HP's VSE Shows the Practical Side of Server Virtualization

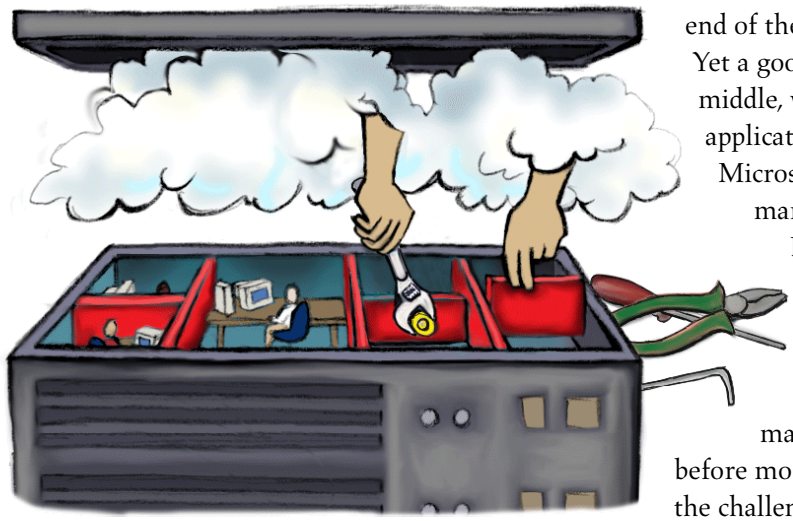
Research Note

Jonathan Eunice
27 April 2006

Virtualization[1] is hot not because the technology *per se* is particularly new or revolutionary, but because now that it has matured, the outcomes it enables are so desirable and so central to what IT shops need. Lower costs, increased flexibility, and reduced risk—who can really argue with those? Given a clear shot at achieving such laudable objectives, who could decline?

The approach goes back decades, deep into the mainframe era. If there is anything revolutionary, it is how broadly applicable and how democratized virtualization has become. In a few short years, what was a specialist function unique to Big Iron has developed into a capability for systems of all sizes—indeed, a capability increasingly built in as a foundation layer of even the highest-volume server, storage, and network devices.

Customer thinking and usage models have rapidly embraced virtualized infrastructure. Partitions, workload managers, and similar tools were until a few years ago known only to mainframers and a few of the largest Unix shops. Today they're being enthusiastically embraced and rapidly adopted throughout IT. While it's still too early to call virtualization "pervasive," it's getting close; the approach has clearly "crossed the chasm" into the mainstream.



Server virtualization is often thought about at one end of the spectrum (x86) or the other (mainframes). Yet a good deal of the action is actually in the middle, with Unix servers running core business applications. In recent years, both IBM and Sun Microsystems have aggressively developed and marketed virtualization in one form or another. But it's actually Hewlett-Packard that has been virtualizing Unix servers the longest, and that has the most diverse experience with deploying it in the field. HP introduced Big Iron resource management to Unix over a decade ago—long before most competitors even thought much about the challenges of scaling Unix datacenters. Year-by-year since, HP has steadily extended and deployed its virtualization toolset, now known as the Virtual Server Environment (VSE).

[1] See our *What is Virtualization?*

From the Labs to VSE

HP was the original thought leader in Utility Computing, the concept that computing power and resources can be delivered flexibly, as a service, flowing on demand to where they're needed. Now known by marketing slogans like "Adaptive Enterprise" and "On Demand," this concept contrasts sharply with the historical model—and, in many places, current practice—of purchasing physical systems, statically configuring them, and devoting them to one application for their entire useful lifetimes.[2] This approach carries with it a very limited ability to adapt to changing customer requirements and business conditions, combined with the Scale Out approach[3] of the Client-Server and Internet Ages—which in turn has led to much of today's high cost-of-operations server sprawl.

Server consolidation, virtualization, and utility computing—or variations on these themes—are the natural answer. Such visions have become commonplace in the last few years, as have production implementations. But they were rare to the point of unique in the *Eighties* when HP Labs was first working them, and even into the mid-Nineties when HP was essentially alone in espousing a fluid utility approach. Indeed, at the time, the concept was sufficiently visionary as to seem futuristic, even disconnected from achievable practice. Yet, here a few years later, utility approaches power much of today's scientific and technical computing (often under the rubric of "grid computing"), and are rapidly spreading throughout commercial IT (though often clothed as "virtualization" and "Service Oriented Architecture").

To be sure, the flavor of utility desired varies by customer size, target workload, and other characteristics. Research scientists will happily disperse their workloads widely in order to get absolutely the largest possible compute farm. Commercial enterprises are much more concerned about both technical and legal risks, and their applications are much more bound by I/O bandwidth and latency—so they are more focused on building their own IT infrastructures and keeping most of their data and activities in-house. For their part, service providers want very large infrastructures (enterprise-scale and above) that also come with extensive tools for provisioning, managing, and billing in an explicitly multi-tenant environment. Finally, folks running just a handful of servers still want utility-like flexibility, but they are most concerned with how that flexibility scales down to individual servers, storage devices, and network appliances—or, at most, a few such devices connected together. As usual, horses for courses.

Opening the Toolbox

HP's Virtual Server Environment is not a single product or feature, but rather a "flexibility and control" toolbox for Integrity servers running some combination of HP-UX (HP's "Big Iron" Unix), Linux, and Windows.[4] Customers pick and choose the VSE components—and combinations of components —that make sense for them. Key mechanisms enable:

Partitioning system resources – The ability to "slice and dice" expensive server gear into smaller "virtual servers" is the starting point for the way many datacenters use virtualization. There are numerous approaches in the industry,[5] and VSE essentially offers one of each. It starts with hard, electrically isolated partitions (nPars) based on its Cell system boards[6]; logical partitions that scale down to single-CPU granularity (vPars); and most recently, Integrity Virtual Machines with a sub-CPU granularity. Finally, VSE includes Process Resource Manager (PRM) and, now, Secure Resource Partitions (SRP). Both are resource limitation and containment techniques that can, for the right workloads, be used in lieu of other kinds

[2] We're focused here on the server side of things, but a similar story applies to storage and network devices.

[3] "Hey, just buy more boxes!"

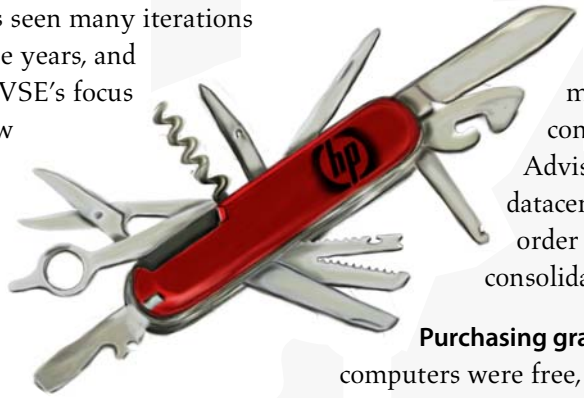
[4] Some VSE parts, such as Serviceguard and SIM, can span beyond Integrity to include ProLiant servers; other parts like gWLM also include OpenVMS; Integrity and its mainstream operating environments remain VSE's heartland, however.

[5] See our *Walling Off Workloads with Partitions*

[6] See our *Arches: Bridging HP Servers to the Future*.

of partitioning.[7] But it is rarely an either/or proposition; these mechanisms can be and often are used together—especially vPars or Integrity VMs running inside nPars, or PRM workload groups and SRPs running within any of the other forms of partitioning.

Dynamically allocating application workload – Partitions create containers within which workloads run. Looking at the problem from the workload perspective, however, gives a different view—one focused on application performance, rather than the carving up of resource elements. This higher-level view meshes nicely with current thinking on service level management (SLM) and service-oriented architecture (SOA). VSE started, lo those many years ago, with the Process Resource Manager (PRM). PRM has seen many iterations and improvements over the years, and remains in active use. But VSE's focus product in this space is now WLM, its "goal-directed" workload manager. It can take service level objectives (SLOs)—essentially, application performance targets and thresholds—and convert them into resource allocation and de-allocation commands. That is, by twiddling the control knobs of PRM, nPars, vPars, TiCAP (Temporary instant CAPacity), and many other VSE components, WLM can dynamically adjust the amount of resources devoted to a job until it appropriately meets its SLO. Global WLM (gWLM) extends this thought across multiple systems, helping to ease administration.



Ensuring application availability – Every server manufacturer works to ensure the reliability and uptime of its products. Integrity servers come from an HP group known as Business Critical Servers, known for taking great care in this regard. But no matter how reliable the design, components still break, and systems can still fail. For applications that can't easily be taken down—and how many "business critical applications" can, in this 24*x*7*x*forever day and age?—one must make even more heroic efforts to ensure they don't go down on their own accord. Thus Serviceguard, high-availability clustering software that HP users have trusted for well over a decade to "catch them when they fall."

Managing ongoing operations – HP has an entire management software operation centered on its OpenView lineup. But managing VSE starts closer to home with more focused system tools. One such is Virtualization Manager, a plug-in for HP's System Insight Manager (SIM) consolidated element manager and administrator console.[8] Another is Capacity Advisor, a tool that examines a datacenter's workload patterns in order to help plan resource consolidation.

Purchasing granular IT capacity – If computers were free, virtualization would be solely about managing change and complexity. But they're not free, so virtualization gets tasked to also help keep another critical axis—cost—under control. Beyond "meat and potatoes," product-oriented functions, VSE includes economic offers that help manage capital acquisitions. These include capacity-on-demand (COD) offers such as the iCAP ("instant CAPacity") TiCAP (Temporary iCAP), and the new GiCAP (Global iCAP, which shares iCAP usage rights among multiple servers); and PPU (Pay Per Use) for utility pricing.

Cross-integration - No vendor happily admits that there are functional gaps or poor integration between the parts of its product line. But there always are. And with virtualization relatively new to most, it's not hard to find components that are, at best, weakly integrated. This is an area in which

[7] Historically, PRM did explicitly that. The advent and maturation of nPars and vPars has reduced PRM's role. Going forward, SRPs and Integrity VMs provide successively greater isolation than PRM, yet a similar sub-CPU granularity of resource allocation; over time, they will take additional workloads from PRM.

[8] See our *HP Systems Insight Manager: One Console to Rule Them All*.

VSE shines. A real virtue of HP's years on station is that it has had time make the parts work together. WLM's automated workload management, for example, works well not just on a single system; it also plays nicely with Serviceguard to maintain workload definitions and activities *across* systems, even during failure scenarios. vPars and Integrity VMs aren't singleton mechanisms; they work with either raw systems or within nPars. And HP's capacity on demand tools (iCAP and TiCAP) can overlap nPars, vPars, and other mechanisms to provide synergistic effects. While there will always be some gaps between and within such tools, HP's level of cross-integration has long been top-notch.

We're not going to replicate the full book on VSE here.[9] It has far too many pieces, parts, modes, and options. We'll net it out by saying that VSE is not just a hopeful strategy or pretty vision statement. It's a rich, polished, often-deployed product suite. Its practical, well-integrated virtualization is a major reason that HP's Integrity servers are such strong production computing platforms.

But however virtuous VSE may be, HP has no unique lock on platform virtualization. Every server vendor has bought or built at least some of the same capabilities. In an era when customers are rapidly shifting their concerns from technologic inputs to business outcomes, what's more interesting than VSE's features and functions is how it's being used in practice.

Cut to the Tape

As often happens with "the Next Big Thing," those developing and selling virtualization have tended to eagerly promote it[10] as a Utopian paradise that effortlessly evaporates all cares and concerns. It takes nothing away from virtualization's commendable progress or fundamental Goodness to say that it isn't perfect, isn't yet ubiquitous, and won't be the solution for every problem.

So what aspects of server virtualization are practical today? What are the effective use models, *versus* what are the simply theoretical, someday-soons, or one-offs? HP kindly referred us to datacenter managers at several of its leading VSE customers. We talked with them about what they've deployed, how they think about and use virtualization, and where they're headed. HP also provided us with VSE case studies that they use internally, but don't typically share beyond their corporate firewall. Obviously, these are going to be cases where VSE has been successful, and users who are happy with VSE. That said, there's a lot to be learned by successful examples. And we've also talked with a fair few that HP didn't suggest.

VSE is used everywhere from some of the largest commercial datacenters—shops that consume Superdomes in the very plural—down to companies deploying a few mid-range Integrity servers, running one or two specific applications. Such different use cases have obviously different needs, preferences, and priorities. One thing they manifestly share is that they trust HP to run their production workloads. But the pieces they deploy vary widely.

It's impossible to summarize the world's diversity in short order. So we've picked a few particular customers that we believe are good exemplars of deployment classes.

A Fortune 100 Datacenter

At the high end is a Fortune 100 enterprise. Let's call them "Consolidated Goodness, Inc."[11] ConGood has been early with, and highly committed to, virtualization as an organizing principle for their server resources. Their datacenter uses a combination of nPars and vPars to manage a consolidated server pool that includes both midranges and Superdomes. Appropriate slices are "carved" out of this pool to run various

[9] Dan Herington and Bryan Jacquot's *The HP Virtual Server Environment: Making the Adaptive Enterprise Vision a Reality in Your Datacenter*, despite its fifteen-word title and 529 pages, is readily accessible. Recommended reading for those who want to dive deeper into VSE or its many components.

[10] Along with its fellow travelers such as Service Oriented Architecture (SOA), grid computing, and utility computing.

[11] No, not their real name. But you already knew that.

enterprise applications. ConGood uses nPars for isolation, but then runs vPars (up to four, currently) inside of nPars for finer granularity. They aim to grow the number of OS instances per physical server over time, as they gain track record and confidence with this approach. They also use Serviceguard to ensure availability.

This essentially is a "Unix mainframe" example. The physical servers are large and very well configured, and each runs many applications simultaneously. Over time, as VSE enables finer granularity (progressing from nPar to vPar and eventually Integrity VMs, as they prove themselves), as ConGood continues to gain confidence in the "many eggs in one basket" approach, and as the power of individual processors and other components purchased grows, ConGood plans to increase the number of applications run on every system apace. Actually, this is a "Unix mainframe plus" example, as the resource pool enabled and managed by VSE spans *multiple* systems.

At the time when ConGood chose HP infrastructure, they also considered both IBM and Sun. Their IT staff felt that HP had a good combination play: good granularity of partitioning (through vPars) and highly isolated, hard partitions (nPars)—with the ability to use both in concert. IBM was in the running (with LPARs), but Sun at the time offered only coarser granularity, though highly isolated, hard partitions (Dynamic System Domains).

ConGood does not use VSE's workload management functions, feeling that PRM, WLM and such would require dedicated teams to understand and tune. They prefer to use the more isolated, and less invasive, partitioning mechanisms. This choice is more idiosyncratic; other companies of similar size do use workload management, especially WLM for automating workload management.

Nor does ConGood use any of HP's utility pricing or capacity-on-demand offers. This choice is typical of those who run a large server farm. They already have their own infrastructure for allocating parts of their server pool, and feel they can manage their own spare resources and their own capital acquisitions just fine by themselves without any vendor-provided capacity-on-demand offer.

Finally, ConGood—as is typical of this size customer—doesn't just deploy a product. Instead, they are a development partner with HP, with direct insight and requirements input into HP's R&D functions. They are looking into both gWLM and Integrity VMs as future options.

A Successful Distributor

A more modestly scaled example comes from "DistroMatic," a successful distributor of business products with nearly fifty warehouses/distribution centers around the United States. DistroMatic runs its order management system and associated applications on a handful of HP servers, which it partitions with nPars. While DistroMatic has been a longtime HP customer, IBM has an especially strong position ("floor to floor") in its corporate parent, and was a competitor for this business as well.

The first thing that pops out about DistroMatic's application is that, if it's not running, the company isn't able to take orders or ship product. Loss of application availability is not just "unfortunate," it grinds the entire business to a halt. On the other hand, this is not a securities trading or online purchasing application that needs to be in operation each and every second. An hour or more down? Very Bad. A few minutes? Not so much. This is illustrated by the fact that DistroMatic previously used manual application/disaster recovery (DR) failover techniques. DistroMatic has, however, recently upgraded to Serviceguard to automate failover, both to make it more timely and more reliable.[12]

Coupling single-system and single-site availability with the ability to do disaster recovery at a different site (the headquarters of their corporate

[12] Delightful understatement of the day: "Humans make mistakes."

parent) was important. They are working to further automate this process over time.

DistroMatic also uses TiCAP for performance protection, "just in case" their growth forecasts or projections of capacity requirements "miss the mark." They have consumed a bit of their temporary capacity, and are talking about buying some of it outright.[13] They are also considering how to automate the use of temporary capacity. They certainly can turn it on and off manually, but where they really want to get is to an automation layer that knows when meeting service level objectives (SLOs) requires more horsepower—and can bring those processors online without intervention. While they are not yet far down the road of implementing WLM for this purpose, that is a next step. They view functions like TiCAP, WLM, gWLM as either unique to HP, or uniquely strong at HP.

Interestingly, while TiCAP appealed considerably, DistroMatic was not interested in traditional capacity (upgrade) on demand (COD) offers; they felt that what they call "capacity on purchase order" was just not a help. This and other examples suggest that evolutions of COD programs to make them more systematic and more granular, far from being minor refinements, may be quite essential to reaching user "tipping points" at which the offers substantially appeal.

It's also clear that what sealed the deal for HP was getting the right "solution architect" involved and including some services. Virtualization addresses complexity, but it also—especially when getting started—adds additional layers, tools, and architectural approaches that the user must accommodate. Getting well-trained, experienced people involved, both in the pre-sales architecture and planning, and in post-delivery setup, training, and configuration, pays dividends for everyone involved.

[13] Temporary offers can be attractive to use occasionally, primarily as a reserve or "insurance policy." But if you begin using it routinely, you're financially better off to own the resources yourself.

DistroMatic likes the ability to run Unix, Linux, and Windows on the same box, but they run primarily HP-UX now. Similarly, they like VSE's breadth, but they use only a few specific parts of it today. Retail and distribution companies are famously focused on efficiency and not frills. DistroMatic offers a perfect illustration that, while VSE is useful shorthand for a complex of capabilities around what HP calls Adaptive Infrastructure, saving money and doing specific practical things is what's most important for customers.

A Medium Business

Our final example is "MPro," a division of a well-known company. MPro runs a couple of rx8620 midrange servers for SAP production instances, along with some test and development images. They use nPars as their primary system partitioning mechanism, and Serviceguard for availability.

Not only is the scale of their operation much smaller than ConGood or DistroMatic, so is the diversity of tools and the level of dynamism targeted. They have a fairly static configuration; while they have extended it with new cell boards, they manage stable servers rather than a dynamic resource pool. This is a typical trade-off. If you have a large number of jobs to run, with new applications frequently entering the mix, it makes sense to establish a systematic platform for running a complex and shifting mix of changing applications. In contrast, if you have just a few key jobs to run, limiting yourself to a few mechanisms makes everything simpler.

This smaller environment has led MPro to deploy capacity-on-demand (iCAP and TiCAP). Servers are purchased fully populated, but with some processors de-configured. They sit idle, awaiting use for peak demand, such as end-of-month work or extraordinary events. MPro is happy to pay the small premium to have these flex resources available rather than purchase outright excess capacity that they may never use. And, in fact,

MPro has not so far had to use their capacity-on-demand facility. This, too, is typical.

No one buys Serviceguard because they *want* something to fail (or even necessarily expect it to), but they are eager to see Serviceguard leap into action and save the day by moving workloads elsewhere. Customers buy Serviceguard because they know that something *might* fail, and they dislike—or outright cannot tolerate—the consequences. So they put in place appropriate insurance so that if and when something were to fail, they'd be protected.

Capacity-on-demand provides a similar insurance, but from a performance rather than an availability perspective. It reassures MPro that if they've wrongly guessed about their peak workloads or their business growth rate, then they have on tape an easy and effective way to compensate. If MPro expected to use capacity-on-demand resources regularly, they'd be better off financially to just buy them. But for unexpected peaks or growth, COD functions make good sense. Our discussions indicate that "we haven't needed it, but if/when we do, we know it's there" is a common sentiment and reassurance.

MPro is a more of a nuts-and-bolts user than ConGood. They have much less need or desire to use varied technologies, or to directly engage with HP's R&D functions on forthcoming products. This is typical of smaller shops.

Lessons Learned

A number of common expectations and approaches pop out of just about all virtualization customers we've talked to, and use cases we've reviewed. They include:

Virtualization Comes With – Unlike the x86 space, where virtualization is often purchased separately from a third party, in larger servers virtualization is provided almost solely by the server vendor. A piece here or there might occasionally be purchased from the likes of Symantec (Veritas), but once you've bought HP Integrity servers, you pretty much need HP's Virtual Server Environment for virtualization. A similar story holds for IBM, Sun, etc. While this may "limit choice," the vastly more important outcome is that it fosters product integration. It also provides "one throat to choke"—and more practically, one number to call—should something go wrong.

Partitioning Resources is the Start – The ability to slice and dice large server resources into smaller usable chunks is what gets the ball rolling. The details of which techniques are used vary by customer and requirement. Some start with workload management, others with hard, electrically isolated partitions. In all cases, what counts is the ability to subdivide and allocate pieces of an expensive capital asset, thus making an "infrastructure" rather than a "point product" purchase.

Scale Within Drives Scale Up – We saw at ConGood and elsewhere that virtualization provides a reason, a justification, and a means for scaling up the basic server platform. What might have been a handful of 4-, 8-, or however-many-processor servers instead becomes a larger server that is partitioned as needed (aka "Scaled Within"). The additional cost of Scale Up infrastructure could not be justified absent virtualization, but with VSE's combination of partitioning, workload management, and availability tools, users felt comfortable with the bigger hunks of infrastructure, and indeed felt that they could achieve better manageability, greater flexibility, and reduced operational expense by going that route.

Availability is All – Virtualization fundamentally asks the owners of business applications to share infrastructure resources. You cannot "put all of your eggs in one basket" unless you've reinforced the basket. Thus it comes as little surprise that essentially every substantial VSE deployment uses high availability techniques such as Serviceguard to ensure application availability. Customers large and small depend on Serviceguard to automatically "fail over" workloads should some system or application component fail.

One Size Does Not Fit All – Businesses differ. Their priorities, policies, and expectations—and how they run their IT operations—differ as well. Even within a business, there's often enormous variation across lines of business or application areas. Virtualization appeals more-or-less across the board, but there's enormous fluctuation in which parts, features, or functions are considered appropriate and attractive. Some users, for example, go ga-ga over capacity-on-demand (especially the temporary, can turn on or off at any time variety, *à la* TiCAP); others, frankly, couldn't care less. Sub-CPU partitioning granularity, wide-area clustering, automated workload rebalancing—name an advanced feature, and there's some datacenters that deeply, thoroughly want that function. There's also a complementary group that wouldn't buy it if it were free, wouldn't install it if they had it, and wouldn't use it if it came pre-installed. This variation is not a negative, but a natural outcome of having a lot of tools in the toolbox, and a lot of different jobs to accomplish with those tools.

Incremental Steps Key – While datacenters may someday run seamlessly virtualized across their entire infrastructure, and while we have seen some synergistic examples of multiple disciplines working in concert (servers and storage, say), most of the practical instances with which we are familiar largely separate their myriad concerns. Even within a given area, such as server virtualization, customers prefer to implement one approach (say partitioning) at a time. They like to see that others have done similar things—and done them *successfully*. After dipping their toes in and getting their first forays working, they gain confidence, and then proceed to add secondary and subsequent approaches. "Big Bang" projects that attempt to virtualize everything in one fluid motion are not only rare, they're seen as risky, impolitic, and largely antithetical to the "optimization" theme that drives virtualization in the first place. Customers value flexibility, and having options; they just don't try to use everything at once, especially right off the bat.

Conclusion

HP has worked for two decades on the vision of Utility Computing, and has been in the development, rather than the research, phase for over a decade—starting from resource management and high availability extensions for HP-UX; moving through partitioning and automated workload management; finally culminating in today's Virtual Server Environment. While its engineer-led culture and its transitions over the past several years have perhaps kept HP from promoting VSE as avidly as IBM and Sun have their respective approaches, HP has had strong capabilities in this area essentially forever.

That history has resulted in the highly-capable VSE suite. But in a maturing industry, what we find even more interesting than its features and functions are its customer use cases, and the lessons one can learn from them. Beyond further endorsing the already widely-accepted idea "virtualization is good," these profiles show specific modes and steps through which virtualization provides tangible business value.