WIND RIVER

Wind River Hypervisor

Table of Contents

Wind River Hypervisor is one of the building blocks of Wind River's comprehensive embedded software solutions. It is an embedded hypervisor that provides a virtualization layer that partitions a single or multi-core chip into multiple partitions with varying levels of protection and capabilities. The effective adoption and optimization of virtualization and multi-core will be key differentiating factors in the competitive marketplace for next-generation embedded devices. Wind River provides a comprehensive solution with the breadth and flexibility to provide a future-proof path on the way to multi-core and virtualization adoption.

Virtualization is used in information technology (IT) and enterprises to deliver higher CPU utilization rates by consolidating many individual systems onto a single compute platform. This delivers significant benefit from a management and cost perspective, especially with the increases in compute capacity delivered in multi-core processors.

Virtualization is rapidly making its way into embedded systems as well, as increasingly more powerful single and multi-core processors continue to gain popularity. Virtualization for embedded devices provides new opportunities for companies building next-generation products using single and multi-core devices. Device makers can utilize virtualization to consolidate their systems by replacing multiple boards or CPUs with a single board or a single CPU. They can use multiple operating systems (such as a real-time operating system and a general purpose operating system) cooperatively to provide innovative device functionality and adopt multi-core processors with improved scalability and reliability.

Adopting and optimizing virtualization and multi-core in the embedded industry requires a wide array of technologies and skills. It requires the virtualization technology itself (the hypervisor), operating systems, support for varying processor architectures and boards, debugging and analysis tools, and test capabilities. These technologies need to be integrated, easy to use, and supported by a group of embedded experts in your geographical region who are ready to assist customers.

Wind River Hypervisor is deterministic, event-driven, small, and scalable, provides direct access to devices, and is processor, architecture, and OS agnostic. It is a Type 1 hypervisor that runs directly on the hardware, with a small memory footprint. It is custom developed with the demands of real-time and safety systems in mind.

Multi-core Software Configurations

Until recently, configuring embedded systems was relatively simple: The processor had a single core that hosted a single operating system. Depending on the product requirements, a general-purpose or real-time operating system was chosen. If both were needed, the design would have to accommodate the two processors.

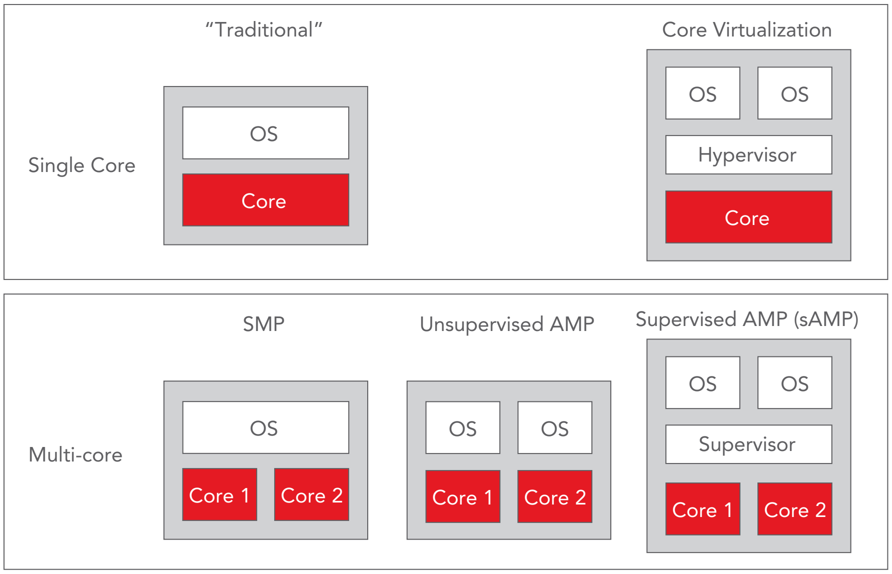

Figure 1: Primary multi-core configurations

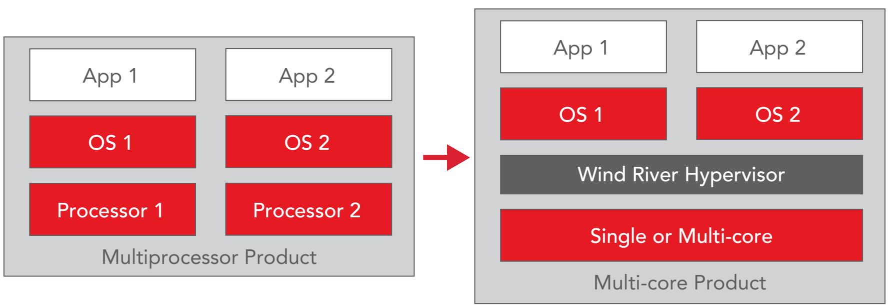

Figure 2: Consolidation of multiple single cores to a single multi-core

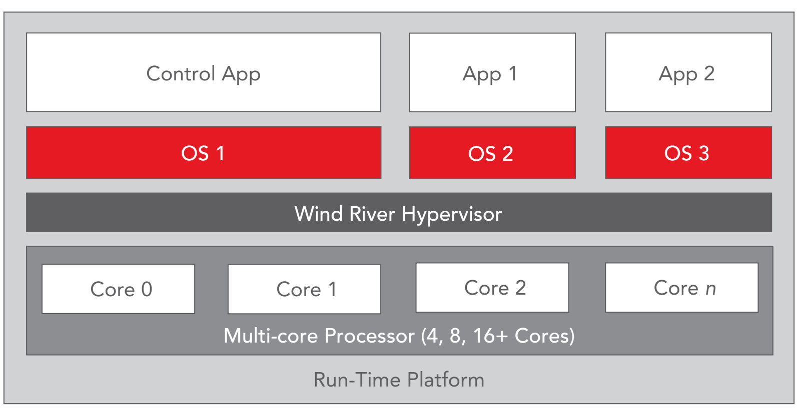

Figure 3: Scalability to high core counts

Today's powerful single and multi-core processors can be used in many different configurations.

A multi-core processor can be managed by a single symmetric multiprocessing (SMP) operating system that manages all the cores. Alternatively each core can be given to a separate operating system in an asymmetric multiprocessing configuration (AMP). Both SMP and AMP have their challenges and advantages. SMP does not always scale well, depending on workload; and AMP can be difficult to configure in regard to which operating system gets access to which shared system device.

The usage of virtualization in the form of a hypervisor enables a wide variety of configurations including mixes of AMP, SMP, and core virtualization. The hypervisor manages the hardware and creates partitions in which operating systems execute. Each partition is given access to resources (processing cores, memory, devices) as specified by the development team. Each partition can hold an operating system (also known as the guest OS) and is protected from the other partitions. The hypervisor can execute a single partition on a single core, a single partition across multiple cores, or multiple partitions on a single core.

The combination of SMP, AMP, and core virtualization provides unprecedented opportunities for device developers to innovate and manufacturers to deliver differentiated products.

Use Cases

Embedded virtualization provides new and exciting capabilities to developers and equipment manufacturers. The following are some use cases that are enabled when employing an embedded hypervisor.

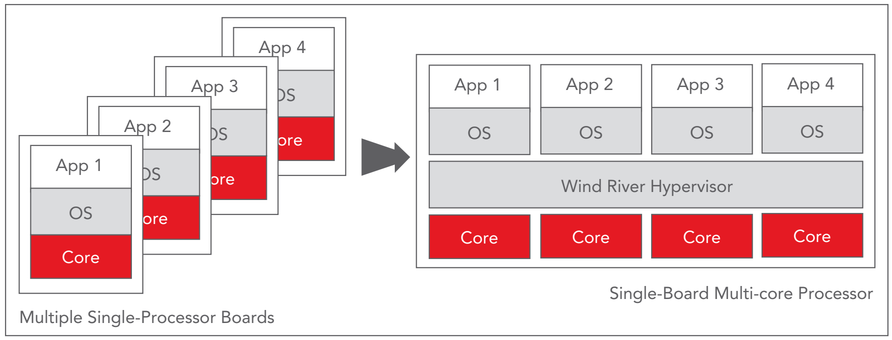

Figure 4: Reduce component costs by consolidating multiple single-processor boards onto a single multi-core CPU

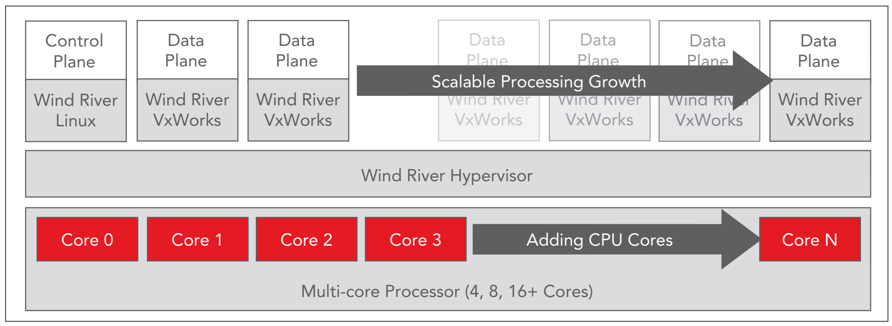

Figure 5: Achieve scalable performance growth to meet scaling performance demands

Consolidation

Many current systems use multiple processors, either on the same compute board or on multiple boards in a rack. The rationale for using multiple processors includes performance demands and the need for separation between different types of functionality (real-time and general purpose, safety, security, etc.).

Embedded virtualization can be used to maintain necessary separation, whether on a single or multi-core processor, allowing the previously separate and disparate functions to be consolidated onto a single compute platform. Multi-core processors offer increases in compute performance with lowered power consumption, allowing for this consolidation while maintaining—or increasing—the compute performance available for the individual functions.

The benefits of such consolidation include reducing the bill of materials (cost of goods sold) and reducing the amount of power used (operating costs). Each of the operating systems executing on a separate compute platform on the original system can be migrated onto a separate partition on the multi-core processor with embedded virtualization. The embedded hypervisor enforces the partitioning that provides separation and fault containment. Figure 4 shows an example of a migration from a multiprocessor system to a multi-core processor on a single board.

Performance and Functionality

Typically, systems that require scalable performance increases meet that requirement by adding incremental compute platforms to the entire solution; by adding compute or data processing boards within a rack that comprises the equipment, current product designs can realize increases in processing growth. This approach is very inefficient in many aspects. The hardware necessary to increase the compute power is expensive, draws a lot of power, and requires a lot of physical space. Among this hardware there are a lot of replicated components such as power supplies and other ancillary silicon that are not directly adding to the increase in performance but increase costs, power, and required space.

Networking Example

Embedded virtualization offers equipment vendors the opportunity to meet scalable performance requirements with significant savings in hardware costs, size, weight, and power consumption as well as significant increases in time-to-market for deployment of the services.

By migrating existing designs to multi-core CPUs and consolidating the separate compute functions into partitions enforced by an embedded hypervisor, the total number of physical boards, or physical CPUs, can be significantly reduced.

Wind River's embedded hypervisor supports the real-time requirements necessary for many high-performance processing engines, while providing a platform by which equipment manufacturers can achieve the performance gains offered by multi-core CPUs.

With the ability to dynamically deploy new instances of data processing engines through software commands—during the run-time of the equipment—the time-to-market for service delivery collapses to administrative provisioning scale. There is no need to deploy a truck and technician to insert new processing cards in an equipment rack located remotely; the commands can be provisioned at the central management and operations center. This is shown in Figure 5.

Also, disparate compute functions such as control plane management processing can be consolidated on the same multi-core hardware, in isolated partitions executing on dissimilar operating systems to meet the differing needs of the application. Control and management capabilities may require a general-purpose operating system or a real-time operating system, while data processing engines may only require lightweight processing executives without the need for a complete operating system. These differences in operating requirements can be met with the isolation provided by the hypervisor and the increased compute capacity of multi-core processors.

Industrial Example

Multi-core processors offer industrial equipment vendors the opportunity to increase product performance and functionality. When migrating existing designs to multi-core CPUs, there often can be available CPU cores remaining on which new functionality can be added. Such new functionality could be the addition of a graphics-rich interface for an industrial controller, for example, or the addition of a standards-based protocol stack. Each of these new functions can be added to the existing product while maintaining the necessary spatial and temporal separation needed to ensure that the real-time application is not impacted by the non-real-time applications. This is shown in Figure 6.

Adding new applications and code to the product's software image and hosting that new content on separate CPU cores provides the ability to add general-purpose applications without impacting the existing application. This can greatly reduce retesting efforts when adopting new hardware platforms.

Migration

Multi-core processors yield increased compute performance with reduced power consumption. Taking advantage of these benefits is attractive for device manufacturers; however, migrating existing middleware stacks and applications to new multi-core hardware is resource-intensive and can require significant redesign and retesting efforts. Depending on age, operating systems hosting existing applications may not necessarily support multi-core CPUs. Completely rewriting middleware and application stacks in new operating environments can prevent vendors from taking advantage of the benefits of multi-core CPUs.

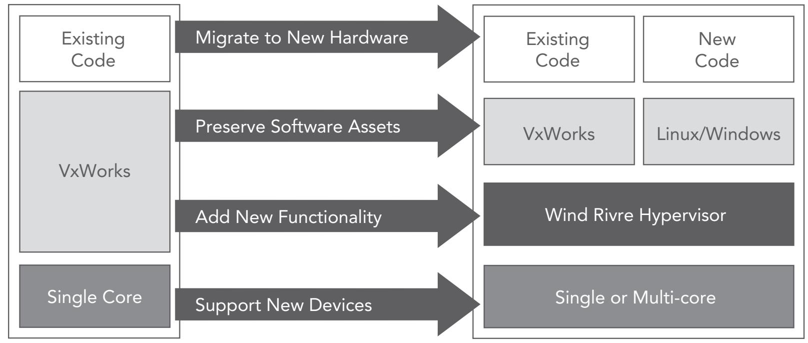

Figure 6: Migrate to new multi-core hardware and leverage new OS features

Embedded virtualization allows vendors to migrate their existing software assets from older single-core CPUs to multi-core CPUs to benefit from performance and power improvements. By leveraging embedded virtualization to isolate legacy applications into a single virtualized partition running on a single core, developers can port the existing code base with minimal effort to the new hardware.

Wind River's complete multi-core software solution of embedded operating systems—such as VxWorks and Linux—embedded virtualization, and a proven services organization helps customers move their software assets from current processors to new, advanced multi-core processors.

The challenge of porting the code base can be greatly mitigated by using an embedded hypervisor that supports unmodified guest operating systems, allowing the existing code to execute in the environment in which it was designed. The underlying hardware is abstracted by the hypervisor. A single-core virtualized partition can be created for existing applications and operating systems that do not support multi-core CPUs. The specific hardware devices required by the existing application can be presented to the partition in such a way that the migrated application detects the devices that it requires and behaves as it does on the older single-core hardware.

New applications can be added in other virtualized partitions executing on unused CPU cores alongside the legacy unmodified guest operating system, allowing developers and vendors to add to the functionality of the device with minimal effort.

By porting unmodified legacy applications in virtualized partitions, device manufacturers can retain their software investment and leverage general-purpose operating systems to provide enhanced human-machine interfaces, or offer a scaling range of product features, for example.

The real-time behavior of Wind River Hypervisor allows the new product to maintain the real-time determinism while providing exciting new capabilities, allowing device manufacturers the opportunity to deliver innovation and differentiation.

Figure 6 depicts a situation where an existing OS and application can be taken from an existing product and be augmented with a new application and operating system to develop a new, differentiated next-generation product.

Unmodified Guest OS

Having the ability to migrate an existing application along with its operating system from older, lower performing hardware to newer multi-core CPUs is an attractive proposition for device manufacturers; however, not all operating systems support multiple cores. Migrating an application to a new version of the OS can be prohibitive.

Embedded virtualization can be used to migrate the existing application and operating system to new hardware providing that the hypervisor supports *unmodified guests* and the hardware provides the necessary support.

Examples of CPUs that offer this hardware support for privilege levels that allow for unmodified guest operating systems to run include Intel CPUs with vPro technology and Freescale CPUs with e500mc CPU cores, such as the P3 and P4 families of CPUs.

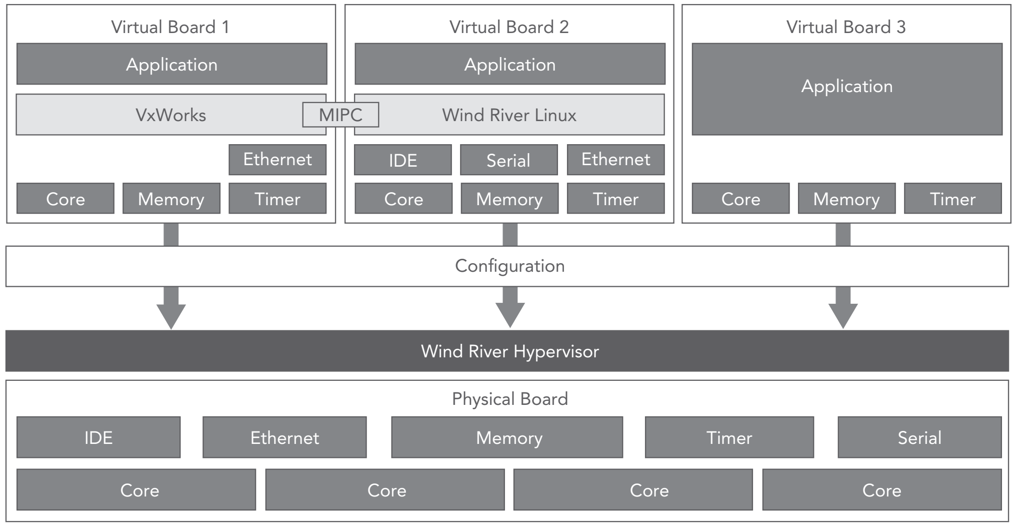

Figure 7: Wind River Hypervisor devices, virtual boards, configuration

Matching the hypervisor with a CPU that provides hardware support for virtualization puts the board designer in a position where migration from older CPU technology to newer technology can be seamless, offering the ability to leverage newer CPU devices with increased core counts, computational power, and lower power consumption.

Modified Guest OS

While the possibility to migrate an existing application to new CPU hardware without making modifications to the OS kernel is attractive, it is not always the case. The developer may have access to the OS kernel source code, offering the ability to recompile the kernel and OS for a new CPU target. To maintain the privilege levels necessary for the OS kernel to execute under hypervisor control, the system calls within the kernel can be instrumented, or augmented, with API calls to the hypervisor rather than the BIOS or hardware directly.

When faced with CPU selection, this approach is an attractive alternative to unmodified guests because it allows for a wider variety of CPUs to which migration can be targeted. Since the set of CPUs that offers the hardware support for unmodified guests is relatively small, the option of instrumenting and recompiling the kernel allows for a wider range of processors for selection.

Features

Virtual Board

Wind River Hypervisor separates a system into multiple virtual boards (partitions). The virtual board can contain a guest operating system with applications or an application without an operating system (virtual board application). The virtual board is managed by the hypervisor. The hypervisor controls which cores the virtual board executes on and which memory and devices it can access. A virtual board can share a single core with another virtual board, run dedicated on a single core, or span multiple cores.

The virtual board is a fault container. The hypervisor prevents faults that occur within a single virtual board from affecting other virtual boards and the rest of the system. The hypervisor can be designated as the fault handler, or the fault can be propagated to the virtual board for handling within the guest operating system.

Virtual boards can be individually created, paused, resumed, restarted, and migrated between cores, providing an easy way to recover from critical failures inside a virtual board. This provides the ability to dynamically scale applications to meet changes in demand. For example, as throughput or demand increases, new instances of virtual boards can be created on available CPU cores to meet the demand. Similarly, as demand wanes, instances can be paused or co-located on a single CPU core, allowing a subset of CPU cores to be freed for other processing tasks or be powered down completely, resulting in overall power savings.

Guest Operating System

Wind River Hypervisor executes guest operating systems, which host applications. A guest operating system executes with near native performance. Wind River Hypervisor uses a mix of hardware assist and paravirtualization of the guest operating system to deliver optimized performance and determinism.

Wind River Hypervisor supports real-time operating systems such as Wind River Linux and VxWorks, and its open interface allows other operating systems and executives to run, including those that are open-source-based, proprietary, or internally developed.

Unmodified Operating Systems

Wind River Hypervisor supports the execution of *unmodified* guest operating systems—operating systems and kernels that do not need to be modified to run on top of a hypervisor. This is in contrast to *paravirtualization,* discussed in a later section.

It is assumed all operating system kernels have complete write access to all hardware that is detected at boot time. In this manner, the OS kernel has the highest level of *privilege*. With virtualization and hypervisors, the hypervisor has the highest level of privilege. This level must be higher than even the OS kernel.

To continue to behave as intended, the guest OS must be able to continue to execute with these necessary changes in privilege levels, without recompiling the OS kernel. This is the intent of support for unmodified guest OSes.

Latest generation CPUs offer hardware support that allows the hypervisor to retain the highest level of privilege and the guest OS to retain complete write access to the hardware devices within its partition that it detects at boot time.

Wind River Hypervisor supports the use of Microsoft Windows inside a virtual board when running on top of Intel processors. This enables the use of Windows alongside real-time content hosted by VxWorks or Wind River Linux.

Wind River Hypervisor supports 32- and 64-bit single-core and multi-core processors and can host 32- and 64-bit single-core and multi-core guest operating systems, including 32-bit guest operating systems on 64-bit processors.

Bare-Metal Executives

Wind River Hypervisor can host the execution of a virtual board without an operating system. This is useful for certain tasks that don't need the capabilities and code footprint of a complete operating system, such as data processing engines or a fast polling loop, but require a subset of services typically provided by operating systems, such as interrupt control and scheduling.

Wind River Hypervisor provides programmable APIs that present these bare-metal executives with access to system resources such as the memory management unit (MMU), interrupts and exceptions, CPU cores, and other hardware devices necessary to operate as if the executives were hosted by a complete operating system. These APIs allow the data processing engines to benefit from virtualization while retaining the necessary real-time responsiveness. Completely event driven and lock free, the hypervisor is only as involved as needed when providing scheduling and hardware access to all operating systems on all CPUs.

Hardware Assist and Paravirtualization

Some processor architectures provide features in hardware that facilitate virtualization (e.g., Intel VT-x, PowerPC e500mc). The hypervisor can use these features to provide virtualization services.

On CPUs where these features do not exist, or if these features impact performance and determinism, the hypervisor implements them in software, and paravirtualization of the operating system is required.

Paravirtualization is the modification of an operating system's privileged system calls to collaborate with a hypervisor. The hypervisor must execute the privileged instructions on behalf of the guest operating system. This collaboration works through programmatic interfaces executed from within the guest operating system. The amount of modification depends highly on the processor architecture. It typically concerns the following:

- **Privilege levels:** A virtualized system typically requires three privilege levels: the hypervisor; the guest operating system; and the application. Many CPU architectures that do not support hardware assist have only two privilege levels. The task of paravirtualization is to emulate the missing privilege level. Privileged instructions in the guest operating system need to be replaced by hypervisor APIs.
- **Device access:** A driver in a native operating system is able to access any device. In a virtualized environment, the hypervisor arbitrates whether a virtual board has access to a device. This includes device interrupts, device I/O, device register access, and direct memory access (DMA). Device access includes access to timers, network cards, graphic cards, and so forth. Environments with hardware assist often allow device drivers to operate without modification.
- **MMU:** The hypervisor controls the MMU in a virtualized environment. Many processors with hardware-assist features allow a guest to modify the virtual board MMU. If hardware assist is unavailable, the guest may have to collaborate with the hypervisor to modify the MMU.

Flexible Configuration

Wind River Hypervisor provides flexible tooling to configure virtual boards on single-core and multi-core processors. The configuration defines how the virtual boards are distributed over the available processing cores; how they are scheduled; and how the physical hardware is partitioned between the virtual boards. There are a number of basic building blocks that the developer can use to assemble a system:

- **One virtual board allocated to a single dedicated core:** This provides the same performance as if the guest OS is running natively on that core.
- **Multiple virtual boards allocated to a single core:** The hypervisor provides scheduling services. The hypervisor's scheduler decides how processor cycles are provided to each virtual board. These decisions are either done on a priority basis or a (time slice) partition

basis. The scheduler in Wind River Hypervisor is pluggable; so, if needed, teams can design their own schedulers for integration with the hypervisor.

- **Multiple cores allocated to a single virtual board:** The guest operating system is executing in SMP mode across a number of cores.

System designers can decide how to use these building blocks in their designs. They can, for example, partition an eight-core processor in a virtual board spanning multiple cores and running an SMP guest operating system. They can run several virtual boards, each using a single core, and they can run multiple virtual boards on a single core.

Having the ability to explicitly configure which hardware devices are presented to each virtual board via the hypervisor, and in which manner the devices are accessed, delivers the benefits of virtualization while maintaining the real-time determinism of a virtualized platform.

Graphical Configuration

The configuration choices made while architecting an embedded device can be graphically selected, viewed, and verified by using Wind River Workbench development tools that are tightly integrated with Wind River Hypervisor. Graphical configuration allows developers to select interrupts and device offsets, for hardware devices, for example, and graphically view the physical board configuration and how the physical devices are allocated among the virtual boards. Through this capability, developers are also able to graphically view the complete memory map of the generated system image to verify virtual board placement in system memory.

Device Model

Wind River Hypervisor offers a unique device driver model that lets developers specifically assign devices to virtual boards in one of several access configurations depending on the requirements of the device. Device access can be *direct, shared, virtualized,* or *emulated.* Each device access method has characteristics that allow developers to craft their embedded product exactly to requirements.

Direct

With direct access to hardware devices, as the name states, the physical device is mapped into the guest's memory space; the guest has full control of and direct access to the device; and there is no virtualization layer between the guest and the physical device. The guest and application have bare-metal access to the device, and other virtual boards do not detect the presence of the device in their hardware scans. This provides the highest level of performance in guest behavior.

Figure 8: Direct guest access to devices

Shared

Shared device access is desirable when one guest "owns" access to the device but there is a need for other guests to have access to the data presented through the device. This might be the case when one guest has access to a hard drive partition, for example, but other guests need to access the file system of the hard drive. The native device driver resides in the "owning" guest while a stub device driver resides in the guest that shares the device.

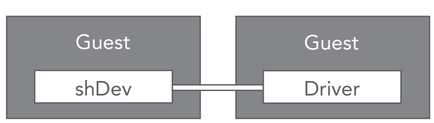

Figure 9: Shared device access

Virtualized

Virtualized access to devices is beneficial when more than one guest needs individual access to a device. In this manner, the physical native device driver is placed within the hypervisor and each guest that needs individual access to the device loads a "stub" device driver that communicates with the driver in the hypervisor.

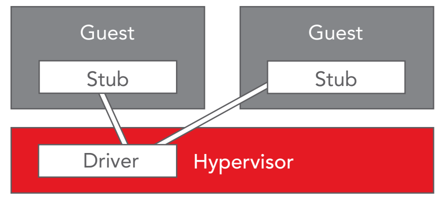

Figure 10: Virtualized access to devices

Emulated

Emulated device access is necessary when the device driver in the guest cannot be replaced with a stub driver, or if it is easier to not replace the driver. When the driver in the guest is the native default device driver, the hypervisor emulates a hardware device instead of accessing a true hardware device. The hypervisor can emulate devices for multiple virtual boards and handle the physical device for all of them. Emulated devices provide the highest level of flexibility for the virtual boards, but there is a performance penalty compared to virtualized or shared devices.

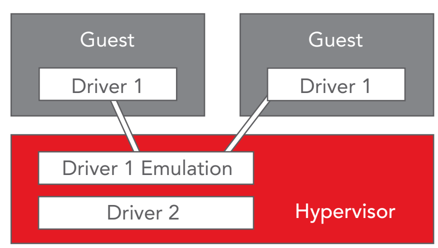

Figure 11: Emulated access to devices

Multi-OS Inter-process Communication

Communication between virtual boards is an important capability when building embedded systems. Multi-OS inter-process communication (MIPC) provides a socketlike API to send messages between virtual boards inside of a multi-core environment. Sending messages uses hardware acceleration where possible and zero copy. MIPC can also be used as an inter-process communication feature to debug other virtual boards, either as an extended serial port or as a sophisticated debug connection for Wind River Workbench.

Layer 2 Ethernet Switch

Wind River Hypervisor provides an internal virtual Layer 2 Ethernet switch that provides standards-based IP connectivity among the guests hosted by the hypervisor.

Based on needs, guests can be configured to have one or more virtual network interface card (NIC), each directly connected to a virtual port on the switch. Multiple internal subnets can be configured and connected to multiple isolated internal switches. Guests can belong to one more virtual internal network. MAC addresses are assigned by the virtual NIC drivers or can be configured along with IP addresses through the board configuration.

Architectural Design

Wind River has decades of experience in developing real-time embedded operating environments and partitioned systems, and this knowledge has been leveraged in the architecture of Wind River Hypervisor.

Deterministic

Embedded systems need to be fast and deterministic. They need to respond within known time bounds without fail. Embedded developers rely on a deterministic operating environment that needs to include the hypervisor as well as guest operating systems.

Besides the requirement for deterministic response times, Wind River Hypervisor is optimized to provide low overhead, even on multi-core processors with a large amount of cores.

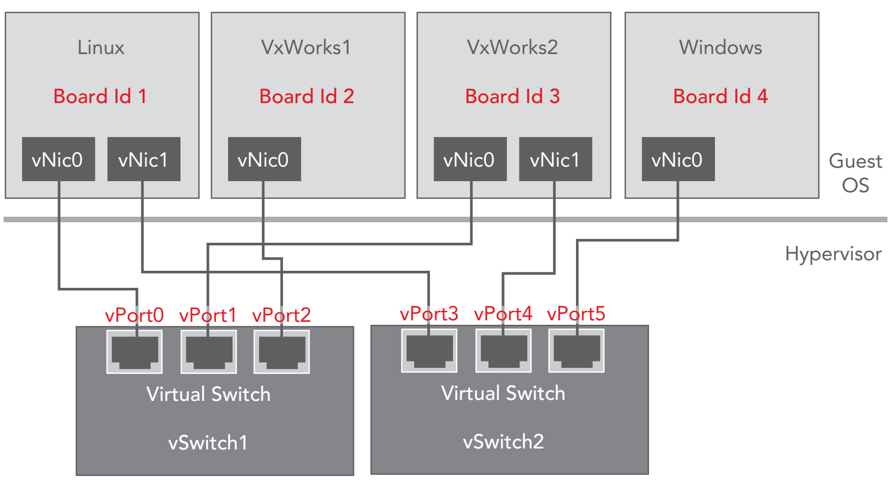

Figure 12: Virtual Layer 2 Ethernet switch

Event Driven

Wind River Hypervisor works as passively as possible. It provides separation, configures resources, and allows the guest operating systems to run at full speed. Wind River Hypervisor has no active threads. It is completely event-driven; there is no overhead or activity unless requested by the guest operating systems. Examples of events are system calls from the guest OS or a driver, interrupt delivery, and so forth.

Minimal Footprint

The footprint of Wind River Hypervisor is strategically kept to a minimum. This ensures that unnecessary code is not executed as part of a hypervisor-based system run-time, so that the hypervisor has minimal impact on system operations. With minimal code size, applications' determinism and safety are retained without impact to expected behavior. The virtualization layer is scaled down to the level where it contains only the minimal functionality required to provide its services. Any additional code, such as device drivers, is not contained inside the hypervisor but resides directly inside the guests that require it.

Scalable

Multi-core processors are making their entry into embedded systems. Top-of-the-line systems (typically in the networking space) may use up to 32 cores. This is certain to increase in the coming years. Many hypervisors and SMP operating systems will not scale up to this number of cores. But Wind River Hypervisor was designed from the ground up to be scalable, ensuring that performance is not compromised as the core count increases.

Direct Access to Devices

When developing embedded products, specific attention must be paid to local hardware device access. In some devices I/O throughput is critical; in others latency or timing constraints are critical.

IT virtualization products separate the guest from the devices they need to access. This is not so with Wind River Hypervisor. Memory, PCI attributes, and interrupts can be directly mapped into a specific guest. The hypervisor is not involved in the data path to or from the device in this case. This is equal to native performance while delivering the development and operational benefits of embedded virtualization.

OS Agnostic

Fragmentation of processor types and operating systems is common in the embedded industry, particularly across a product line. Many device developers use low-cost processors for small devices and higher-performing processors for their high-end products. Wind River Hypervisor, together with Wind River's operating systems, provides a single API and environment that can be used across many different devices. The hypervisor itself is completely OS agnostic, meaning that other operating systems, either third-party or in-house, can be added as a guest.

Development Tools

An integrated development environment is typically the cockpit that the development team leverages to build embedded systems. Wind River Workbench provides an Eclipse-based development environment that improves developer productivity for tasks such as configuring, diagnosing, and analyzing hypervisor-based systems.

Build and Debug

Workbench provides single-click system build capabilities that include the virtual boards and the complete system file. Once the build process completes, the system image can be automatically downloaded for debugging and analysis. Debugging in a multi-core and virtualized environment often requires the developer to keep track of many things at the same time. Workbench enables this through the multi-context debugger.

Workbench provides a single integrated environment to configure, build, debug, and analyze systems based on Wind River Linux, VxWorks, and Wind River Hypervisor, which simplifies a developer's workflow and increases developer productivity in comparison to switching back and forth between tools from different vendors.

JTAG On-Chip Debugging

While embedded development is challenging, debugging a virtualized embedded system is even more so. Being able to have fine-grained control over an embedded target is paramount for success. Wind River Workbench On-Chip Debugging allows JTAG debugging of drivers, interrupt routines, board bring-up, and other low-level scenarios. It also allows a developer to stop an entire multi-core chip. Often when debugging a single thread, it is necessary to stop other threads at the same time to avoid buffer overruns or underruns that would change the debug scenario. Workbench On-Chip Debugging allows the developer to stop all cores of a multi-core target with short lag time, providing an overview of the entire context.

Integration

Wind River Hypervisor is part of the Wind River multi-core software solution, integrating with the other components: VxWorks, Wind River Linux, Wind River Workbench, Wind River Simics, and Wind River Test Management.

Wind River Hypervisor supports an open and well-defined interface to virtual boards. Any operating system can be executed inside a virtual board, with the appropriate paravirtualization modifications. Wind River Workbench provides the interface to configure, build, and debug small and large hypervisor-based systems through graphical editors, elaborate build configurations, and multi-context debuggers.

MIPC can be used to communicate between the virtual boards during run-time. MIPC provides a fast, zero-copy communication channel with a socket-like API. MIPC can also be used as an extended serial connection to virtual boards that do not have access to a serial port or can be used to connect to Wind River Workbench for more sophisticated debugging scenarios.

The Larger Picture

Virtualization is one part of a comprehensive solution consisting of virtualization capabilities, operating systems, development tools, and testing solutions. Wind River has recognized the importance of multi-core and virtualization and has defined a broad and flexible product portfolio to support it.

The flexibility in the portfolio stems from Wind River providing virtualized and non-virtualized alternatives, to effectively take advantage of the multi-core processors of today and tomorrow. Wind River's portfolio allows a device developer to start, for example, with a VxWorks SMP configuration on a dual core processor and in a later device evolution migrate to a VxWorks and Linux combined solution in a supervised AMP configuration. The *same application* can also be used in a virtualized configuration, either on a single core or spread out over multiple cores.

Wind River offers the same technologies over the breadth of its product portfolio. That is what gives designers the flexibility to choose an appropriate multi-core software configuration and know it is future-proof, as processor technology and device requirements evolve over time.

Professional Services

The introduction of multi-core and virtualization provides many more choices to design teams delivering next-generation devices. But there is often a lack of experience and expertise in these new technologies. Wind River Professional Services is on the leading-edge of the multi-core and virtualization revolution and can help accelerate the introduction of these technologies while reducing risk.

Wind River Professional Services, a CMMI Level 3–rated organization, enables you to focus on development activities that add value and differentiate your design. Wind River offers industry-specific services practices, with focused offerings that help you meet strict market deadlines while keeping development costs down. Our experienced team delivers device software expertise globally to solve key development challenges and directly contributes to our clients' success.

Backed by our commercial-grade project methodology, Wind River Professional Services includes the following:

- Multi-core/virtualization architectural review
- Requirements discovery and definition
- Multi-core and virtualized board support package (BSP) and driver optimization
- Software system and middle integration
- Application and infrastructure development

Typical projects range from two to four man-weeks for driver and BSP implementation; one man-month to one man-year for hardware design or extensions to an existing software solution; and multi-man-year programs that bring customer concepts to reality through design, creation, and system test and verification.

Professional Services has extensive experience with platform design, including safety-critical systems and navigation/infotainment systems. Professional Services has implemented both hardware and software solutions for the embedded device market and continues to work with standards organizations to establish the next-generation platforms.

Installation and Orientation Services

The Wind River multi-core software solution provides a host of technologies that may seem overwhelming at first. Wind River offers Installation and Orientation Services to ensure your project starts on time and without hassle by delivering the following:

- **Onsite installation:** Guided install on your hardware and host platform, along with a sample multi-core software configuration, demonstrations, and examples of customizations
- **Hands-on orientation:** Architecture, development file system, open source packages, porting drivers, design issues
- **Advice:** Introduction to Wind River support channels and processes, additional services, project reviews, and consultation

Wind River Installation and Orientation Services will expedite your path to productivity, assuring that we have eliminated common sources of user error and helping you maximize the benefit of multi-core and virtualization.

Embedded Development Kits

Wind River has partnered with leading embedded board vendors to provide embedded development kits that enable developers to begin application development within minutes of opening the kit. They provide a technically comprehensive platform to address complex development requirements.

Each embedded development kit includes a bootable USB flash drive that immediately turns any host computer into a fully integrated development environment, with absolutely no installation required. Each board comes with a pre-flashed, 30-day run-time trial version of Wind River's VxWorks real-time operating system or Wind River Linux, along with Wind River Hypervisor.

Education Services

Education is fundamentally connected not only to individual performance but also to the success of a project or an entire company. Lack of product knowledge can translate into longer development schedules, poor quality, and higher costs. The ability to learn—and to convert that learning into improved performance—creates extraordinary value for individuals, teams, and organizations. To help your team achieve that result, Wind River offers flexible approaches to delivering product education that best fits your time, budget, and skills development requirements.

Public Courses

Wind River's public courses are scheduled for your geographical convenience. They are conducted over one to five days, using a mixed lecture and interactive lab classroom format that leverages the experience of Wind River instructors and other course participants. Courses provide a fast, cost-effective way for students to become more productive in Wind River technology.

Benefits of public courses include the following:

- A conceptual introduction that orients students to the subject matter
- A selective examination of the details, focusing on the most commonly used areas, or on areas with which users tend to be least familiar
- Personal guidance and hands-on application of individual tools and course concepts
- The chance to grasp device software concepts, as well as the fundamental issues involved in real-time design
- The knowledge needed to develop device drivers, perform hardware porting, or develop applications
- Answers to specific questions about topics addressed in the course

Consult your local Wind River sales representative for course schedules and fees.

Onsite Education

If you have a large project team or a number of new users, you may benefit from custom onsite education. Instructors will consult with you and, based on the workshop series curriculum, determine which topics should be included and emphasized. This type of education offers an opportunity for one-on-one discussions with our instructors about your specific project needs, technical requirements, and challenges, all in the comfort of your own office.

Advantages of onsite education include the following:

- Your entire team gains a common knowledge base.
- Knowledge and skills are transferred directly from the classroom to your workplace.
- Use of your location saves employees travel expenses and time away from the office.

Consult your local Wind River sales representative for further information about onsite education.

Support Services

Wind River Customer Support, a Service Capability and Performance (SCP)–certified organization, provides support for Wind River Hypervisor.

Your subscription includes full maintenance and support delivered through Wind River's Online Support (OLS) website and our worldwide technical support team. While under subscription or support agreement, customers receive both maintenance updates and major updates.

Visit Wind River Online Support at www.windriver.com/support for fast access to product manuals, download-able software, and other problem-solving resources. OLS offers a comprehensive knowledge base with robust search features for locating information quickly.

Additional support features, including proactive email alerts covering particular technologies, platforms, or product patches and technical tips for common problems, are available for all customers by subscription. OLS visitors can also access a community of developers to discuss their issues and experiences.

If you cannot find the information you need through Online Support, contact our global support team for access to the industry's most knowledgeable and experienced support staff.

For more details on our support processes including escalations and defect resolution, consult Wind River's Customer Support User's Guide at www.windriver.com/support/resources/csug.pdf.

Wind River Customer Support contact details are at www.windriver.com/support/.

North America, South America, Asia/Pacific

support@windriver.com
Toll-free tel.: 800-872-4977 (800-USA-4WRS)
Tel.: 510-748-4100
Fax: 510-749-2164
Hours: 6:00 a.m.–5:00 p.m. (Pacific time)

Japan

support-jp@windriver.com
Tel.: +81 3 5778 6001
Fax: +81 3 5778 6003
Hours: 9:00 a.m.–5:30 p.m. (local time)

Europe, Middle East, Africa

support-ec@windriver.com
Toll-free tel.: +800 4977 4977
France tel.: +33 1 64 86 66 66
France fax: +33 1 64 86 66 10
Germany tel.: +49 899 624 45 444
Germany fax: +49 899 624 45 999
Italy tel.: +39 011 2448 411
Italy fax: +39 011 2448 499
Middle East region tel.: +972 9741 9561
Middle East region fax: +972 9746 0867
Nordic tel.: +46 9 594 611 20
Nordic fax: +46 8 594 611 49
UK tel.: +44 1793 831 393
UK fax: +44 1793 831 808
Hours: 9:00 a.m.–6:00 p.m. (local time)

WIND RIVER

Wind River is a leader in embedded and mobile software. We enable companies to develop, run, and manage device software faster, better, at lower cost, and more reliably. www.windriver.com