

© 2019 Shutterstock

VIRTUALISIERUNGSTECHNOLOGIE

Virtuelle Software Plattform für das Auto

Die Virtualisierungstechnologie gewinnt an Akzeptanz in der Automobilindustrie. Denn damit kann eine Architektur entwickelt werden, ohne auf Hardware oder Portierungsaufwand Rücksicht nehmen zu müssen. Erst nach der Erstellung des Software-Designs wird entschieden, welche Hardware eingesetzt werden soll und welche bzw. wieviele SoCs notwendig sind, um die nötige Rechenleistung aufzubringen.

Neue Fahrerassistenzsysteme, digitale Cockpits, Beifahrer- und Rücksitz-Displays liegen im Trend. Sie erfordern sehr leistungsfähige Hardware, aber gleichzeitig steht die wachsende Zahl von Hardwarekomponenten im Widerspruch zum erklärten Ziel der Automobilindustrie, leichtere und sparsamere Fahrzeuge zu produzieren. Die Virtualisierung eröffnet einen Weg, um Funktionen zu konsolidieren und die Anzahl der Steuergeräte drastisch zu reduzieren.

Ein wesentlicher Bestandteil von Virtualisierungsarchitekturen ist der Hypervisor. Er läuft direkt auf dem System-on-Chip (SoC)-Prozessor und ermöglicht es, mehrere virtuelle Maschinen (VM) mit unterschiedlichen Betriebssystemen parallel auszuführen. Jede VM ist von den

anderen isoliert. Diese Trennung (die ISO 26262 nennt es „Störungsfreiheit“) ermöglicht es, mehrere unterschiedliche Funktionalitäten auf einem einzigen SoC auszuführen, auch wenn die Systeme unterschiedlicher Kritikalität zugeordnet werden (ASIL-Levels wie QM, A, B, C und D). So ermöglicht der Hypervisor den Betrieb von Gastbetriebssystemen wie Linux, Android, AUTOSAR oder andere Echtzeit-Betriebssysteme in separaten virtuellen Maschinen.

Die so ermöglichte Konsolidierung der Funktionen erfordert sehr leistungsstarke Prozessoren (SoCs). Allerdings bieten die Hardwarehersteller Betriebssysteme mit proprietären Treiberlösungen an, die sie in einem Board Support Package (BSP) liefern. Die Anzahl und die Versionen der jeweiligen Betriebs-

systeme sind begrenzt. Deshalb müssen Systementwickler sicherstellen, dass ihre Anwendungen jeweils auf den vom Siliziumhersteller unterstützten Betriebssystemen ausgeführt werden können. Daher wünschen sich OEMs und Erstausrüster, dass sie die Betriebssysteme, die oft genau auf ihre Anwendungen zugeschnitten sind, über die gesamte Lebensdauer des Fahrzeugs einsetzen und aktualisieren können. Sie möchten die gleiche Architektur und die Software-Bausteine über mehrere Fahrzeuglinien hinweg wiederverwenden, um noch mehr Entwicklungszeit, -kosten und -risiko zu sparen.

Außerdem möchten die Hersteller anstelle von System-Software, die meist von einem Zulieferer bereitgestellt wird, Software aus vielen Quellen

beziehen, so z. B. von ihren eigenen internen Softwarehäusern und von Drittanbietern. Hinzu kommt, dass Over-the-Air-Software-Updates und Gerätemanagement ein Muss in allen neuen Fahrzeugsystemen sind. Nur so können Fehler in Bezug auf Funktionalität, Sicherheit oder Schutz behoben werden und längst schon erwarteten Fahrzeugbesitzer, dass sie neue Funktionen nach dem Kauf hinzufügen können. Dafür ist es wichtig, dass modulare Software-Updates möglich sind. Schließlich wollen OEMs nicht das gesamte Gerät nach dem Update re-qualifizieren, sondern nur die hinzugefügte oder aktualisierte Anwendung.

Virtuelle Treiber

Die auf offenen Standards basierende Virtualisierung bietet genau diese gewünschten Optionen. Dabei verwenden die Betriebssysteme virtuelle Treiber, die standardisiert und damit unabhängig vom Hypervisor sowie der darunterliegenden Hardware sind. Diese Treiber

befindet sich im Geräte-Manager eine Anpassungsschicht, die die physikalischen Treiber über eine standardisierte Schnittstelle zur Verfügung stellt. Die Betriebssysteme, die in anderen virtuellen Maschinen laufen, verwenden die virtuellen Treiber. Über die standardisierte Schnittstelle, die von der Anpassungsschicht im physischen Geräte-Manager bereitgestellt wird, greifen sie auf physische Geräte zu. Da die virtuellen Treiber in den anderen Gast-VMs nur auf standardisierte Schnittstellen für den Hardware-Zugriff angewiesen sind, sind sie völlig unabhängig von den hardware-spezifischen Gerätetreibern.

Wiederverwendung der Software

Wenn die Automobilindustrie insgesamt davon überzeugt werden kann, solche virtuellen Treiber zu verwenden, die alle dem gleichen offenen Standard entsprechen, sind sie frei bei der Wahl von Hardware, Software und der Treiberanpassungsschicht für ihre Softwaresysteme.

Industriestandard VIRTIO

Der Standard, der sich sehr gut für die Entkoppelung zwischen Software und Hardware eignet und sich auch bereits in Serie bewährt hat, ist der offene Industriestandard VIRTIO. VIRTIO bietet ein sehr ausgereiftes Framework für die gemeinsame Nutzung von Geräten mit verschiedenen Betriebssystemen an, die auf einem oder mehreren SoCs inklusive einem Hypervisor ausgeführt werden.

Der 2008 entwickelte Standard spezifiziert die Schnittstelle zwischen virtuellen Maschinen mit Zugriff auf Ein-/Ausgabegeräten und Interaktionen zwischen virtuellen Maschinen und Hypervisoren. Anstelle von physischer Hardware stellen VIRTIO-Geräte abstrakte Gerätefunktionalität über virtuelle Treiber für Gastbetriebssysteme bereit. Diese virtuellen Treiber sind in allen wichtigen Betriebssystemen bereits verfügbar: Linux, Android, BSD, Windows sowie einigen proprietären Echtzeitbetriebssystemen. Der Standard, der offi-

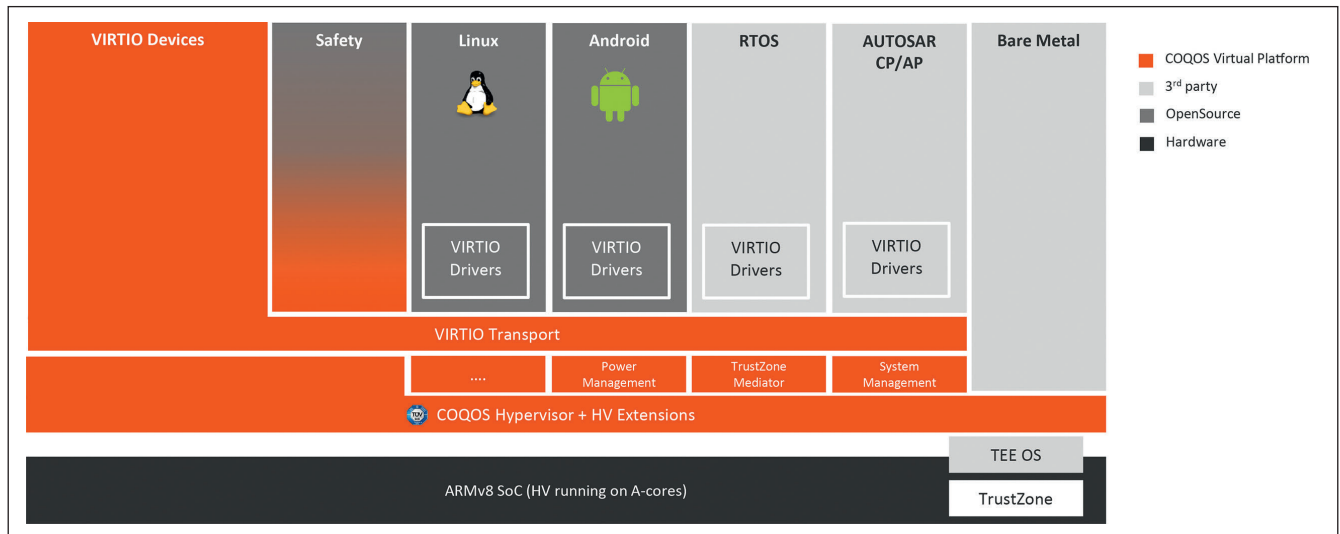


Bild1 : Architekturdarstellung einer virtuellen Plattform mit Hypervisor und VIRTIO. © OpenSynergy

sind für die Kommunikation zwischen dem jeweiligen Gast-Betriebssystem und der Hardware zuständig, z. B. Display-Controller, GPU, Soundgerät, usw.. Überlicherweise dient eines der Betriebssysteme, das in einer virtuellen Maschine läuft, als physischer Geräte-Manager. Der Geräte-Manager enthält physische Treiber, die der Siliziumhersteller für die eigentliche Hardware bereitstellt oder die vom Kunden entwickelt bzw. angepasst werden. Zusätzlich

me. Die Wiederverwendung bestehender Softwaresysteme auf anderer Hardware oder der Einsatz anderer Software auf bestehender Hardware ist ebenso einfach. Hersteller können die Anpassungsschicht zwischen den physikalischen Treibern der Hardware und den standardisierten virtuellen Treibern von jeder Firma beziehen, die mit dem gemeinsamen Standard arbeitet, oder die Anpassungsschicht sogar selbst entwickeln bzw. anpassen.

ziell „OASIS Virtual I/O Devices“ heißt, wird vom OASIS-Konsortium gefördert und gepflegt. Er wird von einer großen Gemeinschaft von Entwicklern weltweit unterstützt und kontinuierlich für den Einsatz in der Automobilindustrie erweitert. Dank dieses Engagements werden zurzeit laufend neue Geräte entsprechend dem Standard virtualisiert, die für die Automobilindustrie unverzichtbar sind (z. B. Audio, Sensoren, GPU-Support für 2D und 3D, USB, CAN, usw.).

VIRTIO-net	Paravirtualisiertes Ethernet
VIRTIO-input	Eingaben für Touch
VIRTIO-scmi	Plattformtakte und Regler für nahtloses Pass-Through
VIRTIO-snd	Audio
VIRTIO-blk	Universal Flash Storage
VIRTIO-gpu	Grafikeinheit
VIRTIO-rpmb	Geschützter Speicherbereich
VIRTIO-console	Inter-VM- Kommunikation
VIRTIO-rng	Zufallsgenerator für Verschlüsselung
VIRTIO-vsock	Punkt-zu-Punkt Verbindung zwischen VMs

Tabelle 1: Wenn Softwareunternehmen in der Automobilindustrie ihre vorhandenen Geräte durch VIRTIO-kompatible Implementierungen ersetzen, wird der Gewinn an Flexibilität immens. In Kürze stehen diese Geräte für die Automobilindustrie in Version 1.2 zur Verfügung. © OpenSynergy

Weil die Eigenschaften der virtuellen Treiber dem Direct Memory Access (DMA) ähnlich sind, erreichen die Implementierungen die gleiche oder nahezu die gleiche Leistung wie hardwareunterstützte I/O-Virtualisierungsmodelle. Dies liegt daran, dass VIRTIO aufwendige Speicherkopieroperationen vermeidet. Anstatt Speicherinhalte zu übertragen, was langsam wäre und die Allzweck-Rechenkerne des Systems übermäßig belasten würde, überträgt VIRTIO Referenzen auf die tatsächlichen Daten über virtuelle Warteschlangen. Um beispielsweise einen großen Datenblock auf das physische Flash-Gerät zu schreiben, fügt der virtuelle Geräte-Manager nur einen kleinen Verweis auf den Speicher ein, in dem sich der Datenblock befindet. Der physikalische Geräte-Manager überträgt dann den großen Datenblock direkt von der Gast-VM an den Controller des Flash-Geräts.

GPUs in einem Cockpit-Controller

Ein anschauliches Beispiel bei dem VIRTIO hohe Leistungsanforderungen erfüllt, ist die gemeinsame Nutzung von GPUs in einem Cockpit-Controller. Der Cockpit-Controller steuert mehrere Displays, typischerweise das Kombiinstrument-Display und das Infotainment-Display. Manchmal umfasst es auch zusätzliche Screens wie ein Head-up-Display, das wichtige Informationen für den Fahrer projiziert oder Bildschirme für Passagiere im vorderen oder hinteren Teil des Fahrzeugs. Dafür wird auf einem System-on-Chip eine große Anzahl sehr unterschiedlichen Funktionen integriert: von Infotainment-Anwendungen bis hin zu Fahrerinformationssystemen. All diese Systeme, ausgeführt auf derselben

virtuellen Plattform, können die verfügbare GPU-Leistung mit minimalem Performance-Overhead gemeinsam nutzen.

Die neuen Varianten von Googles Android Automotive OS werden über ein vorintegriertes VIRTIO-Framework verfügen. So kann Android Automotive OS auf allen vorhandenen Systemen ausgeführt werden, die VIRTIO unterstützen. Die VIRTIO-basierte Virtualisierungsplattform garantiert außerdem das sichere Nebeneinander mit anderen Betriebssystemen, wie Linux, einem RTOS oder anderen Android-Instanzen.

Ausblick

In Zukunft ermöglicht diese Technologie einen ganz neuen Ansatz für die Konzeption von Software-getriebenen Funktionen im Fahrzeug: die Software Defined Architecture (SDA).

Traditionell haben Automobilhersteller ihre Modelle i.d.R. von den einzelnen Komponenten des Fahrzeuges ausgehend konzipiert. Auch die Software eines Autos haben sie somit ausgehend von der Hardware entwickelt bzw. über Zulieferer entwickeln lassen. Das führte und führt überwiegend bis heute zu hardwarespezifischen Software-Stacks und zu Lieferantenbindung.

Der SDA-Ansatz trennt die Hardware von der Anwendungssoftware durch die in diesem Artikel beschriebene Virtualisierungstechnologie. Es kann zunächst eine Architektur designt werden, die alle gewünschten Funktionen eines Systems, z. B. eines Cockpits umfasst. Die virtuelle Plattform wird entsprechend dieser Architektur entwickelt, ohne auf Hardware oder Portierungsaufwand Rücksicht zu nehmen. Erst nach der Er-

stellung des Software-Designs entscheiden die Hersteller, welche Hardware eingesetzt werden soll und welche bzw. wieviele SoCs notwendig sind, um die nötige Rechenleistung aufzubringen. Das erleichtert den Herstellern, dieselbe Software über viele Fahrzeuglinien mit unterschiedlichen Hardwarekonfigurationen hinweg zu nutzen. Der Ansatz der Software Defined Architecture erfüllt somit die bestehenden Wünsche an Flexibilität: OEMs haben die freie Auswahl bei der Hardware (SoCs, MCUs, GPUs, Sensoren, Speicherlösungen und Netzwerke) und bei den Softwarekomponenten (Betriebssysteme und Hypervisoren).

Fazit

Infotainment-Geräte, die auf Software Defined Architecture basieren und den VIRTIO-Standard 1.1 nutzen, sind bereits in Produktion. Die Entwicklung mit fortschrittlicheren VIRTIO-Geräten, z. B. Grafikeinheiten auf einem einzigen SoC, hat bereits begonnen und sollte in den nächsten 1 bis 3 Jahren in Serie gehen. Dieser Paradigmen-Wechsel, die Fahrzeugentwicklung von der Software-Perspektive aus zu denken, ist ein enormer Fortschritt für die Automobilindustrie. Es ist an der Zeit, dass sich Hersteller wie Zulieferer von proprietären Lösungen auf offene Standards zubewegen. Dieses ermöglicht ihnen, großartige Innovationen zu schaffen und dabei Zeit und gleichzeitig Materialkosten einzusparen.

■ (oe) www.opensynergy.com

Isaac Trefz ist Produktmanager für COQOS Hypervisor SDK bei der OpenSynergy GmbH.