

Virtualisierung in der Produktionstechnik

Marc Fischer, Christan Müller, Christopher Kolb

Durch den Einsatz von Virtualisierungstechnologien in Produktionsumgebungen kann Hardware konsolidiert werden, wodurch sich eine Vielzahl von Vorteilen ergibt. Das Prinzip der Virtualisierung wird in anderen Domänen schon seit vielen Jahren eingesetzt, allerdings stellt die Produktionstechnik zusätzliche Anforderungen, die berücksichtigt werden müssen.

Warum eigentlich Virtualisierung?

In vielen Unternehmen wird die Produktion stets erweitert, optimiert und verändert, wobei sich diese Veränderungen möglichst problemlos in die vorhandenen Gegebenheiten integrieren lassen müssen. Dabei kann es leicht zu einer heterogenen Ressourcenlandschaft kommen, da vorhandene Ressourcen und Systeme unangetastet bleiben und neue Ressourcen hinzukommen. Dies zeigt sich beispielsweise in großen Produktionsumgebungen wie denen des Automobilbaus. Hier werden z. B. in der Montage eine Vielzahl von verschiedenen PCs, Steuerungen und Spezialrechnern eingesetzt. Die hohe Anzahl und Verschiedenartigkeit an Hardware verursacht nicht nur bei der Anschaffung hohe Kosten, sondern durch Energie-

kosten auch im Betrieb und ist ein kostspieliger Mehraufwand bei der Wartung. Darüber hinaus wird die Rechenleistung der einzelnen Geräte nur geringfügig ausgenutzt. Dies führt dazu, dass mehr Geräte als nötig eingesetzt werden.

Die Reduktion der Hardwareanzahl wird in anderen Domänen wie der Office-IT und dem Rechenzentrums-/Serverbetrieb durch Virtualisierung gelöst. Dabei werden existierende Systeme von der vorhandenen Hardware durch eine Zwischenschicht, auch *Hypervisor* genannt, abstrahiert. Damit können vorhandene Systeme auf Systeme mit Hypervisor migriert werden, ohne dass sich für den Endbenutzer etwas ändert. Außerdem ist es möglich, mehrere Systeme parallel auf einer Hardware auszuführen. Mit der Virtualisierung ist folglich die Konsolidierung von Systemen auf eine geringere Anzahl von Hardware möglich.

Anforderungen der Produktionstechnik

Im Unterschied zu den Domänen der Office-IT und des Serverbetriebs gibt es in der Produktion Geräte mit besonderen Anforderungen an Software und Hardware. Systeme wie die

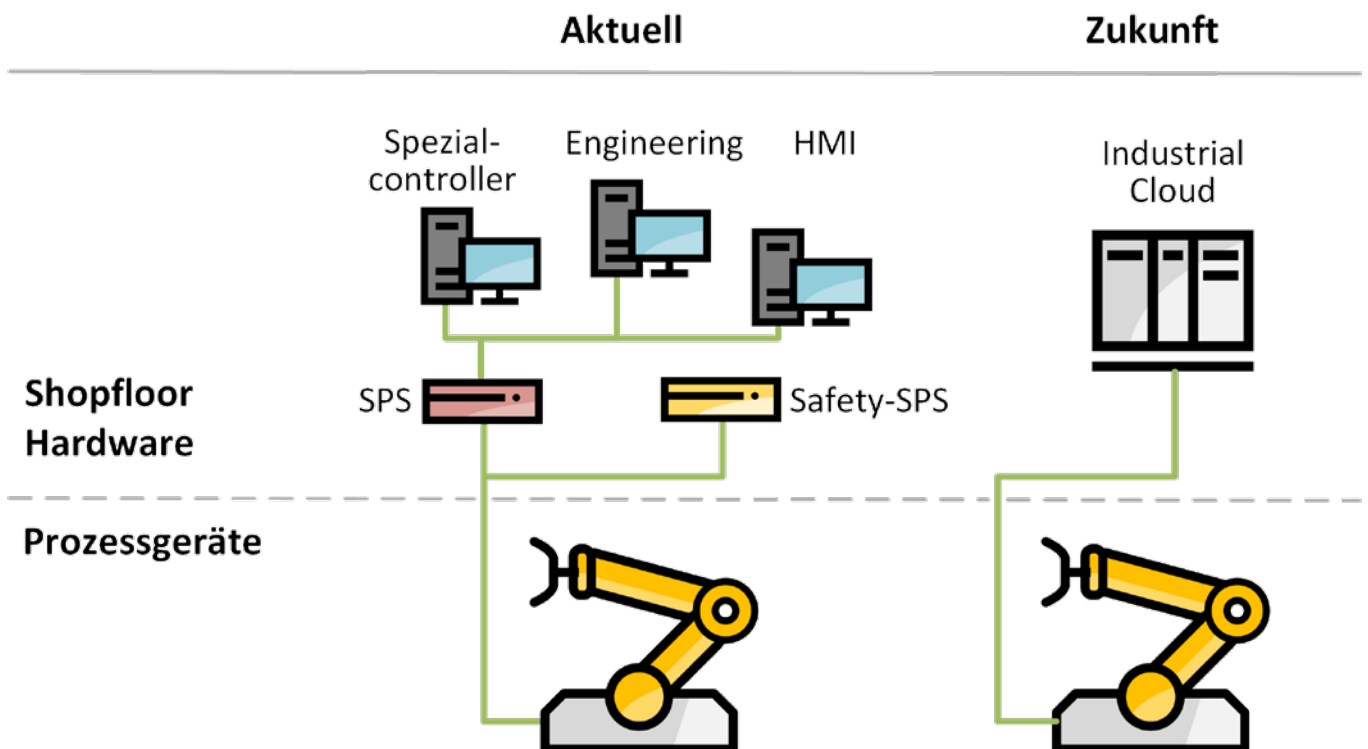


Abbildung 1: Konsolidierung von Hardware in der Produktion.

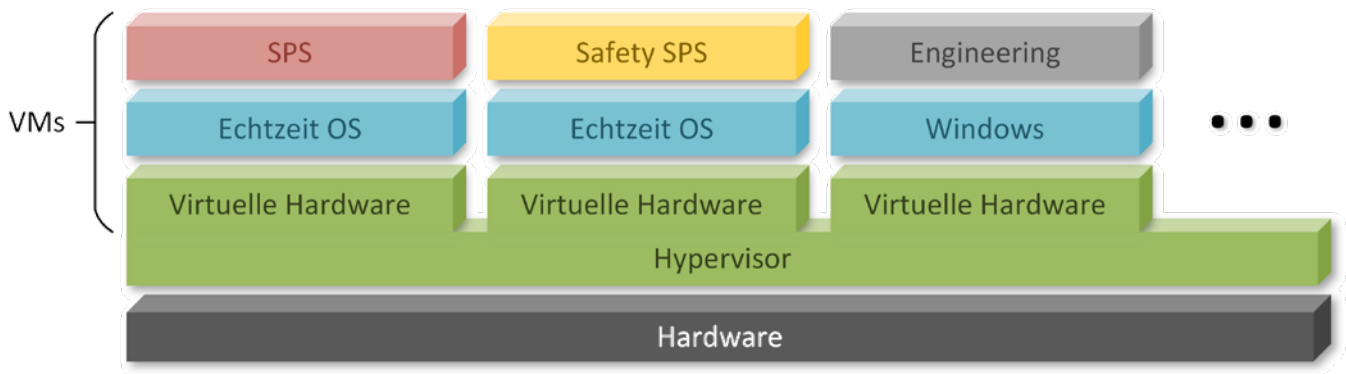


Abbildung 2: Beispielhafter Aufbau von Hardware-Virtualisierung mit typischen VMs aus der Produktion.

speicherprogrammierbare Steuerung (SPS) und andere Prozesssteuergeräte erfordern die Ausführung ihrer Programme in Echtzeit. Zudem werden in Anlagen Sicherheitssteuerungen benötigt, die eine hohe Betriebssicherheit erfordern, die typischerweise durch spezielle redundante Hardware realisiert wird. Je nach Maschine und Einsatzzweck werden oft Safety-Level von SIL 3 oder PL-e erforderlich. Die Stufen geben den Grad der erforderlichen Ausfallsicherheit an und sind international standardisiert.

Abseits der Embedded-Geräte werden in der Produktion häufig Windows-PCs eingesetzt, die vor allem im Engineering weit verbreitet sind. Endanwendersoftware ist daher meist auf Windows mit x86-CPU zugeschnitten, weshalb eine Unterstützung dieses CPU-Typs erforderlich ist.

Was ist Virtualisierung?

Virtualisierung bedeutet in der Informationstechnik zunächst die Abstraktion von Ressourcen. Aufgrund der Allgemeinheit des Begriffs werden häufig verschiedene Arten der Virtualisierung wie Hardware, Desktop, Speicher oder Netzwerk-Virtualisierung unterschieden, die unterschiedliche Ziele verfolgen. Bei der Konsolidierung von Systemen auf eine Hardware wird daher von *Hardware-Virtualisierung* gesprochen, sodass mehrere Systeme parallel auf die abstrahierte Hardware zugreifen können.

Die parallel ausgeführten Systeme werden üblicherweise als *Virtual Machine (VM)* bezeichnet. Wenn mehrere Betriebssysteme auf einer Hardware parallel ausgeführt werden sollen, müssen die Hardwareressourcen (CPU, Arbeitsspeicher, I/O-Geräte, Interrupts, Netzwerkschnittstelle, GPU, ...) aufgeteilt werden, sodass alle Betriebssysteme Zugriff auf die Hardwareressourcen erhalten.

Die Aufteilung der Hardwareressourcen für den gemeinsamen Betrieb mehrerer Betriebssysteme erfolgt durch eine Zwischenschicht, auch Hypervisor genannt, und kann auf Software- oder Hardwareebene erfolgen. Je nach Prozessortyp und Hardware erfolgt die Aufteilung unterschiedlich.

Aufgrund der geforderten Unterstützung einer x86-CPU, werden hier nur die Methoden zur Aufteilung der Hardware bei x86-CPU's gezeigt. Die x86-CPU ist von ihrer historischen Entwicklung her zunächst nicht für den Virtualisierungsbetrieb konzipiert worden, weshalb hardwareseitige Mechanismen

fehlten, um eine Umsetzung von Virtualisierung zu ermöglichen. Außerdem erschwerte die Umsetzung des Zugriffsschutzes im Befehlssatz die Behandlung von Befehlen im Hypervisor, die einen Speicherzugriff oder Zustandsveränderung des Prozessors auslösen. Diese Befehle müssen im Hypervisor behandelt werden, damit getrennte Speicherbereiche und Zustände für die einzelnen VMs realisierbar sind. Es wurden deshalb vier Möglichkeiten entwickelt, diesem Problem entgegen zu treten:

1. **Hardwareemulation:** Emuliert die komplette Hardware. Dies hat Nachteile in der Ausführungsgeschwindigkeit.
2. **Typ 1 Hypervisor:** Virtualisierung wird durch Hardware unterstützt. Angepasste Struktur im Befehlssatz. Der Hypervisor läuft direkt auf der Hardware.
3. **Typ 2 Hypervisor:** Virtualisierung durch Emulation der Speicherbefehle. Der Hypervisor läuft in einem Betriebssystem, auch Host-Betriebssystem genannt, und führt die Emulation der Befehle durch.
4. **Para-Virtualisierung:** Das Gastbetriebssystem wird verändert, sodass alle kritischen Befehle durch sogenannte *Hypercalls* ersetzt werden und im Hypervisor behandelt werden. Dies erfordert keine Unterstützung der Hardware, aber eine Anpassung der Gastbetriebssystems.

Echtzeitvirtualisierung

Die Erfüllung von Echtzeitanforderungen hängt immer vom gesamten System mit allen Ebenen von Hardware- bis Anwendungsebene ab. Daher wird eine echtzeitfähige Hardware, ein echtzeitfähiger Hypervisor und ein echtzeitfähiges VM-Betriebssystem benötigt, damit eine Echtzeitanwendung in einer VM ohne Echtzeitverletzungen ausführbar ist. Die Hardware verursacht nicht zu unterschätzende Verzögerungen durch z. B. das BIOS, die CPU-Optimierung, den Speicherzugriff oder den Cache. Eine intensive Prüfung der Hardware ist daher unerlässlich.

Der Hypervisor ist für die Zuteilung der Ressourcen zuständig. Um ein deterministisches Zeitverhalten bei der Ressourcenzuteilung zu garantieren, gibt es zwei Ansätze.

Räumliche Teilung: Hier wird jeder VM eine bestimmte Speicherregion, ein bestimmter Cache-Bereich und ein bestimmter CPU-Kern exklusiv zugeordnet. Dadurch ist der Einfluss des Hypervisors auf das Zeitverhalten bis auf einen konstanten Overhead gering. Dies kann auch für die Netzwerkschnittstelle durch ein *PCI(e)-Passthrough* erreicht werden, indem jeder VM eine eigene Netzwerkkarte zugeordnet wird.

Zeitbasierte Teilung: Bei einem zeitbasierten Verfahren werden die Ressourcen nach einem bestimmten Plan (auch *Scheduling* genannt) den VMs zugewiesen. Je nach Anzahl der VMs steigt die Latenz bei zeitbasierten Verfahren. Auch der Ort der Umsetzung hat einen Einfluss auf das Latenzverhalten. Umsetzungen in Hardware sind im Allgemeinen schneller.

Typische Echtzeitanforderungen in der Produktion sind die Einhaltung von Zykluszeiten bis zu einer Millisekunde. Daher müssen möglichst kleine Latenzen erreicht werden. Deswegen ist die Verwendung von räumlicher Teilung und einem Typ 1 Hypervisor mit Hardwareunterstützung notwendig. Dies zeigt sich z. B. beim Vergleich der Ressourcenzuteilung der CPU. Der Hypervisor KVM ermöglicht eine räumliche Teilung und erreicht eine maximale Schedulinglatenz von 71 µs, wohingegen der Hypervisor XEN mit einer zeitlichen Teilung eine Schedulinglatenz von über 1 s erreicht (s. Tabelle 1). Der Einsatz von Hypervisoren mit der Option zur räumlichen Teilung, wie z. B. *KVM*, *ACRN* oder *WindRiver*, ermöglicht daher echtzeitfähige Virtualisierung für die Produktion. Es gibt noch weitere x86-basierende Hypervisoren, die laut Produktbeschreibung echtzeitfähig sind. Allerdings stehen dafür keine öffentlichen Messdaten zur Verfügung.

Tabelle 1: Übersicht über Schedulinglatenzen in einer Linux-Echtzeit VM mit unterschiedlichen Hypervisoren.

Hypervisor	Min in µs	Avg in µs	Max in µs	Quelle
KVM	-	-	71	[1]
ACRN	2,4	4,8	12,5	[2]
XEN	-	-	855145	[2]
	-	-	1260548	[1]
WindRiver	2,17	2,3	5,8	[3]

Hardwareunabhängige Safety

Sicherheitssteuerungen sind Steuerungen, die zur Reduktion des Gefahrenrisikos durch einen Ausfall eingesetzt werden. Sie gehören damit zur funktionalen Sicherheit, da die korrekte Funktionsweise der Sicherheitsteuerung maßgeblich zur Sicherheit der gesamten Maschine beiträgt. Die Realisierung funktionaler Sicherheit erfolgt durch einen Nachweis, der durch Normen vorgegeben wird. Hier sind die Normen DIN EN 62061 und DIN EN ISO 13849 für Anlagen und Maschinen relevant. In diesen Normen wird sowohl zufälligen als auch systematischen Fehlern begegnet.

Es gibt verschiedene Möglichkeiten wie zufälligen Fehlern auf Hardwareebene begegnet werden kann. Diese basieren immer auf Redundanz wie z. B. Hardware-Redundanz, Zeitredundanz

oder Informationsredundanz. In der Produktionstechnik ist die Verwendung von Hardwareredundanz weit verbreitet, bei der zwei CPUs z. B. im Lockstep-Verfahren arbeiten. Es kommt also spezialisierte Hardware zum Einsatz, die nur durch eine komplette Anpassung des Hypervisors auf die spezialisierte Hardware verwendbar wäre.

Ein alternativer Ansatz ist die Informationsredundanz. Dabei hat sich in den letzten Jahren die *arithmetische Codierung* als vielversprechender Ansatz durchgesetzt. Diese, rein auf Softwareebene implementierbare Methode, bietet umfassende Möglichkeiten, um zufällige Hardwarefehler zu detektieren. Aufgrund der hardwareunabhängigen Implementierung eignet sich dieser Ansatz optimal für den Einsatz in einer virtualisierten Umgebung.

Fazit

Die Reduktion von Hardware in der Produktionstechnik ist vielversprechend hinsichtlich Anschaffungskosten, Betriebskosten und reduziertem Wartungsaufwand. Der Einsatz von Virtualisierung, die in anderen Domänen schon viele Jahre im Einsatz ist, eignet sich auch für die Produktion, da die Anforderungen hinsichtlich Echtzeit und Safety erfüllbar sind. Die Cloud in der Produktion ist somit keine ferne Zukunft mehr, sondern kann theoretisch schon morgen zum Einsatz kommen.

Referenzen

- [1] Abeni, L., & Faggioli, D. (2020). Using Xen and KVM as real-time hypervisors. *Journal of Systems Architecture*, 106, 101709.
- [2] Li, H., Xu, X., Ren, J., Dong, Y. (2019). ACRN: a big little hypervisor for IoT development. In *Proceedings of the 15th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments* (pp. 31-44). DOI: 10.1145/3313808.3313816.
- [3] Intel, Inc. (2014). *Achieving Real-Time Performance on a Virtualized Industrial Control Platform*. Abgerufen von: <https://www.intel.de/content/www/de/de/embedded/industrial/industrial-solutions-real-time-performance-white-paper.html>.



Marc Fischer

Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen (ISW)
 Universität Stuttgart
 +49 711 685-82534
 marc.fischer@isw.uni-stuttgart.de



Christian Klaus Müller

Team IIoT | Audi Production Lab
 Audi AG
 85045 Ingolstadt
 christian-klaus.mueller@audi.de



Christopher Kolb

Team IIoT | Audi Production Lab
 Audi AG
 85045 Ingolstadt
 christopher.kolb@audi.de