Private Clouds: Selecting the Right Hardware for a Scalable Virtual Infrastructure

Greg Shields

sponsored by

hp

Realtime publishers

Copyright Statement

Chapter 3: Modularizing Cloud Hardware for a Scalable Virtual Infrastructure

Did your head spin just a bit after reading the previous chapter? Containing copious detail on virtual processing and memory, storage, and backups, Chapter 2's intent was to document the technical underpinnings that comprise a private cloud infrastructure. Those underpinnings, as you've discovered, are large in number, rich in dependencies, and complex in construction.

Their very complexity explains why many businesses don't immediately see the return on their virtual investments. The problem isn't necessarily the technology or its hardware. Today's hypervisors and virtual hardware are veritably bombproof. Rather, the problem lies in IT's sometimes inability to reap business-recognizable reward out of that technology investment. Two recent examples highlight this dissonance.

The first is a study from 2007[1], which reported that *44% of companies are unable to declare their virtualization deployments a success*. A key factor in this realization was reported as the business' inability to quantify the ROI on their virtualization investment. Businesses who reported success said that "being able to measure performance of the virtualized environment" represented a key factor.

The second and more recent study happened in May 2009[2]. That study showed that a majority of IT managers report *experiencing more problems than benefits with virtualization technology*. A quarter of those surveyed noted their negative experience was based on "a lack of visibility and tools to troubleshoot performance problems in virtual environments."

Challenging numbers but important ones to consider when you're making decisions about how to implement virtualization in your organization. I personally love these two studies, and have referred back to them numerous times. Their focus on the integration of virtualization with actual business processes presents a somewhat less-rosy picture than other surveys that focus exclusively on technology aspects alone.

[1] http://www.virtualization.info/2007/03/44-of-companies-unable-to-declare-their.html
[2] http://www.infoworld.com/d/virtualization/virtualization-cost-savings-hard-come-interop-survey-finds-196?source=IFWNLE_nlt_wrapup_2009-05-20

Why? As you can see in their results, two activities in virtual environment management are absolutely required for success: *performance management* and *capacity management*. Unfortunately, these two activities are too often neglected by the technologists responsible for managing that environment. If you do not properly quantify how many resources you have along with how many you need, you're going to experience problems with virtualization. Those problems will manifest as either slower-than-expected performance for virtual machines or, even worse, a shortage of resources when they're needed.

For this reason, this guide has focused on the business process impacts of private clouds over and above any click-by-click installation. With a private cloud, you'll be able to quantify your demand for resources. Incorporating that cloud with the right set of modular hardware enables you to discretely scale your supply of resources to meet that demand.

Defining Modularity

I keep returning to the term *modularity* in reference to your private cloud's hardware. That's because constructing a private cloud out of modular hardware takes much of the guesswork out of those critical performance and capacity management activities. It also makes hardware purchases much more plan-able. With modular hardware, the amount of work an IT engineer needs to accomplish simply to purchase new hardware is greatly diminished. Need more processing, just buy more. The specifications are specifically tailored to the resource needs your virtual machines assert they require. That's an absolute boon for right-sizing investment with actual requirements.

But what really is modular hardware? Why is it so fundamentally different than the hardware you're already using in your data center today? In short, modular hardware represents *abstracting hardware resources into sets of purchasable units*. If you need networking, buy a unit of networking. More processing? Buy a unit of processing. Rather than constructing a virtual environment out of individual components, modular hardware creates a pluggable infrastructure that is comprised of repeatable hardware building blocks.

Death of the White Box, Round Two

If this scenario sounds excessively simplistic, recognize that this evolution towards modularity has happened before in IT. Remember the days of the *white box*? A white box in industry parlance refers to a computer that has been assembled out of individual parts. A person building a white box might purchase a processor from one retailer; they purchase RAM from another; and a motherboard from somewhere else. Collected parts in hand, they will construct a fully-functioning computer out of these individual pieces.

For many years, IT professionals created white boxes out of necessity. In IT's early days, there were no preconfigured servers that one could order off the Internet. Creating a new server meant collecting the necessary pieces, shipping them to your location, and plugging them together before ever installing an operating system (OS).

This analogy is important because many of today's data centers use the white box approach in creating their private cloud infrastructures. In a void of pre-engineered and virtualization-specific hardware, creating one's own private cloud until very recently required constructing it from individual pieces.

Just like individual servers, virtualization environments are now maturing. As you'll learn in a minute, white boxing need no longer be the case with virtual environments and private clouds. In fact, continuing with the do-it-yourself approach can actually represent a risk to your virtual investment.

Let's think for a minute how this scenario might work. It's easiest to understand how you can purchase this modularity by comparing it with some other run-of-the-mill purchase. Let's use an analogy with another product you might pick up on the way home from work tonight.

Virtual Hardware like Milk

My wife used to work in retail sales. There, everything is about the individual SKU or "Stock-Keeping Unit." In retail, sales are made because products are displayed in ways that are easy to purchase. You know what a bottle of milk looks like when you head to the grocery store. So you grab the gallon you need and have it rung up at the checkout stand before you leave.

Milk is an easy product because its SKU represents a single item of capacity that you can purchase. You need a gallon of milk. You buy a gallon of milk.

Most of us don't purchase a cow, then a barn, and then spend time engineering how much milk we can expect to receive out of that integration. Although that engineering might be entertaining, it's not necessarily where our skills lie. Thus, while we probably could connect a cow with a milking machine, we're probably not going to get the most optimized of results. In the end, we just want the milk.

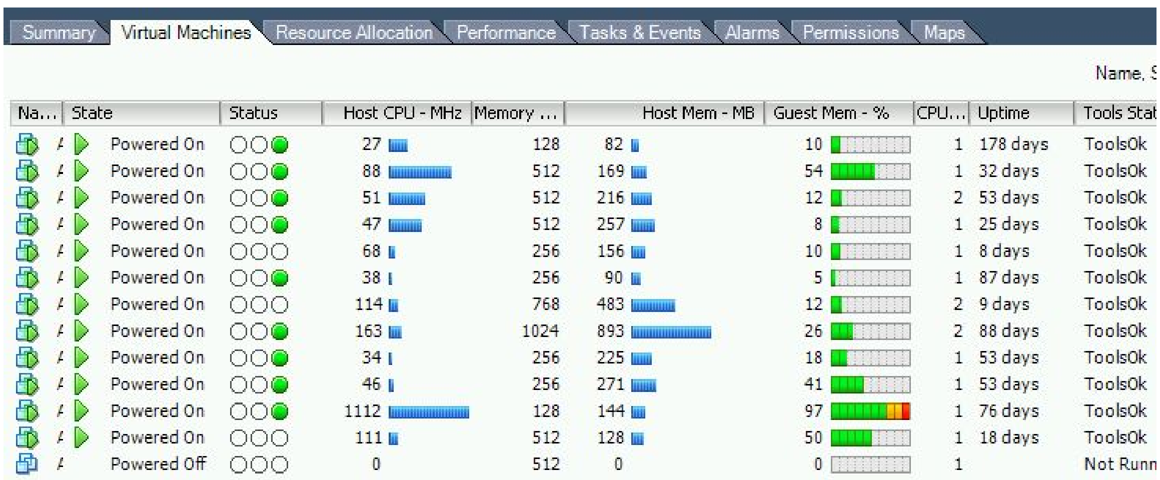
Relating that sidebar back to IT, Chapter 1 has already asserted that users just want access to their applications and data. They really don't care how IT provides it. IT also wants to deliver those services in ways that are easy, get the most out of available hardware, and present the best investment to the business.

You can absolutely deliver those applications and data through a do-it-yourself virtual environment. Many data centers have been doing that for years now. But there's another way to meet this need, using hardware that's been pre-engineered to deliver a known quantity of virtual capacity.

Modularity and the Private Cloud SKU

Can you imagine a day where it is possible to buy an entire private cloud with just a single SKU? If you knew how many resources your virtual machines needed, along with how many you'll need over the coming months, it stands to reason that you should be able to purchase just that amount.

With the right technology, *that day is more or less here*. Figure 3.1 shows a screenshot from the VMware vCenter Client. It shows a set of virtual machines, one per line, along with a series of numerical data about their resource demand. The first virtual machine requires 27MHz of CPU. It is currently using 82MB out of the 128MB of RAM that it has been assigned.



Na...	State	Status	Host CPU - MHz	Memory ...	Host Mem - MB	Guest Mem - %	CPU...	Uptime	Tools Stat
	Powered On		27	128	82	10	1	178 days	ToolsOk
	Powered On		88	512	169	54	1	32 days	ToolsOk
	Powered On		51	512	216	12	2	53 days	ToolsOk
	Powered On		47	512	257	8	1	25 days	ToolsOk
	Powered On		68	256	156	10	1	8 days	ToolsOk
	Powered On		38	256	90	5	1	87 days	ToolsOk
	Powered On		114	768	483	12	2	9 days	ToolsOk
	Powered On		163	1024	893	26	2	88 days	ToolsOk
	Powered On		34	256	225	18	1	53 days	ToolsOk
	Powered On		46	256	271	41	1	53 days	ToolsOk
	Powered On		1112	128	144	97	1	76 days	ToolsOk
	Powered On		111	512	128	50	1	18 days	ToolsOk
	Powered Off		0	512	0	0	1		Not Runn

Figure 3.1: Quantifying capacity in the VMware vCenter Client.

Each of the other virtual machines in this list has their own resource needs as well. The second needs 88MHz and 169MB, and so on. I show this screenshot to prove that today's virtual platform vendors are already reporting on the necessary capacity information. Your IT teams can in many ways sum up this (and other) information to arrive at a reasonable number that represents how many resources your private cloud needs.

Now imagine an SKU whose purchase will deliver a set of servers, storage, networking, and management, all pre-engineered to exactly meet those capacity needs. Even more compelling, imagine extending that environment by simply selecting another SKU on a Web site and checking out. Simple, easy, predictable, and plan-able. Got your credit card handy?

Such a situation presents an entirely new way of doing the business of IT. Are your IT services consuming more resources than you expected? Purchase additional units and connect them in. Once you've received a unit, its connection happens in much the same manner that you would add a preconfigured server into your rack today. Whether that unit represents more networking, more storage, or more processing and memory, the incorporation of additional resources need no longer be an activity that requires significant engineering. *Hence, modularity.*

Modularity Brings Successful Virtualization to IT Generalists

In fact, the argument holds that constructing a private cloud out of individual pieces and parts represents a risk more than a benefit to your data center operations. Just like buying a cow in the early sidebar analogy, think for a minute about the different strengths that your IT teams will need to be successful with the do-it-yourself approach:

- You'll Need Storage Experts—Virtual machines require storage for their disk files, expensive storage on high-performance SAN disks residing in carefully-constructed data centers. Figuring out exactly how much storage you need now and into the future, then combining those needs with budget realities is a constant activity. Matching that combination with the even more important requirement of acceptable virtual machine performance makes storage right-sizing a task for storage experts only.
- You'll Need Networking Experts—Networking is needed in virtual environments at levels of complexity never before seen. Unlike physical servers with their single or dual-teamed connections, virtual hosts require multiple and concurrent levels of redundancy across numerous network connections. Complicating this task is the realization that those connections require throughput rates far beyond the needs of any individual physical server. Getting the mix just right requires network experts that are not often found and rarely afforded.
- You'll Need Server Experts—Memory and processors are necessary in quantities per server that were considered outlandish not too many years ago. Memory and processor density per server regularly doubles (and doubles again) to support the needs of concurrent virtual machines. Special instruction sets for virtualization are now found in multiple versions with complex names and differing capabilities between processor vendors. Simply understanding what you need to support, what you have, and what you will have down the road is a task only server experts can untangle.

Herein lies the problem with virtualization's original promise. By virtualizing, you absolutely can see reductions in space, power, and cooling. You can also get to that flexible infrastructure where available resources and requirements are perfectly matched.

But to get there and actually see a positive return, you'll need to carefully engineer the environment's hardware and software to meet present and future needs. More so than with any other IT project, building a scalable virtual infrastructure requires extreme levels of human expertise to get it right. Lacking those experts, your virtual infrastructure can and likely will cost your business more than benefit it in the long run. Alternatively, you can simply buy a set of pre-engineered SKUs that supply the necessary resources your virtual machines demand.

Capacity Management and the Purchasable Unit

The idea of purchasing pre-engineered SKUs is compelling, but it is also completely without substance unless this hardware actually exists. Fortunately, it does. With that in mind, I'll spend the remainder of this chapter discussing how such hardware accomplishes the task of quantifying resource supply and demand. I'll start by referring back to a statement I made in Chapter 1:

> Think of modular hardware like a set of building blocks that contributes a known quantity of resources to a private cloud. If you need three quantities of processing, you simply purchase three quantities. If you need two, buy two. Modular hardware from top-tier vendors should arrive with an assertion regarding how many resources that hardware will contribute to the private cloud.

Reducing virtual environment capacity to a set of purchasable units requires a couple of different capabilities. First, it requires hardware that is infinitely flexible in how it can be interconnected. You'll need hardware that can plug into an existing environment wherever and whenever necessary to add resources.

Second, such hardware must contain a quantifiable supply of resources—processing megahertz, memory gigabytes, storage gigabytes—that can be used by the virtual infrastructure. By plugging in that hardware when you need it, you'll immediately make available more resources without a data center overhaul.

Figure 3.2 shows a complete architecture where this modularity is in place. You can see how a set of server blades on the right are connected with a quantity of storage in the middle. Storage and servers are networked through a set of network devices at the top and bottom. The entire infrastructure presents its applications and data to users through a second set of network connections.

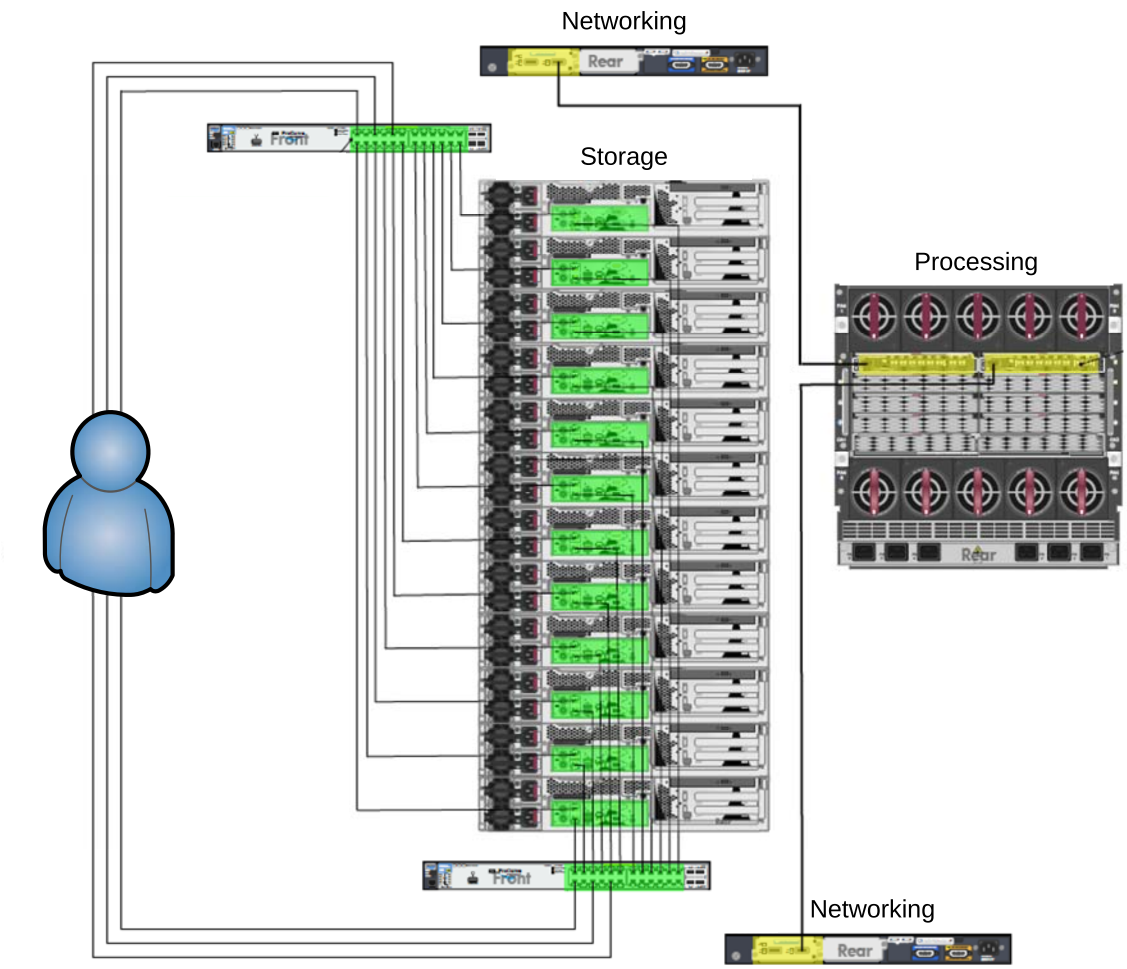


Figure 3.2: An example of actual modular hardware in place.

You'll notice immediately that the hardware seen here isn't revolutionary compared with what you already have in place. The element marked "Processing" has for years been referred to as a blade enclosure with a set of accompanying blades. The networking element is very similar to the switches already in your racks. The storage is comprised of a set (12 in this figure) of individual storage nodes that have been interconnected to create a single storage array.

Although the scale is impressive, the real excitement lies in the boundaries between individual components. Modularity arrives from the fact that these enclosures can be connected as building blocks to increase capacity. Figure 3.3 shows where four enclosures have been connected to create an array of processing power that is assigned to virtual machines.

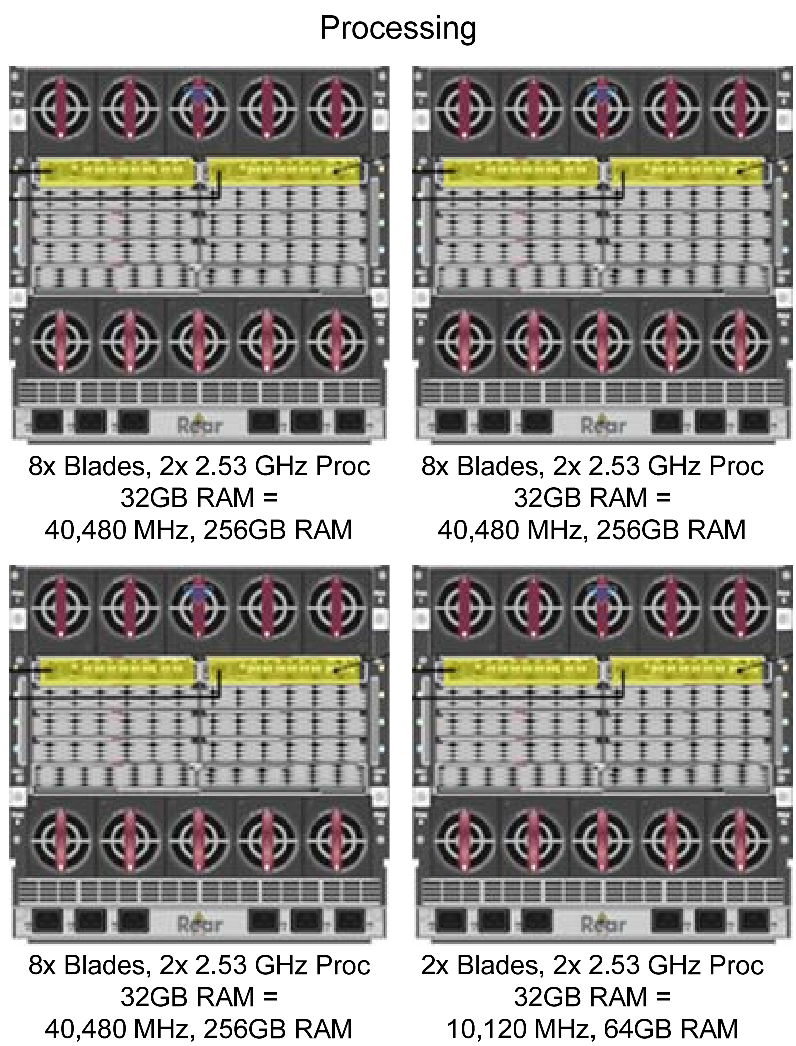


Figure 3.3: Modular enclosures as building blocks for processing capacity.

Take a look at the resources each enclosure contributes to the overall private cloud. The first three in Figure 3.3 are fully-loaded enclosures, complete with eight blades in each node. Each blade is a dual-processor server running at 2.53GHz. Each is also configured with 32GB of RAM.

As mentioned before, processing contribution is additive in virtual environments. Thus, it is possible to add these numbers using a calculation like the one in Figure 3.4. The result from three fully-loaded enclosures is a contribution of 121,440MHz of processing and 768GB of RAM.

$$3\ Nodes\ \left(8\ Blades\ (2\ Processors\ \times 2.53\ GHz)\right) = \ 121{,}440\ MHz$$

$$3\ Nodes\ (8\ Blades\ \times 32\ \text{GB}) = \ 768\ GB$$

Figure 3.4: Calculating processing and memory contribution.

There's a fourth node in this example that has not yet been fully loaded and is shown in the figure to represent how individual processing modules needn't necessarily be fully loaded. Rather than containing eight blades, this node is configured with only two. Perhaps its data center didn't need the extra processing, or maybe they simply ran out of budget. A smart data center might purchase exactly the amount they expect to need, leaving slots available for future expansion. As a result, the incomplete fourth node contributes 10,120MHz of processing and 64GB of RAM.

If you take a look back at Figure 3.1, you'll see the power in these numbers. An environment that knows it will require roughly 100,000MHz of processing and 500GB of RAM could easily purchase three enclosures to gain reasonable assurance that processing and memory supply meet their needed demand.

When those needs exceed available supply, additional enclosures can be purchased or additional blades can be added into existing enclosures. Either situation presents the ability for a data center to measure demand, calculate supply, and plan accordingly before their numbers become upside down.

Planning for Cluster Reserve

These calculations for processing and memory are obviously important, but there's an added factor that sometimes gets missed. Every cluster of virtual hardware must always be prepared for failure. That's why today's hypervisors—such as VMware vSphere, Microsoft Hyper-V, and Citrix XenServer—all require clustering of servers in order to achieve this flexibility of resources.

Clustering is the practice where individual hardware components are logically aggregated into the resource pools first explained in Chapter 2. No matter your hypervisor of choice, you'll need to create a cluster of hardware to fully realize a private cloud.

In addition to creating that resource pool, clusters provide the framework for high availability among individual hardware components. This high availability enables virtual machines on a failed host to automatically relocate onto another at the point of failure. Many types of high availability exist; however, each requires a set of spare capacity that your cluster must plan for. *This is your cluster reserve.*

Cluster reserve works on the notion that the simultaneous failure of some number of individual hardware components must be protected. In a smaller cluster, you might protect against losing only a single piece of hardware at a time. This is an acceptable risk because the number of processing components is small. Larger clusters will need to protect against greater risks, such as multiple nodes failing at once. A cluster reserve represents *a quantity of spare capacity, spread across all cluster nodes, that is equal to the amount of resources used by that lost server or servers.*

In Figure 3.3, any single blade is configured with two processors of 2.53GHz and 32GB of RAM. Thus, if your risk requires protecting against a single blade failure, you'll need to reserve exactly that quantity of resources. Multiple failures will require multiples of this number.

Recognize that cluster reserve resources must remain unused at all times. Thus, clusters with greater numbers of nodes tend to pay a lower percentage of "wasted" resources than those with a smaller number of nodes.

Many hypervisor management tools include automation components that automatically reserve the correct amount. That said, be sure to plan your hardware carefully, particularly as your resource needs go up over time.

Constructing Modular Storage

This notion of modular processing is an advancement our industry has enjoyed only recently. It has required the aggregation of virtual platform technologies with virtualization-specific hardware to manifest something vendors can actually sell. However, modular storage involves tactics that have been around for a number of years. You may already have this in place today.

Modular storage enjoys the best parts of SAN-based storage centralization and provisioning with none of its previous hard limitations. You're probably aware of some of those limitations, particularly if you were an early adopter for SAN storage.

SAN Storage, the Hard Way

I remember a data center that was very proud of their early SAN storage architecture. This architecture could support a jaw-dropping (at the time) 5TB of protected online storage. That storage also enjoyed the use of business continuity volumes, which enabled an early form of snapshot technology while easing backups.

The problem with this early infrastructure was in its hard lines around the storage frame, the enclosure in which its hard drives and interconnecting fabric were contained. The storage frame could support a total of around 15TB of raw storage, but no more. This wasn't a problem as long as the data center's needs never grew past the SAN frame's hard lines.

Unfortunately, and as you can expect where this story is going, that day eventually came. The available storage eventually grew very close to full consumption. Due to the limitations intrinsic to the storage frame, this data center's only options were to replace the existing hard drives with larger capacity drives—a situation that wasn't operationally possible due to the high utilization—or the purchase of a second storage device.

In the past, the primary culprit of SAN consumption was users and their documents and files. But today, virtualization and its virtual machines can become an even greater source of storage consumption. Needed to facilitate their ever-growing needs is a storage architecture that supports flexible expansion at will. This *node-oriented storage architecture* (seen in Figure 3.5) represents another area where modularity brings significant benefit to the private cloud.

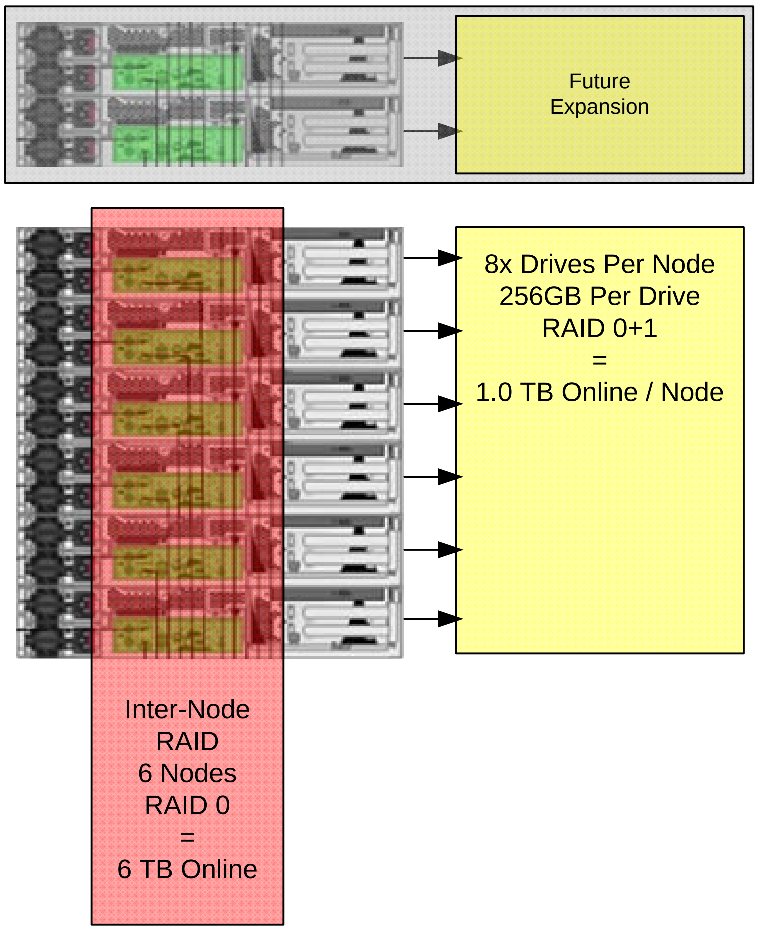


Figure 3.5: Node-oriented storage scales as needed without limitations.

You can see in Figure 3.5 that six individual storage nodes have been aggregated to present a total of 6TB to the data center. This 6TB is calculated by multiplying the drives in each node by their capacity, then removing the per-node overhead for whatever intra-node RAID level, and finally summing each node's contribution to the data center.

You'll notice that two levels of RAID protection are incorporated into this architecture. The first uses striping and mirroring, and occurs at the per-node level. The drives within each node are aggregated into a RAID 0+1 array to ensure the protection of data against a disk failure. A second level of RAID protection is gained by striping that data again across each of the nodes in the stack. The result is the creation and presentation of a single volume that can be provisioned wherever necessary inside the data center.

Aggregating individual nodes into a larger data volume is a great feature. It gets around those hard lines seen by early monolithic SAN architecture. Far more important, however, is the expansion potential this architecture enjoys. That expansion is represented by the two grayed-out drives at the top of Figure 3.5. Those drives suggest the capacity for expansion that such an architecture brings. Each node contributes a quantity of storage to the overall infrastructure, so it stands to reason that additional nodes can be added as necessary to expand the volume being presented.

Only two steps to accomplish this are required: Simply connect and power on the storage nodes, then add them into the volume. With the right hardware, the storage management software should take care of the rest.

Avoiding the Perils of Technology Generations

Processing and storage represent two of the major components in any private cloud, with networking representing the third. By embracing the modular philosophy possible with each of these three elements, your data center avoids much of the pain in hardware life cycle management—the purchasing, lifespan, and eventual decommissioning of server hardware.

One aspect of that pain that merits additional discussion relates to the various generations of technology hardware that you'll eventually acquire. Even with individual servers, storage, or networking equipment, the typical business purchasing cycle eventually acquires hardware that spans generations. This is particularly true inside virtual environments, which have a tendency to outlast typical physical servers due to their resource abstraction. The problem is that sometimes different hardware generations don't necessarily integrate well. You'll need to compensate for this reality as you plan your private cloud construction.

An explanation here will help. Vendor-engineered technology tends to come in generations: this server is from "Generation 5," that server was from "Generation 4," and so on. Each of the major technology vendors has its own nomenclature for representing their generations, but each tends to follow the same behavior in how they release new products.

In non-virtualized environments, the generational aspect of each technology wave causes an increase in overall administrative complexity. In plainer terms, a "Generation 5" server might have different drivers than a "Generation 4" server. Or hardware from "Generation 6" might not play well with anything from generations before. Keeping straight each generation's configurations and idiosyncrasies is an administrative headache, particularly when workloads are directly installed to servers in the all-physical world.

A private cloud naturally abstracts most of the generational aspects of each purchasing cycle, *but not all*. Indeed, each virtual machine has the same configuration no matter which virtual host it runs atop. But those virtual hosts might not always play nice with each other. A couple of real-world examples immediately come to mind:

- Changes in processor classes can prevent workloads from failing over to different virtual hosts without downtime.
- SAN storage arrays that won't communicate with each other can't be abstracted into a seamless storage layer.
- Networking equipment from different manufacturers uses different protocols or management approaches that don't integrate.

Scaling your private cloud is a trivial activity if these idiosyncrasies are planned for. *But you must plan for them.* Today's modular hardware enjoys a scale of compatibility that you just don't get with regular servers, storage, and networking all alone.

Modular hardware also tends to abstract the entire private cloud construction activity a step further than with the build-it-yourself approach. Essentially, you gain extra benefits by buying modular. For example, consider each processing module to be its own self-contained processing node in the private cloud. It will include one or more blades, which are generally of the same generation. Thus, failover can occur within the module, even if it cannot outside the module.

If, for example, Modules A and G don't work well together because their generations are too far apart, their virtual workloads can still fail over within their modules. You still retain the high availability your private cloud needs, albeit in a slightly more limited fashion as technology ages. In the end, your technology generations have less of an impact on operations *because high availability always remains a possibility*.

Managing the Private Cloud

In the end, all this modularity requires a management framework within which to operate your private cloud. Aggregating all your virtual workloads into a singular environment won't scale well if there isn't a centralized way to manage it as a whole.

At issue are the management toolsets that are currently available by the hypervisor vendors. These toolsets are most commonly used in managing virtual environments. Yet they suffer from limitations in visibility. For example, if your environment uses VMware vSphere as its hypervisor of choice, you may be using VMware's vSphere Server to manage your environment. Hyper-V environments may use System Center Virtual Machine Manager (VMM) to accomplish the same tasks.

Although these management toolsets indeed enable performing common actions on virtual machines—power on, power off, connect to console, reset, and so on—their scope of visibility is limited to things that are accessible at the hypervisor layer. It is difficult and likely impossible to, for example, use VMM to create a storage-level snapshot of an exposed SAN volume. The instrumentation simply isn't available for VMM to enact that change.

Environments that rely exclusively on these first-party tools for private cloud management will find themselves forced into additional and separate solutions for some of their other management activities. These segregated toolsets typically cannot share data between platforms, reducing the ability to truly monitor the environment.

That's a bad thing. This chapter began with the assertion that two activities are vital for private cloud management: performance management and capacity management. Completing both of these activities successfully requires management solutions that can see and interact with every part of the private cloud environment—from processing, to storage, to networking, and everything in-between.

You'll find that these toolsets are perhaps best served by the hardware manufacturers themselves. Only at the hardware layer can the necessary metrics be presented to a management tool to effectively measure capacity and keep tabs on resource demand. And, as you've already standardized on modular hardware selections, these tools are automatically available to you.

Such a solution might provide heads-up displays similar to Figure 3.6. There you can see how physical servers, virtual servers, SAN and virtual storage, as well as network components are aggregated under a single pane of glass. There, the activities of every component can be most-effectively monitored, and actions can be invoked via a central console. Your flexibility needs demand this detail in instrumentation across every piece of hardware.

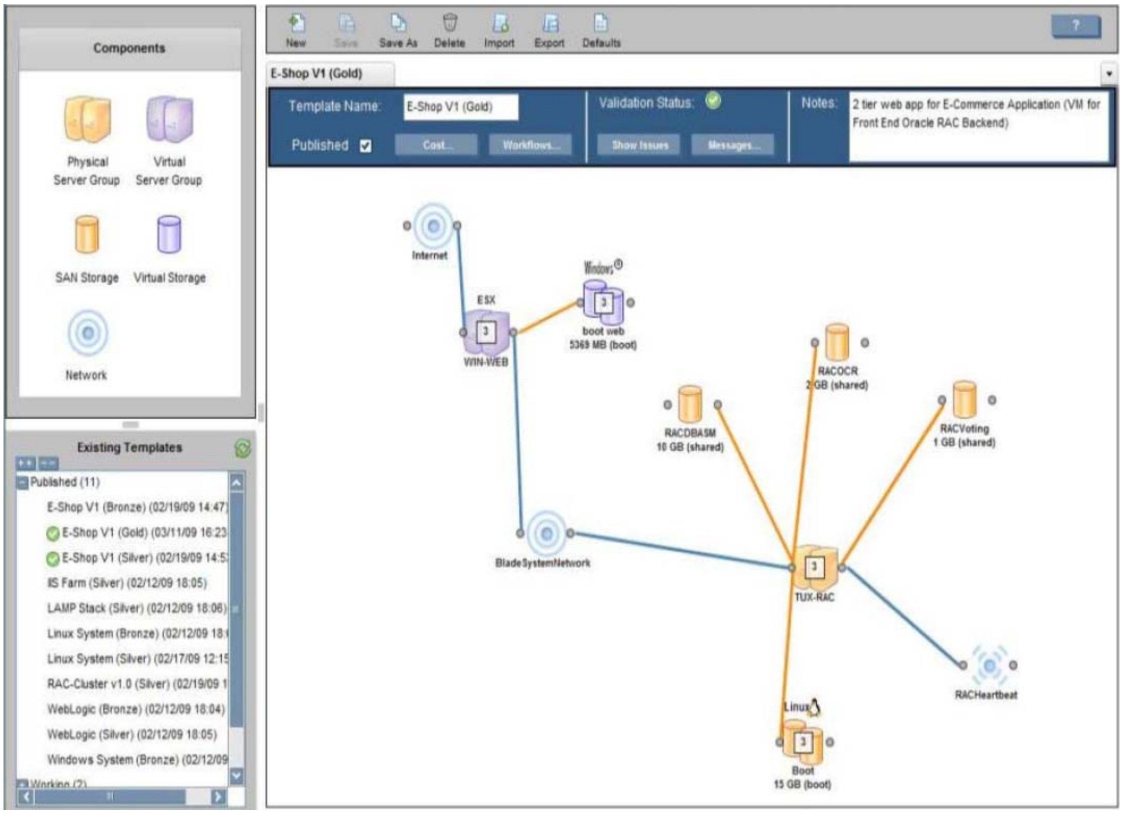


Figure 3.6: Creating workflows and templates across every private cloud hardware component.

Virtualization 2.0 Is Modular Virtualization

You can see where virtualization is headed. Virtualization's technologies are only today beginning to truly manifest the promise of the IT workload that runs "in the cloud." To get there, this chapter has asserted that you'll need three things: First, you'll need a mechanism to quantify resource supply and demand. Second, you'll need a set of hardware that is flexible in its implementation, supplying resources as they're needed and expanding when necessary. And third, you'll need a universal management solution that can peer into every layer to measure performance and enact change as it is needed.

You can absolutely achieve that today, with the right approach.

There's one more chapter to this book. That chapter, coming up, takes the conversation to the next level, helping you build the business case for scalable virtualization using a private cloud. It will help you identify the business requirements you'll need to collect in constructing one. It will also supply you with a set of useful metrics to employ in gauging your own success. With it, you won't be one of those 44% of companies who can't declare their virtualization project a success.

Download Additional Books from Realtime Nexus!