Evaluation verschiedener Container-Technologien

Corvin Schapöhler*, Kai Warendorf, Dominik Schoop

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2018

Der Trend zur Cloud ist unumkehrbar. Bereits heute setzen Firmen wie Microsoft, Goole und Amazon verstärkt auf das Cloud-geschäft [1]. Durch die einfache Portierbarkeit sind Container einer der treibende Technologie hinter diesem Trend. Im folgenden wird diese Technologie genauer betrachtet. Dabei soll die Frage beantwortet werden, wie Docker die populärste Technologie wurde, welche aktuellen Probleme bestehen und wie versucht wird, diese zu lösen.

Container und virtuelle Maschinen

Container dienen der Isolation von Prozessen. Dabei sind sie ressourcensparender und deutlich schneller als Virtuelle Machinen. Wie in Abbildung 1 zu sehen geschieht dies, indem Container nur einzelne Systemressourcen isolieren statt ein gesamten Betriebssystem mit Kernel zu virtualisieren.

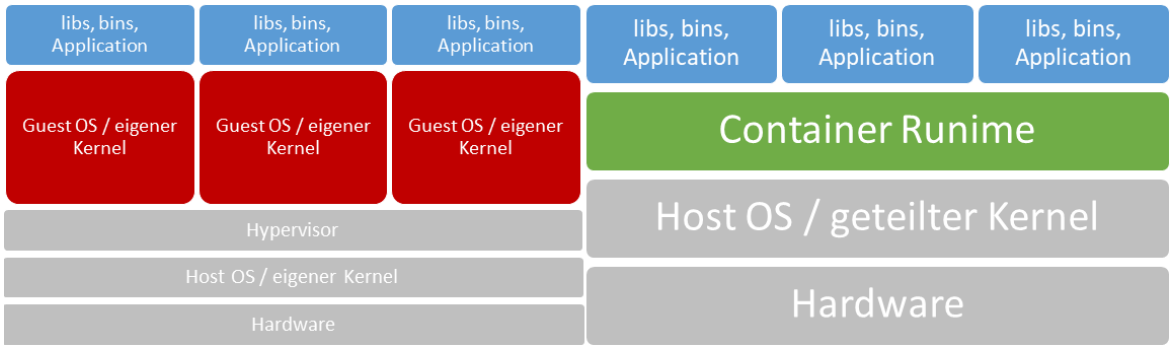

Abbildung 1: Virtualisierung durch VMs (links) im vergleich zu Isolation durch Container (recht)

Durch diese Isolation sind Container deutlich skalierbarer als Virtuelle Maschinen. Container benötigen nur Bruchteile von Sekunden um gestartet oder zerstört zu werden und können somit bei benötigter Leistung zugeschaltet werden. Zudem sind Container von Natur aus unabänderlich. Dies sorgt dafür, dass Container keinen Zustand speichern und somit neue Container die Plätze alter ersetzen können.

Historische Entwicklung

Die grundlegenden Konzepte zur Isolation wurde bereits in den späten 70er Jahren mit der Einführung des Unix Systemaufrufs chroot implementiert. Der erste große Durchbruch kam allerdings erst 2008 mit LXC [2]. Dieses implementiert eine vollständige Isolation des Linux-Kernels, ohne dabei von Kernelpatches abhängig zu sein.

*Diese Arbeit wurde durchgeführt bei der Firma NovaTec GmbH, Leinfelden-Echterdingen

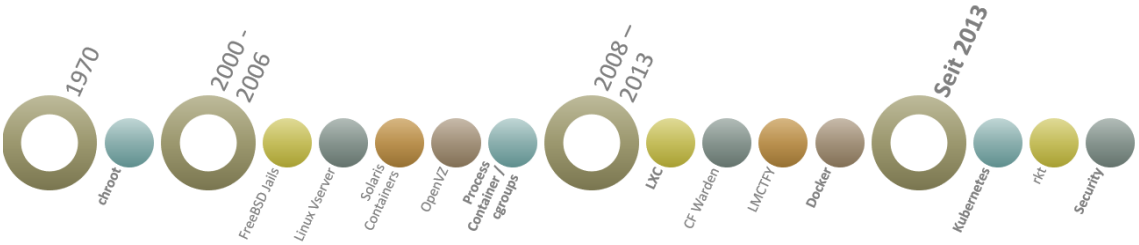

Abbildung 2: Veröffentlichung verschiedener Container-Technologien von 1970 bis heute

In den folgenden Jahren wurden die unterliegenden Konzepte erweitert und vereinfacht. Große Popularität gewann die Technologie allerdings erst 2013 mit dem Release der Container-Plattform Docker. Diese setze zu Beginn auf LXC auf, wechselte bald aber zu einer eigenen Implementierung. Zudem bietet Docker mit dem Docker Hub eine SaaS-Plattform, die die Widerverwendung von Containern ermöglicht. Nach dem Release von Docker wurden neue Konzepte wie rkt oder LXD veröffentlicht (siehe Abbildung 2)

Alternativen zu Docker

Neben Docker hat sich ein großer Markt an Alternativen Container-Runtimes gebildet. So bietet CoreOS mit rkt eine Alternative, die auf Sicherheit und Widerverwendbarkeit einzelner Container setzt. Zudem setzt rkt darauf, eine der ersten Container-Runtimes zu sein, die den AppContainer Standard implementiert. Eine weitere Alternative bietet Canonical mit LXD, einer Weiterführung der Container-Runtime LXC. LXD setzt, recht ähnlich zu ersten Versionen von Docker auf eine RESTful API zur Steuerung von LXC. Im Gegensatz zu rkt oder Docker ist LXD dafür gedacht, komplette Linux-Distributionen zu isolieren.

Durch das rapide Wachstum am Interesse für Container wurden Standards für diese gefordert. Neben AppContainer wurde 2015 unter der Obhut der Linux Foundation die Open Container Initiative (OCI) gegründet [3]. Diese definiert Standards für Container-Images und Runtimes. Zudem stellt die OCI eine beispielhafte Implementierung des Standards, runC, zur Verfügung. Mittlerweile nutzen Docker und viele weitere Tools die Spezifikation und bauen ihre Angebote auf runc um.

Aktuelle Probleme und Lösungen

Durch die enorme Popularität von Container und die ansteigende Nutzung werden auch viele Probleme mit dieser Art der Isolation erkenntlich. Bereits 2014 startete Google alle Dienste in Containern und musste somit jede Woche zwei Milliarden Container verwalten [4]. Zudem werden zunehmend Sicherheitslücken im Linux-Kernel bekannt, die es ermöglichen aus der Isolation auszubrechen. Um diesen Problemen entgegenzuwirken werden Orchestrierungstools wie Kubernetes und Container-Runtimes wie gVisor entwickelt, die mehr Sicherheit und vereinfachte Verwaltung versprechen.

[1] Sataya Nadella. *Annual Report 2017*. Financial Report. Microsoft, 2017. url: https://www.microsoft.com/investor/reports/ar17/index.html (besucht am 09.05.2018).
[2] Rani Osnat. *A Brief History of Containers: From the 1970s to 2017*. Blog. Mar. 21, 2018. url: https://blog.aquasec.com/a- brief- history- of- containersfrom-1970s-chroot-to-docker-2016 (besucht am 09.05.2018).
[3] Open Container Initiative. *Open Container Initiative*. The Linux Foundation. 2018. url: https://www.opencontainers.org/ (besucht am 09.05.2018)
[4] Joe Beda. *Containers At Scale. At Google, the Google Cloud Platform and Be yond*. In: GlueCon 2014. May 22, 2014. url: https://speakerdeck.com/jbeda/containers-at-scale (besucht am 09.05.2018).

Bildquellen:

- Abbildung 1: https://www.serverpronto.com/spu/wp-content/uploads/2016/05/MJHfm1c.jpg
- Abbildung 2: erstellt auf Basis von https://blog.aquasec.com/a-brief-history-of-containersfrom-1970s-chroot-to-docker-2016