



VIRTUTECH CUSTOMER PROFILE: HIGH-END SERVER COMPANY

JUNE 2002

INFO@VIRTUTECH.COM

WWW.VIRTUTECH.COM



BUILDING LARGE SERVERS

Virtutech's customers come in many sizes, from the academic research department and small start-up to the largest corporations in the computer industry. This is the story of how one division within one of the major computer hardware and software companies accelerates its products time to market with Simics.

The division has responsibility for all the platform-dependent hardware and software pieces in a high-end server. Each individual server contains from a handful to many dozens of processors. The servers may be clustered together, creating massive structures able to tackle the toughest industrial-strength problems.

The division's responsibilities include the design and implementation of numerous ASICs that support memory and caches, I/O buses, and high-speed system interconnects. They do not produce the processors themselves, but their chips support communication among processors and between the processors and other system resources like memory, storage, and networks. They also write and test all the boot, management, and device support software needed to make the systems run.

These tasks are so complex that each generation of the server architecture has previously taken four or five years. Simics was first brought in to provide a development platform when physical ASICs and working prototypes are not available. The old way of producing platform-dependent low level software was to perform the major development, debug, and test once real platforms were available.

That started to change with the last generation of the servers. Then, the lead engineer of the boot PROM team realized that if he could gain access to an environment that looked like the real hardware, months in advance of real hardware availability, the parallel effort would pull in the whole schedule. He discovered that Simics would give him that capability. So they set to work developing Simics modules representing the software-visible part of the ASIC

designs. Then they wrote the supporting software. With that parallel preparation, bringing up the first system with new hardware was finished in hours, not months.

Now, in the current development cycle, company management has further directed the operating system, dynamic configuration software, system management software, and quality assurance groups to embrace the Simics way.

Simics also provides them a highly effective test environment for validation of the ASICs themselves, about which more later.

One comment by the central manager of the business relationship between the customer Virtutech says it all: “If I were to take a guess, we’re probably looking at nine months to a year in development time savings than if we were not given access to a simulation environment to develop software.”

Nine months to a year out of a four or five year product cycle. Simics delivers time to market.

Acknowledgments

Since the computer industry is so competitive, especially within the world of high-end servers, the customer understandably prefers that their identity not be shared. The relationship with Virtutech began four years ago.

We spoke with four managers and staff members having key responsibilities for supporting and applying Simics throughout the company. They kindly took time out of their schedules to talk with us for about five hours total, for which we thank them gratefully. Owing to space limitations, this report will review only the most essential points.

The Company

The company is one of the primary providers of Internet and business data center servers, server clusters, and related technologies. It is a multi-national company with an engineering and business presence worldwide.

The Product

The division designs and builds high-end servers ranging from eight CPUs on up to one hundred in a single machine. The servers can be grouped in clusters

including as many as eight-hundred CPUs. All platform-dependent software is their responsibility as well.

They design all the ASICs that are not the processor itself. These include memory control, system backplane, and all I/O interfaces. One of the most ambitious chips was an interprocessor communication ASIC providing communication orders of magnitude faster than 100MBit Ethernet with much lower latency. It provides the capability to create one operating system image across chassis. This chip was the central focus of the earliest Simics models being the first implemented as a simulated module.

The Case for Simics

Delayed product introduction misses revenue opportunities as the product's design ages and becomes less competitive. Therefore, there is strong economic motivation to introduce new products as early as possible. Since Simics offers a way to accelerate software development through parallelism with hardware development, that acceleration is the main justification for the expense of Simics.

The primary technical attributes of simulation are availability, stability, and functionality.

Simulated systems can be available long before new hardware. When an accurate simulation is available, waiting for system power-on to test your software is unnecessary. At the time the boot PROM engineers went looking for simulation solutions, the real hardware was not going to be available for six or seven months. Then they discovered Simics. It solved the time-to-market problems and others they hadn't foreseen at that time.

Simulated systems are stable. Even when lab prototype systems are available, they are usually found in quite small numbers, especially on the scale of the systems discussed here. This forces development teams to share the available hardware among many teams, even around the world. Each use of the system causes perturbations like new hardware modifications, new firmware, or new component configurations. Most of a test time slot can be absorbed just by putting the test system back the way you need it, an inefficient process.

Immaturity of the hardware also contributes to delays in product introduction. Early versions of hardware lack many of the features that mature hardware

will include. Simics offers virtual hardware with as many features as have been programmed into the extensible modules.

Adopting Simics also brought other advantages. In large companies, there are often a variety of internally-developed tools that might be pressed into service. Simics was much more attractive as a commercial product since the customer could pay for support rather than rely on the goodwill of sister organizations within the company.

TWO IMPORTANT APPLICATIONS

Simics supplies virtual hardware for a low-level software development platform in new high-end server system designs. It also drives the highly critical process of validating the design of the custom ASICs that make up the new servers. This section describes both of these efforts.

Developing Software before Hardware Arrives

The business case for accelerating software development in advance of new hardware availability was presented above. Here, the main technical points will be reviewed.

Scope of the Problem

The scope of the problem may be measured in several ways.

First, there is the amount of time involved in the development. From project inception, hardware may be unavailable for a year and a half or more. Without a useful test platform, progress on software development will be slow and incomplete.

Second, support software development can itself be a lengthy process. These large servers aren't booted with a simple PROM program. Developing these management agents may take a couple of years since the effort depends on reading and interpreting the complete specifications of the hardware being managed. Since the design is in flux in the early stages, these codependent subsystems take a long time to evolve.

Third, there are many groups that depend on access to hardware to complete and validate their work. These include the service processor for machine configuration, management software, fault tolerant and Reliability/

Accessibility/Serviceability software, device drivers, operating system kernel, dynamic reconfiguration software, and performance testing. The multiplicity of these groups magnifies the competition for test systems that may become available. It also amplifies the positive effect of any reduction in project time.

Simics can significantly reduce the exposure of a project to the uncertainties inherent in hardware development of complex systems. By providing access to mature, fully-functional virtual hardware from the earliest stages of a server development cycle, everyone benefits, especially the end user.

The Testing Challenge

One of the driving forces shaping the testing of high-end servers is their rarity during development. Prototype development machines may cost a million dollars apiece to build even though the system's final target cost may be one fifth of that. Therefore, there are few provided for testing.

For example, during the previous generation's development, only two test machines were available. Developers would sign up for two-hour blocks of access time. Since these test systems were in use twenty-four hours a day, anyone assigned a time slot had better use it, whenever it occurred. The prospect of living on a three-shift schedule motivated the boot PROM team to look for more efficient ways to use their time.

Boot PROM Leads the Way

The boot PROM sits in the middle of the release process, since all the higher levels depend on it. The boot PROM team searched the industry for solutions to this problem, and found Simics. To apply it to the bring-up of these larger servers, however, required the construction of simulation modules for every software-visible chip. Before the command "boot" could be issued, the map of the complete device tree had to be constructed.

The Programmer's Reference Manual, or PRM, is the public definition of the interface and functionality of the chips. Though the central ASIC of interest was the new interprocessor communication chip, every component in the server visible to software had to be represented for the simulation model to be useful.

For the new chips, the available specifications were in a very preliminary state, so most of the time was spent trying to figure out how the hardware

works. Implementing the modules in Simics was straightforward, however, once the chip design was understood. After development, the customer's modules were folded into future Simics releases by Virtutech, making them available to all the teams.

Simulation Significantly Speeds System Bring-up

Experience with Simics simulations taught the boot PROM team what should be happening, accelerating debugging. Initial bring-up of the first system took only four hours since the software was almost ready owing to Simics work.

Since the early versions of management software knew how to deal with one machine only, bringing up two machines was a challenge. The boot PROM team compiled custom versions of the management software for both machines so that the memory was mapped properly, then more custom software joined the memory together into a big virtual system. Simics made this approach feasible since the custom versions could be tested before use on the real hardware where visibility into the system is limited. With that preparation, the first clustered system came up in just ten days, a record at that time.

Reliability / Servicibility Applications

The outlook of the Quality Assurance (QA) team differs from the development team. Development engineers typically are quite satisfied to create just one working system. QA people, on the other hand, worry about how multiple server systems cooperate with each other. They care about booting one CPU board in slot 0, then in slot 2. Later, they might check slots 1 and 5, and so on.

Handling exceptions properly in all circumstances is central to what the QA team is all about. What should happen when an ECC exception is generated, and the ECC hardware can't take care of it? Testing the system's response to these conditions is difficult, since, in a physical system, hard errors cannot be reliably triggered. So when a hard error occurs, without having tested the software's response, most likely the kernel will hang, and the developer will have no visibility into the causes.

Simics enables the software developers to fabricate adequate software responses to almost any hardware problem. It significantly broadens the domain of fault testing by providing a mechanism for hardware fault injection

on demand, enabling testing of the way those error conditions are handled. With preparation, a wide range of hardware errors may be found to be correctable.

Indeed, the QA people are now totally dedicated to approaching this type of problem with simulation. To support the QA effort, the engineering team are presently creating a fault-injection test harness for adapting Simics models for testing. Fault injection is certainly one of the features that we're looking to utilize heavily on the coming version too, the manager said.

Seeds Planted in This Generation Bear Fruit in the Next Generation

Investing in Simics models of last generation's systems has laid the foundation for an evolutionary approach to development. Trying another MMU works just by plugging in a new MMU model. No soldering is required for using a different CPU. All of a new architecture does not have to be implemented in the model right away. Efficiency in development increases with each generation.

Several groups will reap significant benefits from the ground laying efforts of the boot PROM group. For example, the operating system can explore all machine-dependencies and tame them one by one, by introducing new hardware chip by chip. Currently, they are developing software using the simulation basis. As one of the Simics support team said, "They're happy. They're excited".

Now all the ASIC groups can check their chip's performance and validity in a full system context much earlier in the design cycle. Once the hardware designers supply a usable PRM document, they have effectively released the chip to the full development community. This is truly a revolutionary development.

ASIC Verification: Proving the Design

Simics also tackles the problem of testing the compatibility of the full set of custom ASICs built for the highend server.

The Problem

The problem is that the functionality of the real implementations of the various chips has to be validated as correct for any circumstance.

The assumption is that normal cases will work correctly since the lowest level tests of individual chip structures will probably find most normal problems. Typically the first step in validation will be module verification that looks at a small piece of an ASIC like a buffer. Since the buffer doesn't have so many states, testing can exhaustively cover all of its behavior. At the ASIC level, the process becomes one of testing corner cases, unusual conditions, where chips interact. This is the only aspect commonly studied in verification.

In validation testing, then, the chips are represented as RTL modules, taken as completely accurate. The challenge in the exercise then becomes a question of how to drive all of these chips together with just the right corner case tests that prove the functionality.

It would be possible to use the full RTL implementation of the CPU to drive the bus transactions. But, this is not attractive since an RTL model of a CPU is extremely slow.

Another approach is to use a live system to produce a trace of transactions that can be run through the RTL models in turn. But this isn't adequate since a live system does a few things over and over again. It doesn't capture corner cases, for example, where an error on a bus might occur in just the time period when a particular processor is resetting.

Validation cannot be exhaustive, so testing has to target corner cases and look at possibly problematic conditions, and unusual states in the cache coherency protocol. The tests need to be crafted and hand written just to excite those problematical conditions.

A final alternative would be to drive the memory-transaction-based interface, performing particular patterns of read and write. The problem with that approach is that the interface is too low, that too many operations would have to be organized to produce an accurate sequence of interactions. A much better way is to drive these interactions from a high level, much more efficiently than the RTL version of the CPU can do. In that way lies Simics.

The Solution

The customer has solved these problems by using Simics to drive the memory transaction patterns that test corner cases. Simics validates ASICs by executing just the right patterns written in assembly code to excite those corner conditions in the memory subsystem. For example, the test program is crafted to create a lot of cache conflict, not just random events, to exercise the protocol. Test cases know the line size, tags, protocol, all part of the knowledge that goes into creating the right tests. By working at the semantic level that really counts, Simics produces a better high level picture so that the design matures much faster.

As opposed to the possibility of simply driving memory transactions randomly, a major benefit of the Simics approach is taking a C program and running it on the ASIC's RTL. One doesn't run the OS, just a little assembly language program that runs as a kernel running stand alone. The test starts, running the processor in privileged mode, and doesn't use virtual memory. To run this way, the portability of the source assembly code is important.

"We use Simics as a replacement for the RTL of the next generation processor", said a validation test engineer. "Don't run RTL everywhere: just replace the processor RTL model with the Simics model, which is much more convenient in many cases."

In practice, the total length of the test is on the order of thousands of instructions crafted according to the carefully specified patterns. Instead of taking a trace, Simics runs just those special memory instructions that drive the whole memory subsystem.

Verification in Practice

After system verification is complete, the chips will tape out, and after the silicon is available, the testers perform lab testing. A few bugs will be left that could in theory have been found in lab testing. But the test groups just doesn't have enough test or CPU cycles available to have found them. The number of problems is about a dozen, small compared to other things, like electrical or mechanical problems.

It takes a few hours to compile and run a serious validation test. Building a Verilog model containing one instance of each chip takes an hour and a half. Building the Verilog model means building all those parts together.

As these tests are folded into the next version of testing, accumulated coverage grows.

PRACTICAL MATTERS

Logistics of Simics Solution

Staffing

The core staff that supports the definition of the architecture of the simulations and its implementations number about ten people. As described above, there is a wide span of groups that are involved in using Simics. The whole division and company benefit from Simics.

Host Hardware Setup for Running Simulations

There are a large number of large, very high performance lab machines that support the Simics software development and hardware validation work. One of the current management issues is to determine how best to organize server clusters to provide the highest degree of availability.

Working with Virtutech

The highly accurate CPU module used for system verification and software development came about by giving Virtutech a copy of the Programmer's Reference Manual with the request, "We need a module that represents this CPU". Internally, the teams were required to generate Simics modules for the switch ASIC, MMU, etc. Once the customer became committed to system level simulation as a strategy to reach time to market goals, efforts to produce Simics modules have grown. They contract for things they can't efficiently generate in house, so features are poured into the tool in that way.

Bugs in Simics code were evident in the CPU model because it was produced as a joint development project. When the customer found a problem, Virtutech knew there was a problem and they fixed it in the right way.

One difficulty was that much time was spent porting between Simics release versions. Virtutech was a company spawned from academia where different standards of software evolution obtain compared to industry. That is being addressed now head on and is not expected to be a problem in the future.

Going Forward

Going forward from here there will be a bigger involvement for simulation. A department manager convinced upper management that simulation should be an important part of the bring up. As a result, upper management has decreed that there will be simulation. Teams that should be interested have been notified. If they aren't already committed to Simics simulation, they must make sure that they become interested.

Simulation simplifies the process of validation of compatibility. The only need will be to close the loop by checking that the simulation prototype agrees with what the hardware team has built. Then the burden shifts to the hardware team to prove that they have actually built what they said they were going to produce. Later on, when hardware becomes more stable, simulation plays a smaller role. Now there is a simulation implementation team that is actually implementing the technology.

In short, Simics has become a vital part of the development and testing effort, now officially sanctioned by the full management team. Where Simics can rightfully be applied, it has succeeded brilliantly.

As the Simics business relationship manager said, "If I were to take a guess we're probably looking at nine months to a year in development time savings than if we were not given access to a simulation environment to develop software".

What more needs to be said?



CONTACT INFORMATION

Corporate Headquarters	Virtutech, Inc. 1740 Technology Drive #460 San Jose, CA 95110 Tel: 408- 392-9150 Fax: 408- 608-0430
European Sales and Support Office	Virtutech AB Norr tullsgatan 15 SE-113 27 STOCKHOLM SWEDEN Tel: +46 8-690 07 20 Fax: +46 8-690 07 29
Internet	info@virtutech.com http://www.virtutech.com

© Copyright 2002 Metrica Labs, Inc. All Rights Reserved.

TRADEMARKS. Virtutech, Simics, Hindsight, and the logos thereof, are trademarks or registered trademarks of Virtutech, Inc. and/or its subsidiaries, in the United States and/or other countries.

THIS PUBLICATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS PUBLICATION COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THE PUBLICATION. VIRTUTECH MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS PUBLICATION AT ANY TIME.

Virtutech, Inc., 1740 Technology Drive #460, San Jose, CA 95110, USA