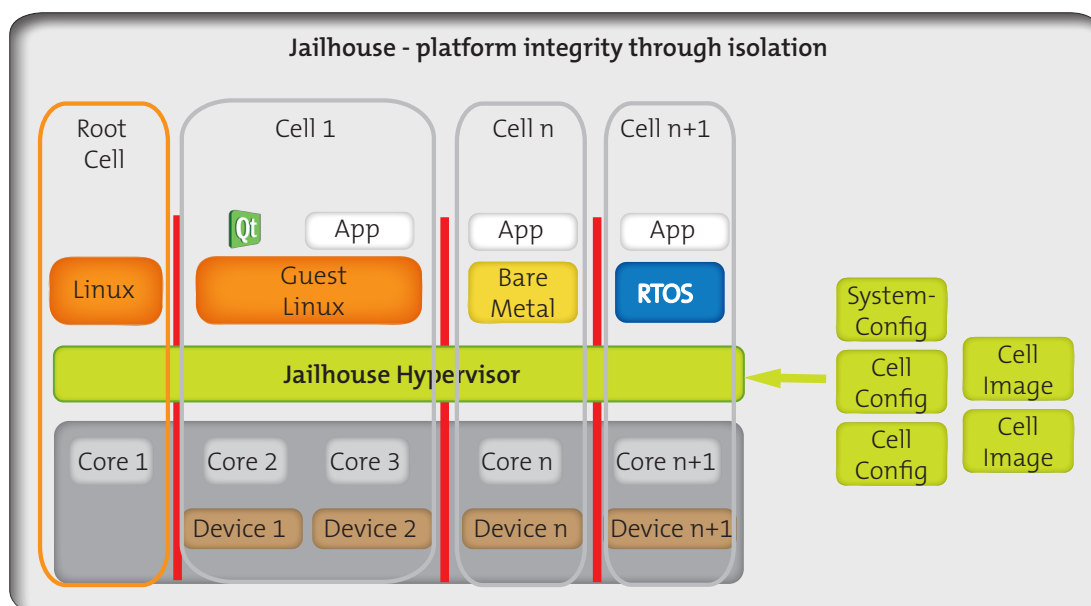


Jailhouse Realtime Hypervisor



Open Source Hypervisor mit Echtzeit Eigenschaft

Im Jahre 2014 wurde die Entwicklung des quelloffenen Hypervisor „jailhouse“ gestartet. Jailhouse ist speziell für die Nutzung von isolierten Echtzeit-Anwendungen, sei es bare-metal oder RTOS basiert, oder zertifizierbarer Software entwickelt worden.

Seit Beginn dieser Dekade sind Multi-Core-CPUs aufgrund ihrer technischen und kommerziellen Parameter für den Betrieb von Automatisierungslösungen interessant geworden. Im Steuerungsbereich sind aufgrund der Anforderungen an Zykluszeiten und Echtzeit (Jitter, Latenzzeit) bisher kaum Virtualisierungslösungen zum Einsatz gekommen. Die bisherigen Hypervisor Ansätze basieren auf einem Mikrokern, der dann als „separation hypervisor kernel“ arbeitet. Dieser Ansatz bedingt eine relativ komplexe Struktur in Form des Mikrokern, der für die Verwaltung aller Gastsysteme und deren Zugriffe auf die HW Komponenten zuständig ist.

Im Gegensatz dazu steht der jailhouse Ansatz, der ein asymmetrisches Multi-Processing (AMP) nutzt, um seine Ziele zu erreichen. Bei jailhouse wird die Hardware statisch partitioniert und

dabei eine strikte Trennung zwischen einzelnen CPUs und Geräten durchgeführt.

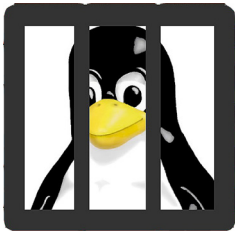
Das übergeordnete Designziel ist Einfachheit. Es soll so wenig Code wie möglich zum Einsatz kommen, um die zwei wichtigsten Anforderungen an jailhouse erfüllen zu können:

- ☐ Echtzeitfähigkeit und
- ☐ Zertifizierbarkeit (Verifizierbarkeit).

Und es sollte eine Open Source Lösung (GPLv2) werden, damit für jeden einsehbar ist, was wie gemacht wird. Das Ergebnis ist überzeugend – jailhouse unterstützt heute x86 und ARM (v7 und v8) Prozessoren, der Code ist jeweils deutlich kleiner als 10 000 LoC und die Echtzeitfähigkeit ist mit einer zusätzlichen, auf den Hypervisor zurückzuführenden Latenz in der Größenordnung von ca. 1 µsek. gegeben bzw. bleibt erhalten.

Reduzierung auf das Wesentliche

Jailhouse ist ein Typ 2 Hypervisor, setzt also auf einem Betriebssystem, in diesem Fall Linux, auf. Der jailhouse Code wird als Linux Treibermodul geladen. Dieses initiale Linuxsystem bleibt weiterhin auf dem System und dient der Überwachung und Administration der einzelnen Gast-



systeme. Es kann aber auch als ganz normales OS genutzt werden. Als Gastsysteme können andere Instanzen von Linux, Echtzeitbetriebssysteme (Linux Preempt-RT oder FreeRTOS oder ...) oder bare-metal Anwendungen in Betracht kommen.

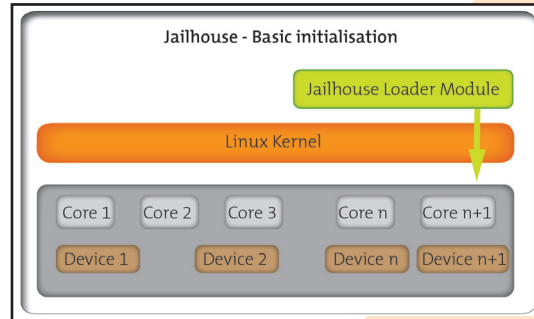


Abb. 1: Jailhouse Basis Initialisierung

Jailhouse teilt das Multi-Core-System in sogenannte „Cells“ auf, wobei eine cell aus mindestens einem Core besteht. Eine cell ist der jailhouse Terminus für die virtuelle Maschine und beinhaltet die Kombination aus CPU, Speicher und I/O. Zu jedem I/O-Gerät gehört ein definierter Speicherbereich, in dem die zugehörigen I/O Register abgebildet werden (memory mapped I/O). Jailhouse kontrolliert die Zugriffe auf die einzelnen Speicherbereiche entsprechend der Definition der jeweiligen cell. Die in den cells laufenden Anwendungen/Gastsysteme werden „inmate“ genannt.

Das Linuxsystem, das den Hypervisor kontrolliert und für die Konfiguration des Systems zuständig ist, wird „root cell“ genannt. Dieser Ansatz mit der Initialisierung der HW durch ein Linuxsystem ist der Grund dafür, den Hypervisor so schlank halten zu können. Und es erlaubt die Verwaltung und Kontrolle der einzelnen cells durch eine normale Linuxshell. Cells können während des Betriebs gestartet und gestoppt werden.

Für das Management der cells gibt es zwei unterschiedliche Betriebsmodelle: das Safety und das normale Modell. Während beim letzteren die root cell alleine verantwortlich ist und die komplette Hoheit über alle anderen cells besitzt, werden im Safety Modell bestimmte cells definiert, die aktiv an der Entscheidungsfindung beteiligt werden und ggfs. auch die root cell überstimmen können. Dies ist ein für mögliche sicherheitskritische Anwendungen notwendiger Funktionsblock.

Verwaltung

Die in der IT Welt weitverbreitete Schnittstelle libvirt zur Verwaltung wurde inzwischen in seiner Basisfunktionalität in jailhouse implementiert.

Jailhouse wurde im Rahmen des vom BMBF geförderten Projektes pCASSO eingesetzt und getestet. Seine Eignung für echtzeit-kritische Anwendungen konnte hierbei ebenso wie die Stabilität nachgewiesen werden. Die Nutzung statischer Hardware-Partitionierung ermöglicht neue Dimensionen für den Einsatz von Mutlicore-Prozessoren in der Produktion. Von der Steuerung einer gesamten Produktionslinie bis hin zu Hallen oder Standorten sind Szenarien denkbar, die durch die fortlaufende Entwicklung von Tag zu Tag realistischer werden.

Ausblick

Aktuell steht der Jailhouse-Hypervisor noch am Anfang, lässt sich jedoch durch weitere Entwicklungen immer stärker integrieren und erfährt durch die Anbindung von Libvirt einen klare Ausrichtung in Richtung Nutzerfreundlichkeit.

Jailhouse bietet eine Lösung für die unterschiedlichen Anforderungen, die an ein modernes „IoT“ System gestellt werden. Es trennt die verschiedenen Bereiche (OP, IT), die unterschiedlichen Anforderungen (Echtzeit) und erlaubt die Konsolidierung von Systemen.

Stand heute wird jailhouse unter anderen von Siemens, Huawei, ARM, AMD und anderen, z. Bsp. Linutronix, unterstützt und weiterentwickelt.

Haben wir Ihr Interesse geweckt? Wollen Sie mehr wissen? Rufen Sie einfach an, oder senden Sie uns eine Email.

LINUTRONIX GMBH

Bahnhofstrasse 3 | D-88690 Uhldingen - Mühlhofen
Telefon +49 7556 25 999 0 | Fax +49 7556 25 999 99
sales@linutronix.de | www.linutronix.de