ComConsult Research

Der Netzwerk Insider

April 2018

Systematische Weiterbildung für Netzwerk- und IT-Professionals

VDI aus der Cloud

von Markus Schaub

Virtuelle Desktop-Umgebungen sind nun wirklich nichts Neues. Schon lange gibt es Terminal-Serverdienste für unterschiedliche Anforderungen: sei es die Remote-Administration des eigentlichen Servers oder das Streaming von einzelnen, zentral gehosteten Anwendungen.

Manch einer hält sie sogar in Zeiten von mobilen Apps und Webanwendungen wie Office 365 oder G-Suite für eine aussterbende Art. Andere wiederum sehen sie als wichtigen Baustein für eine vollständig „cloudifizierte“ IT mit dummen Clients der Marke Chrome OS und Co. „Cloud“ ist für diesen Artikel das Stichwort, denn in der Tat beobachten wir, dass immer mehr Anwendungen in die Cloud verlagert werden und es zudem Kunden gibt, die bei der Planung neuer Standorte, eigene Rechenzentren gar nicht mehr berücksichtigen. Das sind aber nur zwei Beispiele, warum es interessant sein kann, virtuelle Desktops nicht selbst zu hosten, sondern ihren Betrieb in die Cloud zu verlagern. Dabei gibt es nicht nur verschiedene Angebote der Cloud Provider, sondern auch Anforderungen wie ausreichend Bandbreite, die kundenseitig erfüllt werden müssen. Dieser Artikel zeigt exemplarisch auf, welche Stärken aber auch Schwächen eine VDI-Struktur in der Cloud mit sich bringt und welche Voraussetzungen vorhanden sein müssen.

weiter ab Seite 8

ONL, ONIE und P4
Drei Bausteine zum Aufbau moderner Layer 3 Switche

von Markus Geller

Was sich zunächst nach einer Amerikanischen Sportliga, einer ostasiatischen Gottheit und einem veralteten CPU Typ anhört, ist ein noch relativ junger Trend im Netzwerkumfeld.

Wer heute eine PC oder einen Server in Betrieb nimmt, der stellt sich zunächst einmal die Frage: welches Betriebssystem wähle ich aus? Diese Frage ist von verschiedenen Faktoren abhängig. Zum einen hängt die Auswahl an den genutzten Anwendungen, die im Nachgang installiert werden sollen, aber auch von persönlichen Vorlieben ab. So hat sich im privaten wie im betrieblichen Umfeld Microsoft Windows in all seinen Varianten als Betriebssystem für PC und Serversysteme etabliert. Das liegt natürlich ursächlich an der großen Auswahl von Anwendungen, die hierfür zur Verfügung stehen. Im Server Umfeld sehen wir parallel eine starke Verbreitung von Linux Distributionen, die ebenfalls ihren berechtigten Platz als Serverbetriebssystem einnehmen.

weiter auf Seite 16

Geleit

Strategien gegen die Cloud? Wie hoch sind die Erfolgsaussichten?

auf Seite 2

Standpunkt

Enterprise IT im Sumpf des IoT

auf Seite 23

Aktuelles Seminar

Information Security Management mit ISO 27001 und BSI-Grundschutz

auf Seite 22

Aktueller Kongress

ComConsult Netzwerk Forum 2018

ab Seite 4

Geleit

Strategien gegen die Cloud? Wie hoch sind die Erfolgsaussichten?

Als Cloud-Technologie in den Markt eingeführt wurde, gehörte ComConsult Research auch zu den Kritikern dieser Entwicklung. Wir haben immer den Mehrwert von SaaS gesehen und betont, aber der Nutzen von Infrastructure as a Service IaaS war in der Breite für uns nicht nachvollziehbar. 30% theoretische Ersparnis plus Mehrkosten für die Anbindung und den Betrieb können keine Motivation sein, die Unternehmens-IT komplett umzustellen. Da ist es besser, die lokalen Einspar-Potenziale zu heben und einige erfolgreiche Cloud-Technologien intern zu nutzen. Und für den Erfolg brauchen wir eine breite Akzeptanz und nicht einzelne Ausnahmefälle, die den Nutzen plausibel machen.

Nun, inzwischen hat sich die Situation geändert. Aber warum ist das so? Warum ist heute die ursprüngliche Ablehnung der Cloud nicht mehr durchsetzbar? Was hat sich geändert? Generell kann man sagen, dass wir mehr und mehr Lösungen in der Cloud bekommen, die wir so lokal nicht haben. Hinzu kommen Anwendungen, die von ihrer Natur her die Cloud zwingend erfordern. Es gibt zwei Beispiele, bei denen wir im Moment ein Umdenken bei den Top-Führungskräften unserer Kunden beobachten:

- Building Information Modelling BIM: hier entsteht in vielen Fällen ein digitaler Zwilling in der Cloud, der die Basis für die Optimierung des Betriebs legt und eine maximal effiziente Nutzung eines Gebäudes gestattet. 70% der Gesamtkosten eines Gebäudes liegen im Betrieb und nicht im Bau. Plötzlich haben wir hier eine Technologie, die dazu beiträgt, dass nicht nur Kosten gesenkt werden können, sondern auch ein Garant für eine einfachere Vermietbarkeit eines Gebäudes ist. Kombiniert man das mit neuen Technologien im digitalen Gebäude der Zukunft und deren Abbildung im Zwilling, dann wird die Sache rund. Und was passiert: selbst die Hardliner, die Cloud um jeden Preis bisher vermieden haben, springen auf einmal und vor allem ohne zu zögern auf den Zug. Wer hier im Wettbewerb um den Mieter nicht verlieren will, der wird diesen Weg gehen. Zwar gibt es Lösungen mit einem lokalen Zwilling, aber immer mehr Hersteller von Sensortechnik für Gebäude verbinden ihre Lösungen mit Cloud-Servern. Dies ist ein typisches Beispiel dafür, dass wir auf Dauer keine lokale Alternative haben werden.

- Unified Communications and Collaboration UCC: nach einem zögerlichen Start rollt jetzt die Welle in diese Richtung, noch langsam, aber sie rollt. Interessanterweise haben kleine Startups wie Slack und Trello die Tür geöffnet. Dann folgten Cisco und Unify und seit Microsoft dies zu einem Kern von Office 365 ernannt hat, kann niemand diese Welle mehr aufhalten. UCC ist ein gutes Beispiel für einen Anwendungsbereich, der ohne Cloud gar nicht funktionieren kann. Die Integration mobiler Endgeräte an beliebigen Orten und die Kommunikation mit externen Teilnehmern in Kombination mit Dokumenten-Kollaboration geht funktional nur mit einer Cloud-Drehscheibe für die Kommunikation.

Wenn wir diese Beispiele betrachten, was ist so anders als zum Beginn der Entwicklung? Nun, bei Infrastructure as a Service hatten wir eine direkte Konkurrenz zwischen Rechenleistung und Speicher in der Cloud und im lokalen Rechenzentrum. Und für 90% der Anwendungen war nicht erkennbar, wo der Vorteil der Cloud liegt. Zwar konnte die Cloud einige Leistungen preiswerter erbringen als wir das lokal können, aber die Schnittstellen- und Betriebskosten haben diese Vorteile wieder aufgehoben. IaaS in seiner Reinform konnte und kann bis auf Ausnahmen keinen so großen Mehrwert für Unternehmen bieten, dass sich ein Umstieg in die Cloud lohnt.

Was ist also so anders an den genannten Beispielen: ganz einfach, hier leistet die Cloud etwas, das wir lokal nicht haben. Sie liefert nicht nur einen signifikanten Mehrwert, sie generiert fast einen Zwang, diese Dienste zu nutzen. BIM geht auf Dauer nicht ohne Cloud. Und Kollaboration geht nicht ohne Cloud. Und genau an dieser Stelle könnten wir jetzt eine inzwischen lange Liste von Diensten und Produkten starten, die es einfach lokal nicht gibt. Und sobald man anfängt Cloud-Produkte zu nutzen, hat man ein kleines Problem. Diese Produkte sind konzeptionell darauf ausgelegt, mit anderen Cloud-Produkten zusammen eingesetzt zu werden. Wer Salesforce einsetzt, der wird mindestens Box und Zoom evaluieren. Wer Goggles G-Suite und Slack einsetzt, der wird Okta als eine Möglichkeit evaluieren. Die Konsequenz: es gibt dieses “ein wenig Cloud” auf Dauer nicht. Zumindest bei SaaS gilt: ganz oder gar nicht.

Gilt das nur für SaaS? Nein, auch Platform as a Service PaaS muss mittlerweile sehr ernst genommen werden. Zu Beginn der Entwicklung vor mehr als 10 Jahren konnten diese Produkte mit den ausgefeilten und sehr spezialisierten Vor-Ort-Produkten nicht konkurrieren. Das ist heute nicht mehr so. Eine Amazon AWS oder Microsoft Azure haben so viele Alleinstellungsmerkmale, dass sich nicht mehr die Frage stellt, ob ich für bestimmte Anwendungen ein PaaS Produkt nutze, sondern mehr welches ich nutze. Wir haben in einem sehr aufwendigen Test Amazon und Microsoft verglichen, indem wir unser Study.tv Angebot komplett in beide Welten portiert haben. Und study.tv ist komplexer als es aussieht. In dem relativ kleinen Produkt stecken alle Technologien, die sie auch für beliebig große Lösungen brauchen. Wir werden in den nächsten Monaten über den Vergleich und seine Ergebnisse sprechen. Aber so viel vorweg: Study.tv ist heute in der Cloud. Die Testergebnisse waren so eindeutig positiv, dass eine rein gehostete Lösung auch mit den neuesten Technologien keinerlei Sinn mehr machte. Zumindest für uns ist klar: sie können mit einer rein lokalen Umgebung nicht mit einem PaaS-Ansatz konkurrieren. Dies war vor einigen Jahren noch anders, aber inzwischen wird die Lücke immer größer. Microsoft unterstützt mit Azure einen Hybridansatz und das macht für Microsoft-Umgebungen Sinn. Aber für Produkte wie Study.tv, die sich komplett an externe Benutzer wenden, ist jeder Hybridansatz sinnlos. Für solche Anwendungen gehört die Zukunft der Cloud. Und wir werden dies in den nächsten Monaten in diversen Vorträgen und Analysen belegen.

VDI aus der Cloud

Fortsetzung von Seite 1

Markus Schaub ist seit 2009 Leiter von ComConsult-Study.tv. Er verfügt über umfangreiche Berufserfahrung in den Bereichen Netzwerken und VoIP und ist seit mehr als 13 Jahren bei ComConsult beschäftigt. Seine Schwerpunkte liegen im Netzwerk-Design, IP-Infrastrukturdiensten und SIP, zu denen er viele Vorträge auf Kongressen hielt, erfolgreich Seminare durchführte und zahlreiche Veröffentlichungen schrieb.

Dazu werden die zwei Varianten „Virtual Desktop“ und „Application Streaming“ anhand der AWS Implementierung vorgestellt und verglichen. Das Augenmerk liegt dabei auf den infrastrukturellen Voraussetzungen. Die Verwaltung der virtuellen Umgebungen und der Software würde den Rahmen eines einzelnen Artikels weit übersteigen.

Desktop

Virtuelle Desktops heißen “WorkSpaces” bei AWS.

Wer glaubt, bei WorkSpaces handelt es sich nur um das Starten eines virtuellen Rechners und der Verbindung per Browser dorthin, täuscht sich. Bevor man die eigentlichen virtuellen Desktops aufsetzen kann, sind einige Vorbereitungen notwendig: Netzwerke müssen angelegt, Zugiffsbeschränkungen und Freigaben vorbereitet sowie der spätere Internet-Access der WorkSpaces, so denn gewünscht, sichergestellt werden.

Voraussetzung: Netzwerk

Zunächst sollte man prüfen, ob die Virtual Private Cloud (VPC), in der die WorkSpaces laufen sollen, die notwendigen Voraussetzungen erfüllt und wenn nicht, diese einpflegen. Dabei geht es konkret um drei Punkte: Netzwerk-Design, Internet-Access und Sicherheit/Zugriffbeschränkung.

Die virtuellen Desktops werden netzwerktechnisch später Subnetzen zugeordnet werden. Um ihnen bei Bedarf im weiteren Verlauf Zugriff auf das Internet zu gewähren, gibt es drei Varianten:

- NAT
- Dynamische IP
- Manuelle IP

Empfohlen wird NAT. Für eine Handvoll oder gar nur einen einzelnen WorkSpace wäre das natürlich übertrieben, jedoch macht es durchaus Sinn, eine ganze Farm von WorkSpaces via NAT mit dem Internet zu verbinden, da so IPv4 Adressen eingespart werden. Also genau so, wie es bei realen Desktops im Unternehmen auch gehandhabt würde. Grundsätzlich ist auch IPv6 möglich, jedoch gibt es da noch (?) Beschränkungen was die Ausstattungsvarianten der Virtuellen Desktops angeht: die beiden leistungsfähigsten Bundles „Performance“ und „Graphics“ unterstützen keine automatische IPv6-Zuweisung.

Abbildung 1 zeigt, wie das Design mittels NAT-Gateway aussieht. Die in der Abbildung dargestellten Tabellen entsprechen den Routingtabellen der Subnetze: während die WorkSpaces das NAT-Gateway als Default Gateway eingestellt haben, nutzt das Default Gateway selbst natürlich das Internet Gateway des VPC.

Gemäß der Sicherheitsmaxim „Microsegmentierung“ legt man als nächstes die Access-Listen und Security Groups für die WorkSpaces an. Dabei müssen eine Reihe von Ports geöffnet werden, damit die verschiedenen Remote-Zugänge funktionieren. Der Remote-Client benötigt für das PCoIP den Port 4172 für UDP und TCP, für den Webzugang braucht man die UDP (!) und TCP Ports 80 und 443. Hinzu kommt noch inbound der TCP Port 8200 für das Management und UDP Port 55002 für PCoIP Streaming.

Zu diesen Sicherheitsregeln, die nur für die Verwaltung und den Zugriff auf die WorkSpaces benötigt werden, kommen natürlich noch sämtliche Sicherheitsregeln, die man für andere Systemdienste innerhalb der Subnetze benötigt (bspw.

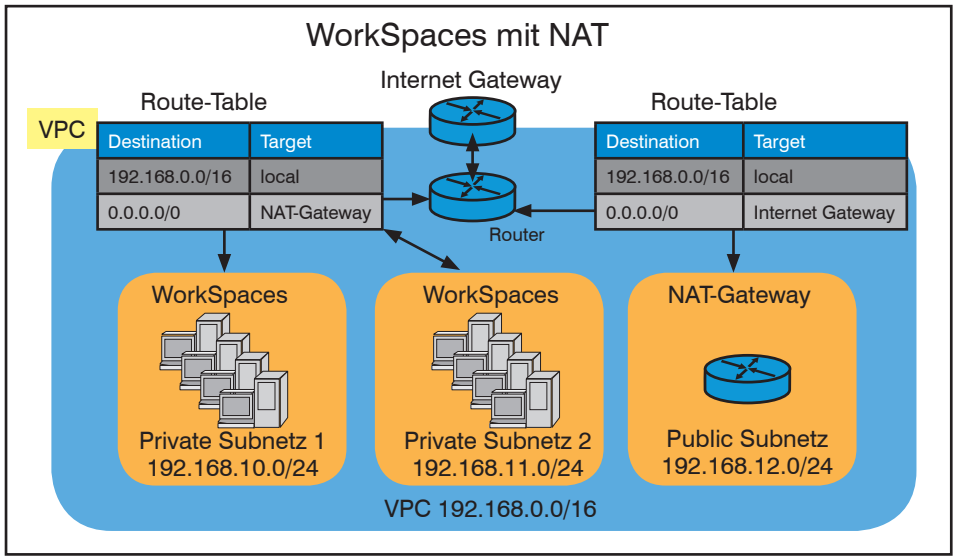
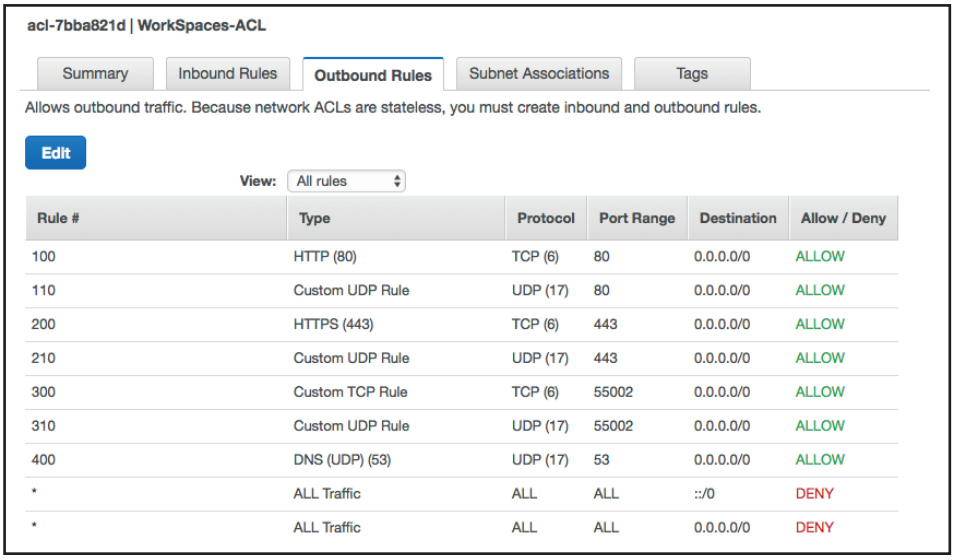


Abbildung 1: WorkSpaces Design mit NAT

für das Active Directory, bzw. den AD-Connector, DNS, NTP, etc.). Außerdem muss man ggf. Ports für die Anwendungen freigeben, die später auf den virtuellen Desktops laufen.

Abbildung 2 zeigt ein Beispiel für eine ausgehende Access-Liste. Man kann schon an diesem minimalistischen Beispiel sehen, dass das sehr schnell sehr unübersichtlich werden kann. Insbesondere wenn man sowohl mit Access-Listen als auch mit Security Groups arbeitet. Eine aktuelle und übersichtliche Dokumentation ist hier Pflicht!



Rule #	Type	Protocol	Port Range	Destination	Allow / Deny
100	HTTP (80)	TCP (6)	80	0.0.0.0/0	ALLOW
110	Custom UDP Rule	UDP (17)	80	0.0.0.0/0	ALLOW
200	HTTPS (443)	TCP (6)	443	0.0.0.0/0	ALLOW
210	Custom UDP Rule	UDP (17)	443	0.0.0.0/0	ALLOW
300	Custom TCP Rule	TCP (6)	55002	0.0.0.0/0	ALLOW
310	Custom UDP Rule	UDP (17)	55002	0.0.0.0/0	ALLOW
400	DNS (UDP) (53)	UDP (17)	53	0.0.0.0/0	ALLOW
*	ALL Traffic	ALL	ALL	::/0	DENY
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY

Abbildung 2: Beispiel für eine ausgehende ACL für WorkSpaces

Will man die Sicherheit weiter erhöhen, so kann man den Zugriff auf die WorkSpaces einschränken. Per Default ist der Zugriff von allen Geräten mittels Client aus möglich, der Zugriff über Webbrowser hingegen nicht. Das kann jedoch leicht auf bestimmte Clients eingeschränkt werden. Insb. ist es ratsam, Tablets und erst recht Smartphones auszuschließen, da die meisten Windows Anwendungen dafür nicht geeignet sind. Das spart dann jede Menge Nerven beim Support. Ein wirklicher Sicherheitsgewinn ist das jedoch noch nicht. Der nächste Schritt ist die Arbeit mit Zertifikaten: man kann den Zugriff auf die WorkSpaces so beschränken, dass der Client beim Verbindungsaufbau ein gültiges, bekanntes Zertifikat vorweisen muss, dessen Gegenpart zuvor im Directory hinterlegt wurde. So kann man sicherstellen, dass nur „zertifizierte“ Clients Zugriff auf die eigenen WorkSpaces haben. Insbesondere für mobile User, bei denen eine Zugriffsbeschränkung über die IP Adressen mittels Access-Listen oder Security Group nicht möglich ist, ist das eine gute Alternative.

Voraussetzung: Directory Service

Wer Workspaces betreiben möchte, muss die späteren User der virtuellen Desktops in einem Active Directory speichern und dort deren Zugriffsrechte verwalten. Hat man für AWS noch keine AD-Funktionalitäten konfiguriert, muss man das zwingend tun, bevor man mit den WorkSpaces beginnen kann. Dafür stehen einem mehrere Varianten zur Verfügung:

- „Microsoft AD“
  Damit meint Amazon ein bei AWS gehostetes AD.
- „AD Connector“
  Wer ein eigenes, on-premise AD betreibt, kann dieses mit der AWS Cloud verknüpfen und so seine User im eigenen RZ pflegen.
- „Cross trust“
  In diesem Fall betreibt man zwei Microsoft AD, eines on-premise, eines in der Cloud. Zwischen beiden kann dann ein Vertrauensverhältnis etabliert werden, so dass man letztlich wieder die lokalen User für die WorkSpaces nutzen kann.
- „Simple AD“
  Hinter „Simple AD“ verbirgt sich ein Microsoft kompatibles AD, das jedoch auf Samba 4 gehostet wird. Dieses läuft dann ebenso wie das Microsoft AD in der AWS Cloud.

  Vorteil des Simple AD ist, dass es kostengünstiger ist, der Nachteil ist, dass es zur Zeit in Frankfurt nicht zur Verfügung steht.

Wer WorkSpaces zunächst einmal ausprobieren möchte, ohne großen Aufwand in das AD zu stecken, sollte die Simple AD Variante nutzen. Innerhalb der EU kann man das Stand heute allerdings nur in der Region Irland. Die Performance dürfte auch für den Realbetrieb in den meisten Fällen völlig ok sein. Unsere eigene Erfahrung ist, dass das für Tests allemal ausreicht.

Legt man seinen ersten WorkSpace an, so hat man die Option Quick und Advanced. Wie schon in einem anderen Artikel zum Thema Netzwerkdesign in der Cloud erwähnt, sollte man keinesfalls auf den Gedanken kommen, die Quick Variante zu nutzen, die ist nämlich wirklich „dirty“. Egal ob man schon eigene Subnetze, VPCs oder ein AD eingerichtet hat, bei Quick wird all das nochmal angelegt. Damit braucht man nachher viel Zeit, um all das wieder los zu werden, außerdem zahlt man für den Nutzungszeitraum doppelt: für das Test AD und für das bereits in Betrieb genommene.

Wer also noch kein AD oder AD Verküpfung für AWS angelegt hat, sollte damit beginnen, ein Simple-AD anzulegen. Um ein AD erfolgreich in Betrieb zu nehmen, benötigt man mindestens zwei Subnetze jeweils in einer anderen Availability Zone. Auch diese Subnetze sollte man, falls nicht bereits vorhanden, besser vorher bereits angelegt haben. Die User kann man später noch hinzufügen.

Anlegen der WorkSpaces

Ist alles vorbereitet, kann man seinen ersten WorkSpace anlegen: zunächst wählt man das AD aus, dem der virtuelle Client zugeordnet werden soll. Anschließend bestimmt man einen oder mehrere User als Nutzer des Clients. Dabei ist es sowohl möglich neue User anzulegen als auch bestehende aus dem AD-Verzeichnis auszuwählen. Danach wählt man das Betriebssystem des Clients und dessen virtuelle Hardwareumgebung aus. Zur Verfügung stehen Stand heute nur die Betriebssysteme Windows 7 und Windows 10. Windows 8 sucht man vergebens, ebenso wie Linux. Bei der virtuellen Hardwareumgebung werden verschiedene Klassen unterschieden:

- Standard
  2 virtuelle CPUs, 4 GB RAM, 80 GB Root-Partition, 50 GB User-Partition
- Value
  1 virtuelle CPUs, 2 GB RAM, 80 GB Root-Partition, 10 GB User-Partition
- Performance
  2 virtuelle CPUs, 7.5 GB RAM, 80 GB Root-Partition, 100 GB User-Partition
- Graphic
  8 virtuelle CPUs, 15 GB RAM, 1 GPU, 4 GB Videospeicher, 100 GB Root-Partition, 100 GB User-Partition
- Power
  4 virtuelle CPUs, 16 GB RAM, 175 GB Root-Partition, 100 GB User-Partition

ONL, ONIE und P4

Drei Bausteine zum Aufbau moderner Layer 3 Switche

Fortsetzung von Seite 1

Seit über 10 Jahren ist Markus Geller bei der ComConsult Research GmbH erster Ansprechpartner für die Themen VoIP und Lokale Netze. Der Schwerpunkt seiner Trainer Tätigkeit liegt dabei auf den Gebieten SIP, PSTN Migration, WebRTC sowie Layer 2 und 3 Techniken für MAN und LAN. Markus Geller verfügt über eine langjährige Erfahrung beim Aufbau und der Planung von Netzwerken im large Enterprise Umfeld, inkl. RZ-Netzwerken, WLAN und Multicastverfahren. In seiner über 20-jährigen IT-Laufbahn beschäftigt er sich mit der Evaluierung neuer Technologien und deren Einsatz in der Praxis. Zudem ist er als Autor diverser Fachartikel für den ComConsult Netzwerk Insider und das Wissensportal tätig.

Diese Art der Infrastrukturbereitstellung hat sowohl Vor- als auch Nachteile. Betrachten wir dazu einen Gegenentwurf wie ihn Apple verfolgt. Durch die Kopplung der Hardware an das Betriebssystem - in diesem Fall MacOS - kann durchaus eine optimierte Lösung zur Verfügung gestellt werden, aber zu dem Preis, dass aufgrund der geringeren Marktverbreitung nicht jede Anwendung zur Verfügung steht. Dies bedeutet für den Nutzer: Er hat die Wahl zwischen einer flexiblen Lösung, die weniger Hardware optimiert ist oder einem hardwareoptimierten System, welches im Zweifelsfall weniger flexibel ist.

Dies bringt uns nun zu dem eigentlichen Anliegen dieses Artikels. Die eingangs geschilderte Situation der freien Wahl des Betriebssystems galt eben bisher nur für Server und PC Systeme. Im Netzwerkumfeld wurden solche Diskussionen jedoch nie geführt - bis heute. Denn seit ein paar Jahren gibt es Softwareunternehmen wie BigSwitch oder Cumulus, die sich auf die Entwicklung eines Netzwerkbetriebssystems für Switche und Router spezialisiert haben. Die daraus resultierenden Lösungen geben einem jetzt die Möglichkeit, auch Router und Switche mit alternativen, flexiblen Betriebssystemen auszurüsten. Voraussetzung ist jedoch, dass die bekannten und etablierten Enterprise Netzwerkausrüster die Möglichkeit schaffen, solche Alternativen auch nutzen zu dürfen, was Stand heute eher die Ausnahme als die Regel ist.

Neben diesen kommerziellen Switch-OS Alternativen hat sich seit 2014 noch eine weitere Variante am Markt manifestiert: das ONL.

ONL steht für Open Network Linux und stellt eine Quellcode-offene Möglichkeit dar, einen Switch mit einem Betriebssystem zu versorgen. Die zugrunde liegende Linux Distribution basiert auf Debian 7 (Wheezy), allerdings wurden gravierende Änderungen vorgenommen, da eine grundlegende Anpassung an die verwendete Switch-Hardware notwendig wurde. Als Beispiel sei hier nur die Verwendung von Switch ASICs und SFP's genannt, die so in klassischen PC Systemen nicht zum Einsatz kommen.

Grundvoraussetzung, dass solche OS-Varianten - egal ob quelloffen oder kommerziell - genutzt werden können, ist natürlich die Verfügbarkeit von Hardware Produkten. Diese „Boxen", gerne auch „White Label Switch" genannt, basieren auf „open standard, bare-metal hardware". Dieser Markt, der laut Gartner im Jahr 2020 rund 22% der Data Center Infrastrukturen ausmacht, war bisher ein geschlossener Kosmos mit Herstellern wie Accton/EdgeCore, Alpha Networks, Penguin oder Quanta. Allerdings gibt es auch in diesem Feld zunehmend Bewegung, indem bekannte Enterprise Akteure wie Dell (ehemals Force10), HPE, Mellanox oder Arista vorstoßen.

Die eigentlich spannende Frage ist jedoch: Warum sollte ich zum Beispiel meinen Cisco Switch von seinem CatOS, IOS oder NX-OS befreien? Der Grund hierfür liegt in der Forderung nach Flexibilisierung.

Das erste, was den meisten Netzwerkern beim Begriff Flexibilisierung einfällt, ist SDN: Software Defined Networking, also die dynamische Steuerung und Erkennung von Datenströmen bzw. Anwendungen, in Abhängigkeit von ihrer Priorität und der bereitgestellten Netzinfrastruktur (Bandbreite, Delay, Paketloss, usw.)

Diese Sichtweise ist jedoch zu kurz gegriffen. Durch die Nutzung eines Linux Kernels als Grundlage einer Software Architektur sind flexibel steuerbare Anwendungen oder Dienste auf einem Netzwerkknoten keine Utopie mehr. Ein Beispiel für eine solche Vorgehensweise wäre die Implementierung einer Firewall oder eines Session Border Controllers auf der vorhandenen Switch-Hardware, ohne Änderungen an der physikalischen Struktur vornehmen zu müssen.

Jetzt wir der ein oder andere Leser einwenden, dass dies ja heute schon mittels NFV (Network Function Virtualization) möglich ist, indem man Produkte wie VMware, Microsoft Hyper-V oder KVM hierfür heranzieht. Dabei verkennen sie aber die Situation, dass diese Lösungen zum einen nur im Rechenzentrum zum Einsatz kommen und eben immer den Server als Host, nutzen um Netzwerkdienste anbieten zu können. Der Ansatz mit ONL als Basis zielt aber auf ein anderes Einsatz-

ONL, ONIE und P4 - Drei Bausteine zum Aufbau moderner Layer 3 Switche

feld, dort, wo kein Server zur Verfügung steht, um Netzwerkservices zu erbringen. Das trifft in erster Linie natürlich die großen Netzprovider oder aber auch die Anbieter von (Hyperscale) Rechenzentren, die eine Vielzahl von Netzwerkknoten betreiben, ohne dabei auf die klassischen Virtualisierungsprodukte aufzusetzen.

Hier sollen mittels des Linuxkernels Services zur Verfügung gestellt werden, ohne Änderungen an der Netzwerkhardware vornehmen zu müssen. Im Grunde ist der dahinterstehende Gedanke, Netzwerkfunktionen wieder aus der Overlay-Welt der Virtualisierung zurück in die Netzwerkwelt zu holen, ohne dabei die Switch Infrastruktur durch Serverhosts zu ergänzen.

Die Entwicklung von ONL ist dabei in einem größeren Kontext zu betrachten. Im Jahr 2011 haben sich führende Anbieter der IT Industrie zusammengeschlossen, um das Open Compute Project (OCP) zu starten. Mitglieder sind u.a.: Facebook, Intel, Nokia, Google, Apple, Microsoft, Seagate Technology, Dell, Rackspace, Ericsson, Cisco, Juniper Networks, Goldman Sachs, Fidelity, Lenovo und die Bank of America.

Ziel des Projektes ist es, ein effizientes Design zu entwickeln, welches wenn immer möglich auf standardisierten, offenen Schnittstellen bezüglich der verwendeten Soft- und Hardware Architekturen beruht. Das Konzept orientiert sich dabei an Lösungen für das Rechenzentrum, also Storage, Server und Netzwerk. Dies beinhaltet z.B. Lösungen, die sich CPU-seitig an Intel/AMD oder ARM Infrastrukturen orientieren oder eben auf der Softwareseite Linux als Betriebssystem Basis einsetzen. Als Beispiel sollen hier die Datacenter Infrastrukturen von Facebook dienen, die zu 100% OCP konform sind.

Nun betrachten wir in diesem Artikel speziell die Auswirkungen auf unsere Datennetze. Um also den Blick noch einmal auf ONL zu werfen, hier ein paar grundsätzliche Funktionen bzw. Module, die bei ONL zum Einsatz kommen.

Beginnen wir mit dem generischen Aufbau eines ONL basierten Switchsystems. (siehe Abbildung 1)

Wie man unschwer erkennen kann, orientiert sich der Aufbau an einer ganz allgemeinen Struktur, wie sie auch in der Serverwelt genutzt wird. Erst im Detail erkennt man dann, welche speziellen Funktionen gefordert werden. (siehe Abbildung 2)

Die besonderen Erweiterungen sind zum einen spezielle Hardware Treiber und

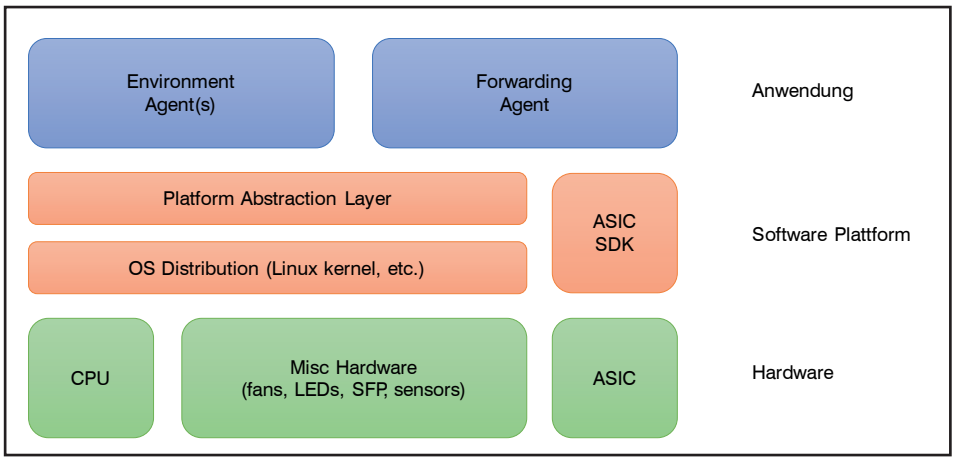


Abbildung 1: Switch Architektur 1 Quelle: Rob Sherwood

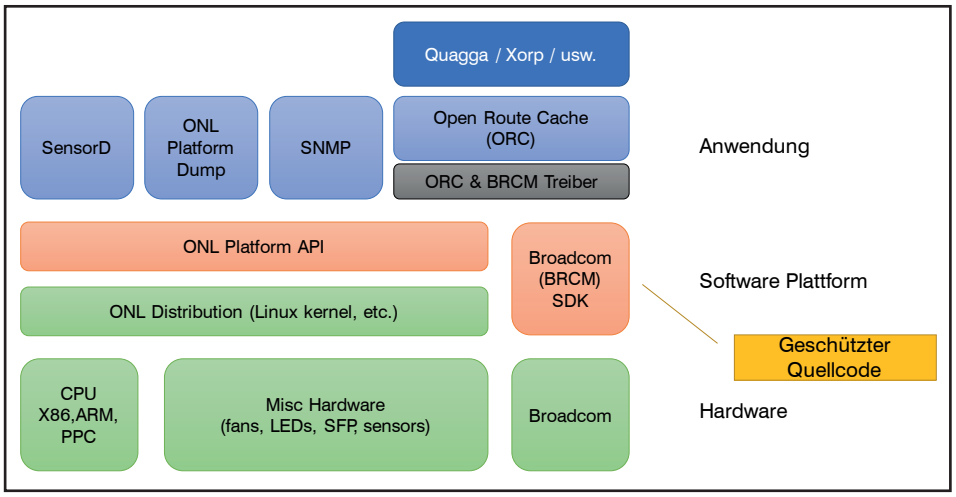


Abbildung 2: Switch Architektur 2 Quelle: Rob Sherwood

SDKs, die auf die Switch ASICs, z.B. von Broadcom, abgestimmt sind, welche im Allgemeinen nicht der Offenheit des Quellcodes unterliegen. Diese bilden neben der ONL Distribution und den damit verbundenen Plattform APIs die Softwarebasis unseres Switches. Darauf aufbauend schließen sich die gewünschten Anwendungen an. Diese können - wie der Sensor Daemon - Bestandteile der Distribution sein oder aber auch durch zusätzliche Softwarepakete wie Quagga oder Xorp (eXtensible Open Router Platform) abgebildet werden, welche die eigentliche Layer 3 Funktionalitäten wie IPv4/v6, OSPF, IS-IS, BGPv4 oder auch RIP bereitstellen. Optional wäre natürlich auch vorstellbar, dass bei ausreichender CPU- und Speicherausstattung des Switches mittels Virtualisierung Dienste auf diesen Switchsystemen temporär gestartet werden können, die z.B. bei Serverausfällen deren Funktion übernehmen können.

Der entscheidende Punkt ist jedoch, dass durch die Verwendung eines Linux Betriebssystems auf dem Netzwerkknoten dieser sich administrativ genauso behandeln lässt wie ein Server. Dies bedeutet: alle Tools, die ich zur Verwaltung meiner Linux Systeme nutze, kann ich jetzt auch für meine Netzinfrastruktur einsetzen. Die bisherige Vorgehensweise über CLI Scripte, die auf die Befehlsstruktur des verwendeten Switch-Systems beruhen, oder die Nutzung spezieller APIs, die vom Hersteller zur Verfügung gestellt werden, entfällt. Gerade in heterogenen Umgebungen erleichtert dieser Umstand die Administration der Infrastruktur.

Auf Seiten des Netzwerkadministrators erfordert dies natürlich ein Umdenken. Er muss sich nun mit neuen Techniken einer zentralisierten, SDN basierten Konfiguration vertraut machen und vielleicht eine neue Scriptsprache wie Ruby, Pearl oder P4 lernen, um seiner Arbeit nachkommen zu können. Aber ist dies ein Nachteil? Sicher nicht. Wer heute schon Produkte verschiedener Hersteller oder Produktlinien im Einsatz hat, muss sich auch jetzt schon auf die unterschiedlichen CLI Befehle der einzelnen Systeme einstellen und muss