Servers, Storage, Virtualization, and You: It's Complicated

Simplicity isn't normally a hallmark of any of these three technologies, but it can be. Find out here how to break down the complexity of these components and keep them simple.

By Nick Cavalancia

SCALE COMPUTING

TABLE OF CONTENTS

SCALE COMPUTING

Every time a new application needs to be added, you are repeatedly reminded just how complex this delicate balance of servers, storage and virtualization of yours really is.

You know it all too well. When you think of even any one component in your virtual environment, the word simple just doesn't come to mind. Servers come in many form factors today with varying hardware limitations. Storage is available in so many flavors of sizes, redundancy, and virtualization capabilities. And virtualization is so advanced that each of the major vendors has its own customer training and partner ecosystem.

So what else should you expect when you combine all three of these already complex components, but something even more complicated.

It's not because of you. You run in a small team with many responsibilities and focuses. Virtualization and storage only play a limited role in your daily schedule, resulting in limited in-house experience and making you a "Jack of all Trades." You've had to rely on multiple vendors to setup and support a best of breed solution, which, while providing great performance, has only increased the number of tools to master, support phone numbers to call, and paths to the finger-pointing when things go wrong.

You're doing all you can to ensure the critical applications and services running on your virtual infrastructure remain running. But every time a new application needs to be added, or another piece of hardware needs to be bolted on, you are repeatedly reminded just how complex this delicate balance of servers, storage and virtualization of yours really is.

But does it really have to be?

While you start ripping out the pieces you think are to blame, there are some steps you can take to identify the complexity in your environment, plan for how to remove it while providing the same level of service, and then make the necessary changes in a non-disruptive manner.

In this whitepaper, we'll cover three ways in which your servers, storage and virtualization are unnecessarily complex, and discuss ways to help you move towards simplifying them.

Complexity #1: Architecture

Other than those of you having a single Windows Server 2012 server hosting a few VMs using dedicated storage, the rest of you would probably categorize your architecture as somewhat complex. You may have fallen into the trap of addressing servers, storage, and virtualization separately without regard for the overall impact on your architecture, but it's not entirely your fault.

There are so many choices from an equally vast number of vendors, each promoting how their corner of the virtualization world is going to revolutionize the performance of your VMs. So you purchase this server, and that SAN (two for redundancy!), and those switches, piecing together what you believe will be the ultimate VM environment.

There are so many choices from a vast number of vendors, each promoting how their corner of the virtualization world is going to revolutionize the performance of your VMs.

So, when did it (or does it) actually become complicated?

It you want to pinpoint a specific time, it's usually when you inject shared storage into the environment that it immediately becomes complex. Of course there's a reason you implemented shared storage - decoupling processing power from storage, better storage expandability quickly, and better manageability of VMs come to mind.

It's very cool the first time you do a live migration from one physical host to another because you have shared storage, but the getting to that point took your simple single server environment and took the level of complexity to orders of magnitude higher.

Most organizations work to address this by using a traditional architecture that may loosely follow the 3-2-1 rule — 3 servers, utilizing 2 switches, connected to 1 SAN. The architecture seems easy enough to understand, and provides you with the utility to perform live VM migrations, implement maintenance cycles, and the like, keeping all of your applications running. So, now you've gone from just having virtualization to actually being able to manage it.

While this is the way "it's always been done," is it the simple way?

Even with the redundancy of server and switches present, there's only one SAN. Now, it's obvious that SANs do come with redundancy built in, but there's still a single point of failure in that 3-2-1 architecture. Additionally, it's likely your servers are from one vendor, your switches from another, and your SAN from still another. So when a problem does arise, you're going to hear vendors blame one another wasting your time and not solving your problem.

No one wants to operate in this kind of environment, so it is possible to unwind this and eliminate some of the complexity?

Buying new hardware and performing V2V and P2V migrations to a more simply architected environment is an obvious but expensive option.

Simplifying the Architecture

There is no single simple answer here, as everyone's environment looks completely different from one another. And since your environment's architecture exists because of system, application, and storage requirements, there's also no really good answer around simply ripping out some of the pieces to make it simpler.

Buying new hardware and performing V2V and P2V migrations to a more simply architected environment is an obvious but expensive option. Even so, the thought process is on the right track.

Your best plan is to take advantage of hardware refreshes and replace with less complex solutions. It's not sexy, but it's the least expensive option that gets you there as quickly as possible. By identifying which parts of the architecture are overly complex, your hardware refreshes become more a re-architecting exercise than simply upgrading of capacity and speed.

You can't just architect an environment in the vacuum of "what applications do I need to run"; you need to be thinking about where those applications need to go next.

Complexity #2: Scalability

No business plans on getting smaller, so you know your environment is going to grow. Without even looking at a roadmap, you can count on needing to support more applications, additional storage requirements, all supporting more users.

Now, when you purchased those best-in-class pieces of your environment previously discussed, you did address scalability. The vendor likely talked about an ability to easily grow over the next number of years, giving you a sense of confidence that you've got this area covered.

So your environment is scalable… isn't it?

Here's the real problem. If you purchase hardware that allows you to scale well into years from now, when you finally outgrow it, the migration to something even larger is going to be riddled with even more complexity. Additionally, because you don't know exactly how the business is going to change over the next number of years, how can you be certain you will have the flexibility to scale in the direction needed by the business?

The challenge with scalability in virtualization is that it isn't linear.

The challenge with scalability in virtualization is that it isn't linear. So predicting which part of your architecture is going to hit a growth point first, and at what cost is extremely difficult. Sure, your 16-bay SAN will allow you to add on additional drives… until you need drive number 17. And it's predicting when that exact point in time will be for your SAN (and other components) that makes scalability complex.

It's obvious the answer isn't to simply add more, buy more, or migrate to something bigger today, so how can you simplify scalability?

Simplifying Scalability

The answer lies in focusing on three key areas: cost, utilization, and non-disruptive migrations. Your purchases today should focus on what you need today over purchasing for the next 3-5 years. The reason? There are three of them. First, because your only buying what you need today, the cost is lower, reducing your company's CapEx budget. Second, the utilization of that hardware is going to be much greater, proving its' necessity. And last, your ability to move to the next level without disruption is greatly increased.

While addressing scalability this way will cause you to perform more migrations over the years, it actually provides you with more control. Since you don't know what your application needs will be next year, let

alone 3 years from now, by focusing on today's business needs, your hardware choices today will more easily be able to migrate to the next evolution of your environment.

As you plan for scalability, remember even the addition of a single drive requires some way to ensure the availability of the data on that drive. So, as you scale your environment, you also increase the complexity around how to keep that environment running.

Complexity #3: Availability

It's important to remember that disasters take many forms.

One of the reasons you moved your servers to a virtual environment was for the resilience virtualization provides. Being able to move your virtual servers from one physical server to another (by using that shared storage we've been talking about) gives you a layer of availability you may need someday in the face of a disaster.

It's important to remember that disasters take many forms. Everyone thinks of Disasters (with a capital D) — earthquakes, fires, hurricanes, and the like, but there are also many, many little "disasters" as well, such as power outages or flooding from a broken water pipe. And, regardless of the disaster, the focus is usually on one or more hardware components failing.

But is that the only availability scenario you need to plan for?

Sometimes the disaster isn't a failure of any part of the virtual architecture at all — the next disaster could simply be a downed Internet connection from the municipal worker outside with the misplaced backhoe. For the simplest of availability scenarios — a loss of hardware — the answer is easy: have redundancy in place or an ability to quickly replace the component. But when the scenario isn't so simple, like the backhoe example, the level of complexity rises dramatically. Now availability takes on a new meaning and includes failover and failback, adding yet another level of complexity.

And virtual environments in the cloud aren't always the easy answer, as it depends on the application. Some apps run well in the cloud, like email and your CRM. But other applications, specifically where large

amounts of data are transferred as part of every request, such as computer-aided design, or digital x-rays in a hospital, need to run on-premises to be useful.

No single availability plan will address every possible disaster, so how can you ensure availability, regardless of the scenario?

Simplifying Availability

The first step is to identify the disasters you want to plan for. It's impossible to plan for every scenario, but you can cover some broad categories where maintaining availability will require different steps.

Once you have the disaster scenarios identified, it's a matter of building not just an execution plan, but a plan for failover.

The simple hardware failure is one that your environment likely already addresses — your SAN has data striped and mirrored, your 2 switches create the necessary redundancy, and your vSphere or Hyper-V environments have been configured to take certain actions should a failure occur.

But take it a few steps further and decide as an organization, based on the applications being virtualized and their criticality to the business, two or three other availability scenarios you need to have a plan for.

Once you have the disaster scenarios identified, it's a matter of building not just an execution plan, but a plan for failover — which may be scripted, automated, or performed manually — as well as for failback, which is often overlooked completely. Identifying whether local hardware or a redundant environment in the cloud is necessary is a part of the process, as failover and back to the cloud is a very different beast from doing the same thing locally.

The last step is probably the hardest — testing. Because of the inherent complexities of your architecture, this just doesn't get done. And that's bad. Really bad — given the importance of the virtualized systems, your testing really needs to be done so often it looks far more like practice than just a simple one-off test. But if you not just create the plan, but actually put it to the test repeatedly, the once complex planning of your virtual environment's availability is now a simple and familiar task, ready in waiting.

Keeping it Uncomplicated

Some of you aren't at "complicated" just yet, while others have knowingly been there for quite some time. What you both share is a desire to build and maintain an environment of disparate server, storage and virtualization components, with them somehow retaining an element of simplicity.

By fixing your virtual environment focus to as close to today's business needs as is possible, you'll drive your architecture to be as simple as is can be.

A lot in this whitepaper is about how you think about addressing the complexity in your environment, rather than the specific "how-to". Part of the reason is no two readers will have the identical components or application needs, so it's difficult to provide specific guidance.
Your environment is complicated because it was allowed to grow out of control. By fixing your virtual environment focus to as close to today's business needs as is possible, you'll drive your architecture to be as simple as is can be, which only helps reduce the complexity around maintaining its availability, and scaling it in the future (only when the business needs change!). ■

With nearly 20 years of enterprise IT experience, Nick Cavalancia is an accomplished consultant, speaker, trainer, writer, and columnist and has achieved certifications including MCSE, MCT, MCNE and MCNI. He has authored, co-authored and contributed to over a dozen books on Windows, Active Directory, Exchange and other Microsoft technologies. He has spoken at conferences such as the Microsoft Exchange Conference, TechEd, Exchange Connections, and on countless webinars and at tradeshows around the world.