BlackBerry | QNX

PRODUCT BRIEF

QNX Hypervisor

By enabling the secure separation and isolation of operating systems on a single compute platform, virtualization – employed using a Type 1 hypervisor – allows embedded system developers to reduce the cost, size, weight, and power consumption of the system.

The QNX® Hypervisor enables developers to partition, separate, and isolate safety-critical environments from non-safety critical environments reliably and securely; and to do so with the precision needed in an embedded production system.

The QNX Hypervisor provides broad design flexibility. At one end of the spectrum, guest operating systems (OSes) can be pinned to CPU cores and given exclusive access to hardware. At the other end of the spectrum, guest OSes can share CPU cores and hardware devices using priority-based scheduling and standards-based VirtIO interfaces – all with full hardware optimization.

Best-in-class technology

The QNX Hypervisor is a real-time, Type 1 hypervisor that is ideal for use in automotive domain controllers, industrial automation systems, and medical equipment. System designers can safely and securely consolidate instrument cluster and infotainment for a cockpit domain controller as well as design and deliver the next generation high-performance compute platforms that will power the next generation vehicle architectures. The QNX Hypervisor is a foundational element of a safe and secure domain controller.

Field-proven BlackBerry QNX operating system technology provides the core of the hypervisor runtime environment. This enables developers to use trusted BlackBerry QNX services (e.g. fast boot, splash screen display, instant device activation, secure boot) along with the award winning graphical QNX Momentics Tool Suite for analysis and debug.

An application running in a virtualized environment has a performance overhead typically less than 2% when compared to the same application running in a native environment. This extremely small overhead illustrates the efficiency of the design and hardware optimization support of the QNX Hypervisor. Boot times for guests will vary but can be reduced to tens of milliseconds.

The QNX Hypervisor supports hardware optimization on Intel x86_64 VT-x and ARMv8 AArch64 hardware. Hypervisor-enabled board support packages exist for automotive reference boards such as Intel® AtomTM processor C3000 product family, Intel® AtomTM A3900, Renesas R-Car H3, Qualcomm® Snapdragon™ 820A, and NXP i.MX 8.

Preserve safety certifications

The QNX Hypervisor facilitates safety certifications by separating safety-critical components from non-critical components in separate guest OSes. Safety certifications can be achieved on components selectively. Different parts of the system can then be updated independently without impacting certifications. The QNX Hypervisor is itself built from a safety and security pedigree (it complies with IEC 61508 SIL 3 for industrial safety, IEC 62304 for medical device software, and ISO 26262 ASIL D for automotive safety; certification in progress).

Virtual CPU model

QNX Hypervisor follows a priority-based virtual CPU (vCPU) sharing model. Each vCPU has a priority and scheduling policy, ensuring that a higher priority guest OS will always preempt a lower priority guest OS when sharing a physical CPU (pCPU). Oversubscribing of vCPUs allows system designers to maximize all cores. In addition, a vCPU may be pinned to a pCPU and given exclusive access to that core. vCPUs can be given CPU budgets using QNX adaptive partitioning. This partitioning enforces guaranteed CPU time for a set of vCPUs even when the system is completely busy. This flexibility of pinning and floating of vCPUs allows the system designer to build dependable systems without wasting hardware resources.

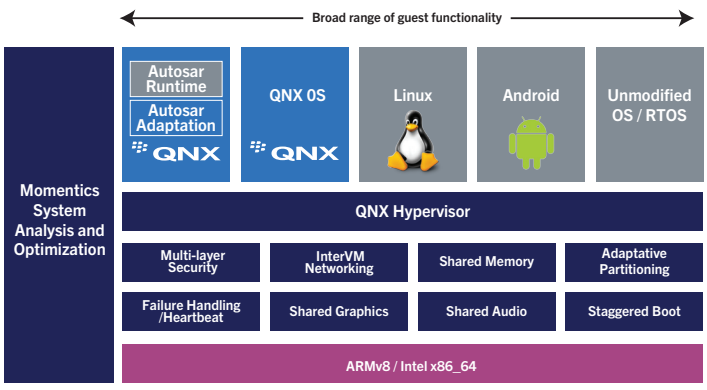
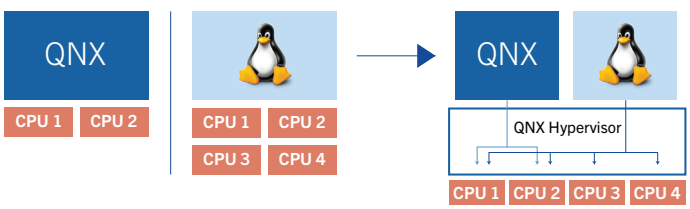


Figure 1: QNX Hypervisor software stack shared devices, multiple guest OSes, integrated toolchain.



Separate SoCs/Hardware

Hypervisor microkernel scheduler:
- vCPU has priority, processor affinity mask, and processor budget

Figure 2: Sample scenario - consolidating a QNX digital instrument cluster (a safety certified guest OS) and a Linux OS based Infotainment system on the same hardware (in this case a System-On-Chip with 4 cores). The QNX Safety certified guest with 2 vCPUs is given higher priority than the Linux Infotainment that has 4 vCPUs.

Device Interaction

In embedded systems that use a hypervisor, it is desirable to have exclusive access to certain devices while sharing other devices among guests. Sharing provides cost savings, reduced development time, and operational efficiency. A guest OS can use a mix of hardware interfaces: emulated devices such as timers and serial ports, hardware pass-through drivers (e.g. CAN bus drivers), and industry-standard VirtIO drivers for sharing Ethernet and block based filesystem devices. Guest OSes communicate through shared memory and peer-to-peer Ethernet connections. Guest OSes are launched, removed, paused, and restarted on demand and managed with built-in monitoring and failure handling services.

Shared graphics

The QNX Hypervisor provides several solutions for sharing a graphics processing unit (GPU) among multiple guest OSes with each solution making use of integrated hardware optimizations.

One option is to have a guest OS (usually a safety certified guest OS) own the graphics hardware along with the hardware-acceleration graphics support. Other guest OSes will send draw commands to the safety certified guest OS for rendering. The draw stream can target a separate display or a surface on a shared display. Another supported option involves the creation of virtual Graphics Processing Units (vGPUs). Many guest OSes can then use hardware-accelerated graphics commands at the same time. Virtual GPUs are properly coordinated and fault monitored by trusted mediation software.

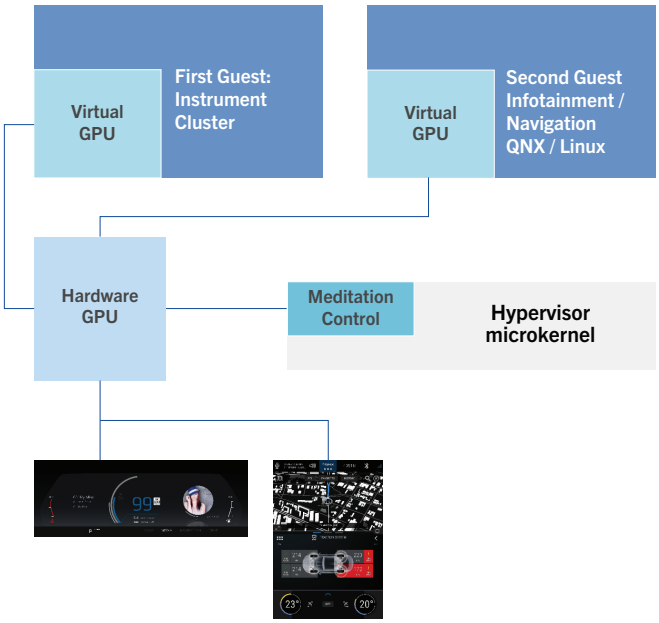


Figure 4: Mediated sharing of a (GPU) to drive a QNX digital instrument cluster and an Infotainment system at the same time.

Integration with QNX Momentics®

The QNX Hypervisor is integrated with the QNX Momentics Tool Suite, enabling developers to see and capture system-wide events such as interrupts, context switches, and shared interfaces between virtual machines. This greatly improves integration and debugging capabilities for virtualized platforms and cannot be done using typical debuggers, which are only aware of a single operating system.

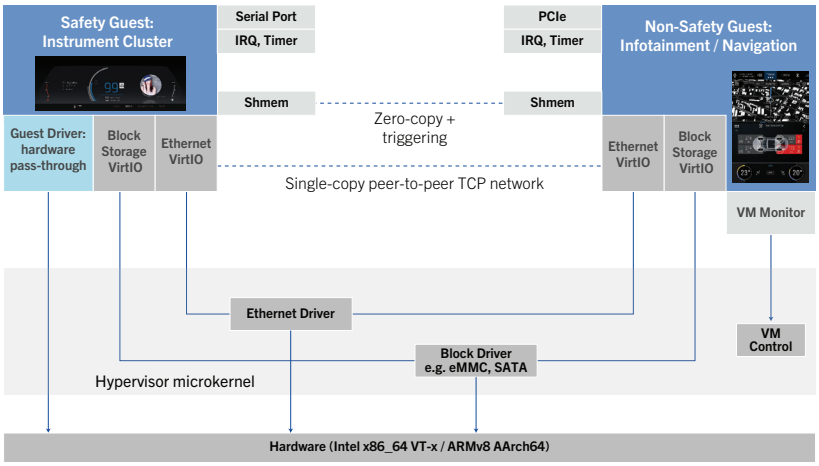


Figure 3: Example scenario depicting virtualization of a safety component (digital instrument cluster) and a non-safety component (infotainment). Services include shared memory, Ethernet, and VirtIO.

QNX Hypervisor Features

- Type 1 Hypervisor
- Safety certification pedigree
- Virtual CPU model
- Pin to cores or share cores based on priority
- Adaptive partitioning. Allows for CPU guarantees of guest runtime
- 64-bit and 32-bit guests: QNX, Linux, Android, RTOS
- Shared memory with triggering
- VirtIO (1.0) device sharing
- TAP and peer-to-peer networking with bridging
- Failure detection and restart of guests
- Virtual watchdog for guest integrity checking
- Low overhead (typical < 2%)
- Graphical tools for analysis and debug

About BlackBerry QNX

BlackBerry QNX, is a leading supplier of safe, secure, and trusted operating systems, development tools, and professional services for connected embedded systems. Global leaders such as Ford, Audi, Cisco, General Electric, Lockheed Martin, and Siemens depend on BlackBerry QNX technologies for their next generation of secure vehicle software platforms, network routers, medical devices, industrial automation systems, security and defense systems, and other mission and/or life-critical applications. This includes full software lifecycle management via secure over the air software updates. Founded in 1980, BlackBerry QNX is headquartered in Ottawa, Canada, with its products distributed in over 100 countries worldwide.