System Development & Debug

Is it real, or is it virtual?

Virtual platforms can offer significant productivity gains, but what exactly are they and how do you use them? By David N. Kleidermacher, Chief Technology Officer, Green Hills Software

Virtual computing, virtual machines and virtual prototypes are terms that, while representing important and promising technologies, are often misused and misunderstood. Much of the confusion is caused by the application of similar technologies to achieve vastly different goals.

This article aims to provide an overview of the major virtualisation initiatives while presenting a way of thinking and talking about virtualisation that will demystify the concept, for both computing professionals and interested non-professionals.

Virtualisation involves the creation of a virtual platform that mimics a true system. A virtual computing platform (also known as a virtual machine) is capable of hosting the same software environment that runs on the true hardware, with the software remaining generally unaware that it is executing in a virtual environment. By encapsulating the underlying physical implementation, virtualisation provides opportunities to modify, duplicate, or otherwise transform the environment seen by software.

Virtual computing technology has been around for a long time, but recent evolution of desktop and server hardware platforms has made the use of virtualisation practical for a variety of important activities, including software development and simulation, server consolidation and provisioning, and a variety of compelling hybrid computing architectures. To help simplify discussion, we propose to use the following naming scheme (note that these names are not claimed to be wholly original) for the aforementioned major application concepts of virtualisation: virtual IT (information technology), virtual prototypes and virtual hybrids.

Virtual IT

The majority of press coverage, investment, and worldwide interest in virtualisation today is aimed at the first of these; virtual IT.

VMware (now part of EMC Corporation) has made virtual IT into a big business. Virtual IT, powered by virtual machines, enables multiple operating environments (Windows, Linux) to run on a single hardware platform (Figure 1) with the goal of improving the flexibility and availability of IT resources.

With high powered server computers, multiple server functions running on typical server operating systems – such as Windows Server and UNIX – can be consolidated onto a single platform using virtual machine technology, that turns the server operating system into a guest of the underlying virtual machine software.

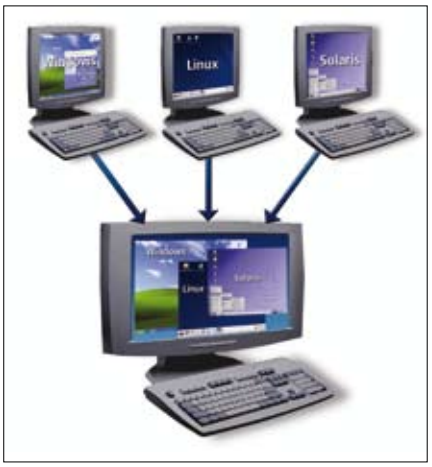

Fig 1: ***Virtual IT enables multiple operating environments to run on a single hardware platform.***

Failures in a server may be handled by restarting or moving a guest instance to another computer, with minimal impact on downtime.

Outside of server applications virtual IT provides flexibility, of which consumers are taking increasing advantage. One example is the use of the Parallels virtual machine technology to run Windows alongside the native Mac OSX environment, running natively on Intel-based Apple desktops and laptops. This allows Apple fans to use (when necessary) the Windows environment without requiring a separate PC.

On contemporary PC platforms, another use of the virtualisation moniker may add to the confusion: Intel's Virtualisation Technology (VT).

Given the name, one may think that PCs containing Intel VT provide a built-in virtual machine of the same class as VMware, but that is not the case.

VT (more recently called VT-x to denote virtualised execution) is a set of hardware acceleration capabilities, added to Intel Architecture chips and chipsets, that make it easier for virtual machine software (such as Green Hills' Padded Cell, VMware, and Parallels) to provide a fully virtualised PC platform.

Within such a platform, one or more guest operating systems – for example Windows, Linux, or Solaris – can execute, unmodified, with good performance on top of the Intel hardware/virtual machine software platform.

This concept of 'hardware assisted virtualisation' is not unique to Intel; IBM has designed hardware assisted virtualisation support for its Power Architecture-based server computers, and AMD has the similar Pacifica technology. The ubiquitous availability of this technology on standard PCs, made possible by Intel and others, is helping the world to realise a much wider range of virtualisation applications. With the advent of hardware-assisted virtualisation, the software has become simpler. Subsequently, some hardware manufacturers have introduced the term 'hypervisor', to distinguish hardware-accelerated virtual machines from the pure software type.

Unfortunately, this has caused confusion; the different name would seem to imply a potentially different function, so it is perhaps more helpful to simply replace the term hypervisor with virtual machine.

Virtual prototypes

One of the main goals of virtualisation is to use it in instances where the actual environment is unavailable, due to lack of hardware for example.

Virtual prototypes aim to provide this environment, to enable early development of software that will ultimately run on the real hardware. Because the virtualisation need only support the software prototyping function, the accuracy of the virtual machine does not necessarily need to match that of the hardware platform.

Virtual prototypes only need to support those aspects of the software application that are being developed and instruction set simulators certainly fall into this category. A simulator can range from simulating just a CPU core, to all of the peripherals that make up a complete system on chip (SoC) or board design.

Often, the software simulator provides a plug-in interface, enabling developers to add custom peripheral simulations, in cases where the virtual machine does not support a particular device.

In recent years, instruction set simulators designed for embedded software application prototyping have reached a new capability level, due to advances in virtualisation software technology and the speed of off-the-shelf PCs that execute

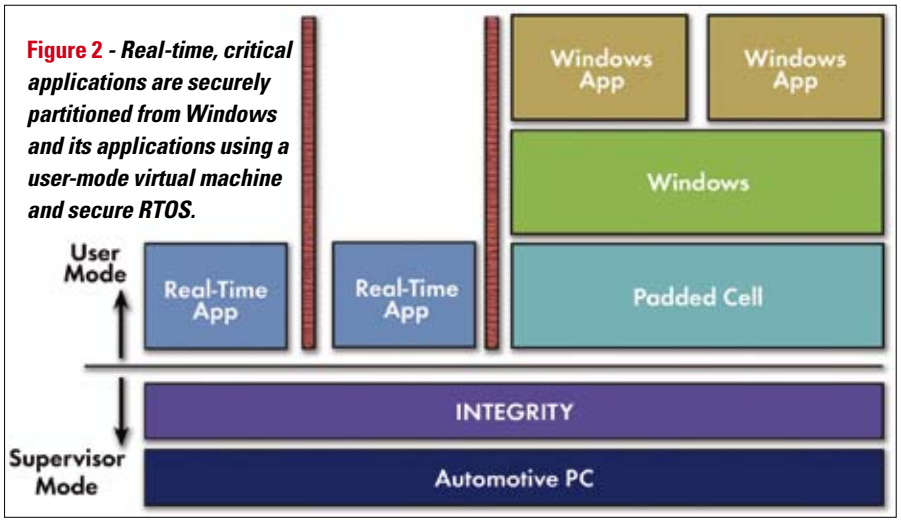

Figure 2 - ***Real-time, critical applications are securely partitioned from Windows and its applications using a user-mode virtual machine and secure RTOS.***

the virtual prototypes.

Similar virtualisation technologies used to support virtual IT platforms are now also used to make virtual prototypes faster. This in turn enables software developers to simulate more complex applications, in some cases the very same complete system that will run on the physical hardware target.

A number of companies, including Green Hills, Virtutech, Virtio and Vast, now supply these high speed virtual prototyping environments to embedded software developers, enabling them to develop and integrate software before hardware is available or when it is in short supply, saving time to market.

Virtual prototypes have also long been used for hardware prototyping. In this realm, virtual prototypes enable hardware developers to execute a model of the eventual hardware product – often written in a hardware programming language such as VHDL or SystemC – enabling the model (and sometimes the software running on it) to be simulated and debugged. This is a necessity for practically all hardware development; the cost of ASIC production is now high that virtual prototyping is required to ensure that the hardware design has no flaws, which would require an expensive respin.

A confusion we see today is that both the vendors of hardware modelling environments and software simulation environments have adopted the term virtual prototype to describe their products. This confuses customers and the media, since the two technologies target different markets and goals, with different execution time characteristics.

Confusion could be alleviated, by referring to them as either hardware prototyping tools or software prototyping tools. In some cases, a single tool (or a combination of two integrated tools) can provide both hardware modelling and software simulation. These could be referred to as [hardware/software] co-prototyping tools.

Virtual prototypes can record the execution of the machine and then use this recording to provide a virtual debugging environment that allows software developers to move backwards as well as forwards in time. A software developer can set a watchpoint on a memory location and then run the debugger backwards, to find when that location was corrupted. This new class of time machine debugger can provide a compelling productivity advantage over traditional unidirectional debuggers.

Virtual hybrids

The final class of virtualisation technology involves the combination of one or more virtualised or guest operating system environments, with software applications that run directly on the real hardware. To understand the great variety of applications for this, we provide the example of the future in-car infotainment system.

Microsoft Windows XP has found its way into some high-end infotainment systems, such as the BMW 7 series. This is accomplished using two computers, one for the traditional audio and navigation functions in the front of the car (called the head unit), and the second a PC housed in the back of the car. With Windows in the car, the rear seat passenger now has access to a familiar office environment, including electronic mail, office applications such as spreadsheets and word processors, and an Internet browser. Although BMW has proven that desktop Windows can be used in an infotainment system, we do not have proof that Windows can be used effectively in low or even mid-range infotainment platforms that cannot bear the cost, size, weight, and power overhead of two separate computers.

Traditionally, head units have relied on real-time operating systems to provide instant-on access, real-time response for audio and other multimedia functions, and bullet proof reliability (a necessity for the car due to the prohibitive cost of recalls). Windows is unable to meet these requirements. Therefore, in order to add Windows rear seat functionality to a single computer, for providing infotainment applications while also serving as the head unit, designers could employ virtualisation in a way that guarantees absolutely secure partitioning between the head-unit functions and the rear-seat Windows environment.

A powerful hybrid solution involves using a secure, real-time operating system to run on the bare hardware, and a virtual machine that runs as an application on top of the real-time operating system. One example of this architecture can be found in Green Hills' INTEGRITY PC technology. In INTEGRITY PC, the separation of the guest (one or more) from real-time, critical applications is provided by the INTEGRITY real-time operating system. The virtual machine component is Green Hills' Padded Cell technology, which provides a platform virtual machine for the x86 and can run Windows and other x86-based desktop environments.

Unlike other commercial virtual machines, such as VMware, Padded Cell runs completely in user-mode along with the guest (Windows) operating system and its applications. Therefore, none of Windows, its applications or the virtual machine (itself a complex piece of software) can affect the stability of the real-time, secure components that run directly on top of the real-time microkernel **(Figure 2)**. Audio and video applications running under control of the real-time microkernel are guaranteed to perform flawlessly.

Paravirtualisation

The virtual technology discussed earlier concentrated on running multiple operating systems unmodified on a single platform. Another approach is paravirtualisation, where the guest operating system is modified in order to improve the ability of the underlying virtual machine to achieve its intended function.

Although paravirtualisation provides some performance optimisation opportunities that traditional platform virtualisation lacks, the downside of using customised guest operating systems can outweigh the benefits. There could be too much overhead required to maintain customised versions of guest operating systems and device drivers, that are continually going through release and upgrade cycles. For this reason, paravirtualisation has more utility on architectures that lack hardware-assisted virtualisation.

Virtualisation is enjoying a renaissance in computing, providing compelling advantage in many disciplines. This adoption has spawned confusion due to lack of a consistent nomenclature. A top level classification of virtual computing applications breaks virtual computing into three main categories that focus technology on its target application: virtual IT, virtual prototypes, and virtual hybrids. Within these categories, we need to further distinguish technologies: platform vs. paravirtualisation; hardware prototyping tools vs. software prototyping tools. If nothing else, we simply need to be more careful when discussing virtualisation to make sure all parties are on the same page. One thing is for certain, the future of virtual computing is bright. <Ends>

www.ghs.com