

KVM on Z vs. ESXi

Dane Wicki

4. Oktober 2017

Zusammenfassung

Virtualisierung hat seinen Ursprung schon in den Frühen 1960er Jahren, als die Computer systeme noch gross und teuer zu warten waren. Der Ursprung findet sich bei den Maneframes, eine platform, welche auch heute noch in gebrauch ist, obschon sie totgeglaubt wurde. IBM entwickelte die Virtualisierung, damit die users die Computer resourcen untereinander teilen konnten, und diese Trotzdem noch parallel gebraucht werden konnten. Heutzutage erfreut sich die Virtualisierung eines enormen Hypes, welcher zu vielen verschiedenen Softwarelösungen geführt hat.

Die Virtualisierung ermöglicht das ausführen mehreren Virtuellen machinen, welche auf einem einzigen host laufen. Diese Arbeit soll nun einen Vergleich zweier moderner Virtualisierungslösungen aufzeigen. KVM on Z ist die Erste lösung, auf welche eingegangen werden wird. Es ist eine Opensource lösung, welche in den Linux Kernel integriert wurde. ESXi, die 2. Lösung, ist eine Weitere möglichkeit um zu Virtualisieren. Diese Lösung wird von VMWare vertrieben und ist nicht ganz so Quelloffen wie seine zu gegenüberstellende lösung.

Inhaltsverzeichnis

| | | |
|----------|--|-----------|
| 1 | Einleitung | 2 |
| 2 | Grundlagen | 3 |
| 2.1 | Grundlagen der Virtualisierung | 3 |
| 2.1.1 | Hypervisor | 5 |
| 2.2 | KVM | 6 |
| 2.2.1 | Funktionsumfang | 7 |
| 2.2.2 | KVM on System Z | 8 |
| 2.3 | ESXi | 8 |
| 2.3.1 | Funktionsumfang | 8 |
| 3 | Vergleich | 9 |
| 4 | Fazit | 10 |
| 5 | Anhang | 11 |
| 6 | Wayne | 12 |
| 7 | Was ist Virtualisierung | 13 |
| 7.1 | Grundlagen der Virtualisierung | 14 |
| 8 | KVM | 15 |

Kapitel 1

Einleitung

TODO

Kapitel 2

Grundlagen

2.1 Grundlagen der Virtualisierung

Virtuell ist laut Duden etwas, was nicht echt, nicht in der Wirklichkeit vorhanden, also nur scheinbar vorhanden ist. Wenn wir nun von einer Virtualisierung sprechen im Bereich der IT hat dies viel mit dieser Definition zu tun. Wenn wir nun eine Betriebssysteminstanz auf einem nicht vorhandenen, also virtuellen Rechner installieren, nennen wir das Virtualisierung. Aber was meine ich nun mit dem virtuellen Rechner und wie kann ich darauf etwas laufen lassen wenn er doch nicht existiert?

Nehmen wir doch mal an ich habe einen nicht virtuellen Rechner, also einen wirklich vorhandenen Rechner vor mir. Nun installiere ich das Betriebssystem auf eben jenem Rechner, so habe ich einen uns gewonnenen Rechner. Nun könnte ich jedoch auf diesem Rechner eine Software laufen lassen, die einen weiteren Rechner herstellt, also einen, welcher nicht wirklich vorhanden ist, nur scheinbar vorhanden ist, so ist dieser Rechner virtuell. Und wie auf einem vorhandenen Rechner, kann ich auf diesem virtuellen Rechner eine Betriebssysteminstanz installieren, so

habe ich einen Virtuellen rechner.

Also versteht man unter der Informatik unter dem begriff Virtualisierung unter anderem die Technologie, bei welcher das Betriebssystem nicht unmittelbar auf einer physischen Instanz, einem physisch vorhanden rechner installiert ist, sondern auf der Hardware über einer Zwischenschicht. Diese Zwischenschicht abstrahiert die reale physische hardware, ist also nicht wirklich vorhanden. Die auf dieser Zwischenschicht laufende Betriebssysteminstanz wird als Gastssystem bezeichnet.

Auf der realen physikalischen Hardware muss jedoch auch eine software laufen, welche diese virtuellen systeme erstellt, diese Software werden als Hypervisor bezeichnet. Der Hypervisor dient in erster linie der verwaltung und Zuteilung der Ressourcen für die einzelnen Gast-systeme. Dabei soll der Hypervisor die Ressourcen so verteilen, dass das Gastssystem die Ressourcen zur verfügung gestellt bekommt, wenn es diese auch benötigt.

Dank der Virtualisierung lassen sich also mehrere Gastssysteme auf einem einzelnen Hypervisor zum laufen zu bringen. Diese Gastssystem sind zudem nicht direkt abhängig von der bestehenden hardware, so dass es auch möglich ist, das Selbe gastsystem auf mehreren verschiedenen physikalischen rechner zum laufen zu bringen sofern der Hypervisor auf dieser neuen physikalischen instanz zum laufen gebracht werden kann. Dies bringt den Enormen vorteil, dass ein virtualisiertes betriebssystem auf einfache art und weise auf einen neuen rechner gezogen werden kann. Weiter bietet die möglichkeit, dass mehrere Gastssysteme auf einer physikalischen instanz laufen können den vorteil, dass die Ressourcen besser ausgenutzt werden können. Dies wird vor allem bei Servern und Mainframes geschätzt, da die Hardware in diesen Bereichen teuer ist und man diese, wenn man schon viel geld

bezahlt auch ausnutzen möchte.

2.1.1 Hypervisor

Hypervisoren werden auch Virtual-Machine-Monitor (kurz. VMM) genannt. Diese Hypervisoren sind eine Klasse von Systemen, die als eine abstrahierende Schicht zwischen tatsächlicher vorhandener Hardware und weiteren zu installierenden Betriebssysteme dienen.

Sie dienen der Verwaltung der Ressourcen für die einzelnen Gastssysteme. Im Bereich der Hypervisoren gibt es ein Klassifizierungssystem, bei welchem diese in 2. Typen unterteilt werden.

Type 1

Ein Typ-1-Hypervisor (bare-metal oder auch native genannt) setzt direkt auf der Hardware auf. Es benötigt kein Betriebssystem, auf welchem dieses läuft. Es wird jedoch vorausgesetzt, dass die Hardware des Hostsystems vom Typ-1-Hypervisor durch entsprechende Treiber unterstützt wird.

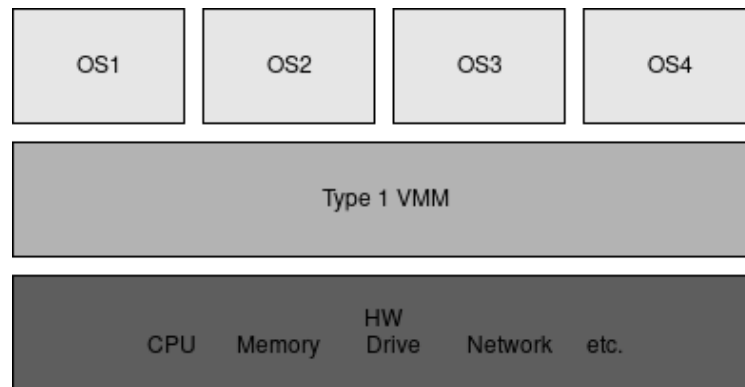


Abbildung 2.1: Aufruf des FilterVIs

Type 2

Ein Typ-2-Hypervisor setzt auf einem vollwertigen Betriebssystem, welches auf dem Hostsystem installiert ist, auf und nutzt die Treiber, des Betriebssystems, um auf die Hardwareressourcen des Hostsystemes zuzugreifen zu können.

Dementsprechend sind alle Typ-2-Hypervisoren auf einem Hostsystem lauffähig, sofern sich auf diesem das unterstützte Hostbetriebssystem installieren lässt.

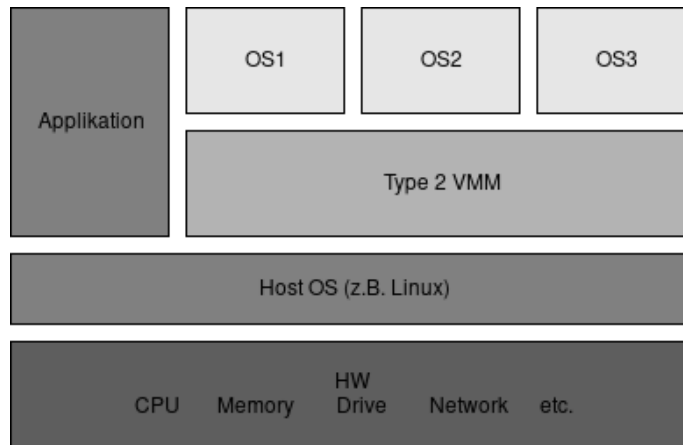


Abbildung 2.2: Aufruf des asd

2.2 KVM

KVM steht für Kernel Based Virtualmachine. bei KVM handelt es sich um eine Infrastruktur des Linux-Kernels zur Virtualisierung. Die Virtualisierung auf verschiedenen Plattformen unterstützt, unter anderem die folgenden Intel, AMD, ARM, PowerPC und System Z. KVM selbst nimmt keine Emulation vor, es ist also nicht möglich eine Bestimmte hardware zu emulieren mit KVM selbst. so würde es sich als enorm Schwer erweisen ein vollwertiges Betriebssystem auf dem Hostsystem nur mit hilfe von KVM zu installieren. Um die Emulation

zu ermöglichen hat sich QEMU als defacto standard etabliert. QEMU steht für Quick Emulator und stellt für virtualisierte Gastsysteme die notwendigen Geräte wie Festplatten, Netzwerk-, Sound- und Grafikkarten zur Verfügung.

KVM ist desweiteren auch eine Quelloffene Lösung, und ist neben XEN der weitverbreitetste Quelloffene Hypervisor. Trotz einigen Aussagen ist KVM jedoch ein Type 2 Hypervisor und nicht wie XEN ein Type 1.

2.2.1 Funktionsumfang

KVM ist bereits bei der Installation eine sehr schlanke und einfache Umgebung. Es sind im Wesentlichen nur die KVM Kernel Module dem bestehenden System dazuzustallieren sowie Qemu und Management-Tools einzurichten. Der Vorteil liegt hier auch, dass viele schon bestehende Linux-Server nachträglich zu einem Virtualisierungssystem aufgerüstet werden kann ohne damit eine komplette Neuinstallation tätigen zu müssen. Weiter verhält sich jede Virtuelle CPU (kurz VCPU) im Gastsystem wie ein gewöhnlicher Linux-Process und kann so beispielsweise auch über normale Kommandos wie kill oder top kontrolliert und gesteuert werden. Das selbe gilt auch für die Gerätelandschaft. Da mit KVM und Qemu die normalen Linux-Treiber genutzt werden können, ist eine Umgewöhnung für einen Linux-Administrator nicht nötig.

KVM in Zusammenarbeit mit Qemu unterstützt eine enorme Anzahl an Gast-Betriebssysteme, darunter finden sich fast sämtliche wichtigen Windows-Varianten sowie Linux, Solaris, BSD, FreeBSD und einige exotischere wie ReactOS.

Ein Nachteil der KVM ist jedoch, dass sie nicht auf Graphische Systeme ausgerichtet ist, es ist also die Idee, dass ein Gastsystem, welches auf einer KVM Instanz läuft, nur über ein Terminal bedient wird.

Beim Management zeigt sich, wie stark sich die Marktposition dieses Quelloffenen Open-Source-Hypervisors verbessert hat. Es ist inzwischen ein Fülle von Administrationswerkzeugen verfügbar, zudem wird KVM in vielen Cloud-Plattformen eingebaut. Durch die simplen tools virt-manager sowie virsh ist schon eine Remote-Management möglich. Für vortgeschrittene Funktionen wie Orchestrierung ganzer Pools von Virtuellen Maschinen mit weitergehenden funktionen wie Failover, High Availability und dergleichen gibt es weitere Lösungen. Hier springen Drittobjekte sowie Softwarehersteller in dir Bresche, normalerweise auch Open Source.

Es gibt zudem für auf KVM aufbauende Komplettlösungen für die Servervirtualisierung. allen voran RHEV (Red Hat Virtualization).

Für die Verwaltung der Gastsysteme gibt im Wesentlichen viele verschiedene Tools, von einfachen verwaltungssoftware für den Desktop PC wie zum Beispiel Gnome-Boxe

2.2.2 KVM on System Z

| | | |
|-------|-------|-------|
| Test1 | Test2 | Test3 |
| TestA | TestB | TestB |

<http://www.redbooks.ibm.com/redbooks/pdfs/sg248332.pdf>

2.3 ESXi

2.3.1 Funktionsumfang

Kapitel 3

Vergleich

Kapitel 4

Fazit

Kapitel 5

Anhang

Kapitel 6

Wayne

Ein vergleich im eigentlichen sinne stellt sich hier als enorm Schwierig dar. Da Zwei unterschiedliche einsatzgebiete miteinander verglichen werden sollen. Die Systeme, welche verglichen werden sollen, werden gemäs der Aufgabenstellung nicht einmal auf der Selben hardware verglichen sondern auf 2 für verschiedene Anwendungen gedachte lösungen. Der Auftrag an sich ist als ob ich eine Luxuskreuzfahrtschiff mit einem Transportschiff verglichen werden soll. So gibt es schon Gemeinsamkeiten, jedoch kann kein wahrer, endgültiger und absoluter vergleich aufgestellt werden. Es kann auch kein wahrer gewinner aufgezeigt werden, da beide Lösungen in sich gut sind. Es soll mehr für den jeweiligen einsatzzweck sowie den gewünschten Funktionen einen individuellen entscheid getroffen werden. Ich möchte hiermit trotzdem darlegen, welches die ungefähren unterschiede der beiden lösungen darstellt, damit eine Entscheidungsfindung unter umsten vereinfacht werden kann.

++ Wie man den vorangegangenen Kapitel nun entnehmne kann, sind beide Lösungsansätze eine gute wahl.¹

¹Timo Hirt. *KVM - The kernel-based virtual machine*. Feb. 2010. URL: <https://www.cs.hs-rm.de/~linn/fachsem0910/hirt/KVM.pdf>.

Kapitel 7

Was ist Virtualisierung

Unter der Virtualisierung versteht man in der Informatik unter anderem die Technologie, bei welcher das Betriebssystem nicht unmittelbar auf einer physischen Instanz wie einem Server, oder Desktop PCs installiert ist, sondern auf der Hardware über einer Zwischenschicht, welche die reale physische Hardware abstrahiert, oder aber in einem Gastgeber-Betriebssystem mittels einer Virtualisierungssoftware wie VirtualBox. Dieses auf der virtuellen Umgebung laufende Betriebssystem nennt man Gastsysteme. Die tatsächliche vorhandene Hardware-Umgebung wird als Hostsystem bezeichnet. Die Softwareschicht, welche diese Gastsysteme verwaltet, nennt man Hypervisor (Klassifikationen siehe Kapitel XYZ). Dieser Hypervisor verwaltet die Ressourcen und deren Zuteilung für die einzelnen Gastsysteme. Dabei verteilt der Hypervisor die Ressourcen so, dass alle Ressourcen für das Gastsystem verfügbar sind, wenn dieses es benötigt, so als ob nur jenes Gastsystem vorhanden wäre. Dem Gastsystem wird dank des Hypervisors eine eigener vollständiger Rechner vorgetäuscht, mit allen Hardware-Elementen, welche dazugehören. Das Abstrahieren des Hypervisors und Vortäuschen einer realen Hardware führt dazu, dass das Gastsystem nicht nur auf der realen Hardware lauffähig ist, in der es installiert wurde, sondern auf allen, in denen der Hypervisor installiert bzw. lauffähig

ist.

++Kurze geschichte zur Virtualisierung Die ersten Hypervisoren, die Virtualisierung ermöglichten,

Unter Virtualisierung versteht man unter anderem eine Technologie, bei der eine Betriebssysteminstanz nicht unmittelbar auf einem Physischen Rechner, z.B Server, installiert ist, sondern auf der Hardware über einer Zwischenschicht, die von der Hardware abstrahiert, oder aber in einem Gastgeber-Betriebssystem mittels Virtualisierungssoftware, z.B. VMWare oder Virtual Box.

In der Informatik bezeichnet Virtualisierung das Nachbilden eines Hardware- Softwareobjektes, welches vom selben typ wie die tatsächliche darunterliegende Hardware/Software ist. Dieses Nachbilden geschieht mit hilfe eines Abstraktionslayers.

7.1 Grundlagen der Virtualisierung

Die Geschichte der Virtualisierung liegt schon eine lange Zeit zurück. Bereits in den frühen 1960er Jahre hat IBM

++Hypervisor klassifikationen. Ein Hypervisor lässt sich typischerweise in einen von 2 Typen unterteilen. +Typ1 Native or bare metal Native Hypervisoren sind softwaresysteme, welche direkt auf der Hardware des Hostsystemes laufen um die Hardware zu kontrollieren und die Gastsysteme zu überwachen. .

Diesens auf der abstrahierten, virtuellen umgebung laufende Betriebssystem nennt man Gastssystem.

Kapitel 8

KVM

Die Kernel-based Virtual Machine ist eine Virtualisierungslösung welche in den Linuxkernel integriert wurde.

Die Kernel-based Virtual Machine ist eine Infrastruktur des Linux-Kernel zur Virtualisierung

<https://www.cs.hs-rm.de/~linn/fachsem0910/hirt/KVM.pdf> <https://www.tecchannel.de/a/kostenlos-virtualisierungssoftware-im-vergleich,2051909,4> <https://www.tecchannel.de/a/kostenlos-virtualisierungssoftware-im-vergleich,2051909,5>

Literatur

- [1] Timo Hirt. *KVM - The kernel-based virtual machine*. Feb. 2010.
URL: <https://www.cs.hs-rm.de/~linn/fachsem0910/hirt/KVM.pdf>.