The Virtual Test Lab

Peter S. Magnusson, Virtutech

Virtualization can ease software development in an increasingly interconnected world.

Systems companies today face increasingly complex and expensive testing challenges that will require innovative solutions.

Historically, the only option for comprehensively testing an electronic system required first building the system and then running it in a test lab filled with racks of equipment. In large companies, these test laboratories cost hundreds of millions of dollars each year in capital equipment and maintenance.

Embedded systems companies now confront these same challenges: Systems have grown rapidly, and testing their software content in real-world environments has become more complex and difficult.

Faced with these increasingly prohibitive constraints, many companies are considering alternative approaches to building the necessary test labs.

One approach to addressing this problem uses full-system simulation with virtual hardware early on and throughout the development process. By doing so, a systems company can move toward replacing hardware test racks with virtual racks, thereby employing a significantly less expensive and unwieldy solution. Moreover, full system simulation on virtual hardware creates a more productive test environment.

Currently, system-level design is undergoing a major transformation. As processors get faster, developers are implementing a larger portion of electronic and embedded systems in software. This has been the case not only for products in data centers such as enterprise servers, routers, and switches, but also for a wide variety of equally complex systems, including avionics, automobiles, cell phones, and videogame consoles.

Sophisticated software development has become the focus in many electronic systems. For example, automobile warranty costs related to software problems have increased dramatically in recent years. Software developers now face unforeseen pressures as project costs, schedule risks, and even end-product safety fall on their shoulders.

TESTING CRISIS

Software's role in providing the critical path to shipping the product has exposed the inefficiencies and impracticality of traditional development methods.

Except when developing PC software on a desktop platform, software testing requires an indirect approach. One conventional method involves testing the software on a PC with specially created test scaffolding. This approach provides severely limited accuracy, however, especially considering that it cannot replicate the real-time operating system. Software designed for a MIPS-based set-top box, for example, but cross-compiled on a PC with test scaffolding, typically experiences many hardware and software integration problems in the product launch's final phases.

The inaccuracies that arise during PC-based testing frequently force companies to employ real hardware instead. Understandably, many developers would prefer to test software using the exact object code that will ship with the final product, which usually means using the actual hardware as well.

But working with hardware raises a new set of challenges. For one, the hardware might still be in development. Often, software developers must wait months before getting access to hardware prototypes. Additionally, hardware is expensive. Companies cannot afford to supply a $500,000 piece of equipment to every engineer working on software development. Forced economies leave some developers without essential hardware, which lowers their productivity.

The increasing interconnectedness of systems aggravates testing difficulty. Today, systems rarely work in isolation. Problems that show up late in development or even in the field after deployment usually involve complex interactions between the system and its environment. This exponentially multiplies the software development challenges and further drives the requirement for extensive test hardware to recreate at least some of the environment in which the system will run—an often extremely expensive and unwieldy proposition.

VIRTUALIZATION TO THE RESCUE

Despite these challenges, real hardware has been the only trusted method to test software for critical applications—until now.

Recent technological advances in virtualization now satisfy the demands of software development, enabling the construction of a software model of the complete system that can run on the developer's desktop PC. Virtualization is so accurate it can run the fabled *golden code*—the actual binary that ships in the final product.

Because developers base the virtual platform on a system specification, the software must be tested on the real hardware before deployment. But during the development and debugging phase and most of the QA phase, software can be tested on virtual models.

Thanks to great improvements in hardware-system modeling technology, virtualization has transformed into a viable and attractive solution during the past few years. Just-in-time compilation techniques and time compression let simulations run at peak speeds as high as billions of simulated instructions per second. With virtualization, performance has become high enough to make running a full software load of applications practical. Indeed, in some cases, the simulation executes faster than hardware.

Amazing benefits

Companies can now extend the benefits of virtualization to create complete test racks. By building a virtual test laboratory, companies can simulate not only the system being tested but also the other systems it interacts with.

Virtualization at this level provides amazing benefits, including a significant time savings because virtual models can be engineered before hardware exists. Although creating the virtual platform requires writing models of the underlying hardware, the requirements can be satisfied more easily than in the actual hardware design because it isn't necessary to understand the hardware internals.

With virtualization, developers need only programmers' reference manuals to model even complex hardware. This means that software and hardware development can overlap. The planned time to market can be advanced by months, giving the company a tremendous business advantage and increased profitability.

A virtual test laboratory can simulate not only the system being tested but also the other systems it interacts with.

Reducing the quantity of hardware required offers another significant and tangible cost savings. Virtual platforms can run on any hardware, including a relatively inexpensive commodity-level PC with significantly lower capabilities than the system being simulated. That off-the-shelf PCs continually improve while becoming less expensive only increases virtualization's benefits.

Large systems companies with an extensive heterogeneous test environment of server farms and room after room of test racks can now replace some of their unwieldy hardware with virtualized models. This makes it possible to model millions of dollars of hardware using only thousands of dollars of personal computing power.

A virtual test laboratory also lowers marginal costs while still creating value. A virtual test rack scales like software. Replicating whole virtual test racks costs only a slight fraction of the prohibitively expensive cost of replicating real hardware test racks. This means that every engineer in the company can use a virtual test rack. This scalability significantly reduces capital requirements and operational expenses.

Quick and painless debugging

Companies employing full system virtualization also benefit from advanced debugging and inspection abilities that would never be possible in hardware, particularly in real-time embedded systems. Debugging real-time systems involves overcoming two main problems: the difficulty of stopping the entire system safely and, after stopping the system, the difficulty involved in recreating bugs because the system's state has changed.

Virtualization's deterministic and nonintrusive qualities, however, give software developers complete control over real-time code. The simulation infrastructure underneath a virtualization platform manages time, which lets a developer single-step or stop the system to examine internals. This remains true even in a multiprocessor or networked environment.

Imagine a third-generation base station that contains dozens of processors and has a bug. With real hardware, a developer must rely on open-ended trial and error to diagnose the problem. With virtualization, the developer can stop the entire system at any point, freezing the complete state. Given that developers synchronize all processors in a virtual hardware system, testers can then advance the system in single lock-step until they identify the problem.

Virtualization can also address problems stemming from interactions between different products or multiple instances in the same product. Solving these problems, which are often among the most difficult encountered in real hardware, requires intrusive debugging tools. These tools perturb the system so that the problem no longer manifests itself, even though the testers have done nothing to solve the problem.

When bugs depend on subtle timing interactions, repeatedly running the full system will still fail to isolate the problem. Virtual hardware models are deterministic and therefore remove this problem of repeatability. Once the problem has been seen once, testers can reproduce it on demand for further investigation and diagnosis.

Because virtualization runs faster than real time in lightly loaded systems, software developers can quickly test how the system will react after weeks or months of intermittent use. A technique called *hypersimulation* or *time compression* allows the system to be run over one hundred times faster than real time, allowing several months of runtime to be simulated overnight.

For companies dealing with expensive and unmanageable testing and debugging processes, the virtual platform offers an increasingly attractive option. Using virtualization throughout the development process will improve quality, time, and cost savings. Fast and accurate, virtualization can change how companies approach development because a virtual platform's debugging capabilities far exceed those available with real hardware.

As software becomes an ever larger component of complex electronic and embedded systems, and as it provides an ever greater capability to simulate models of these systems on commodity computers, all trends point toward virtualization software moving forcefully into the mainstream. ■

***Peter S. Magnusson** is the founder and CEO of Virtutech. Contact him at psm@virtutech.com.*

Editor: Wayne Wolf, Dept. of Electrical Engineering, Princeton University, Princeton NJ; wolf@princeton.edu

How to Reach *Computer*

Writers

We welcome submissions. For detailed information, visit www.computer.org/computer/author.htm.

News Ideas

Contact Lee Garber at lgarber@ computer.org with ideas for news features or news briefs.

Products and Books

Send product announcements to products@computer.org. Contact computer-ma@computer.org with book announcements.

Letters to the Editor

Please provide an e-mail address with your letter. Send letters to computer@computer.org.

On the Web

Explore www.computer.org/computer/ for free articles and general information about *Computer* magazine.

Magazine Change of Address

Send change-of-address requests for magazine subscriptions to address.change@ieee.org. Make sure to specify *Computer*.

Missing or Damaged Copies

If you are missing an issue or received a damaged copy, contact membership@computer.org.

Reprint Permission

To obtain permission to reprint an article, contact William Hagen, IEEE Copyrights and Trademarks Manager, at whagen@ieee.org. To buy a reprint, send a query to computer@computer.org.

Innovative Technology for Computer Professionals
Computer