

Virtualisierung für die „Kleinen“

Warum auch eingebettete Systeme von Virtualisierungstechniken profitieren können, um die Leistungsfähigkeit von Multi-Core-Prozessoren optimal zu nutzen

Die Entwicklung leistungsfähiger Prozessoren war in den letzten Jahren durch die Abkehr von der Leistungssteigerung durch Taktfrequenzerhöhung gekennzeichnet. Der Grund dafür liegt u.a. in der Power Wall, die eine physikalische und ökonomische Grenze bei der Prozessor-Entwicklung beschreibt. Der einzige Weg zur Steigerung der Leistungsfähigkeit eines Systems liegt daher nicht in der Erhöhung der Maximalleistung, sondern viel mehr in der Steigerung des Durchsatzes eines Prozessors mit Hilfe von mehreren, unabhängig arbeitenden Rechenkernen.

Von Robert Hilbrich

Multi-Core-Prozessoren, ob als Dual- oder Quad-Core, gehören mittlerweile im Server- und auch im Desktop-Bereich zum Standard. Der Einsatz dieser neuen Prozessorgeneration in komplexen, software-intensiven, eingebetteten Systemen verläuft dagegen wesentlich langsamer. Während bereits zertifizierte Steuerungen für Gelenkarmroboter auf der Basis von Dual-Core-Prozessoren verfügbar sind, konnten sich Mehrkernprozessoren in sicherheitskritischen Domänen, wie der Avionik oder im Automotive-Bereich, noch nicht in der Breite durchsetzen. Ein wesentlicher Grund dafür liegt in den notwendigen Anpassungen der Entwicklungsprozesse für die Software, um den notwendigen, aktiven

Umstieg auf Multi-Core-Systeme zu bewerkstelligen.

▣ Geballte Rechenleistung auch in eingebetteten Systemen

Bei der Betrachtung des theoretisch möglichen Durchsatzes von aktuellen Multi-Core-Prozessoren wird deutlich, dass mittlerweile auch eingebettete Systeme mit einer Rechenleistung ausgestattet werden können, die bis vor wenigen Jahren noch Servern vorbehalten war. Während dort jedoch die parallele Verarbeitung von Algorithmen von Anfang an ein fester Bestandteil der Architektur und des Software-Designs war, wird der Bereich der eingebetteten Systeme nun an vielen Stellen erstmalig mit Parallelverarbei-

tung und den damit verbundenen Herausforderungen konfrontiert.

Viele etablierte Entwicklungsmuster zur Implementierung von komplexen Steuerungsaufgaben basieren auf einer seriellen Ausführung mit einem zumindest temporär exklusiven Zugriff auf alle Hardware-Ressourcen, so dass beispielsweise kritische Code-Abschnitte durch ein einfaches Abschalten der Interrupts vor unerwünschten Unterbrechungen geschützt werden können. Diese Entwicklungsmuster können bei paralleler Ausführung auf mehreren Rechenkernen nicht mehr eingesetzt werden. Langjährig erprobte Software-Entwürfe und -Architekturen, verwendete Bibliotheken, aber auch die Testprozesse zur Qualitätssicherung müssen deshalb mit der Einführung von Mehrkernprozessoren erneut auf den Prüfstand.

▣ Parallele Ausführung ist nicht neu

Im Server-Bereich wurden parallele Architekturen bereits seit den 80er-Jahren erfolgreich eingesetzt, z.B. im Cray-2 von 1986. Etablierte und erfolgreiche Software-Techniken aus diesem Bereich werden daher mittlerweile auch hinsichtlich ihrer Tauglichkeit für eingebettete Systeme untersucht. Zu diesen Techniken zählt auch Software-Virtualisierung, deren Ursprünge bis zur IBM-S/360-67-Mainframe-Serie aus den 60er-Jahren zurückzuverfolgen sind.

Die Entwicklung von eingebetteten Systemen auf Multi-Core-Basis könnte damit auf fundierte Erfahrungen

aus der Entwicklung von effizienten Software-Schichten zur Abstraktion von der physischen Hardware (Hypervisoren) und Mechanismen zur Isolation zurückgreifen. In Verbindung mit Multi-Core-Prozessoren bietet sich damit das Potential zur Erhöhung der Funktionsdichte bei geringerem Platz-, Gewichts- und Energieverbrauch. Eine zertifizierbare Nutzung mehrerer Rechenkerne in sicherheitskritischen, eingebetteten Systemen stellt daher einen signifikanten Wettbewerbsvorteil dar (Bild 1).

Ein eingebettetes System ist kein Mainframe

Als direkte Folge der Untersuchung von Techniken zur Software-Virtualisierung im Embedded-Bereich wird die Virtualisierung mit neuen Anforderungen konfrontiert, die sich unmittelbar aus dem neuen Anwendungsbereich ergeben. Dazu gehört die Umsetzung von erforderlichen Systemeigenschaften wie Echtzeitfähigkeit und Determinismus, aber auch die wesentlich größere Bedeutung von Energiesparmechanismen sowie bei mobilen Geräten die Fähigkeit zum adäquaten Umgang mit wechselnden Umgebungskontexten.

Virtualisierung in der Form, wie sie bisher in ihren klassischen Anwendungsgebieten zum Einsatz kam, kann daher nicht nahtlos in den Bereich eingebetteter Systeme übernommen werden. Es ist vielmehr erforderlich, eine neue Spezialisierung dieser Techniken für den Embedded-Bereich zu entwickeln, die auch den neuen Anforderungen Rechnung trägt.

„It's the complexity, stupid!“

Die Hauptursache für den stetig steigenden Entwicklungsaufwand von sicherheitskritischen, eingebetteten Systemen liegt in der ansteigenden Komplexität des Systems. Sie ist das Ergebnis von steigenden Anforderungen an den Funktionsumfang, der neben den Kosten ein wesentliches Unterscheidungsmerkmal zu konkurrierenden Produkten am Markt darstellt.

Im Bereich der Avionik können die hochkomplexen Interaktionsmuster zwischen den beteiligten Steuergeräten mittlerweile nur noch mit Hilfe

von Software gesteuert werden. Auch im Automotive-Bereich ist ein vergleichbarer Trend zu erkennen. Fahrerassistenzsysteme werden auch bei Kleinwagen zum Standard und entwickeln sich von ehemals Komfortfunktionen zu sicherheitsrelevanten Komponenten, die an immer mehr Stellen in den Prozess des Fahrens eingreifen. Beispielsweise sind die verbauten Radar-Systeme in neueren Oberklas-

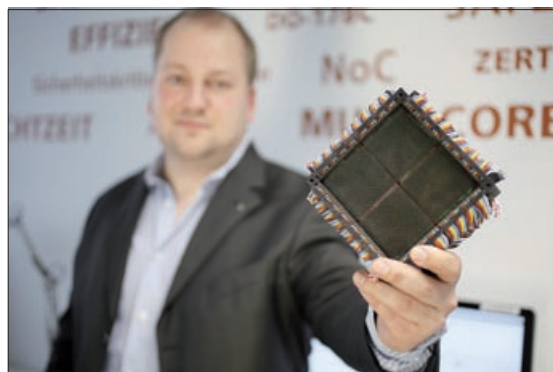


Bild 1. Parallele Architekturen werden bereits seit den 80er-Jahren eingesetzt. Sie bieten heute für sicherheitskritische, eingebettete Systeme einen signifikanten Wettbewerbsvorteil.

(Bild: Heyde/Fraunhofer FIRST)

sFahrzeugen nicht nur in der Lage, den Abstand zum nächsten Hindernis zu bestimmen. Mittlerweile können diese mehrere vorausfahrende Fahrzeuge einzeln identifizieren und deren Positionsänderung in Echtzeit verfolgen.

Komplexität folgt aus Funktionsvielfalt und Dynamik

Bei genauer Betrachtung des Komplexitätsbegriffs wird deutlich, dass eine hohe Komplexität das Resultat vom gleichzeitigen Zusammentreffen zweier Parameter ist: Funktionsvielfalt und Dynamik. Umfangreiche Funktionen können problemlos realisiert werden, wenn dafür ausreichend zeitliche Reserven zur Verfügung stehen. Andererseits führt die Forderung nach schnellen Reaktio-

nen bei einfachen Aufgaben auch nicht zu hohen Komplexitäts-Niveaus. Erst wenn komplizierte Entscheidungsprozesse und Steuerungsaufgaben sehr schnell durchgeführt werden müssen, ergibt sich ein hohes Komplexitäts-Niveau.

Für eingebettete Systeme bedeutet dies, dass eine verstärkte Verwendung von Mehrkernprozessoren die Funktionsdichte weiter ansteigen lässt, ohne dass dabei Echtzeiteigenschaften vernachlässigt werden dürfen. Damit wird sich die Komplexität des Systems zwangsläufig erhöhen und zu einer Steigerung der Wahrscheinlichkeit von unentdeckten Fehlern im Systementwurf, einer höheren Anfälligkeit für zufällige Fehler zur Laufzeit, aber auch zu mehr Fehlern in der Bedienung führen.



PCB-power also in 3 dimensions

www.wirelaid.de

Logik und Leistung auf einem PCB-Layer.
Einfach und günstig.

empower your PCB.

 WIRELAID®



Bild 2. Virtualisierung bietet Schutz und Isolation – gerade bei der Integration von Infotainment- und Automotive-Funktionen.

(Bild: Schurian/Fraunhofer FIRST)

Virtualisierung als Kompromiss zwischen Isolation und Integration

Komplexität lässt sich oft nur schwer vermeiden und muss daher im Systementwurf durch ein hohes Maß an Integration, zum Beispiel durch eng vernetzte Komponenten, abgebildet werden. Gleichzeitig soll das System aber eine unerwünschte Fehlerweiterleitung verhindern und muss daher ein hinreichendes Maß an Isolation zwischen den einzelnen Komponenten bieten (**Bild 2**).

Dieser Grundkonflikt im Entwurf hat signifikante Auswirkungen auf weitere Architektureigenschaften und Entwicklungsentscheidungen. So erhöht eine verstärkte Integration zwischen den Einzelkomponenten zwar das Niveau der damit realisierbaren Funktionsvielfalt und senkt in der Regel auch die Materialkosten, gleichzeitig geht dies aber auf Kosten der Zuverlässigkeit, da die Wahrscheinlichkeit für eine unbemerkte Fehlerweiterleitung steigt, und erhöht zudem auch den Entwicklungsaufwand durch eine aufwändige Integrationsphase.

Bisher wurde der Konflikt bei sicherheitskritischen, eingebetteten Systemen verstärkt zugunsten der Isolation aufgelöst. Dabei führt die Nutzung von durchgängig heterogenen Hardware-Komponenten und Kommunikationskanälen automatisch zu einem hohen Isolations-Niveau, da die Wahrscheinlichkeit für ein vollständiges Systemversagen als Folge eines unentdeckten Entwurfsfehlers – genauer gesagt, der Weiterleitung eines Fehlers über gemeinsame Ressourcen hinweg – minimiert wird.

Mittlerweile bekommt der Aspekt der Integration allerdings ein stärkeres Gewicht. Durch standardisierte und von mehreren Applikationen gemeinsam verwendete Hardware-Komponenten, können die Entwicklungs-, Beschaffungs- und Wartungskosten spürbar gesenkt werden. Sichtbar wird diese Entwicklung im Automobilbereich bei der Motivation hinter AUTOSAR sowie im Luftfahrtbereich beim Konzept der Integrated Modular Avionics (IMA). IMA ist eine gemeinsam genutzte Computer-Plattform für verschiedene Avionik-Systeme, die zwar logisch gesehen eine zentrale Plattform darstellt, jedoch aus Gründen der Zuverlässigkeit physisch auf verschiedene Zonen im Flugzeug verteilt ist.

Da nun eine Isolation zwischen den Systemen auf der Hardware-Ebene durch die Verwendung gleichartiger Hardware-Komponenten nicht mehr gegeben ist, muss der Forderung nach Isolation in der Software-Ebene Rechnung getragen werden. An dieser Stelle präsentiert sich Virtualisierung als vielversprechender Ansatz, um Sicher-



Bild 3. Navigationsgeräte benötigen oft spezielle 3D-Beschleuniger-Hardware – ein Hindernis für den Einsatz von Virtualisierungstechnologien.

(Bild: Schurian/Fraunhofer FIRST)

heitsmechanismen zu realisieren, so dass trotz einer hoch integrierten Hardware-Basis ein hinreichendes Maß an Isolation zwischen den Software-Komponenten erreicht werden kann.

Dies zeigt die vielfältigen Einsatzmöglichkeiten der Virtualisierung, deutet aber auch die neuen Anforderungen an, die sich durch den Einsatz bei sicherheitskritischen, eingebetteten Systemen ergeben. Eine wichtige Herausforderung ergibt sich aus dem aufwändigen Umgang mit hardware-nah implementierten Komponenten. So nutzen die Steuergeräte im Auto oft prozessorspezifische Funktionen und teilweise auch spezielle ASICs, um die

hohe Komplexität der Signalverarbeitung in einem engen Zeitraster realisieren zu können. Ein weiteres Beispiel ist der Zugriff auf 3D-Beschleuniger, wie sie in Navigationsgeräten zur graphischen Anzeige der Routen-Informationen verwendet werden (**Bild 3**). Diese Komponenten erschweren die Verwendung von zusätzlichen Abstraktionsschichten im Rahmen der Virtualisierung, da dann eventuell Echtzeiteigenschaften nur noch schwer garantiert werden können.

Multi-Core und Funktionale Sicherheit – auch ohne Virtualisierung?

Der Nutzen von Virtualisierungstechnologien, insbesondere im Umfeld von eingebetteten Systemen – ein traditionell sehr kostensensitiver Markt – wird oft hinterfragt. Gerade in Hinblick auf die statisch zur Entwicklungszeit konfigurierten Steuergeräte stellt sich die berechtigte Frage, ob Virtualisierung als dynamisches Konzept zur Laufzeit mit einem zu hohen Zuschlag durch die

Einführung einer weiteren Abstraktionsschicht in der Software verbunden ist.

Virtualisierung muss sich hier mit alternativen Architekturanansätzen messen lassen, bei denen Funktionen statisch zu den einzelnen Kernen eines Multi-Core-Prozessors zugeordnet und dann zur Laufzeit parallel ausgeführt werden. Der

Gedanke der Konsolidierung von Steuergeräten wird dabei an sich konsequent weitergeführt, wobei nun oft ein einzelner Rechenkern die Funktion eines Prozessors des „alten“ Steuergeräts übernimmt. Vereinfacht gesprochen können damit auf einem Prozessor mit n Kernen maximal n Steuergeräte konsolidiert werden (**Bild 4**).

Bei diesem Ansatz wird zudem vereinfachend davon ausgegangen, dass der Durchsatz eines Dual-Core-Prozessors dem doppelten eines gleich schnell getakteten Single-Core-Prozessors entspricht. Dies trifft jedoch nur bedingt zu, da der Zugriff auf den Speicher zwischen beiden Kernen ge-

teilt werden muss. Mit steigender Anzahl an Kernen wird diese Diskrepanz zum Single-Core-Prozessor immer größer, so dass sich ein abweichendes Laufzeitverhalten ergibt.

Ferner greift dieser Ansatz auch hinsichtlich der zukünftigen Entwicklung von Mehrkernprozessoren zu kurz. Zunächst beschränkt er die Anzahl der aktuell konsolidierbaren Funktionen auf die Menge an Rechenkernen, so dass aktuell nur etwa zwei bis acht Funktionen auf einem Prozessor integriert werden können. Allerdings ist absehbar, dass die einzelnen Kerne eines Mehrkernprozessors in Zukunft langsamer als ihre Single-Core-

Vertreter werden, um den rapiden Anstieg der Leistungsaufnahme zu reduzieren. Eine einfache 1:1-Übertragung einer Software-Funktion von einem Single-Core- auf einen Multi-Core-Prozessor wird daher in Zukunft allein schon aufgrund der reduzierten Prozessorleistung zu einer Verschlechterung der Rechenleistung führen.

Der größte Nachteil dieses einfachen Architekturansatzes liegt jedoch im Bereich der Zuverlässigkeit und der Toleranz von Laufzeitfehlern. Einzelne Kerne bieten durch ihren Befehlssatz die Möglichkeit der Trennung von privilegierten und nicht-privilegierten Instruktionen und damit einen gewissen Schutz vor den Auswirkungen von Laufzeitfehlern auf andere, parallel arbeitende Software-Komponenten. Für eine sichere Partitionierung in Raum und Zeit genügt eine einfache Trennung von Instruktionen jedoch nicht, um eine Fehlerweiterleitung über gemeinsam genutzte Ressourcen zu verhindern. Virtualisierung ermöglicht hier die sichere Partitionierung von gemeinsam genutzten Kommunikationskanälen und anderen I/O-Komponenten.

Je weiter die Entwicklung von Mehrkernprozessoren voranschreitet und je stärker damit eine höhere Integration von Komponenten realisiert wird, umso mehr wird die Partitionierung von gemeinsam genutzten Ressourcen zusätzlich zur Prozessorzeit

in den Fokus geraten. Eine höhere Integration ist letztlich das Ergebnis einer gemeinsamen Nutzung von bisher exklusiv genutzten Komponenten. Damit führt eine höhere Integrationsdichte automatisch zu einem Anstieg der Wahrscheinlichkeit einer Fehlerweiterleitung, so dass sichere und zertifizierte

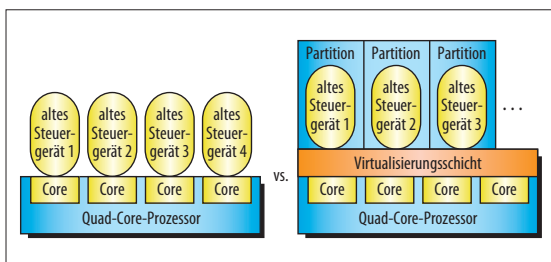


Bild 4. Migrationswege zur Konsolidierung von Steuergeräten auf Multi-Core-Prozessoren. Links: Statische Zuordnung von Steuergerätekfunktionen auf Rechenkerne. Rechts: Flexiblere Zuordnung und Isolation von Steuergerätekfunktionen auf Multi-Core-Prozessoren durch Partitionen und eine Virtualisierungsschicht.

zierbare Schutzmechanismen in der Software in Zukunft von essentieller Bedeutung sind.

■ Virtualisierung als Herausforderung für die Automotive-Domäne

Während Virtualisierung in der Software-Architektur im Luftfahrtbereich bereits seit vielen Jahren eingesetzt wird, ist dieser Ansatz in der Automotive-Domäne noch nicht weit verbreitet und trifft auf verschiedene Hürden, die nicht nur im Bereich der Software-Technik liegen.

Aktuelle Hardware-Architekturen im Automobilbereich sind durch eine gezielte Verteilung und Vernetzung von Prozessoren auf das gesamte Fahrzeug gekennzeichnet, um an geeigneten Stellen eine lokale Vorverarbeitung durchzuführen und damit zentrale Steuerungslogiken zu entlasten.

Gleichzeitig werden die Kabelbäume, die das Rückgrat einer derartigen Architektur darstellen, bereits heute für jedes Fahrzeug entsprechend der vorhandenen Funktionen konfektioniert, um Materialkosten zu senken. Eine fiktive Konzentration aller Funktionen auf einem Steuergerät würde daher zu höheren Kosten führen.

Vor diesem Hintergrund ist die Frage berechtigt, ob – und wenn ja, wo – eine Funktionsintegration auf homogenen Mehrkernprozessoren mit Hilfe

von Virtualisierungstechnologien im Automobil sinnvoll ist. Bei der Head-Unit trifft ein hoher Bedarf an Rechenleistung, zum Beispiel für ein Navigationssystem, auf sicherheitskritische Automotive-Funktionen im Kombiinstrument, so dass sich eine Integration auf der Basis von sicherer Virtualisierung anbietet. Die Head-Unit steht deshalb bei aktuellen Forschungsarbeiten auf diesem Gebiet oft im Vordergrund. Hier könnte die Konsolidierung von Automotive- und Infotainment-Funktionen auf einem Gerät von einer flexibleren Ressourcen-Allokation und einem Anstieg der realisierbaren Funktionskomplexität profitieren.

Bei diesem Ansatz treffen statisch konfigurierte Software-Architekturen, wie AUTOSAR, auf ihre dynamisch agierenden Pendanten im Infotainment-Bereich. Die Anpassung der etablierten Entwicklungsmethoden sowie der vorhandenen Entwicklungswerkzeuge stellen neue Herausforderungen für die Zukunft bereit. Dieses Vorgehen ist jedoch nicht nur mit technischen Herausforderungen verbunden, sondern erfordert vor allem auch neue Integrationsprozesse zwischen unterschiedlichen Entwicklergruppen in langjährig gewachsenen Organisationsstrukturen. *sj*



Dipl.-Inf. Robert Hilbrich

Studierte Informatik an der Humboldt-Universität zu Berlin, bevor er 2009 am Fraunhofer-Institut für Rechnerarchitektur und Software-Technik FIRST als wissenschaftlicher Mitarbeiter begann. Dort ist er seit Anfang 2011 Forschungsleiter für den Bereich Embedded Multi-Core und Business-Developer Avionik. Seine Forschungsinteressen gelten der software-technischen Nutzung des Potentials von Multi- und Many-Core-Prozessoren in sicherheitskritischen Domänen, insbesondere in den Bereichen Avionik und Automotive. Seit Oktober 2009 promoviert Robert Hilbrich an der BTU Cottbus.