THE GORILLA GUIDE TO...®

2018 Edition!

The Fundamentals of Hyperconverged Infrastructure

Inside This Guide:

- Learn about the IT challenges that remain in a post-virtualization data center
- Gain an understanding of all that hyperconverged infrastructure has to offer
- Zero in on the business benefits that come with the adoption of hyperconvergence

James Green, Scott D. Lowe, David M. Davis
ActualTech Media

HELPING YOU NAVIGATE THE TECHNOLOGY JUNGLE!

ActualTech Media
www.actualtechmedia.com

In Partnership With SCALE COMPUTING

The Gorilla Guide To…

The Fundamentals of Hyperconverged Infrastructure

Authors
James Green, ActualTech Media
Scott D. Lowe, ActualTech Media
David M. Davis, ActualTech Media

Editor
Hilary Kirchner, Dream Write Creative, LLC

Layout and Design
Scott D. Lowe, ActualTech Media

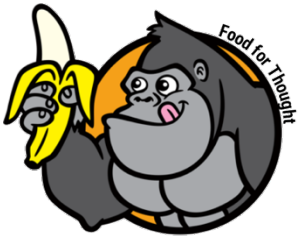
Copyright © 2017 by ActualTech Media. All rights reserved. No portion of this book may be reproduced or used in any manner without the express written permission of the publisher except for the use of brief quotations. The information provided within this eBook is for general informational purposes only. While we try to keep the information up-to-date and correct, there are no representations or warranties, express or implied, about the completeness, accuracy, reliability, suitability or availability with respect to the information, products, services, or related graphics contained in this book for any purpose. Any use of this information is at your own risk.

ActualTech Media
Okatie Village Ste 103-157
Bluffton, SC 29909
www.actualtechmedia.com

Entering the Jungle

Callouts Used in This Book

The Gorilla is the professorial sort that enjoys helping people learn. In the Schoolhouse callout, you'll gain insight into topics that may be outside the main subject but that are still important.



This is a special place where readers can learn a bit more about ancillary topics presented in the book.

Icons Used in This Book

Definition. Defines a word, phrase, or concept.

Skip ahead. We'll help you navigate you to the right place in the book to boost your knowledge.

Watch out! Make sure you read this so you don't make a critical error!

Chapter 1

Virtualization in the Modern Data Center

Consumer technology is constantly changing, and the same goes for the technology used in data centers around the world. Just as consumers are now able to buy a single smartphone device to do just about anything they can dream up, IT buyers can now acquire a single device or solution for just about any infrastructure service they need.

This single device/solution concept is made possible by faster and faster server hardware, virtualization, and hyperconvergence.

In this book, we'll start by briefly introducing virtualization as a concept, in case it's new to you, and then discuss the state of virtualization today. Later on, we'll introduce you to hyperconvergence and the way it solves many of the challenges that virtualization introduces. By the end of the book, you'll understand the various types of hyperconvergence architectures and what sorts of use cases benefit most from hyperconvergence.

Already a Virtualization Expert?

If you already have a good grasp of virtualization and the advantages it brings, as well as the challenges it introduces, feel free to skip this chapter and move on to Chapter 2, "Introduction to Hyperconvergence."

If you haven't started to virtualize your server infrastructure, or if you've started virtualizing but haven't yet achieved 100% virtualization, read this chapter before moving on. In this chapter, you'll learn about virtualization and become motivated to "virtualize everything," as is the norm at more and more companies.

What Is Virtualization?

If you work in an IT organization, surely you have at least heard about *virtualization.* Virtualization has changed the world of technology for large enterprises, small and medium-size businesses (SMBs), IT pros, and even many consumers.

Using software, virtualization abstracts away something that was traditionally physical and runs it as virtual.

But what does that mean, *virtual*? With server virtualization, software emulates hardware for the purpose of abstracting the physical server from the operating system. This abstraction is done using a special piece of software called a *hypervisor*. The hypervisor either runs on top of or inside an operating system (such as Windows Server or a Linux variant) and allows you to run virtualized servers on top of that hypervisor. Those virtualized servers are typically called *virtual machines*, or VMs. It is inside the VMs that you can install just about any guest operating system, applications, and data that you choose.

What companies large and small can do with virtualization is to take their existing physical servers, virtualize them, and run them inside VMs that run on top of a single host server. The result of this type of virtualization is that companies can consolidate many physical servers onto far fewer physical servers. In fact, some organizations can consolidate all their VMs onto a single host. Or preferably, as best practices would dictate, they can run all their virtual machines across two hosts in a cluster, storing virtual machine images on shared storage, so that one host could take over for the other host in the event of failure.

Instead of using the dictionary definition of *abstraction* to describe virtualization, most admins describe it with phrases such as:

- Virtualization allows you to run much, much more on a single physical server (host) than ever before.
- Virtualization allows IT organizations to do more with less.
- Virtualization is when you run your servers on top of software-based virtual machines instead of on hardware machines.

As you can see in **Figure 1-1**, with a single physical server (having its own CPU, memory, and storage I/O resources), you are layering a hypervisor on top in place of the typical server operating system. Then, on top of that hypervisor, you are running 2 VMs, each with its own CPU, memory, and I/O resources, so that you can install your own guest operating system to run applications and storage company data.

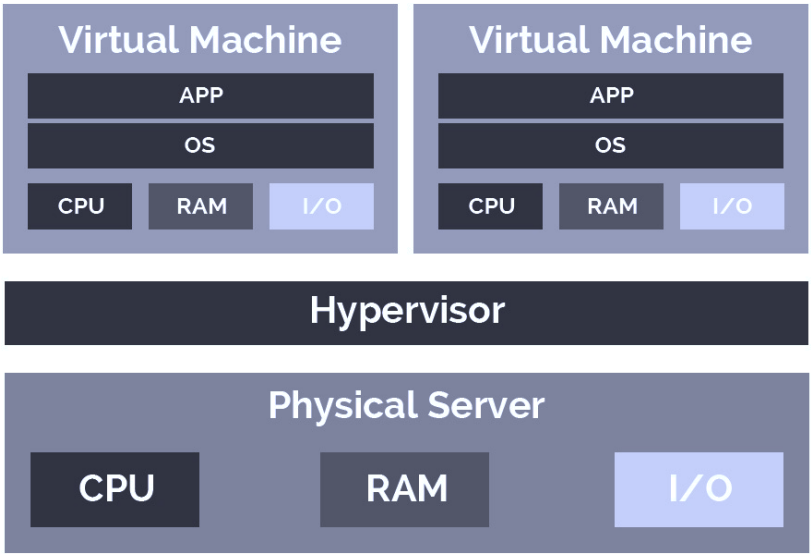


Figure 1-1: Server virtualization diagram

IT professionals have looked to virtualization to save them from some serious challenges. Namely, virtualization has helped to overcome availability challenges and increase operational efficiency.

How Virtualization Was Supposed to Change IT

It's great to look at server virtualization from the perspective of the business and show all the money that can be saved. After all, in many cases, some of that money saved can be used in the IT organization for other uses.

But what if you look at server virtualization from the perspective of the staff who administers the infrastructure? How does server virtualization change the daily life of the infrastructure admin?

Operational Efficiency

Admins are continually pushed to add more and more applications or support more users and devices. However, rarely are they offered any additional resources to manage and support all the additional infrastructure required. No additional administrative staff, no additional budget, and in many cases, not even any additional infrastructure to run the applications.

In other words, admins are simply expected to "do more with what you have… or with less." This is especially true at SMBs, which have inherently smaller budgets than large enterprises.

Server virtualization is one of the few solutions that can actually allow admins to accomplish this "do-more-with-less" goal.

Server virtualization offers far greater efficiency in administration because:

- **Virtualized servers (and VMs) are portable.** They can easily be moved from one server to another, and virtual hardware can be resized when new resources are needed — they can be cloned or copied to add more VMs.

- **Virtualized servers (VMs) are all managed from a single centralized management interface.** Monitoring, performance management, and troubleshooting are all far more efficient than having many physical servers to contend with.
- **By having many fewer servers, admins have fewer servers to keep current.** This is especially helpful when servers need updating (both hardware and software) or when troubleshooting, should the unexpected occur.

Technology Agility

End users expect admins to be able to bring up new applications or VMs within minutes and ensure that applications never go down.

Meeting those expectations is next to impossible with traditional physical servers; however, with server virtualization, admins can meet them easily.

With server virtualization, VMs are hardware independent and can be easily backed-up, replicated, restored, cloned, or moved from one server or site to another.

Server virtualization allows admins to create a library of VM images and spin up new VMs whenever needed.

Finally, VMs can easily be moved from server to server or site to site with without downtime in most cases.

Advanced Availability Features

Server virtualization also allows administrators to leverage more advanced data center functionality than would ever be possible with a purely physical data center. Here are some examples:

- **Virtualization backup.** Virtualization backup makes data protection easy, because it can easily back up only the changed blocks of a VM's disk storage and send them to tape. Protected

VMs can be recovered onto other servers as needed, and the underlying hardware is abstracted. As a result, VMs can easily be restored onto a very different physical server than the original.

- **Replication.** Replication can be done all in software for any VM or group of VMs that need offsite data protection.
- **Balancing resource consumption.** Resource consumption on the virtual infrastructure can be dynamically balanced within the cluster to ensure that every VM gets the resources it needs to run its applications.

Virtualization-Induced Challenges

As you've learned above, server virtualization immediately offers the IT organization numerous benefits. However, data centers rarely shrink. Data center footprints tend to grow over time, creating the need to add more virtualization hosts (physical servers) to provide resources to run more VMs. As the infrastructure and application criticality grows, so does the need for high availability. Availability assurance is one of the most popular features of most server virtualization hypervisors. With server virtualization, when a physical server fails, all VMs that were running on it can be automatically restarted on surviving hosts.

While high availability in server virtualization may be readily available and easy to implement, the same is not true for high availability for storage.

With virtual server host-to-host failover requiring shared storage to work, data center architects must utilize a shared storage system (SAN or NAS) to store the VM disk files on. High availability for server virtualization mitigates a host failure in short order, but offers nothing to mitigate any possible failure of the shared storage. With shared storage often being complex and expensive, the architecture that many server virtualization designs end up with is what's called the "3-2-1 design" (also known as the "inverted pyramid of doom").

The 3-2-1 design (shown in **Figure 1-2**) is when, for example, you have 3 hosts, 2 network switches (for redundancy), and 1 shared storage array where all data is stored. In the 3-2-1 design, the shared storage array is the single point of failure, meaning that if the single shared storage array fails, everything else in the infrastructure goes down as well.

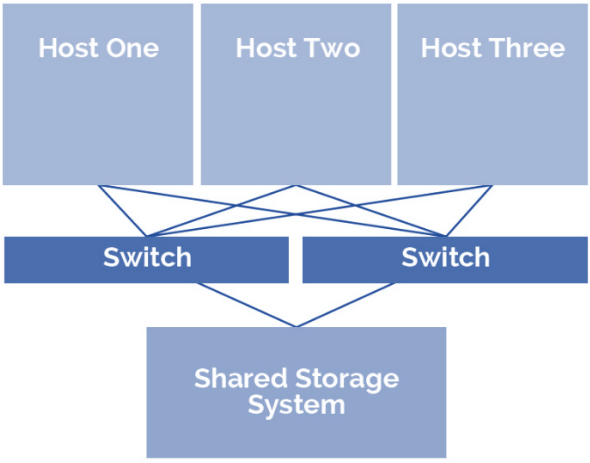


Figure 1-2: The 3-2-1 infrastructure design

Unfortunately, too many organizations are stuck in the 3-2-1 design even as the number of servers in their infrastructure grows well beyond three hosts. Even large companies with 50 or more hosts still use this "inverted pyramid of doom" infrastructure design simply because they can't afford the cost or handle the complexity to move beyond it. That's because, to move to a redundant storage infrastructure where you don't have the shared storage as the single point of failure, you must implement an infrastructure design similar to the one shown in **Figure 1-3**.

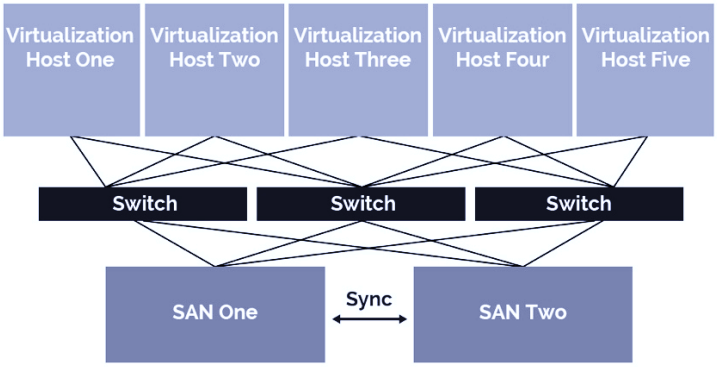


Figure 1-3: Virtual infrastructure with high availability for storage

With this design, you implement a redundant SAN or NAS array. When you do so, that array can eliminate the "1" in the 3-2-1 design architecture and ensure that critical applications running in the virtual infrastructure don't go down should the shared storage suffer a failure.

The first challenge in this infrastructure addition is, of course, the cost. Buying shared storage to begin with is a challenge for many businesses. The idea of buying redundant shared storage and paying essentially double the price is painful.

The second challenge with this design is the tremendous and undesired complexity that it brings to the data center.

Infrastructure admins commonly wear many different hats. Here are just some of the things an average infrastructure admin might have to deal with:

- Hypervisor administration
- Network administration with full redundancy
- SAN (e.g., targets, LUNs, multipathing) or NAS administration
- Hardware compatibility lists (HCL) management
- Patching and updating multiple systems and related firmware

- Dealing with multiple support groups when unexpected trouble occurs, which usually results in finger-pointing and inefficiency
- Maintaining multiple maintenance contracts, which is costly and time-consuming

Virtualization Is a Commodity

We're pushing in to the third decade of x86 virtualization being available to consumers, and into the second decade of server virtualization being utilized in the mainstream data center. At this point, most data center architects just *expect* virtualization. Many organizations have a "virtual first" policy, which essentially dictates that all workloads should be virtualized unless there's a good reason to do otherwise.

Accepting virtualization as a standard data center practice is great, and it provides tremendous benefits to IT organizations and to the businesses they serve. However, the complexity that eats away at growing virtualized environments can start to negate some of the effects that virtualization was adopted for in the first place.

What modern data centers truly need is a solution that offers them all the benefits of virtualization without all the complexities that come along with virtualization, once the virtual infrastructure grows and advanced functionality is added (such as high availability for storage). Hyperconvergence *is* that solution, and you're going to learn about that next.

Up Next

You have a good understanding of what virtualization is and how it is supposed to help your company. But you've also just begun to realize that even while virtualization solves some big problems in the data center, it also introduces some new ones.

In the next chapter, you'll learn about a data center architecture called *hyperconvergence* that leverages virtualization and software-defined storage (SDS) to take efficiency, agility, and cost savings to the next level.

Chapter 2

Introduction to Hyperconvergence

If you haven't been living under a rock, it's very likely that you've heard the term *hyperconvergence* over the past couple of years. Just like almost all IT buzzwords, it has become notorious for being overused and under-defined. Before discussing the value of hyperconvergence and how it can transform IT in your organization, you need to understand the term. So let's define the term *hyperconvergence* and briefly discuss the inception and drivers for it.

What Is Hyperconvergence?

At the highest, most abstract level, *hyperconvergence* can be understood to be the combination (or *convergence*) of many potentially disparate platforms into a single platform. In relation to the physical hardware, this means placing compute (CPU and memory) and storage (spinning disk and solid-state drives [SSDs]) into a single server. From a software perspective, this means that at the very least, all components of the system are managed from a common interface. Depending on the offering, this may be a custom user interface built by the manufacturer, or it could be an add-on or extension to the existing hypervisor management software.

Remember from Chapter 1 that the *hypervisor* is the software that allows multiple virtual machines (VMs) to run on a single piece of physical hardware.

Why is it called hyperconvergence?

The word *hyperconvergence* stems from the combination of the words *hypervisor* + *convergence* = *hyperconvergence*.

Its definition stemmed from an attempt to differentiate from the trend of *convergence* in general. It has taken on a life of its own and now is synonymous with combining many potentially disparate platforms into a single platform for running virtual machines (VMs).

A commonly understood definition of *hyperconvergence* states that it is "a platform that pools direct attached storage and eliminates the need for a storage array." While this is true, it doesn't tell the whole story. Physical hardware is an important piece of the puzzle, but the grander picture of hyperconvergence is really focused on the simplified management of the data center infrastructure. Hyperconvergence aims to eliminate siloes of management.

A characteristic feature of hyperconvergence is that it scales out in predictable, finite portions. These portions are often represented as building blocks. As an admin adds building blocks (also called *nodes*, *bricks*, or various other terms that mean "a single unit") to the cluster, all the relevant infrastructure components scale together.

The *hyperconverged infrastructure* (HCI) model is in contrast to an older data center model where storage capacity might be added at one point, and then additional RAM, CPU nodes, and so on would be added a few months down the road.

With hyperconvergence, all resources can scale at once. However, it is still possible to scale resources independently, if needed.

How Hyperconvergence Came to Be

The story of how hyperconvergence came to be is really a chapter in a bigger story about the cyclical nature of the data center. As you may have noticed, the data center tends to evolve in a cyclical fashion. What was once the standard is continually improved upon until nearly the opposite architecture is the standard. Then it is improved on again and looks very much like it used to a decade ago.

An example of this cyclical nature is end-user computing. As technology developed to allow multiple users to share a single computer, users connected with "dumb" devices that were simply a display and controls to manipulate the computer running in the data center. As technology evolved further, the desktop computer became the standard for end users, and everyone had one on their desk. Today, many organizations choose to use virtual desktop and thin or zero client devices for end-user workstations. This looks very much like the terminal server architecture of the past, but with a new breath of life.

The hyperconvergence story is the same. Once upon a time, in a data center a couple of decades ago, physical servers ran a single workload. They had an operating system and locally attached storage where the operating system and applications were installed. Each of these servers was managed individually.

Then, roughly a decade ago, the concept of a shared storage array was broadly adopted to allow for more efficiency and utilization. While efficiency was realized, the management of this infrastructure became more complex.

Hyperconvergence is the cyclical return to direct-attached storage, with the improvement being the distributed nature of that storage. In hyperconvergence, software-*defined storage* (SDS) technology allows the direct-attached storage to be pooled and managed as if it was shared storage. Leveraged in tandem with virtualization, this architecture allows for tens to hundreds of workloads to run on a handful of physical servers, and managing them is a breeze.

Leveraging Virtualization

Speaking of virtualization, we must not forget that virtualization is the catalyst that allows the creation of hyperconverged infrastructure. Without virtualization software (both compute virtualization and storage virtualization), the hyperconvergence model would not be possible.

You already saw in Chapter 1 why the case for server virtualization is so strong. Now, let's look at how combining a hypervisor with hypervisor-embedded storage can take virtualization to the next level.

By its very nature, virtualization of any kind abstracts a resource from a consumer. In the storage portion of hyperconvergence, each server in the cluster has a given amount of direct-attached storage. That means that the disks are connected to a storage controller inside a particular server. Those servers are all connected by a network fabric. Today, that network fabric is likely IP over 1 or 10 GbE. The faster, the better. However, cost is always a factor. The servers have direct-attached storage and yet can share data among themselves.

One layer up on the logical storage infrastructure stack sits the SDS technology. SDS is delivered in various forms and is outside the scope of this section; suffice it to say that this is where storage virtualization takes place.

Software-Defined Storage

For anything to be dubbed "software-defined," it must have a few characteristics. First, it involves some level of abstraction from the underlying resource. Additionally, the resource is administered and managed via policies. This policy-driven approach (as opposed to an imperative one) is what makes software-defined [anything] so powerful.

In the case of storage, a software-defined approach likely implies storage virtualization as well as policy-based management of provisioning, deduplication, replication, snapshotting, thin provisioning, and backups.

The SDS layer consumes the underlying physical storage and provides it to the workload that will consume it. In most cases, this is a VM. This is the key point to understand about virtualization and its role in hyperconvergence: the workload (the VM in this example) is unaware of, and indifferent to, the underlying resource it is consuming. It just gets what it needs from the virtualization/abstraction layer.

This explains how a VM running anywhere in the cluster can leverage storage that is local to the node it's running on or storage that is remote (residing on another node); it doesn't know or care where the data is stored.

The marriage of compute and storage by building a platform that delivers a hypervisor as well as hypervisor-embedded storage unlocks heretofore unrealized possibilities in data efficiency, availability, scalability, and operational simplicity.

Single Vendor Acquisition and Support

One of the more complex aspects of day-to-day life in the IT world is keeping track of which vendor provides which service or product. Knowing who to call at any point in time can be a real challenge. Plus, finger-pointing between vendors when issues arise is all too common.

Hyperconvergence solves this problem to some degree:

- **There are fewer vendors involved overall.** This makes managing the vendor relationship simpler moving forward. However, hyperconvergence doesn't replace the entire data center, and there are still other components and vendors to keep track of (the Internet service provider [ISP], for example).
- **The procurement process is streamlined.** By having a good portion of the IT infrastructure provided by the same vendor, the hassle of trying to set up new vendors is eliminated.
- **Efficiency is increased.** Doing business with a single vendor can have a major impact on the turnaround time for projects

and on the overall stability of the infrastructure. The new vendor doesn't have to be brought up to speed on the environment or try to unbury details about the IT organization's history.

Lower Cost of Ownership

Another of the major business drivers for hyperconvergence is the lower cost of ownership overall. Compared to an infrastructure with a monolithic SAN, disparate backup and replication systems, and disparate WAN accelerators, a single hyperconverged platform provides the following cost savings:

- **It's cheaper to acquire and maintain over the life of the platform.** Because the platform scales and upgrades so well, the life of the platform could turn out to be quite long.
- **Staffing costs are kept lower.** The cost of IT professionals' salaries is certainly at the front of many business owners' minds, and hyperconvergence helps keep those costs down as well. There is no need to hire an expensive specialist for different areas — the operating costs for the department can be kept low. This is made possible by the simplification inherent in the hyperconverged model.

Next Up

In this chapter, you learned about the technical merits and cost savings of HCI. If the technical merits of hyperconvergence haven't quite sold you, perhaps some dollar signs will get you off the fence!

In the next chapter, we'll be taking a closer look at the technical merits of hyperconvergence. You may be surprised at how much can fit in a single platform.

Chapter 3

Benefits of Hyperconvergence

Now that you know what hyperconvergence is, let's talk about the benefits of it. Some of the higher-level benefits of hyperconvergence are obvious. For example, because everything is virtual (servers and storage) and managed from a single point of control, your management burden is eased.

However, there is more to hyperconvergence than simplified management. Hyperconvergence takes the benefits gained from virtualization and expands them to an exponentially higher degree.

Read on to find out what you'll experience once you've implemented hyperconvergence.

Management Efficiency

Storage arrays are not known for being easy to manage. After all, larger companies hire a storage admin (or a team of them) to learn, obtain certification on, configure, administer, monitor, and troubleshoot complex storage arrays.

Just like when modern software bypasses decades old inefficient solutions (causing market disruption and happiness for end users), hyperconvergence can eliminate that costly, complex, and aging storage infrastructure. When the traditional storage infrastructure and its painful management tools are eliminated, one of the end results is that hyperconvergence provides tremendous management efficiency to small and medium-sized businesses (SMBs) and enterprises alike.

Hyperconvergence allows you to configure, administer, monitor, and troubleshoot your storage infrastructure from the same interface that you already use to configure, administer, monitor, and troubleshoot your virtual infrastructure.

Hyperconvergence also eliminates complex management paradigms of traditional storage, such as LUN striping and RAID configurations, by presenting all the storage in a single, shared storage pool — think of it as a giant C:\ drive for all the company's storage. Additionally, hyperconvergence eliminates complex and time-consuming storage connectivity configurations that are typically required with legacy storage.

With hyperconvergence in place, you don't just eliminate traditional storage infrastructure and gain a single pane of glass; you gain far more efficient management constructs, configurations, monitoring, and troubleshooting. The result is that hyperconvergence is easier and less time-consuming to manage than traditional infrastructure.

Data Efficiency

By pooling all storage into a single shared resource pool, you're able to reclaim overhead that traditionally had to be allocated to each LUN. With storage reclaimed you'll be able to manage the capacity and I/O throughput more efficiently than ever before — and increase return on investment (ROI) while doing it.

Additionally, hyperconvergence offers thin provisioning for all virtual machines (VMs) created, allowing you to only use storage capacity when applications need it.

One of the secrets to the data efficiency, high availability, and high performance of hyperconvergence is how data is stored. With hyperconvergence, new data is striped and mirrored across all disks in a cluster (**Figure 3-1**). Multiple copies of the data mean that the read performance can be improved, and in the event of a failure, data is always available.

IT organizations need the greatest efficiency, performance, and availability that they can get for their infrastructure investment, and hyperconvergence offers exactly that.

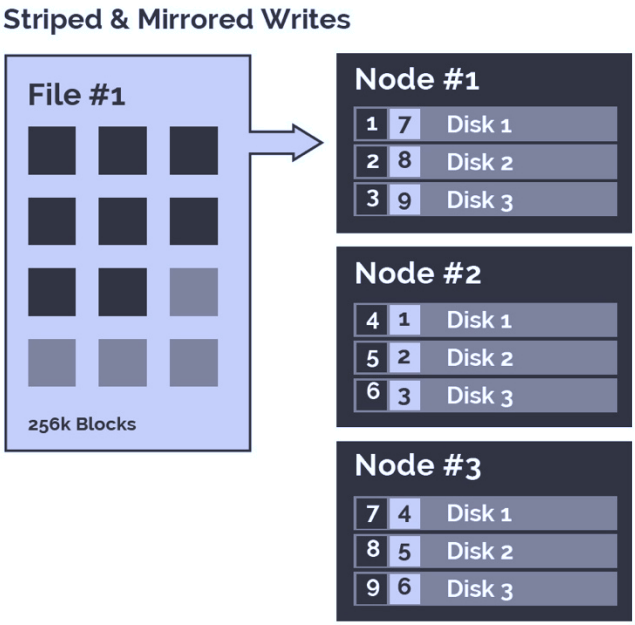


Figure 3-1: Distributed storage offering striping and mirroring of data

High Availability

One of the most common infrastructure struggles of data center architects is storage availability. In order to achieve high availability with traditional storage systems, you typically have to buy a second identical storage system and then perform a bunch of complex configurations to make them highly available (see **Figure 3-2**).

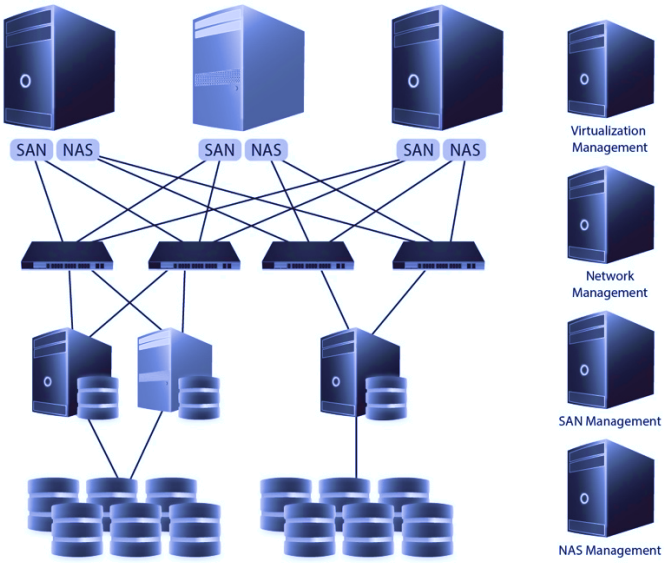


Figure 3-2: Traditional infrastructure with highly available storage

Many organizations don't have the time or the money to perform advanced configurations or to buy fully redundant storage infrastructure (when they likely struggled to buy the infrastructure in the first place). What happens is that most of them use the 3-2-1 design — 3 hosts connected to 2 switches, connected to just 1 SAN or NAS that stores all their data. This is an unfortunate configuration for so many businesses who think that they can't afford highly available storage and are forced to take their chances on such a high-risk design.

Hyperconvergence offers highly available storage *built in*. What that means is that there is no chance that losing either a single disk or an entire node can take down their infrastructure.

When it comes to the high availability of the compute infrastructure, protecting from server failure is also built in with hyperconvergence. As shown in **Figure 3-3,** when a physical server in the HCI fails, no data will be lost and all VMs that were running will be automatically restarted on another host in the cluster. (The data that was on the failed node will be distributed across other nodes in the cluster to ensure that high availability of the storage is maintained.)

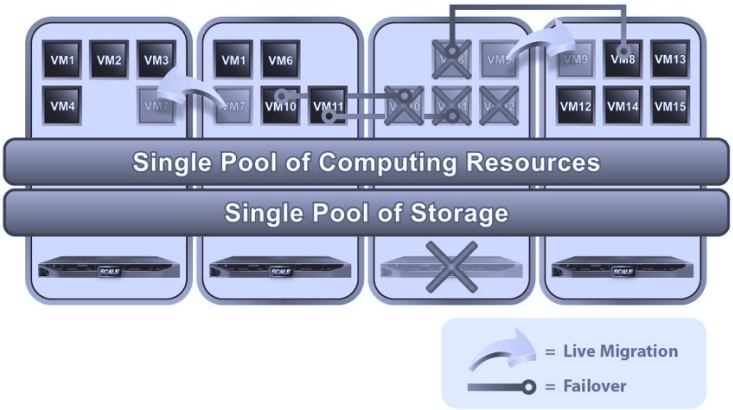


Figure 3-3: Compute high availability with hyperconvergence

Scalability

In the traditional model, when a business needs to add a new application, they usually add a new physical server and, if needed, directly attached storage. This model is inefficient because it's a 1:1 application-to-server-or-storage model where the server and storage usually spend their time mostly idle. As the infrastructure grows, it's also painful to scale the management of the infrastructure.

Server virtualization offers far more efficient utilization of servers and storage. For example:

When you need to add a new VM there is usually capacity available to add it to the existing virtual infrastructure without having to add any new servers or storage.

With storage being in a single, shared resource pool and with thin provisioning, there are usually storage resources available to add a new VMs (and their associated virtual disks).

Hyperconvergence takes the scalability of virtualization to a whole new level. With hyperconvergence, all storage across all hosts is combined into a single shared resource pool, which maximizes the amount of storage available. It also makes it easy see total capacity and how much

is remaining. When a hyperconvergence cluster is truly out of capacity, simply add another hyperconverged server or node and all resources will be immediately expanded. This is because the node, which adds additional CPU, memory, storage capacity, and storage throughput, is extended as well.

In many cases, the licenses to add additional highly available nodes are included, and adding those nodes is as easy as plugging the node into power and having the network to scale up the infrastructure (bring up additional capacity).

Data Protection

Besides management and data efficiency and highly available storage and compute, other advanced features are commonly built in — including data protection.

Data protection in the infrastructure comes in several different forms. We've already discussed the striped and mirrored writes that hyperconvergence provides to distribute the data and protect it in the event of failure.

Another critical data protection feature used to get data offsite is *replication.* Traditional storage arrays offer data protection with a requirement to purchase redundant arrays and replicate entire LUNs.

Many hyperconverged solutions have data replication built in. What that means is that you can select an individual VM, or group of VMs, and replicate those to a secondary data center or from a remote office to a centralized data center. The result is that critical applications can be protected from disaster using the all-in software and built-in replication included in hyperconvergence.

Comparing Traditional Storage and Hyperconvergence

When comparing the differences between traditional storage infrastructure and hyperconvergence, you'll find that hyperconvergence offers the following 5 benefits over traditional infrastructure:

- **Single point of administration**, monitoring, and control for storage, servers, and virtual infrastructure
- **Lower cost infrastructure** due to both the elimination of dedicated SAN/NAS and the greater efficiencies of management
- **Highly available storage and compute** that's built-in and available should a node fail
- **Included data protection** in the form of wide-striping and replication
- **Reduced cost** (through eliminating dedicated storage maintenance and support contracts), greater utilization of servers and storage, increased uptime, and inclusion of advanced features

Up Next

Now you understand the numerous ways that hyperconvergence can help data center infrastructures to become more efficient, more scalable, more available, and more cost-effective.

In the next chapter, you'll learn how different hyperconvergence architectures work and how to select the best hyperconvergence architecture for your needs.

Chapter 4

Hyperconvergence Architecture

In the world of hyperconvergence, the storage architecture can have a dramatic impact on the overall solution. As one of the main differentiators between *hyperconverged infrastructure* (HCI) and a more traditional data center architecture, storage can be the tipping point between a good hyperconverged platform and a great one.

There are as many ways to do storage for hyperconvergence as there are hyperconvergence vendors on the market. The storage architecture does, however, seem to fall into two distinct categories: VSA-based or hypervisor-embedded storage virtualization. Each architecture direction has advantages, and the choice between the two isn't black and white.

A *virtual storage appliance* (VSA), is a fancy way of saying that a virtual machine (VM) consumes the physical storage provided to the hypervisor and allows other VMs to access that storage. This VM is running in the general pool of infrastructure resources and consumes resources in the same way as the rest of the VMs. In this case, the VSA resides directly in the I/O path, and other VMs read and write data through it.

In contrast to the VSA-based model is *hypervisor-embedded storage virtualization.* This is where a process that runs at the hypervisor level is responsible for managing the physical storage and presenting it for access by VMs. In this case, no VM is in the I/O path; rather, the hypervisor consumes additional resources to manage the storage devices.

In either situation, storage virtualization provided by the hyperconverged platform has a stark advantage over storage provided by a third party, in that it allows the HCI vendor control over the entire infrastructure stack. Virtualization is a complex beast, and the more data center components that can be managed by a single platform, the more awareness those components can have of each other.

Virtual Storage Appliances

Let's start by looking at the way a VSA functions. In a mature hyperconverged system that uses VSAs to manage storage, a VM will access data by contacting the local VSA (the one running on the same hypervisor node) without traversing the external network (where *external* means "outside the hypervisor").

That VSA will then directly access the physical disks in the node to retrieve the data and deliver it back to the requesting VM.

Presumably, the VSA may also have that data cached, in which case it would serve the request immediately with no need to access the physical disks. Expensive flash storage and liberal use of DRAM make this method of providing storage possible.

While the VSA-based design is certainly workable, it does present some interesting challenges:

- **The VSA runs alongside other production workloads and is subject to the same contention risks as the other VMs.** While modern hypervisors have mechanisms to account for this and guarantee resources, that sort of special treatment must be configured by the admin. In a market that is begging for simplicity, adding requirements for special configuration of a VSA virtual machine introduces unnecessary complexity. It also tends to be incredibly fragile. Making an improper change to the VM could impact performance for a large number of VMs.

- **A VSA is generally unaware of the actions and state of the hypervisor that is running it.** This means that maintenance operations, upgrades, and the like are not coordinated as gracefully as they could be with hypervisor-embedded storage virtualization.
- **VSAs may need to be upgraded separately (whereas hypervisor-embedded virtualization solutions will be upgraded as a part of the overall hypervisor).** If the storage layer is separate from the hypervisor, this is an extra variable introduced to the upgrade and must be checked against the version compatibility list. Upgrading hypervisor-embedded storage solutions is a less complex endeavor for the admin.
- **The VSA model may consume significantly more resources**. Due to the need to create what amounts to a full storage appliance inside each VSA (which is then duplicated by the number of hosts in the cluster), VSAs for hyperconverged clusters frequently consume tens or even hundreds of gigabytes of RAM and multiple CPU cores. This consumed memory and CPU cannot be put to work with other virtual machines.
- **The VSA introduces latency into the I/O path**. By the time a virtual machine is accessing storage on a VSA, it's performing I/O against an abstraction of an abstraction at best; sometimes the recursion goes even deeper. All these layers introduce latency as the I/O request is handed off from one component to the next. Eventually, when faster storage technology like NVMe becomes mainstream, VSAs could stand in the way of maximizing the potential of underlying storage platforms.

One advantage that a VSA-based approach to storage has over hypervisor-embedded storage virtualization is that it tends to support multiple hypervisors. If the storage virtualization platform is in the hypervisor kernel, it won't be able to work across platforms. For many customers, this flexibility won't matter, but it is an important consideration for a select few.

Hypervisor-Embedded Storage Virtualization

Hypervisor-embedded virtualization and management of storage has advantages over VSA-based storage for a number of reasons:

- **Hypervisor-embedded storage virtualization removes a layer of complexity from the end user — the IT admin.** Hypervisor-embedded storage virtualization may in fact be more complex and challenging to the developers creating the product; however, this complexity is kept from the admin. Simplicity is the name of the game when it comes to hyperconvergence, so removing an area of potential configuration error and troubleshooting focus from the admin is a no-brainer.
- **A higher level of interconnectivity is possible.** When the manufacturer controls and develops the entire stack from the VMs down to the hypervisor, a level of interconnectivity is possible that would just not be attainable if a number of products from disparate vendors were brought together to accomplish the same task. This holds true in regard to storage virtualization and the kernel as well: if the storage system is integrated into the hypervisor kernel, it can also be aware of, and interact directly with, that kernel. This means increased awareness of hypervisor operations, such as maintenance, and increased insight into failures, isolations, partitions, and the like. The VSA-based model, which does not have this level of access to the kernel, is left standing out in the cold.
- **Beyond awareness, the hypervisor-embedded virtualization platform also has potential efficiency benefits.** VMs writing to a local disk can actually write to a shared memory segment where there is no need to copy the data before the kernel writes it to disk. Based on the configuration

and technologies in use, hypervisor-embedded virtualization can be more efficient.

Which Is Better?

Both of these storage models are good options in a hyperconverged architecture, and there are use cases that each would be a better fit for. However, general purpose environments will benefit more in the long run by leveraging a hypervisor-embedded approach to storage virtualization, particularly due to the simplicity it affords when compared to the VSA model.

Up Next

Speaking of use cases, the next chapter will focus on some specific use cases for hyperconvergence as an architecture. Keep reading to learn some good starting points for implementing HCI, as well as business challenges that the HCI model can address.

Chapter 5

Hyperconvergence Use Cases

Hyperconvergence is mature enough as an architecture that it can be a fit for almost all workloads. In the modern data center, there is no shortage of different workload categories, and each one carries its own set of requirements. Amid this broad array of potential use cases, however, are a few distinctly easy wins when it comes to where *hyperconverged infrastructure* (HCI) fits in.

Often, especially in the small and medium-size business (SMB) space, an organization is just moving from physical-based server provisioning to a virtualization-based model. Adding hyperconvergence to the mix as a part of this process makes the overall transition easier. Virtualization can be a complex undertaking, and the simplicity of HCI helps to lighten the burden. One of the main advantages is the freedom from having to integrate and manage components from a variety of vendors to complete a project.

In a broad sense, the industry uses the term *server virtualization* to describe the workload made up of general-purpose servers like directory servers, file servers, web and application servers, and so on. Server virtualization is a project that is particularly suited to hyperconvergence.

Another venue that seems to provide an easy win for HCI is the remote office. Due to physical space constraints, manageability requirements, and a general lack of on-site IT support, hyperconvergence can often suit this environment better than any other design.

Finally, virtual desktop infrastructure and disaster recovery projects are potentially ripe for leveraging hyperconverged architectures. Let's take a look at each of these four use cases in a bit more detail.

Server Virtualization

For background, let's quickly examine a server virtualization project in the pre-hyperconvergence era:

- First, you would calculate the estimated workload, as well as the physical servers, storage, and network infrastructure that would run it.
- Next, when all your gear showed up, you would install a hypervisor on the servers, initialize the storage array, and configure the network and storage protocols to connect the two. The hypervisor management software would typically be installed on a dedicated server.
- Finally, once the physical servers were clustered, the storage was provisioned, and the network segments were plumbed in, you could begin provisioning the server virtual machines (VMs).

For the sake of this example, let's imagine that your environment requires 3 hypervisor hosts, 2 switches, and 1 small storage array. For an outside consultant, the initial setup of all this equipment might take one to two business days. For an inside IT staff with no prior experience with this architecture, it could take a few days to a few weeks. All of this must take place before provisioning any VMs.

This same level of effort is also required in the future for projects such as expanding the system or migrating to a newer generation of hardware.

Hyperconvergence for Server Virtualization

Contrast the previous example with this one:

- You place your order after selecting a few simple specifications — no scientific calculators or pivot tables required.
- When the gear arrives, you unbox it, install it in the rack, and physically connect it.

- You then provide some basic information, such as what it should be named, what networks and IP addresses to use, and a few other basics. The cluster immediately begins initializing itself. Within about 15 minutes, the cluster is ready to use.

It's hardly time for a coffee break on day one, and the VM creation and migration part of the project can begin immediately. Simply log in to the management interface and create VMs in a few clicks.

Once the project is complete, administration retains the same level of simplicity as the initial setup. Day-to-day operations don't require an enterprise storage guru to be on staff. The storage performance, however, rivals any general purpose standalone storage array.

What This Means

Especially in the example of an organization that is adopting virtualization technology for the first time, doesn't the second scenario sound much more approachable? And, indeed, this is what should be expected when deploying a modern HCI solution.

Edge Computing

Edge computing is the practice of running applications and processing data as close to the source of the data—and often the user—as possible. As the volume of data that is generated increases, the problem of data gravity becomes more and more prevalent. In response, many modern architectures call for compute power in remote locations to process data rather than transporting all the data back to the primary data center for processing.

Data Gravity

Data Gravity is a clever analogy that compares data center technology to nature. As in nature, objects with a greater mass exhibit more gravity. With regard to data, as the mass increases (more data is stored), it becomes more and more difficult to move it around. For example, it takes longer to copy a 1 GB file from one site to another than it does to copy a 1 MB file. Edge computing is the result of this figurative gravity pulling the applications ***to*** the data.

Some examples of places edge computing is starting to become common are:

- **Industrial Internet of Things (IoT).** For example, in a remote manufacturing facility where many of the machines are monitored and controlled by embedded sensors, the high latency and low bandwidth back to the primary data center could make it more practical to run some compute local to the facility.
- **Medical equipment systems.** In a healthcare scenario which is generating massive image files and data sets from test results and patient monitoring, it can be much more efficient to run applications at each clinic or branch hospital rather than in the primary data center.
- **Ships and Oil Rigs.** Any of these "small city in the middle of the ocean" scenarios will require some sort of edge computing. The available latency and bandwidth from satellite connectivity is simply insufficient for the amount of data being generated today.
- **Remote Office/Branch Office.** Limited on-site IT staff and lack of affordable high-bandwidth connectivity often drives compute out to ROBO settings today. In many parts of the world, it's more affordable to run some compute locally than

it is to procure a robust enough connection back to the primary data center to run the workloads there. The end user experience is often impacted by the choice to run workloads locally or in the primary data center.

Hyperconvergence in edge computing use cases is often an easy win because the simplicity and elegance of hyperconvergence makes it possible for admins to take the kit out, deploy it, and leave—without staying onsite for a week to do the deployment. Plus, there is no need to bring in a whole team of specialists to configure the different infrastructure components.

Beyond network connectivity which was mentioned numerous times above, there are at least two other constraints that edge computing use cases commonly have that aren't present when doing server virtualization in the data center.

Physical Space

The first constraint is physical space. In the central office, IT likely has a designated space where all the infrastructure components reside. This space has been specially designed to meet the needs of the infrastructure. Special cooling, power, and fire suppression may have been installed, depending on the scale of the infrastructure.

Except in abnormally sizable operations where a branch office might have hundreds of staff members, this typical data center scenario is not the situation in most edge computing deployments. Instead, the infrastructure needs to be deployed in a closet near the back of the office or, in some very unfortunate circumstances, underneath a spare desk.

Hyperconvergence conquers this space challenge. Especially with high-density nodes, a very large infrastructure for most edge settings could fit in an 8U rack.

IT Staff

The second challenge that hyperconvergence addresses in edge computing settings is that of IT staff. Except for unusually sizable operations, most edge environments don't have dedicated IT staff. Most commonly, they have a person or two who are known to be technically savvy.

These couple of folks are the ones IT calls when they need physical help with the infrastructure because these employees at least understand the technology enough to be helpful.

With a lack of trained, proficient IT resources in the remote office, the gear must be simple, run well, and require very little in the way of on-site support. With a hyperconverged solution, the infrastructure is so simple that there would be nothing for an on-site IT resource to do!

Virtual Desktop Infrastructure

Virtual desktops are one of the IT services responsible for making hyperconvergence as mainstream as it is today. Early on in the life of hyperconvergence as a paradigm, HCI proprietors used VDI initiatives as a foot in the door for hyperconvergence. Once organizations got a taste of how simple and scalable HCI is, they began adopting it in other parts of their infrastructure.

The simple reason that VDI is such a slam dunk for hyperconvergence is that virtual desktop environments grow in predictable increments (users), and so does hyperconvergence (nodes). The two pair well together because computing the number of VDI users that a given node configuration can support is pretty straightforward. Once you've got that figure, it's simple division to determine how many nodes you need to onboard a given number of additional users.

Let's say, for example, that you deployed a four-node cluster where each node can support roughly 75 desktops. It's simple to tell the business leaders that the infrastructure has capacity for 300 users. Should the needs change and require support for 400 users, that's no problem. **Figure 5-1** provides you with a look at how this might work in practice.

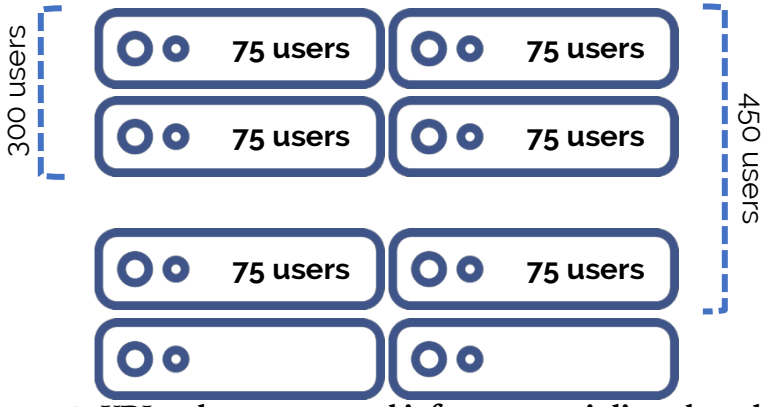


Figure 5-1. VDI on hyperconverged infrastructure is linearly scalable

You already know that you simply need to purchase two additional nodes and add them to the HCI cluster. You'll procure and deploy the new nodes with nary more than a flick of the wrist, and capacity will have increased from 300 to 450. All 100 additional users can be served, and you'll look like a wizard!

Disaster Recovery

Building a disaster recovery (DR) site is an imposing charter — financially, architecturally, and operationally. The DR site of old commonly looked like the little brother of the primary site. If the primary site held 10 hosts, 2 switches, and some shared storage, a common DR site configuration would have been to deploy 6 to 8 hosts, 2 switches, and shared storage in a remote location. In the event of a site-wide natural disaster, a building fire, a fiber cut, or any other imaginable outage scenario, the DR site would have just enough capacity to bring up all the critical business services and allow operations to continue.

The problem with the traditional approach is that IT departments today are already strapped for resources. Administrators don't have enough hours in the day to satisfy all the requirements of the primary site in a timely manner. Asking them to build a replica of the primary site up the road is quite a tall order given the utilization of most IT teams today.

Hyperconvergence can help, however. The operational simplicity that hyperconvergence brings to deployment, management, and site-to-site replication can make a multisite architecture much more approachable, both in terms of cost and complexity.

Data center outages are responsible for astronomical losses in some organizations, and at the very least, an unplanned outage might force you to give a bunch of paid employees the day off. Whatever the cost of an outage might be for you, avoiding a disaster recovery strategy because of the expense or because of the operational challenge is likely to ultimately backfire and cost *more* time and hours than doing it right at the outset. By taking advantage of hyperconvergence at one or, preferably, both sites, the undertaking looks a lot more reasonable.

Up Next

Now that you understand where hyperconvergence might be a good fit and what challenges this architecture solves, the stage is set to look more closely at how hyperconvergence can change the way we think about IT. In the next chapter, we'll look at some of the many ways hyperconvergence can help us to improve the way we do IT.

Chapter 6

Thinking Differently About IT

In the last chapter, you learned about a few low-hanging-fruit use cases for hyperconvergence. In almost any use case, there are a number of ways by which hyperconvergence stands to change the nature of your IT business. Let's look at a few of them.

Single Point of Administration

Pre-Hyperconvergence

Even if you're familiar with administering an environment with an overwhelming number of disparate systems, it's important to realize that even in a data center with a virtualized infrastructure, there are often no less than three different portals or interfaces from which to manage the IT infrastructure.

Even in a barebones environment, there would be:

- Hypervisor management
- Storage administration
- Backup administration

Three different points of management may not seem overwhelming to start with, but as the environment grows, it is rare that the IT department won't have an increasing list of systems and responsibilities. It will all become harder and harder to manage.

How Hyperconvergence Helps

With hyperconvergence, this whole problem of management sprawl goes away. By the very nature of hyperconvergence, the disparate systems are combined into one.

Administration of a *hyperconverged infrastructure* (HCI) takes place from a single management interface. From there, virtual machines (VMs) are created and managed on the automatically provisioned pool of storage resources, backup and replication is configured and administered, and more.

Single Point of Acquisition and Support

The same problem that exists with managing a disparate infrastructure from an administration standpoint also exists from a vendor management standpoint.

Pre-Hyperconvergence

Relying on multiple vendors could result in the following:

- **Costs.** If different products are used for virtualization, servers, storage, backup, and so on, the overhead of keeping up with each vendor grows with every additional infrastructure component.
- **Relationship Management.** Each new vendor relationship must be established, developed, and maintained. The fact that a different vendor is responsible for each component means that support contracts renew independently, and managing support contracts this way can be a full-time job.
- **Finger-pointing.** A given support request could be fielded by a variety of vendors in the infrastructure stack, and finger-pointing between the vendors is likely to ensue if the resolution turns out to be a tricky one.
- **Challenging hardware compatibility.** Hardware compatibility lists must be checked for each component any time an individual piece is changed or updated.

- **Time.** Even in a relatively small environment, all this vendor management can be exhausting and waste precious time.

How Hyperconvergence Helps

Hyperconvergence comes to the rescue here as well. The *hyperconverged-solution vendor* is a single point of acquisition, maintenance and renewals, and technical support.

Now, instead of having to consult a dense spreadsheet to find out which vendor is responsible for each component, what your customer number is, and who your account manager is, there is one vendor, one support phone number, and one customer account number that will identify you. On a normal day, this is quite nice. During an unplanned outage or critical failure, this is invaluable.

High Availability for Storage

In an environment where physical server provisioning is still the norm, a few metrics leave something to be desired. Cooling and the power bill come to mind, but one of the primary drivers for virtualization in many organizations is high availability.

Let's use a file server as an example. In a physical server environment, the file server is run by a single physical server and perhaps a small RAID-5 array of direct-attached storage (internal to the server). In many businesses, everyone in the office depends on this file server.

What happens if the motherboard in the file server fails? Suddenly, everyone in the office is without their files.

Virtualization and shared storage on a standalone storage array can solve this problem. The virtualized workload can move freely between hypervisor hosts, and the storage is highly available thanks to multiple controllers and RAID-protected volumes of disks. In this case, if a physical server fails, it's no problem. All the business-critical workloads, such as the file server, can be restarted on a surviving node, and business

can resume as usual in a matter of minutes. The potential downside of this architecture on its own is the cost and complexity of implementing such a solution.

Hyperconvergence allows for this same level of increased stability without all the complexity of the aforementioned infrastructure. By using integrated *software-defined storage* (SDS) technology to pool direct-attached storage into a volume that is accessible by the whole cluster, high availability is achieved in a hyperconverged footprint.

If a node in the hyperconverged cluster is to fail, the surviving nodes will continue to run the business-critical workloads.

Disaster Recovery

Speaking of critical failures and unplanned outages, isn't protecting the IT organization from these disasters a cumbersome and often frustrating process?

The traditional method of protecting an IT environment from loss in the event of a catastrophe is to install agents in the guest operating systems and back them up over the network using backup software that writes the backup data either to disk or to a tape that will be shipped off-site to a safe location.

In more sophisticated environments, running workloads are also sometimes replicated to a remote site, meaning that a redundant copy of a running workload is transferred to a remote site and made available for use in the event of a disaster. Typically, a single provider manages both backup and replication; however, this provider is not the same one providing storage or hypervisor management.

Hyperconvergence makes disaster recovery (DR) easy.

Backup and Replication

The first of two advancements that make it easy is the way that virtualization impacts backup. By virtualizing the infrastructure,

How to calculate TCO

Total Cost of Ownership is the analysis of the direct and indirect costs of a product or system over the lifetime of its use. Estimating TCO for IT projects can be especially revealing because operational complexity and ongoing support and maintenance costs can make a solution that looks appealing at the outset downright unaffordable in the long term.

When it comes to deploying a solution like hyperconverged infrastructure, it's important to look at TCO because there are many more variables in play that just some hardware and a hypervisor. Consider the following lifetime costs as compared to a traditional infrastructure:

Procurement Costs

- Base hardware
- Supporting networking equipment
- Hypervisor licenses
- Software features like deduplication and replication
- Backup software
- Integration services

Operating Costs

- Training and certification
- Ongoing support time
- Troubleshooting time
- Data center utilities (electricity, floor space, etc)
- Downtime/outage costs
- Support/maintenance contracts

Upgrade Costs

- Integration services
- Rip-and-replace upgrade costs (vs. granular scale-out)

As you can see, there's no small number of factors that comp[illegible] TCO of a data center architecture. When deciding on any scenario, be sure to keep an eye on both the direct and indir[illegible]th immediate and into the future.

Even if a legacy infrastructure and an HCI infrastructure would cost the same to acquire, the ongoing expenses to pay for power, cooling, physical space, and — most importantly — the cost of IT staff to spend hours maintaining the equipment will justify the hyperconverged option. The HCI option is also easier to design and purchase to boot!

That's a Wrap

We hope that you've had fun reading this Gorilla Guide and that you've learned what hyperconvergence is and how it can help you. By now, you know that by leveraging virtualization, software-defined storage (SDS), and commodity hardware, hyperconvergence solutions can fundamentally change the IT organization by allowing them to eliminate the storage management silo as well as its associated costs and complexities.

We hope you have seen the tremendous value that hyperconvergence can bring IT practitioners and data centers of all sizes. We also hope that you won't just learn about hyperconvergence but will also take the next step to evaluate hyperconvergence and see the value of hyperconvergence for yourself! Go forward and hyper convert!