Advances in virtualization aid information assurance

By Peter Carlston and Joe Wlad

Virtualization is a concept that has been tossed around for many years. Today's technology is moving virtualization to the forefront because of its ability to reduce the amount of hardware required to support an application. Previously, virtualization was possible with some creative Operating System (OS) use and user/privileged memory access. However, the newest multicore processors make implementing virtualization much more practical for today's compute platforms. This article looks at what new advances in Intel processors are doing to make virtualization easier to implement in security applications.

Powerful virtualization capabilities have enabled workload consolidation, where multiple heterogeneous OSs and applications run on common hardware, and workload migration, where applications built on one hardware platform can be migrated to another platform in binary form. The cohesive benefits virtualization offers are now being extended to secure environments (for example, military networks), obviating the need for physical hardware separation between secure applications and nonsecure applications. Moreover, the need for fewer physical platforms translates into substantial capital and operating cost savings.

Virtualization involves a software program called a *Virtual Machine Monitor* (VMM) that abstracts hardware to each partition called a *Virtual Machine* (VM). It also coordinates shared hardware between multiple partitions. VMMs date back to the 1970s. Traditionally, a VMM creates a virtual environment indistinguishable from the bare hardware that an OS may run on without modification. Underlying hardware capability emulation allows guest OSs to run in a hardware environment different from their original environment.

Intel Virtualization Technology[1] provides hardware capabilities that enable simpler and more robust VMM designs. It allows for total VMM separation from each VM by creating a new privileged ring structure. Intel Virtualization Technology enables unmodified OS and application migration, thus simplifying legacy software stack porting to virtualized environments. Intel multicore processors make ideal virtualized platforms by combining Virtualization Technology and the compute performance needed for running multiple VMs. Figure 1 shows a notional VMM architecture using Intel Virtualization Technology supporting three guest OSs in different partitions. The VMM operates in root (or privileged) mode, while each guest OS runs in nonroot mode along with any needed application software or middleware, such as networking, file systems, or device drivers.

One emerging Intel Virtualization Technology application addresses the area of information assurance where software and data of varying classification levels can now coexist on a single hardware platform. The architecture for this design approach is called *Multiple Independent Levels of Security* (MILS). MILS requires a separation kernel that provides not only the VMM, but also data isolation and information flow capabilities that help meet high assurance security requirements.

Figure 2 shows the MILS architecture depicting differing levels of security classification separated on a single platform. In this example, a secret application and a top secret application execute in a round-robin fashion along with a mixed application of both secret and top secret classification. Each application is supported by a complete guest OS or minimal runtime library. These applications' communication policies are defined by the system developer and enforced by the separation kernel. Any needed middleware is placed into separate partitions executing in user mode. To guarantee system security, the separation kernel and any high assurance applications are

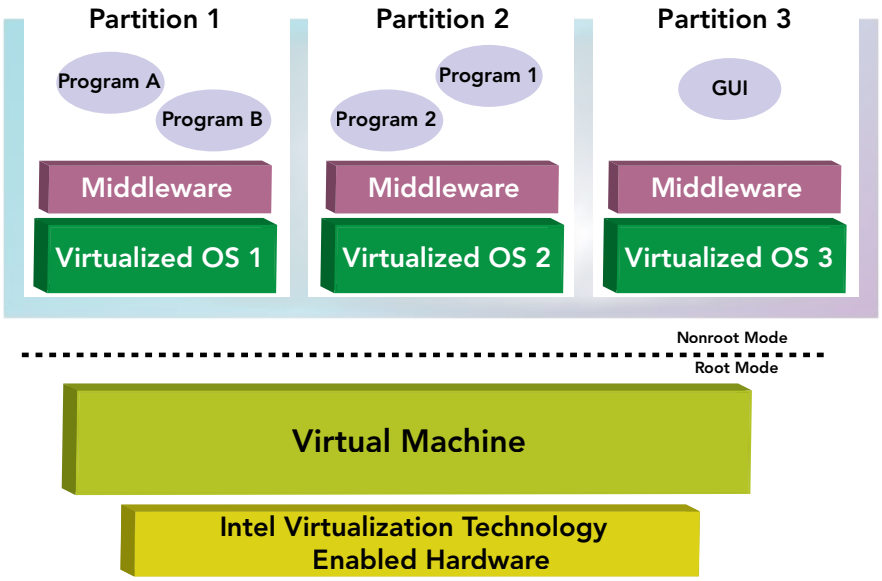

Figure 1

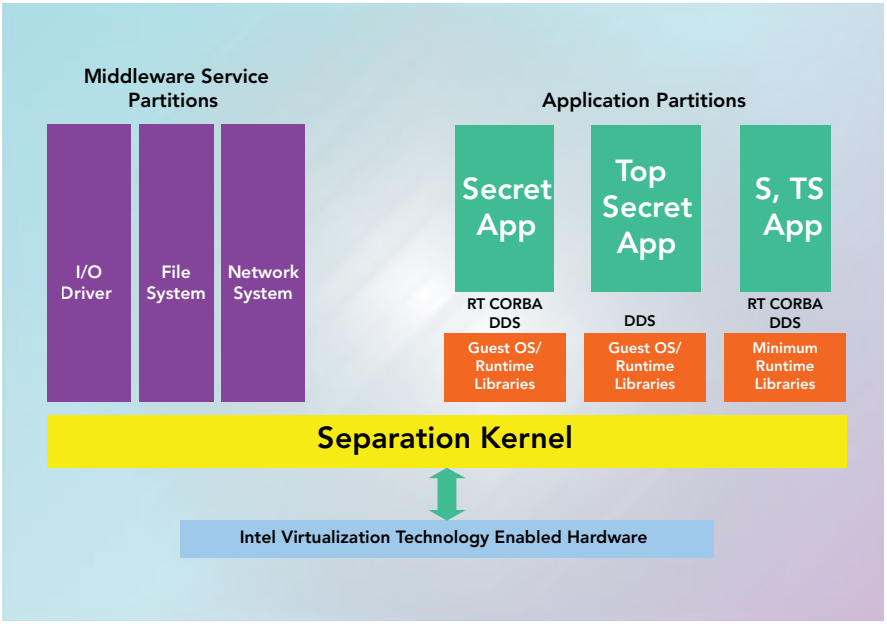

Figure 2

validated against the framework of the Common Criteria, an international standard for computer security that is beyond the scope of this article.

A COTS-based information assurance architecture

Figure 3 shows a conceptual diagram of a COTS-based MILS system. Notice that applications and guest OSs run in separate Top Secret, Secret, and Unclassified VMs. These VMs are independent from one another. A lightweight separation kernel such as LynuxWorks' LynxSecure is used to set up and control the VM environments above it.

The separation kernel allows system hardware resources to be assigned and/or shared in a controlled manner. In this instance, the separation kernel has assigned processor cores 0 and 3 to the Secret and Top Secret domains, respectively. Unclassified applications run on cores 1 and 2, respectively. Separate system memory areas (shown by four different colors) and dedicated Network Interface Cards (NICs) also have been assigned to four domains. Other devices, such as trusted file systems, network stack, and device drivers (for example, keyboards), are shared across all domains according to security policies defined by the system designer and enforced by the separation kernel. The trusted file system, for example, requires safeguards built into the separation kernel and system hardware to prevent unauthorized data access and movement between partitions.

Processor virtualization enhancements

Notice that the separation kernel now runs in a set of hardware *root mode* (also termed *privileged*) ring structures. This new set of structures has been introduced to solve a major difficulty with older-style, software-only VMMs. Historically, microprocessor OSs have always been developed to utilize a processor's Ring 0 registers. This gave the OS control over all system hardware and isolated OS functions from applications, which run only in Rings 1-3. But when software VMMs were developed later after the basic PC architecture had been developed, they had to control system hardware, so the guest OSs had to be de-privileged and made to run in Ring 1. This meant the OS itself had to be changed, for example, *paravirtualized* Linux variants. Or, if OS source code was not available, as with Microsoft or other OSs, the software VMM had to implement binary patches to the guest OSs so that the VMM could intercept and control all access to the underlying hardware. Both of these approaches often led to performance and stability issues.

Using the new root mode processor ring structure means that guest OSs can run often unchanged in Ring 0. But all requests for memory and other system resources are now intercepted by the separation kernel running in the new root mode Ring 0 hardware structures. Many current microprocessors also provide special hardware registers that VMMs and separation kernels use to store context information – file descriptor tables, memory management structures, and so on. This simplifies and speeds up switching from one process to another, so overall system responsiveness is increased even when applications on multiple guest OSs are running concurrently.

Some idea of the benefits of hardware-assisted virtualization can be gleaned from Figure 4, which shows the number of events per million instructions a typical VMM (in this case, Xen) must handle while running two standard PC benchmark applications. Notice that the number of Interrupt and Other events is tremendously reduced when processor virtualization enhancements are utilized. Notice also, though, that the number of I/O Operation events is unchanged.

Chipset virtualization enhancements

Each guest OS has its own virtualized hardware drivers it uses to request services from the VMM. Sharing I/O in a virtualized environment previously meant that the VMM had to emulate I/O access; it multiplexed all the requests from all the guest OSs to the actual I/O device(s) below it. As the left hand side of Figure 5 shows, the sequence for outgoing packets is:

1. The application makes a call to its OS with pointers to the packet(s).
2. The OS calls the virtualized driver for NIC Device "A."
3. The VMM takes the pointers to all the outgoing packets, sorts them, and passes the ones for Device A to the

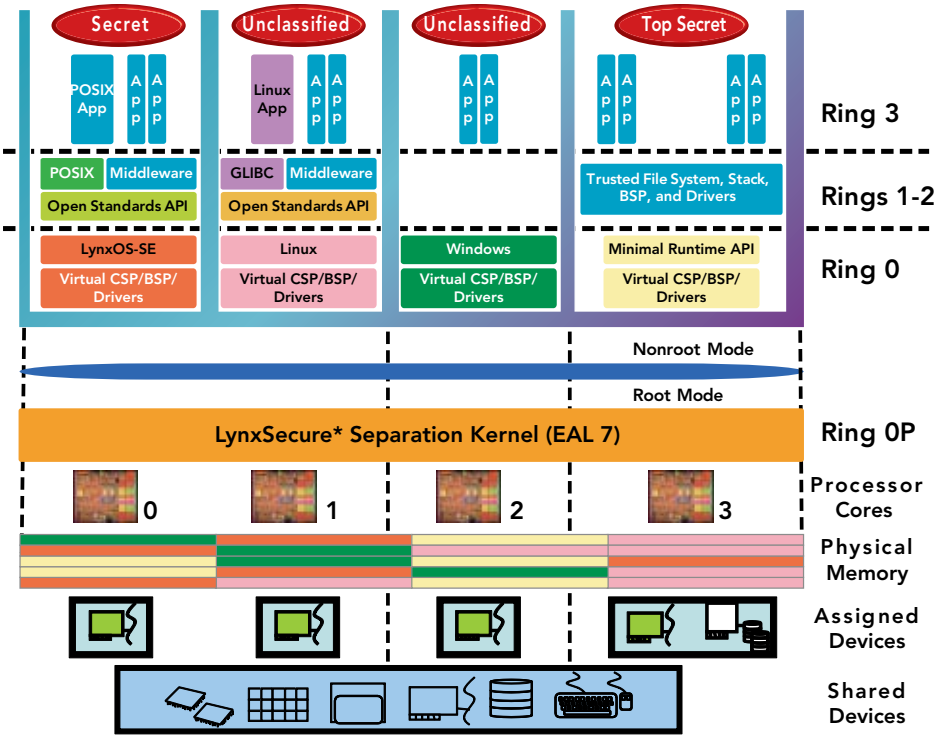

Figure 3

memory-mapped I/O area for that device, and informs that device's driver.
4. Device A's driver interrupts Device A, and the packets are sent over the wire.

The sequence is basically reversed for incoming packets, with the VMM handling the processing to ensure that packets are directed to the proper VM. In both cases, the amount of processing that must take place within the VM/separation kernel can add latency and degrade system performance.

A better solution is for the separation kernel to set up secure memory re-mapping tables directly within the processor's chipset. The trusted application's VM will still use virtualized drivers, but the calls made would be handled directly by translation tables in the chipset. So, in effect, the packets flow through the system with very little separation kernel involvement, as shown on the right hand side of Figure 5.

Any attempt to perform a memory operation outside of a device's or application's assigned memory area will fail. Diagnostic data will be written to new fault recording registers in the chipset, and a fault log in a separate protected area of memory will be updated.

Chipsets that support this type of directed I/O assignment are available. These devices substantially reduce the number of I/O events a separation kernel must handle.

Improving network performance: NIC and system enhancements

Several other system enhancements are helping increase application responsiveness by optimizing virtualized platforms' network I/O performance.

First of all, the TCP/IP software stacks in Microsoft and Linux OSs have been optimized to reduce overhead and increase performance.

Secondly, Ethernet silicon now supports enhancements to increase network throughput in a virtualized multicore processor environment. Intel Gigabit and 10 GbE controllers, for example, can be configured to use a hash table created from IP, TCP, and port addresses to map incoming packets to queues associated with a specific processor core[2].

What's more, the controller can place incoming packets directly into that core's Level 2 cache so that data is actually available before or just as soon as an operation in the processor's instruction pipeline needs it.

Data transmits also utilize similar Direct Cache Access technology. A process running on a specific core posts transmit data into its protected area of memory. The Ethernet controller then reads the data into its appropriate queue, sends it out over the wire, and causes the completion message to be pre-fetched into the L2 cache so it is available more quickly for the application.

Other enhancements such as VM Device Queues improve overall throughput and CPU utilization by allowing the NIC hardware to route packets from specific queues to specific VMs – whichever core(s) they are running on. This is useful for cases where it may be advantageous for VMs to migrate from one core or processor to another. VM Device Queues also help ensure transmit fairness and

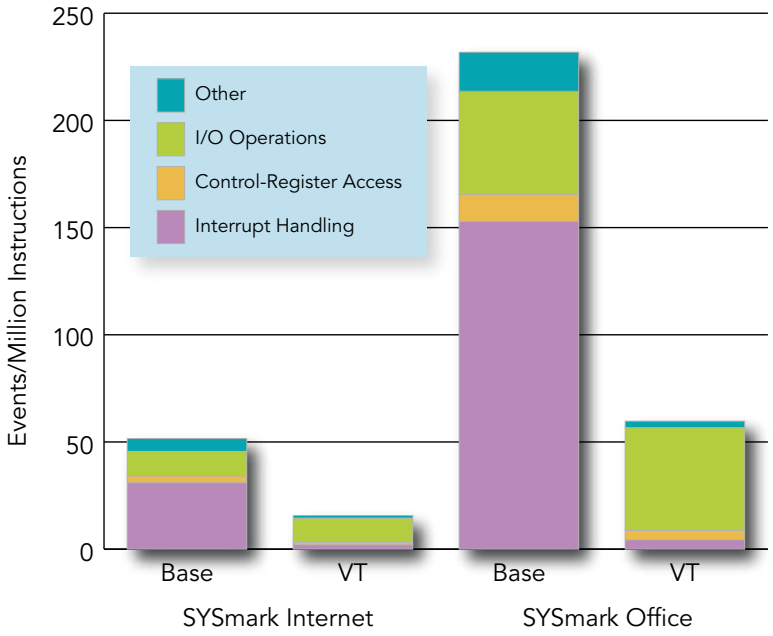

Figure 4

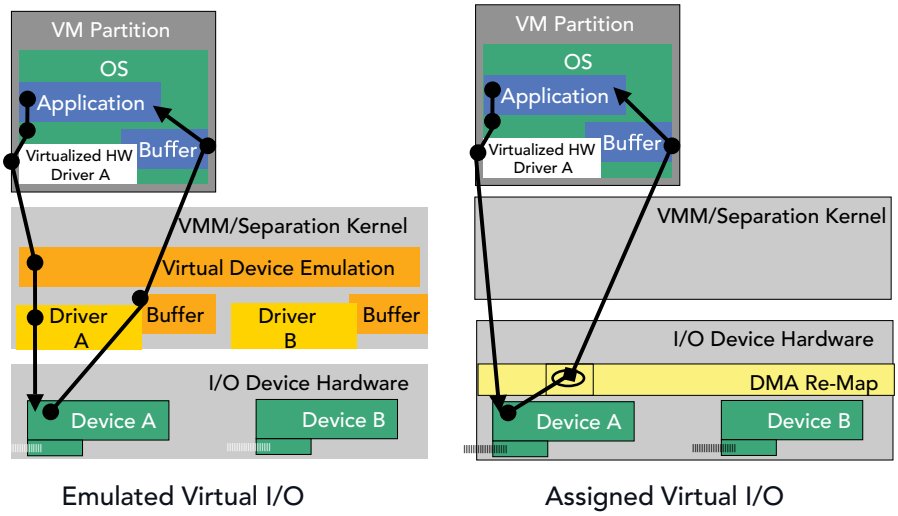

Figure 5

prevent head-of-the-line blocking.

Small, highly trusted separation kernels

These new technologies enhance the stability and reduce the size and complexity of the separation kernel itself. Separation kernels with < 5K code lines are a practical necessity since they must be mathematically verified and formally evaluated before they can be certified for use in government information assurance systems. Kernels much more than 5K code lines are extremely difficult and time-consuming to formally evaluate. It is now more feasible to craft separation kernels with less than 5K lines of code and achieve much higher levels of system performance at the same time.

Taken together, these technologies and techniques allow applications running in virtualized environments to run at near native performance. And an important side benefit of modern virtualized environments is that legacy applications do not have to be changed. Legacy, single-threaded programs, for example, usually do not have to be altered for VMs running on multicore processors.

Trusted platforms

Commercial users face some of the same problems as the U.S. defense community. For example, their high-value, mission-critical commercial business data faces threats from virus-infected drivers or applications, rogue administrators, and users who can load their own malicious drivers, pull off critical data over unsecured USB ports, and so on.

Thus, microprocessor vendors also are enhancing their processors and chipsets to support *Trusted Execution Technology*[3]. Intel Trusted Execution Technology-enabled components work together with Trusted Platform Management silicon defined in the Open Group's specification. In these systems, the Trusted Platform Management silicon provides sealed storage for encryption keys, software measurements, and configuration policies. Other system software ingredients work together with the hardware-hardened framework to enable hardware and software configurations to be locked down so that any tampering is detected and dealt with accordingly during the system's trusted boot process. Intel Trusted Execution Technology-enabled platforms have been demonstrated and are becoming commercially available from major OEMs.

The future

The good news is that some major commercial markets require secure, power-efficient platforms that support virtualized environments, creating a competitive advantage for processor manufacturers and OS vendors to improve their products' virtualized performance and security. MILS platform developers can leverage these technologies to produce a new breed of high-performance, cost-effective information assurance systems. ECD

Footnotes

[1] Intel Virtualization Technology requires a computer system with an enabled Intel processor, BIOS, VMM, and for some uses, certain computer system software enabled for it. Functionality, performance, or other benefits will vary depending on hardware and software configurations and may require a BIOS update. Software applications may not be compatible with all OSs.

[2] Intel 1 GbE NICs currently implement four transmit and four receive queues per port. 10 GbE NICs currently support 32 transmit and 64 receive queues per port, mapped to a maximum of 16 processor cores.

[3] No computer system can provide absolute security under all conditions. Intel Trusted Execution Technology is a security technology under development by Intel that requires a computer system with Intel Virtualization Technology, an Intel Trusted Execution Technology-enabled processor, chipset, BIOS, Authenticated Code Modules, and an Intel or other compatible measured VMM. In addition, Intel Trusted Execution Technology requires the system to contain a TPMv1.2 as defined by the Trusted Computing Group and specific software for some uses. See www.intel.com/technology/security/ for more information.

Peter Carlston *is a platform solutions architect with Intel's Embedded and Communications Processor Division. His primary responsibility is to ensure that Intel's current and future embedded processor products meet requirements derived from the large variety of military, aerospace, and government use cases, form factors, and work loads. Part of his current work involves developing proof of concepts for the types of signal processing applications that now run efficiently across a range of Intel's current- and next-generation processors.*

Intel
480-554-7295
peter.carlston@intel.com
www.intel.com

Joe Wlad *is director of product management at LynuxWorks, Inc., where he is responsible for all safety-critical products in the industrial, medical, and aerospace markets. Joe has more than 23 years of safety-critical systems design, development, test, and evaluation experience. He holds a BS in Aerospace Engineering from the University of Buffalo and an MBA from Santa Clara University.*

LynuxWorks
408-979-4411
jwlad@lnxw.com
www.lynuxworks.com