwende erstmals die Virtualisierung von x86-Server-CPU realisieren zu haben. Die Chip-Hersteller Intel und AMD folgten schnell mit Hardware-seitigen Optimierungen. Server-Systeme wuchsen an. Universitäten waren unter den ersten Kunden, von einem regen Austausch der Akteure ist auszugehen. Auf

ihrer Unternehmensseite feiert das Unternehmen VMWare entsprechend ihre rasant wachsenden Buchungszahlen und ihr erstes Patent mit der ID 6397242, die 2002 für *System and Method for Virtualizing Computer Systems* ausgegeben wurde.

## Arbeit an und mit dem Hypervisor

Wir sehen nunmehr seit 20 Jahren eine Großinvestition in die Technologie der Virtualisierung, mit Universitäten und privaten Unternehmen als Mediatoren (→ Großinvestition). In unserer Universität ist man stolz auf die solide Ausbildung, die man als Organisation mittlerweile den hauseigenen Expert:innen für VMWare bieten kann und über die Jahre geboten hat. Und in den 2020er Jahren sind es mittlerweile Cloud-Unternehmen, die vom Austausch der Expertise am meisten profitieren. Es gibt heute nur einen kleinen Markt an Anbietern, die virtuelle Serverumgebungen für konventionelle Server-CPU's mit x86-Instruktionsset mit vollem Service anbieten. Und das hier genutzte VMWare ist sowohl Pionier als auch Marktführer, mit wenigen großen kommerziellen Konkurrenten wie Oracle, Microsoft und Citrix und offenen Alternativen wie Openstack mit Xen-Chip. Markt-Analysen stellen ein starkes Wachstum des Software-Sektors fest und prognostizieren in den nächsten Jahren einen weiteren Sprung. Auch dank des Fokus auf individualisierte User:innen erkennt hier Tung-Hui Hu die Motivation des Kapitals, ökonomische Verschwendung zu reduzieren und billige (vgl. Patel/Moore 2018) Rechenleistung anzubieten:

»Acting as a sanitary partition between users, virtualization ensures user productivity by removing the «wasted resources» of computing from the equation.

These previously illiquid expenditures of capital include the physical hardware of disk drives and servers, but also the labor involved in assembling them, the labor of maintaining and removing unsanitary elements (malware, trash, spam, even the occasional worm or virus) from the servers, and the physical stream of waste that those computers ultimately produce« (Hu 2015: 64)

Wenn Software über mehrere Maschinen verteilt ist und nur noch ein Hypervisor logisch nachvollziehen kann, wann und wo etwas verarbeitet wird, verlieren Daten aus Sicht der User:innen scheinbar ihren Ort. Bits brechen aus den engen Schluchten von vereinzelt CPU- und GPU-Gravierungen aus, elektrische Signale werden über Server hinweg weitergereicht und Anfragen gemeinsam bearbeitet, Sicherheitsrisiken bleiben isoliert. Software löst sich von Hardware. User:innen von physischen Grenzen. Virtuelle Server, anders gesagt, sind ein verdammt guter Trick. Die informatische Virtualisierung ist so gesehen auf Topologie ausgerichtet (vgl. ebd.: 67), und Admins achten darauf, dass Software über die Maschinen hinweg funktionsfähig bleibt, dass alle Orte auf der Karte erreichbar sind, dass Steuerung gelingt.

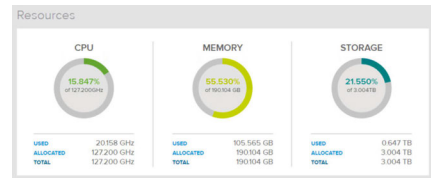
Wir sitzen in unserem Meeting mit den Systemadmins der IT-Infrastruktur und hören ihre Zusammenfassung der Situation. Operationen können flexibel Ressourcen zugewiesen werden, auf einem modernen VMWare-Hypervisor können Anwendungen und Speicherkapazitäten sogar im laufenden Betrieb zwischen Maschinen verschoben werden, automatisiert und je nach Auslastung angepasst; und der Hypervisor kann Leistung simulieren, um lokalen Anwendungen Potenzial und Kapazität zuzusichern, falls sie besondere Anforderungen zugesichert haben

wollen. Auch normale User:innen und ihre Erwartungen werden so gehandhabt. Der Hypervisor spielt den Unterschied zwischen realem, geladenem und virtuellem Speicher voll aus. Kauft sich eine Professorin im Rechenzentrum einen großen Datenspeicher ein, mag ihr System zwar einen entsprechend großen, freien Speicher anzeigen. Das bedeutet jedoch nicht, dass die virtuelle Maschine aktuell *tatsächlich* diesen Speicher okkupiert. Das System glaubt nur, dass es die Kraft und den Speicher hat, den es angezeigt bekommt. Die Professorin glaubt nur, dass die volle Kapazität des Speichers vorgehalten wird. Das System bekommt die Ressourcen nur bei wirklichem Bedarf gestellt. Aus Gesamtsystemsicht ergäbe sich ansonsten eine große Verschwendung. Es ist ein Fake. Und dann doch nicht ganz.

Ein Hypervisor regt zu Planung und Pflege an, dafür braucht es die System-Admins. Und eine digitale Repräsentation der virtuellen Schichten.

VMWare ist nicht nur, aber wesentlich über Dashboards organisiert. Das zeigt uns die Software-Demonstration eindrucksvoll. Der Eintritt in das virtuelle Management ist denkbar einfach. Unsere Gesprächspartner schließen ihren Laptop an einen Beamer an und loggen sich über ein Browser-Fenster in das Management der virtuellen Server ein. Und direkt erscheinen die bunten Farben und ausdrucksstarken Zahlen des Systems auf Dashboards. Das tatsächliche Design der Dashboards ist äußerst individuell, also auf das jeweilige Rechenzentrum zugeschnitten. Zentrale Systemwerte sind bei unserer Demonstration auf der Startseite, hier gibt es etwa allgemeine Werte zu sehen, Dinge wie CPU- und Arbeitsspeicher-Auslastung (Abb. 2).

Abb. 2: Beispiel von rudimentären Dashboard-Daten über einen Blog-Post von VMWare (Screenshot S.L.)



Während des Gesprächs fällt auf, dass die Admins mit einem bestimmten Blick auf das System schauen. Das Dashboard versteht aktuelle Software und voll erreichbare, sprich voll einsatzfähige, virtuelle Maschinen als *Gesundheit* und bietet visuell ansprechende Symbole zur Repräsentation dessen an. Nur was ist und worum geht es bei *Gesundheit*, wie prägt es die Arbeit mit dem Rechenzentrum?

Gesundheit ist zunächst ein Sammelbegriff, der sich durch alltäglichen Betrieb zieht. Zentral ist, dass das System im Hintergrund laufend Updates durchführt, alle Systeme spiegelt (n+1) und so vor Datenverlust schützt. Die Universität hat dafür gleich zwei voneinander getrennte Standorte aufgebaut. Ebenfalls aufschlussreich ist der historische und prozessbasierte Blick auf Auslastung und potenzielle Unter- oder Überlast. Anders gesagt: Auch virtuell können sich Maschinen verlaufen. Wie es im Gespräch hieß: »Wir haben natürlich eine Überwachung für die Server, die uns sagt, wenn CPU zu viel benutzt oder wenn CPU über 100 Prozent, wenn Memory [...] 100 Prozent [erreicht], wenn der Server nicht läuft usw. usf.« Die virtuelle Technik voll auszuschöpfen, bedeutet, dass das System berechnet, wie viel Last ein virtueller Server *eigentlich* gerne hätte. Beim Management und der weiteren Planung hilft der kommerzielle Anbieter gerne aus; so sagt der Interviewte weiter: »Wir haben eigene VMWare-Software,

die [...] Optimierungs- und Kapazitätsplanung kann und Sachen angibt.« Hier geht es um kurz-, wie auch langfristige Effizienz, Ausbau oder Einsparung, wie uns ein konkretes Beispiel zeigt:

»O.k. Hier gibt es jetzt einen Server, der hat im Moment vier CPUs. VMWare selber ist der Meinung, zwei müssten reichen. Und dazu ist der halt in der Lage, das rauszufinden. Und andersrum kann er das halt genauso. Er kann halt hier sagen: O.k. (überlegt kurz) Ja, das ist nicht immer nur CPU, das ist auch RAM [Arbeitsspeicher; S.L.]. So, dass er [...]. Er hat hier 16 Gigabyte RAM und eine Erhöhung um ein Gigabyte wäre sinnvoll, weil seine Metriken halt sagen: O.k. Der RAM-Verbrauch ist zu hoch im Moment.«

Maschinelle und menschliche (Selbst-)Beobachtung verschränken sich und prägen die Lebenswelt des Hypervisors. Im Kern fordert der Begriff Gesundheit dazu auf, den Status einzelner virtueller Maschinen zu beobachten und zu evaluieren. Virtuelle Maschinen laufen bei VMWare mit jeweils individueller Software, die etwa von Wissenschaftler:innen oder dem Verwaltungspersonal genutzt werden. Diese Maschinen haben demnach konkrete Anforderungen an Software-Aktualisierung, die sowohl mit den physischen Maschinen (z.B. Chip-Generationen) als auch der verwendeten VMWare-Software (das gebuchte Service-Paket) im Einklang sein müssen. Die Soft- und Hardware im Sinne des Anbieters aktuell zu halten, ist doppelt entscheidend: Es garantiert, dass Maschinen laufen, und es sichert der IT-Administration zu, dass sie als Kund:in bei VMWare zielgenauen Support bekommen. Es gibt strenge Vorgaben hinsichtlich Kompatibilität, die laut VMWare zu handhaben sind. Und die IT-Administration muss laufend Updates antizipieren. Mit manchen Anbietern wurden Non-Disclosure Agree-

ments geschlossen, um vorab über neue Versionen und Schwachstellen zu lernen, auf die es zu reagieren gilt. Virtualität ist Handarbeit (→ Denkmal, virtuelles). Es wird angezeigt, wenn Hardware nicht kompatibel oder Versionen veraltet sind. Visuelle Aufarbeitung ist gerne gesehen, Skripte und Automatisierung machen die Arbeit leichter.

Das Dashboard lässt uns reisen, lässt Admins zur Pflege der Gesundheit kreativ werden. Ein ethnographischer Protokollauszug aus der Demonstration:

»Die Arbeit an der Umgebung ist spielerisch. Aber abgegrenzt. Die Dashboards der Virtualisierungsumgebung sind spezialisiert, und (der zweite anwende IT-Kollege) Jens Biermann weist mehrmals darauf hin, dass *ein* Dashboard schnell durch weitere Ansichten ergänzt werden muss, durch andere Dashboards, durch eigenen Code, durch Anwendungen, die von der Online-Community entwickelt und eingebaut werden können. Herr Biermann will demonstrieren, schnappt sich den Laptop von Kollege Weiß, und wählt sich mit seinen Daten bei spezifischen Anwendungen ein, die ebenfalls von VMWare angeboten werden oder die über die Website abgerufen werden können. Als Sonderfall stellt er uns die Kompatibilitätslisten vor, die offiziellen Kompatibilitätsmatrizen genannt werden.«

Bemerkenswert ist der Community-Teil der Entwicklung. VMWare ist kommerziell, aber System-Admins auf der ganzen Welt entwickeln kreative Formen, mit der Schnittstelle, den Dashboards oder Datenströmen umzugehen. Unser Gesprächspartner klingt fast begeistert davon, wie virtuelle Server eine eigene Welt entfesseln:

»Also, es gibt immer wieder Community-Produkte oder so Community-Versionen, die auch in die Produktiv Pro-

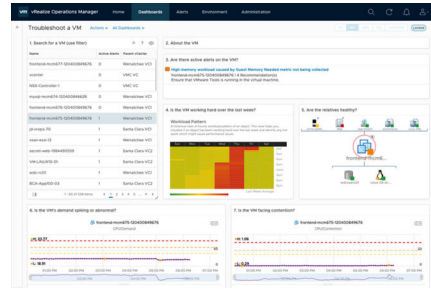
duktion von VMWare mit einfließen. Also, es gibt eine riesige Community bei VMWare, die alle irgendwelche Plug-ins für die Webseiten, für irgendwelche Konfigurationen, die dann mit in die Produkte auch einfließen hinterher. Also, das dauert natürlich ein bisschen, weil das ein Riesenaufwand ist, also auch für VMWare selber, aber die fließen dann hinterher auch, wenn das gute Ideen sind, mit ein.«

VMWare hat zudem eigene *Troubleshooting*-Dashboards, die Admins Lösungsvorschläge unterbreiten, neuerdings mit Künstlicher Intelligenz ausgeschmückt. Abbildung 3 gibt dafür ein Beispiel, bereitgestellt von der Öffentlichkeitsarbeit von VMWare.

*Alles ist aus einer Hand*, dafür schätzt die Leitung der IT-Administration den kommerziellen Anbieter VMWare. Mit nur wenigen Stellschrauben lässt sich das gesamte System koordinieren. Nicht alle an der Universität finden diese enge Anbindung an kommerzielle Software gut, vor allem, weil VMWare zugleich Beiträge aus der Online-Community vereinnahmt. Die Kritiker:innen wünschen sich eine freie und offene Alternative. Aber für die IT-Administration bedeutet der kommerzielle Service eine Sicherheitsgarantie: »Weil, auf der Umgebung da ruht ein Großteil der Infrastruktur der [Organisation]. Und wenn man da jetzt irgendeine Bastellösung macht, was man sicher auch machen kann, wo man aber im Prinzip keinen Support einkaufen kann, und dann hat man ein Problem, das wirklich betriebsverhindernd ist, und man hat dann keinen, der einem das Problem löst.« Das sind unterschiedliche Deutungen von dem, was der *Hyper* im *Hypervisor* leisten soll: größtenteils ausgelagerte vs. eigenständige Übersicht des Systems und ihrer Einzelteile. Dabei wird die Handlungsinitiative des Hypervisors verhandelt. Es

ist ein Akteur, der Überblick in eine Situation bringt, die Menschen aufgrund der virtuellen Qualität nicht überblicken können.

Abb. 3: *Troubleshoot a VM* (Johnson 2019; Screenshot S.L.)



## Der Hypervisor heute und morgen

Der Hypervisor ist angekommen. Aber wird er bleiben? Im universitären Rechenzentrum ist er als Instanz geschätzt und passt offenbar gut zu den Anforderungen der diversen, oft individualisierten User:innen. Aber im weltweiten Cloud-Markt gibt es bereits einen neuen Trend: Virtualisierung. Noch mehr Virtualisierung. Denn: Maschinen und Menschen sind zur Last geworden. Wieso muss ein Rechenzentrums-Management überhaupt einzelne Server handhaben, wozu braucht es mehr oder weniger händisch gesteuerte Updates und Upgrades, manchmal sogar gezielte Allokation von virtuellen Ressourcen? Und warum laufen diverse Betriebssysteme nebeneinander, wenn doch oftmals nur Applikationen gefordert sind? Das sind Fragen, die an die Vergangenheit der virtuellen Speicherverwaltung erinnern (→ Multiplizität), neue alte Fragen, die die praktische Informatik an die Industrie richtet – und die insbesondere von großen Cloud-Anbietern wie Goog-

le oder Microsoft aufgegriffen werden. Für Rechenzentrumsbetreibende sind das bisweilen unangenehme Fragen, denn Aufgaben, Kompetenzen und Ressourcen verschieben sich, erneut. Kurzum: Nach dem Hypervisor stehen nunmehr *Cluster Management Systeme* im Fokus der Aufmerksamkeit, eine neue Art der Virtualisierungsschicht. Ein solcher Cluster zerschneidet die scharfe Trennlinie zwischen Hypervisor *Typ 1* und *2* und setzt alles auf den Einsatz von virtuellen *Containern*, also das Management von Aufgaben und Prozessen in Form von Applikationen. Das Cluster-System betreut die Container über alle Maschinen hinweg, ja sogar über einzelne Rechenzentren hinaus. Ein Interface bleibt übrig. Bereitgestellt von *Docker* oder *Kubernetes*. Konzeptionell betrachtet treibt ein solches System den Grundgedanken des Hypervisors einen Schritt weiter, selbst wenn an der Automatisierung der verschiedenen Prozesse noch hitzig gearbeitet wird, wie Munn (2021) am Beispiel von Alibabas System nachzeichnet. Alles wird zur virtuellen Planungsaufgabe (dem *scheduling*). Die Cluster-Systeme wollen dabei maschinelle Ressourcen verstecken, neue Dashboards betrachten die Effizienz und sozusagen die gesunde Verteilung von Prozesslast, und die Cluster-Systeme sind darauf angelegt, Serversysteme über Orte hinweg miteinander zu kombinieren und ein global integriertes Rechensystem zu entwerfen. Es geht mehr um *infra* denn *hyper*.

Der Streifzug durch die virtuelle Serverlandschaft unserer Universität hinterlässt einen Eindruck von permanenter Ambivalenz. Das Adjektiv *virtuell* war im Lauf der Computer-Geschichte – von den Großrechnern zum heutigen Rechenzentrum – immer wieder eine kleine Provokation. »Wie könnt ihr nur?«, so lesen sich die Reaktionen im Rückblick. Ver-

antwortungen wurden neu verteilt, neue Schichten hinzugezogen, neue Geräte dazwischengeschaltet, alte weggenommen (→ Multiplizität). Und doch trieb Virtualisierung oft Wandel an, verschränkte sich mit Digitalisierungspraktiken unterschiedlicher Art, letztendlich auch mit kommerziellen Plattformlogiken der Programmierbarmachung und Extraktion von digitalen Handlungen (vgl. Helmond 2015). So lässt sich der konzeptionelle Wert des Hypervisors einfangen. Mit Blick auf Computer-Architekturen hat laut Oxford English Dictionary »to virtualize« (O.V. o. J.) zwei Bedeutungen: in eine digitale Form übertragen einerseits und eine virtuelle Version von Computer-Ressourcen erzeugen andererseits. Gemeint ist bei virtuellen Servern definitorisch betrachtet letzteres: das effiziente Nutzen und Zuweisen von Server-Ressourcen. Nichts anderes leistet der Hypervisor: Ressourcen werden scheinbar von ihren materiellen Lasten befreit, kopiert, multipliziert, neue Möglichkeiten erscheinen am Horizont und setzen sich rasant durch. Das Alltagsverständnis eines Rechners wird infrage gestellt und die Hardware selbst bekommt eine virtuelle Komponente. Dabei entwickelt die Virtualisierungsschicht ein Eigenleben. Die Arbeit am Rechenzentrum verschiebt sich auf die Ebene des Codes und laufende Software-Updates, neue Unsicherheiten und neue Unzulänglichkeiten eingeschlossen. Es ist eine Pflegearbeit, die lästig sein kann. Aber in unseren Interviews schwingt auch immer mit, dass über Virtualisierung Digitalisierung möglich gemacht wird. Der Hypervisor treibt die digitale Transformation voran. Damit verweist er auf beide Bedeutungsebenen von Virtualisierung und steht im Zentrum des Begriffs.

## Literatur

- Beck, Stefan/Niewöhner, Jörg/Sørensen, Estrid (2012): »Einleitung. Science and Technology Studies aus sozial- und kulturanthropologischer Perspektive«, In: Stefan Stefan/Jörg Niewöhner/Estrid Sørensen (Hg.), *Science and Technology Studies: Eine sozialanthropologische Einführung*, Bielefeld: transcript, S. 9–48.
- Brodie, Patrick (2020): »Climate extraction and supply chains of data«, in: *Media, Culture & Society* 42(7–8), S. 1095–1114.
- Dommann, Monika/Rickli, Hannes/Stadler, Max (2020): *Data centers: Edges of a wired nation*, Zürich: Lars Müller.
- Edwards, Dustin/Cooper, Zane Griffin Talley/Hogan, Mél (2024): »The making of critical data center studies«, in: *Convergence* (Online First).
- Heintz, Bettina (1993): *Die Herrschaft der Regel: Zur Grundlagengeschichte des Computers*, Frankfurt a.M./New York: Campus.
- Helmond, Anne (2015): »The platformization of the web: Making web data platform ready«, in: *Social Media + Society* 1/2.
- Hu, Tung-Hui (2015): *A prehistory of the cloud*, Cambridge/Massachusetts: MIT Press.
- Johnson, Josh (2019): »5 steps to effective dashboard design«, in: *VMware design*. Online unter: <https://medium.com/vmwaredesign/5-steps-to-effective-dashboard-design-c1813455e159> (letzter Zugriff: 01.04.2024)
- Katzan, Harry (1970): »Operating systems architecture«, In: *Proceedings of the May 5–7, 1970, Spring joint computer conference, AFIPS '70 (Spring)*, New York: Association for Computing Machinery, S. 109–118.
- Laser, Stefan (2020): *Hightech am Ende. Über das globale Recycling von Elektroschrott und die Entstehung neuer Werte*, Wiesbaden: Springer VS.
- Laser, Stefan/Ochs, Carsten (2018): »Kontroversen bewertbar machen. Über die Methode des Mapping of Controversies«, in: Jonathan Kropf/Stefan Laser (Hg.), *Digitale Bewertungspraktiken: Für eine Bewertungssoziologie des Digitalen*, Wiesbaden: Springer VS, S. 97–125.
- Laser, Stefan/Pasek, Anne/Sørensen, Estrid/Hogan, Mél/Ojala, Mace/Fehrenbacher, Jens/Hepach, Maximilian Gregor/Çelik, Leman/Ravi Kumar, Koushik (2022): »The environmental footprint of social media hosting: Tinkering with Mastodon«, in: *EASST Review* 41(3).
- Latour, Bruno (2018): *Aramis: oder Die Liebe zur Technik*, Tübingen: Mohr Siebeck.
- Mackenzie, Adrian (2003): »These things called systems: Collective imaginings and infrastructural software«, in: *Social Studies of Science* 33(3), S. 365–387.
- Masanet, Eric/Shehabi, Arman/Lei, Nuo/Smith, Sarah/Koomey, Jonathan (2020): »Recalibrating global data center energy-use estimates«, in: *Science* 367/6481, S. 984–986.
- Munn, Luke (2021): »Imperfect orchestration: Inside the data center's struggle for efficiency«, in: *Computational Culture* 8.
- OED (o. J.): »to virtualize«, in: *Oxford English Dictionary*.
- Pasek, Anne (2023): »On being anxious about digital carbon emissions«, in: *Social Media + Society* 9(2).
- Pasek, Anne/Lin, Cindy Kaiying/Cooper, Zane Griffin Talley/Kinder, Jordan B. (2023): *Digital energetics*, Minneapolis: Meson Press.

- Patel, Raj/Moore, Jason W. (2018): *Entwertung: Eine Geschichte der Welt in sieben billigen Dingen*, Berlin: Rowohlt.
- Pickren, Graham (2018): »The global assemblage of digital flow«: Critical data studies and the infrastructures of computing«, in: *Progress in Human Geography*, 42(2), S. 225–243.
- Popek, Gerald J./Goldberg, Robert P. (1974): »Formal requirements for virtualizable third generation architectures«, in: *Communications of the ACM* 17(7), S. 412–421.
- Rammert, Werner (2013): *Technik aus soziologischer Perspektive 2: Kultur – Innovation – Virtualität*, Wiesbaden: Springer-Verlag.
- Rella, Ludovico (2023): »Close to the metal: Towards a material political economy of the epistemology of computation«, in: *Social Studies of Science* 54(1), S. 3–29.
- Sørensen, Estrid/Laser, Stefan (2023): »Towards artful sustainable integration of IT infrastructures: A report from the construction of a university data centre«, in: Patricia Jankowski/Anja Höfner/Marja Lena Hoffmann/Friederike Rohde/Rainer Rehak/Johanna Graf (Hg.), *Shaping digital transformation for a sustainable society: Contributions from Bits & Bäume*, Berlin: Technische Universität Berlin, S. 87–90.
- Velkova, Julia/Plantin, Jean-Christophe (2023): »Data centers and the infrastructural temporalities of digital media: An introduction«, in: *New Media & Society*, 25(2), S. 273–286.

## INF

Herausgeber:innen

Abkürzung der Teilprojektvariante INF *Informationsinfrastruktur: Technik und Praxeologien* am SFB 1567 *Virtuelle Lebenswelten*, deren Aufgabe es ist, Probleme zu lösen und zu machen (→ 1567, → Klappkiste, → Text, plain).

## Insel, virtuelle

Ann-Carolyn Hartwig

Warum Insel, wenn Fragen von und über → Virtualität verhandelt werden sollen? Virtualität muss als Phänomen beschreibbar gemacht bzw. beschrieben werden – insbesondere dann, wenn Gegenstände einer vordigitalen Virtualität in den Fokus rücken. In den 1990er Jahren entwickelt sich aus dem Interesse an Inseln ein eigenes Forschungsfeld: die *Island Studies* (vgl. Breuer 2012: 184–185). Dieses Interesse ist v.a. darin begründet, dass Inseln »nicht nur Sehnsuchtsorte, sondern zudem sowohl form- und ordnungsstiftende Denkfiguren als auch epistemologische Analyseinstrumente« (Ramponi/Wendt/Wilkens 2011: 7) sind. Sie leisten also einen Beitrag zu kultur- und gesellschaftswissenschaftlichen sowie wissensgeschichtlichen Untersuchungen.

Vor der Folie der Virtualität ist die Insel bislang noch nicht betrachtet worden. Ihr Potential als Beschreibungsmuster des Virtuellen wird hier nun mit Blick auf zwei Robinsonaden des 18. Jahrhunderts, Daniel Defoes *Robinson Crusoe* (1719) und Johann Gottfried Schnabels *Insel Felsenburg*