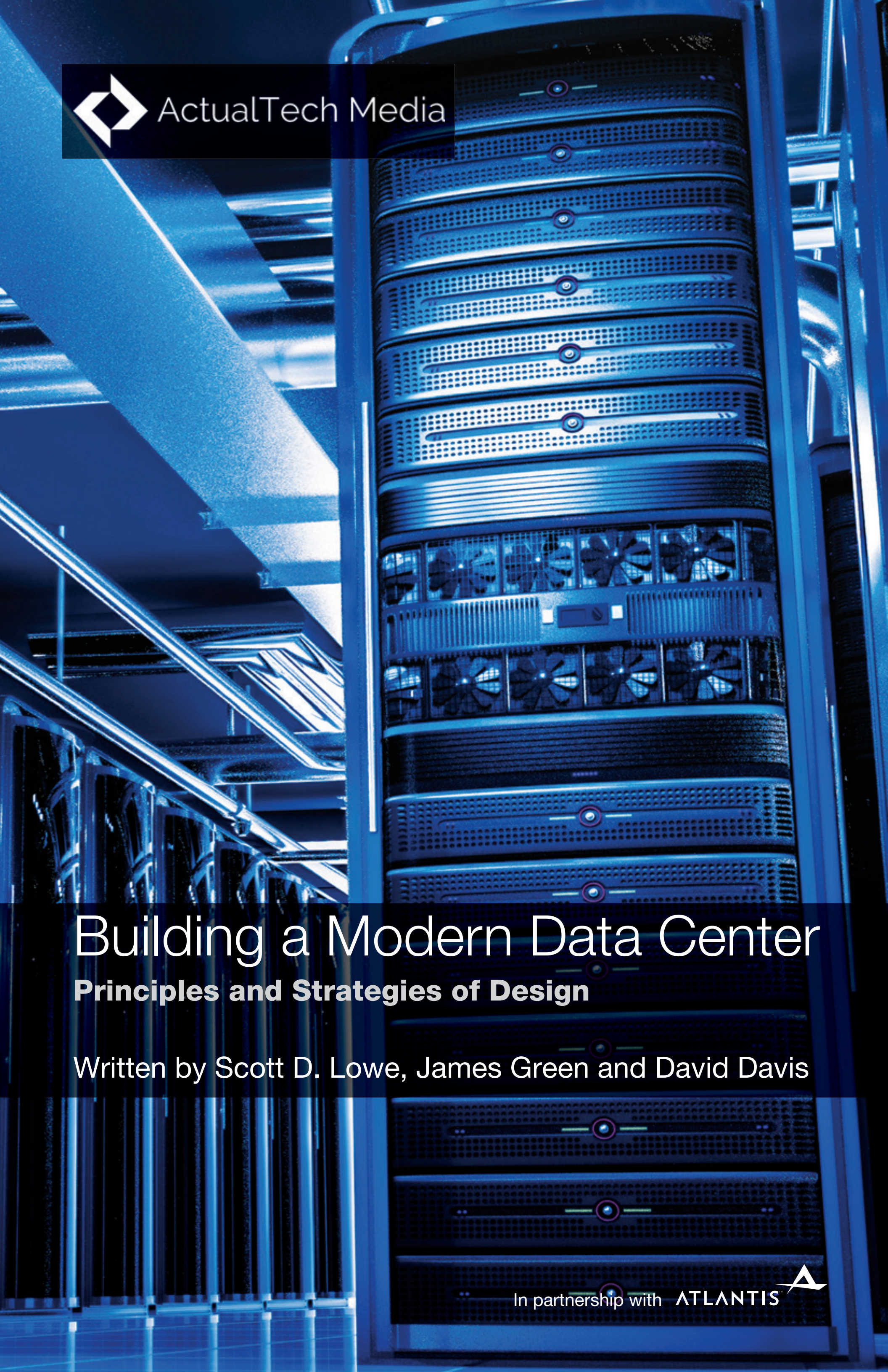


ActualTech Media

Building a Modern Data Center

Principles and Strategies of Design

Written by Scott D. Lowe, James Green and David Davis

In partnership with ATLANTIS

Principles and Strategies of Design

Written by
Scott D. Lowe, James Green and David Davis
Partners, ActualTech Media

ActualTech Media

Building a Modern Data Center Principles and Strategies of Design

Author:	Scott D. Lowe, ActualTech Media James Green, ActualTech Media David Davis, ActualTech Media
Editor:	Hilary Kirchner, Dream Write Creative
Cover Design:	Atlantis Computing
Layout:	Braeden Black, Avalon Media Productions
Project Manager:	Geordie Carswell, ActualTech Media

Printed in the United States of America

First Printing, 2016

ISBN 978-1-943952-07-6

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

Table of Contents

About the Authors

Scott D. Lowe,
Partner, ActualTech Media

Scott Lowe is a vExpert and partner and Co-Founder of ActualTech Media. Scott has been in the IT field for close to twenty years and spent ten of those years in filling the CIO role for various organizations. Scott has written thousands of articles and blog postings and regularly contributes to *www.EnterpriseStorageGuide.com* & www.*ActualTech.io*.

James Green,
Partner, ActualTech Media

James Green is an independent blogger at *www.virtadmin.com*, two-time vExpert, serial Tech Field Day delegate, and works as a virtualization consultant in the Midwest.

David M. Davis,
Global Trainer & vExpert

David Davis is a partner and co-founder of ActualTech Media. With over 20 years in enterprise technology, he has served as an IT Manager and has authored hundreds of papers, ebooks, and video training courses. He's a 6 x vExpert, VCP, VCAP, & CCIE# 9369. You'll find his vSphere video training at *www.Pluralsight.com* and he blogs at *www.VirtualizationSoftware.com* and *www.ActualTech.io*.

About ActualTech Media

ActualTech Media provides enterprise IT decision makers with the information they need to make informed, strategic decisions as they modernize and optimize their IT operations.

Leading 3rd party IT industry influencers Scott D. Lowe, David M. Davis and special technical partners cover hot topics from the software defined data center to hyperconvergence
and virtualization.

Cutting through the hype, noise and claims around new data center technologies isn't easy, but ActualTech Media helps find the signal in the noise. Analysis, authorship and events produced by ActualTech Media provide an essential piece of the technology evaluation puzzle.

More information available at **www.actualtechmedia.com**

ActualTech Media

Foreword

For too long, IT has been thought of as a cost center, where the increased demands and shrinking budgets put it at odds with being responsive to new business requirements. The role of the CIO and IT as a whole has been in a state of flux – outsourcing, cloud computing (often in the form of stealth IT), and a heavy focus on reducing costs typically meant that infrastructure is aging and internal resources spend most of the time fighting fires and optimizing environments, at the cost of any new initiatives or enhancements. Now is the time for IT to take its place as a critical driver to business success by creating a modern data center. The authors of this book have real life experience as both IT practitioners and years of educating IT communities on transformational technologies.

The technologies and strategies discussed are the culmination of a number of trends that have been combined into solutions that have IT at a tipping point. The new solutions build on technologies that have matured over many years, including virtualization and convergence. While these solutions incorporate many of the same building blocks, users will require retraining on both the consumption and operational models. In order to be successful, all stakeholders must be at the table and have a good understanding of both the business goals and challenges of IT transformation. Share this book with the members of the team that will be involved in the project to modernize the data center. Thousands of organizations have gone through this process already, while every data center has its own unique characteristics, the standardization and simplification of IT discussed in this book will allow you to avoid complexity and create the modern data center necessary for your future. This shift is not about reducing headcount; rather it is the opportunity for IT to have a seat at the table to support corporate growth and innovation.

Stuart Miniman
Principal Research Contributor, Wikibon.com
Twitter: @stu

1

Introduction

The modern data center is an exciting place, and it looks nothing like the data center of only 10 years past. The IT industry and the world in general are changing at an exponential pace. In fact, according to Moore's Law (named after the co-founder of Intel, Gordon Moore), computing power doubles every few years. Due to the limitations of the laws of physics, this pace has slowed a bit over the past decade, but processing power still doubles every 2.5 to 3 years.

This means that, in the modern data center, tasks that were accepted as unattainable only a few years ago are commonplace today. One practical example that will be discussed many times in this book is leveraging software defined storage (SDS) to serve I/O to Tier 1 applications. This model would have likely been cost-prohibitive and *really* slow just a few years back. But today, SDS offers IT organizations some of the greatest storage flexibility and ease of use they've ever known.

IT Is Changing . . . and It Must

With this exponential growth, there are more opportunities for entrepreneurs to make a killing, for organizations to multiply their revenues tenfold by leveraging emerging technology, and for top-tier graduate students to create projects that once seemed like pure science fiction. While bleeding edge technology trickles down into the mainstream IT organization, the changing nature of the IT business itself rivals the pace of technological change.

And IT *must* change.

Over the next couple of years, IT organizations and business leaders will have the opportunity to focus on dramatic transformations in the way they conduct themselves. Technological change will create completely new ways of doing business. Those who learn and embrace these new developments will thrive — those who don't, will struggle and possibly fail. The exponential growth in technology in the next decade will generate the rise of entirely new industries and cause everything in our day-to-day experience to be different, from the way we shop to the way we drive our cars.

Simplification Is the New Black

Thanks to the ever-growing number of devices interconnected on private and public networks, and thanks to the Internet, the scope of IT's responsibility continues to expand.

For example, not so many years ago, a printing press would have been operated and maintained by a number of highly specialized printing press operators; IT would have had nothing to do with the print shop, except to provide the computers on which the projects were designed. Today, however, it is not at all unusual that most of the printing press staff are gone, and the press itself is primarily controlled by a computer. The computer is responsible for operating the press at maximum efficiency, and sometimes it does so well that the print shop doubles its profits and the computer becomes critical to the business. All of a sudden, IT is in the bulk-printing business.

This printing press example is just part of the boom of devices, such as sensors, lights, actuators, and medical devices that are being networked and exchanging data. This boom was dubbed the "Internet of Things" (IoT) way back in 1999, believe it or not! Today, we're still right at the beginning of this paradigm shift. The estimations of IoT proliferation in the next few years are staggering. A 2014 Gartner report shows an estimated 25 trillion connected devices by 2020. Because this trend means that IT departments become responsible for more systems and endpoints, one of the primary goals for these departments in the near future is to simplify administration of their systems.

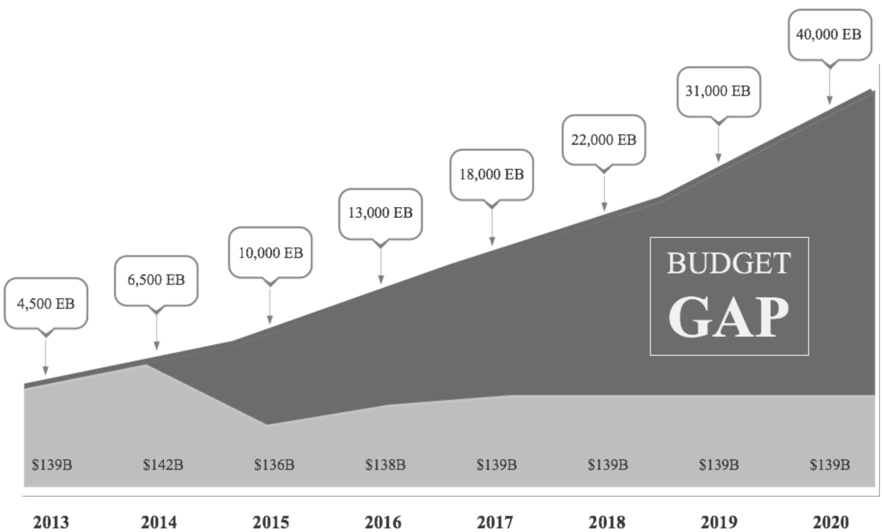


Figure 1-1: The IT Budget Gap

Despite the fact that the number of integrations and the amount of data IT has to manage because of these devices is increasing, in many industries budgets are not. (See **Figure 1-1**) This means that, in order to succeed, CIOs will need to find ways to boost efficiency in major ways. Five years from now, the same team of administrators may be managing 10 times the number of resources that they're managing today, and expectations for stability, performance and availability will continue to increase.

There are two main venues where the push for simplicity must come from: the manufacturers and the IT executives. First of all, manufacturers must begin to design and redesign their products with administrative simplicity in mind. To be clear, this doesn't mean the product must be simple. In fact, the product will almost assuredly be even more complex. However, the products must have the intelligence to self-configure, solve problems, and make decisions so that the administrator doesn't have to.

Secondly, IT Directors and CIOs must survey their entire organization and ruthlessly eliminate complexity. In order to scale to meet the needs of the future, all productivity-sapping complexity must be replaced with elegant simplicity such that the IT staff can spend time on valuable work, rather than on putting out fires and troubleshooting mysterious failures that take hours to resolve. One of the primary ways this might be done is to eliminate the organizational siloes that prevent communication and collaboration.

Focus on Cost Reduction

As shown in **Figure 1-2**, a recent Gartner report shows that IT budgets are predicted to remain nearly flat as far out as 2020. This is despite 56% of respondents reporting that overall revenues are expected to increase. The reality is that many of those IT organizations will be required to do more with less, or at the very least, do more without any increase in budget. As this trend isn't likely to change, the IT department of the future will be focused on reducing expenses where possible and increasing control over the absolutely necessary expenses.

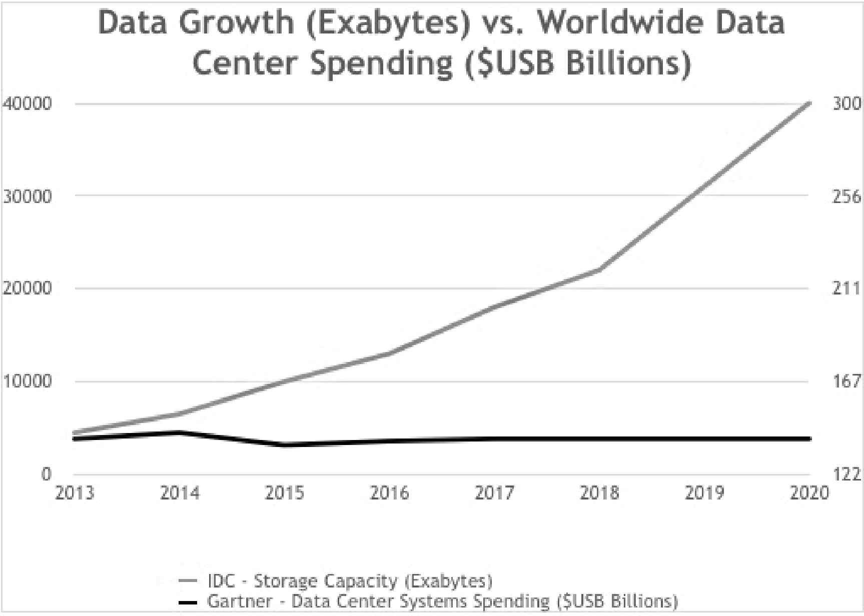


Figure 1-2: Data Growth vs. Data Center Spending

From the Field: Survey Results

Throughout this book, you will see this special callout intended to provide you with information gleaned from a survey conducted by ActualTech Media of more than 1,200 IT pros regarding their thoughts around SDS and HCI.

24.6% of survey respondents have four or less dedicated IT staff supporting their business. Without budget increases, this same small handful of people will be expected to continue supporting the organization's increasing needs.

Thanks again to Moore's Law, it will be possible to complete projects in 2016 for a fraction of the cost that the same project would have cost in 2013. One of the most common examples of this is the cost of storage for a server virtualization project. Due to the challenges of performance and workload characteristics, a properly built server virtualization storage platform has historically been expensive and quite complex. Thanks to denser processors though, cheaper and larger RAM configurations, the falling cost of flash storage (solid state drives [SSDs]), and innovation toward solving the storage problem, a server virtualization project can be completed successfully today for a much lower cost, relatively speaking, and with much more simplicity than ever before.

Organizations continue to look for breakthrough technologies which allow them to accomplish bigger and better projects with the same or less budget than they had previously.

As mentioned in the previous section, the IT department of the future will be looking to manufacturers and consultants to help them build systems that are so simple to manage that less IT staff is required. This is because hiring additional IT staff to manage complex solutions is costly. Of course, this doesn't necessarily mean that IT jobs are at risk; it means that since there's no budget to *grow* the IT staff, the current workforce must accomplish more duties as the responsibility of IT grows. This additional responsibility means that IT jobs are only safe assuming that each IT practitioner is growing and adapting with the technology. Those who do not learn to handle the additional responsibility will be of little use to the organization moving forward.

Focus on Customer Service and the Business

Because of the constant expectation to do more with less, IT departments look to their vendors to provide the highest class of support for their products. IT professionals don't have time to tinker with a solution for days on end to figure out what *might* work. White-glove service from top technical resources is becoming an expectation.

Many leading manufacturers are leveraging increased connectivity and big data analytics to preemptively solve their customers' problems before they have them. After all, the best support call an administrator could ask for is the one that never has to be made. Plus, in the coming years, CIOs and IT Directors will look to shift more of the burden of maintaining infrastructure and software resources to their partners and vendors, while they look to their own IT staff to assume a role more focused on innovation and providing value to the business.

IT departments must focus on providing value to their internal customers as well. As new technology enables business agility on the manufacturer side, IT will have to continue providing the services users need and want, or users will assuredly find them elsewhere.

This means that shadow IT poses significant security, compliance, and control risk to the entire business, and the only way to really stop it is to serve the internal customers *so well* that they don't need to look elsewhere for their technical needs. *Shadow IT* is a term used to describe business units (or individuals) other than IT who deploy technical solutions — not sanctioned or controlled by IT — to solve their business problems. A simple example of this phenomenon is a few individuals in the finance department using personal Dropbox folders to share files while the business' chosen direction is to share company files in SharePoint.

Adapt or Die

The world of IT operates in the same way as the rest of the world: things change over time, and those companies, technologies, or individuals who choose not to keep up with the current state of the industry get left behind. It's unfortunate, but it is reality. The movie rental giant Blockbuster was dominating the market until Netflix and Redbox innovated and Blockbuster failed to adapt. Blockbuster eventually went bankrupt, and is now an afterthought in the movie consumption industry while Netflix's fortunes are at an all-time high.

This lifecycle is exactly the same in IT; there are household names in the industry that are more or less irrelevant (or quickly becoming so) at this point because of their failure to adapt to the changing market. Unfortunate as it may be, this is also happening at the individual level. As IT administrators and architects adapt or do not adapt over the course of time, they either advance in their organization and career or they become irrelevant.

Flattening of the IT Organization

A decade ago, IT administrators prided themselves on their silos. The stereotypical gray-ponytailed storage administrators were the pride and joy of a thriving IT department. However, that organizational mentality has come and gone, and the IT department of the future is becoming much more tightly integrated and highly generalized.

With the ongoing need to do more with less, the most lucrative jobs will be the roles whose responsibilities span a wide variety of technical specializations. The industry will place great value on highly proficient generalists as compared to specialists.

One of the primary disruptions to the industry that's driving this dissolution of silos was the mainstream adoption of x86 virtualization. As organizations across the globe have consolidated rooms full of physical servers down into a few racks, the supporting technologies have necessarily come closer together as well.

Virtualization administrators are also network, storage, and software administrators by association. Because this has historically been quite a full plate, the focus is now on making life easier for these "Jacks of all trades." To keep costs down and agility up, the IT professional of the future will be expected to have a broad skill set and be capable of transitioning quickly between different areas of focus. Data center vendors will need to focus on making sure their incredibly complex, innovative solutions require as little babysitting as possible.

IT as an Operational Expense

Especially in the enterprise environment, getting budgetary approval for operational expenses can prove to be easier than getting approval for large capital expenditures. As such, the operating model of many IT departments is shifting away from capital expenses (CapEx) when possible and toward a primarily operational expense-funded (OpEx-funded) model for completing projects. A large component of recent success in these areas is due to a shift of on-premises, corporately managed resources to public cloud infrastructure and Software-as-a-Service (SaaS) platforms.

Since cloud resources can be billed just like a monthly phone bill, shifting IT resources to the cloud also shifts the way the budget for those resources is allocated. While buying a pile of servers and network equipment to complete projects over the next year is budgeted as a capital expenditure, the organization's "cloud bill" will be slotted as an operational expenditure. This is because, with a capital purchase, once the equipment is purchased, it is owned and depreciating in value from the moment it hits the loading dock. This removes an element of control from IT executives as compared to an operational expense.

If a SaaS application is billed per user on a monthly basis, there's no need to pay for licenses now to accommodate growth in headcount six months down the road. It can also be in use this month and cancelled next month. This is in contrast to the IT director who can't just "cancel" the stack of servers purchased six months ago because the project got cancelled. Due to these advantages from a budgeting and control standpoint, products and services offering a model that will require little or no capital expense and allow budgeting as an operational expense will be preferred.

From the Field: Survey Results

48.5% of survey respondents indicated that their organization is likely to adopt cloud-based storage services.

The survey didn't explore this, but it's likely that a good number of those looking to adopt cloud-based storage services are interested in doing so because of the operational expense associated with the model. They're tired of the storage hardware and maintenance refresh cycle.

What this means for manufacturers is that transparency, granular control, and offering a OpEx-based model like renting or leasing, billing based on monthly usage, and expanding in small, predictable chunks based on need, will position them for adoption by the IT department of the future. Also, offering insight and helping the customer to increase efficiency and accuracy over the course of the billing cycle will create lasting loyalty.

This shifting focus from CapEx to OpEx is also giving rise to new kinds of data center architectures that allow organizations to keep data centers on premises and private, but that enable some economic aspects similar to cloud. Rather than having to overbuy storage, for example, companies can begin to adopt software defined storage (SDS) or hyperconverged infrastructure (HCI) solutions that enable pay-as-you-grow adoption methodologies.

What's Next?

The future always holds a lot of promise, but it also creates trepidation as people struggle to understand emerging offerings and how they personally fit in to the new order. In this book, you will learn about the trends that will shape modern data centers over the next decade and how you can seize the opportunities that come with this transformation.

2

Data Center Evolution

Human history has a curious way of repeating itself, and veteran IT professionals know that technology in the data center tends to be cyclical as well. As technology changes, the data center sometimes reincorporates technology or methodologies that used to work but were phased out in favor of newer options. Then, when higher performing or simplified versions of the old technology are eventually developed, the cycle starts over again. A good example of this is the end user's endpoint device. There was a time where the computing power and logic for end user applications was contained in the data center. A terminal device gave users a display, controls, and a session back to the data center via the network.

Somewhere along the way, as the personal computer matured, IT organizations found that employees could be more productive and IT could be more effective by deploying PCs for each user and running client-server applications where the computing happened at the desktop and only accessed resources in the data center when necessary.

Then — miracle of miracles — 10 or 15 years later, after computing power had grown and new software had been developed, IT was able to provide simpler-to-manage and more cost-effective end user computing resources by placing "dumb" devices as the endpoint and keeping the computing power in the data center.

And so the cycle came full circle.

This happens in all kinds of areas as technology matures. The cyclical nature doesn't mean that progress isn't being made, however. Each time the cycle starts a new iteration, it's substantially improved from the last iteration. Before we get into the state of IT today, and ultimately where it is headed, let's go on a bit of a journey to the past.

A History of the Modern Data Center

The first few decades in the life of the room that eventually became known as the "data center" were characterized by electromechanical computers made from electrical switches and mechanical relays, and later by all electronic computers that used vacuum tubes as switches.

The innovation responsible for the data center as we know it today was the transistorized, integrated circuit based microprocessor. Maturity in this technology eventually led to Intel's 8086 chip, and all of its successors. The x86 instruction set lives on today, and is the foundation of many components of the modern data center. Although none of today's modern processors has an "86" in its name at all, the name "x86" comes from the 8086 and its successors, such as the 80186, 80286, and so on.

As computing technology developed, so did the ability to store the data that was being manipulated and recorded. Tape-based data storage technology began to be displaced when IBM released the first disk-based storage unit in 1956 (**Figure 2-1**). It was capable of storing a whopping 3.75 *mega*bytes — paltry by today's terabyte standards. It weighed over a ton, was moved by forklift, and was delivered by cargo plane. Magnetic, spinning disks continue to increase in capacity to this day, although the form factor and rotation speed have been fairly static in recent years. The last time a new rotational speed was introduced was in 2000 when Seagate introduced the 15,000 RPM Cheetah drive. CPU clock speed and density has increased many times over since then.

Figure 2-1: The spinning disk system for the IBM RAMAC 305

These two constantly developing technologies — the microprocessor/x86 architecture and disk-based storage medium — form the foundation for the modern data center. In the 1990s, the prevailing data center design had each application running on a server, or a set of servers, with locally attached storage media. As the quantity and criticality of line-of-business applications supported by the data center grew, this architecture began to show some dramatic inefficiency when deployed at scale. Plus, the process of addressing that inefficiency has characterized the modern data center for the past two decades.

The Rise of the Monolithic Storage Array

The inefficiency at scale actually had two components. The first is that servers very commonly only used a fraction of the computing power they had available. It would have been totally normal at this time to see a server that regularly ran at 10% CPU utilization, thus wasting massive amounts of resources. (The solution to this problem will be discussed in the next section.) The second problem was that data storage utilization had the same utilization issue. With the many, many islands of storage created by placing direct attached storage with every server, there came a great inefficiency caused by the need to allow room for growth.

As an example, imagine that an enterprise had 800 servers in their data center. If each of those servers had 60 GB of unused storage capacity to allow for growth. That would mean there was 48 TB of unused capacity across the organization. Using the lens of today's data center to look at this problem, paying for 48 TB of capacity to just sit on the shelf seems absurd, but until this problem could be solved, that was the accepted design (**Figure 2-2**).

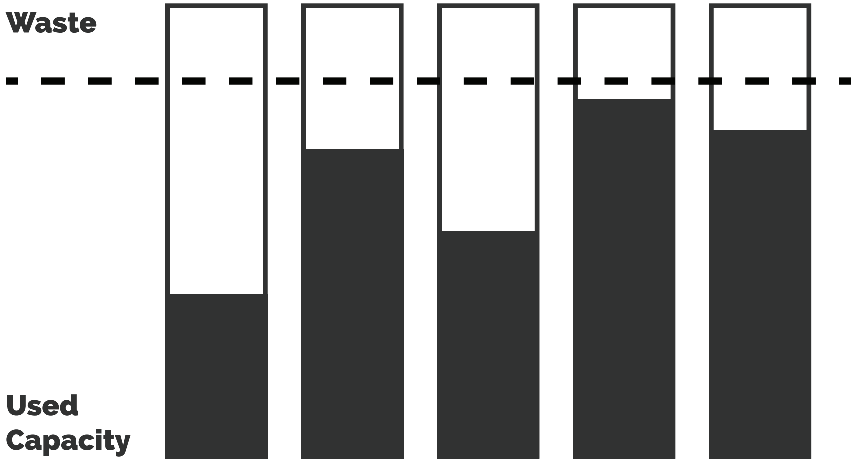


Figure 2-2: Inefficiency of the microprocessor/x86 architecture & Disk-based storage

This problem was relatively easily solved, however. Rather than provision direct-attached storage for each server, disks were pooled and made accessible via the network. This allowed many devices to draw from one capacity pool and increase utilization across the enterprise dramatically. It also decreased the management overhead of storage systems, because it meant that rather than managing 800 storage silos, perhaps there were only 5 or 10.

These arrays of disks ("storage arrays") were connected on a network segregated from the local area network. This network is referred to as a *storage area network*, or SAN, as shown in **Figure 2-3**. The network made use of a different network protocol more suited for storage networking called *Fibre Channel Protocol*. It was more suited for delivering storage because of its "lossless" and high-speed nature. The purpose of the SAN is to direct and store data, and therefore the loss of transmissions is unacceptable. This is why the use of something like TCP/IP networking was not used for the first SANs.

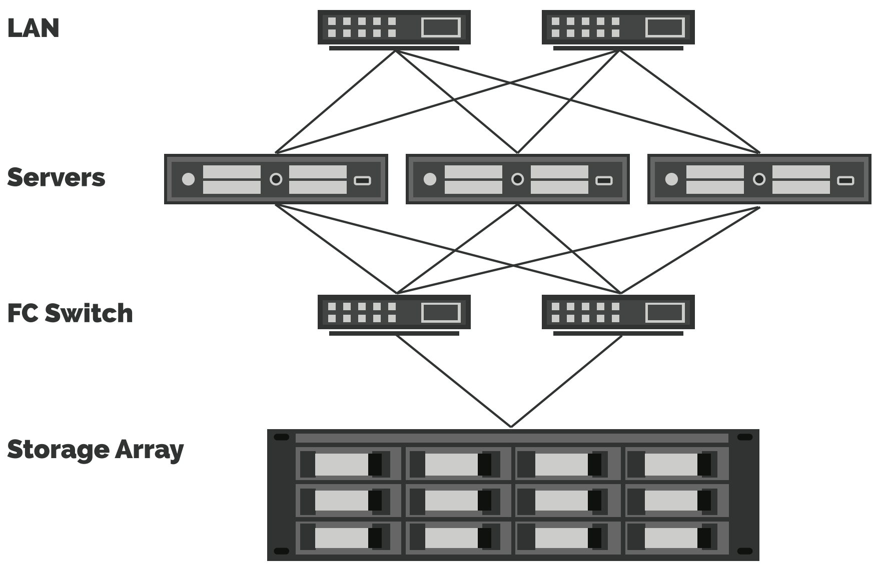


Figure 2-3: A Simple SAN

Since the SAN provided access to the storage array, and likely because of a misunderstanding of the whole architecture by some administrators, the term *SAN* came to be used colloquially to mean "a storage array providing block-level storage." So if you hear someone say, "Data is written to the SAN..." he or she would be referring to the storage array ingesting data, rather than the storage network providing transit services for the data.

Block vs. File Storage

Data stored on a shared storage device is typically accessed in one of two ways: at the block level or at the file level.

File level access means just what it sounds like, "the granularity of access is a full file."

Block level access, on the other hand, sends SCSI commands directly from the initiator (client side) to the target (storage array side).

The determining factor in which method is used is the *storage protocol*. Examples of storage protocols are: Fibre Channel (block), iSCSI (block), NFS (file), and SMB (file). Each type of storage protocol and the resulting access granularity has a use case. File-based protocols may be used where an end user or application will be accessing the files directly — a network share, for example. Block-based protocols are more likely to be used when an operating system or hypervisor is accessing the storage, as direct access to the disk is preferred.

Block-based storage can be formatted by the client with a filesystem like NTFS of VMFS, whereas a file-based volume is already formatted by the storage platform.

Data Services

Most storage platforms come with a variety of different data services that allow the administrator to manipulate and protect the stored data. These are a few of the most common.

Snapshots

A storage snapshot is a storage feature that allows an administrator to capture the state and contents of a volume or object at a certain point in time. A snapshot can be used later to revert to the previous state. Snapshots are also sometimes copied offsite to help with recovery from site-level disasters.

Replication

Replication is a storage feature that allows an administrator to copy a duplicate of a data set to another system. Replication is most commonly a data protection method; copies of data a replicated offsite and available for restore in the event of a disaster. Replication can also have other uses, however, like replicating production data to a testing environment.

Data Reduction

Especially in enterprise environments, there is generally a large amount of duplicate data. Virtualization compounds this issue by allowing administrators to very simply deploy tens to thousands of identical operating systems. Many storage platforms are capable of compression and deduplication, which both involve removing duplicate bits of data. The difference between the two is scope. Compression happens to a single file or object, whereas deduplication happens across an entire data set. By removing duplicate data, often only a fraction of the initial data must be stored.

This incorrect usage of the term *SAN* continues to this day, and is so common that it's accepted nomenclature by all but the most academic storage administrators.

As the industry matured and more organizations adopted a shared storage model, the value of the architecture continued to increase. Manufacturers added features to the management platforms of the storage arrays to allow operations like storage snapshots, replication, and data reduction. Again, rather than 800 places to manage file system snapshots, administrators could make use of volume-level snapshots from just a few (or even one) management console. This created new possibilities for backup and recovery solutions to complete backups faster and more efficiently. Storage systems also contained mechanisms for replicating data from one storage array to another. This meant that a second copy of the data could be kept up-to-date in a safe location, as opposed to backing up and restoring data all the time.

Perhaps one of the greatest efficiencies achieved by adopting the shared storage model was potential for global deduplication of data across the enterprise. Even if deduplication was available in the Direct Attached Storage (DAS) model, deduplicating 800 silos of data individually would not result in high consolidation ratios. However, deduplicating data across all 800 systems that were likely similar would result in much higher consolidation.

By the mid-2000s, average data centers had the efficiency of using shared storage across servers and applications, combined with the added efficiency of being able to globally deduplicate that data. Performance of the shared storage systems grew as manufacturers continued to improve the networking protocols, the physical disk media, and the file systems that governed the storage array. Due to its size and scope in many organizations, managing the storage network and the storage arrays became a job for entire teams of people, each with highly specialized skill sets.

Using shared storage allowed more agility and flexibility with servers than was known with direct-attached storage. During this time, many organizations chose to provision the operating system disk for a server on the storage array and use a "boot from SAN" model. The benefit of deploying operating systems this way was this: if one physical server failed, a new server could replace it, be mapped to the same boot volume, and the same operating system instance and applications could be back up and running in no time.

Blade server form factors became popular in this era as well. Blade servers have a smaller footprint because of their small number of drives (if any); this allows higher density per rack unit.

As effective as all of this consolidation was at driving down costs in the data center, there was still the problem of compute resources. CPU and memory resources were still generally configured far above the actual utilization of the application the server was built for. Eliminating this problem was the second frontier in solving inefficiency in the modern data center.

The Virtualization of Compute — Software Defined Servers

Virtualization as a concept is not a new development. Virtualization has been around since the 1960s when the technology was developed to allow multiple jobs to run simultaneously on a mainframe. This was in contrast to the prior capability of running a single batch process at a given time. Virtualization allowed for multiple workloads to run in tandem on shared hardware, yet be isolated from one another. As mainframes gave way to microcomputers and PCs, virtualization as a technology became less important, at least for a time.

In the late 1980s, as different companies struggled to control the PC market, end users found themselves in a bit of a pickle. Certain applications would be designed only for one platform. If a user owned a Unix-based computer and wanted to run a Microsoft DOS program, they were out of luck. That is, until a company released a technology that allowed the virtualization of the application developed for one operating system to run on an operating system it was not developed for. For approximately 10 years, this technology matured on desktop computers.

That was great for the desktop, but the true power of modern virtualization came in 2001, when VMware released ESX, a bare-metal hypervisor capable of virtualizing *server* workloads in the data center. The *hypervisor*, a term used to describe the software that abstracts physical resources like CPU and memory from the virtual machines, fulfilled the same purpose as the virtualization technology developed for mainframes: running multiple workloads simultaneously and effectively isolated from one another.

Many people were skeptical of this idea at first, but data centers had run into some significant issues as they grew. There's the capacity issue which has been discussed. Further, servers often used on a tiny fraction of the CPU and memory resources allocated. But there were also environmental issues; the electricity and cooling bills were growing, and floor space was becoming an increasingly scarce commodity. CIOs could see that the problem was only getting worse, and server virtualization had the potential to solve this issue.

As the market watched and waited and allowed x86 virtualization to mature, the footprint of physical servers in the data center continued to grow to outrageous numbers and consumed equally outrageous quantities of electricity. An EPA study estimated that the data center industry in 2006 consumed 61 billion kilowatt hours of electricity, which was 1.5% of the total U.S. electricity consumption that year. The total cost for this energy would have been around 4.5 billion dollars*.

Clearly, it was time to make a change.

The next stage in the lifecycle of virtualization was characterized by a massive amounts of effort put into a process known in the industry as "physical-to-virtual" (P2V). The term P2V refers to the process of capturing the identity and entire contents of a physical machine and transferring them to a virtual machine. This process allowed organizations to move their existing, deployed applications to a virtualized platform without the need to rebuild the systems. The process was generally non-disruptive. Without the ability to perform this migration, the shift to primarily virtualized data centers would have been significantly slower. A myriad of problems plaguing the data center were finally solved by virtualizing compute resources with tools like VMware ESX (which is now called ESXi or vSphere) or Microsoft Virtual Server 2005 (now succeeded by Hyper-V).

The electric bill for the data center was also reduced. If physical servers could be consolidated at the rate of 4 virtual machines to 1 physical machine (a 4:1 *consolidation ratio*), then the data center could power off 3 out of 4 physical servers — a huge reduction in the overall power consumption.

Also, if only 1 out of 4 physical servers was running, the cooling and battery backup systems didn't need to be nearly as robust. The physical footprint of the data center was reduced. That reduction was especially economically important in co-location scenarios where square footage is a premium.

* At the time of writing, this EPA report can be downloaded at: www.energystar.gov/ia/partners/prod_development/downloads/EPA_Data center_Report_Congress_Final1.pdf

In another economic boon, overall utilization of compute resources increased. Rather than having 10 physical servers running at 10% utilization, there were now two servers running at 50% or higher utilization. From a cost standpoint, this advantage alone would be enough to justify x86 virtualization.

Consolidation Ratios

The consolidation ratio is a way of referring to the effectiveness of some sort of reduction technique. One example of a consolidation ratio would be the amount physical servers that can be consolidated to virtual machines on one physical host.

Another example is the amount of copies of duplicate data that can be represented by only one copy of the data.

In both cases, the ratio will be expressed as [consolidated amount]:1. For example, 4:1 vCPUs to physical cores would indicate in a P2V project that for every 1 physical CPU core available, 4 vCPUs are allocated and performing to expectations.

But wait! There's more!

Once there were fewer physical servers to manage, the administration of said servers became easier as well. Administrators needed to apply a fraction of the firmware updates, perform a fraction of the hardware upgrades during a refresh cycle, and repair a fraction of the motherboard failures.

The impact of virtualization changed networking as well, all the way down to the physical connections. Where there may have once been two data cables for every server in the environment, in a post-virtualization data center there are perhaps two data cables per hypervisor with many virtual machines utilizing those physical links. This creates a favorable level oversubscription of the network links as compared to the waste from the legacy model. See **Figure 2-4** for an example of this consolidation.

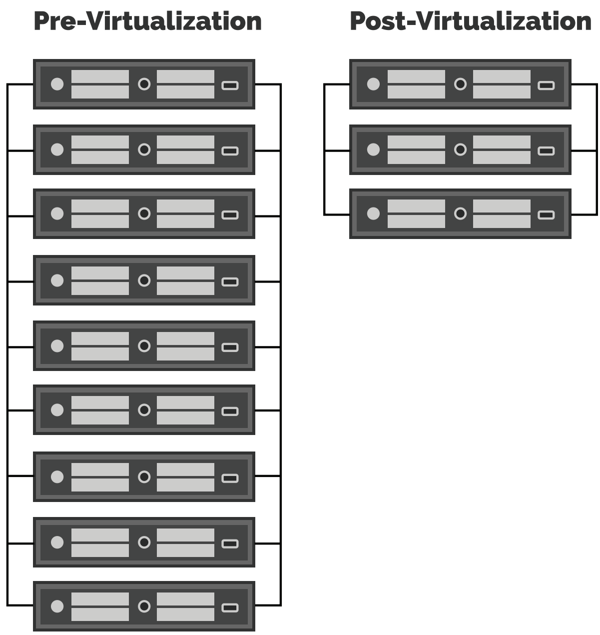


Figure 2-4: Pre- vs. Post-virtualization network consolidation

Hypervisor and virtual machine performance increased, and with it the demands on related infrastructure components. Driving virtual machine density necessitated higher bandwidth networking to allow for the high quantity of traffic sharing a single pipe. It also required higher disk performance and lower latency due to the virtual machines sharing the same storage path.

In a totally physical data center, 10 physical machines all performing I/O operations at 5 milliseconds of latency would be fine. However, when virtualized, the tenth machine in line might see a 50 millisecond I/O latency, which is likely unacceptable. To make matters worse, operating systems weren't written with virtualization or shared storage in mind. They were written to arrange data on a physical disk with the understanding that it could afford to be inefficient at times, since the only machine utilizing the disk it ran on was itself. With virtual machines and shared storage, these inefficiencies got quickly out of hand.

From the Field: Survey Results

By now, administrators have had time to get used to server virtualization. Roughly 95% of survey respondents indicated that they have either "some" or "expert-level" knowledge of server virtualization. Only 5% said they have "little or no" server virtualization knowledge.

This need for better performance led to absurd practices like "short stroking" (the practice of only utilizing the outer edge of a spinning disk and wasting the rest of the capacity) and buying disks solely for the performance impact that extra disks have on a storage array even though capacity was not needed. Clearly, the x86 virtualization movement called for serious innovation in the data storage market.

The No-Spin Zone: The Move from Disk to Flash

Magnetic storage media has been the dominant choice for data storage for the majority of data center history. Spinning disks have served as primary storage, and tape-based storage systems have served higher capacity longer term storage needs. However, the performance of spinning disk eventually leveled off due to physics-induced limitations. The speed by which data on a spinning disk can be accessed is based upon a few factors, but the one that is the biggest problem is the rotation speed of the disk platter. Eventually, the platter can't be spun any faster without damaging it.

Based on what the storage industry has produced in the past few years, it would appear that 15,000 rotations per minute (15K RPM) is the fastest speed that manufacturers have been able to maintain while keeping the disk economically viable to the customer. A 15K SAS drive is a high-performing disk, to be sure. However, the number of I/O operations that any spinning disk can perform in a second doesn't seem to be changing all that much. The fastest, most efficient spinning disks can deliver less than 200 random I/O per second (IOPS). While this is beyond adequate for a use case like a PC, it has left something to be desired when serving I/O to a dense, mixed workload virtual server or virtual desktop environment. The numbers get even trickier when RAID write penalties are factored in; depending on the RAID configuration a number of disks may be needed to achieve 200 IOPS rather than just one.

There's also the issue of latency. Due to the mechanical nature of a spinning disk drive, latency (the time it takes to retrieve or write the data in question) can't be pushed below a certain threshold. Tiny bits of latency added together across many drives becomes an issue at scale.

The solution to both the IOPS problem and the latency problem is found in flash storage. In short, flash storage media makes use of non-volatile memory to store data as opposed to magnetic platters.

Although the use of flash storage was initially troublesome due to durability issues, the performance has always been quite attractive and often worth the risk. Because flash storage is not mechanical in nature, it doesn't suffer from the same limitations as spinning disks. Flash storage is capable of latency on the order of microseconds as opposed to spinning disk's multiple milliseconds. It's also capable of far more I/O operations per second than a handful of spinning disks.

The issue of durability has been solved over time as manufacturers improve the physical memory, storage controllers use intelligence like wear leveling, and different types of flash cells are developed, like single level cell (SLC), multi-level cell and enterprise-grade multi-level cell (MLC/eMLC), and triple level cell (TLC). A typical eMLC drive on the market in 2015 is warrantied for 10 full writes per day over a period of 5 years. Alternatively, some manufacturers specify simply a total amount of data written. The same eMLC drive would probably be warrantied for something like 3.5 PB of data written.

Lastly, because of the non-mechanical (or "solid state") nature of flash storage, it requires *much* less power to operate when compared to spinning disk. As data center power bills have always run high, any way to reduce power consumption is attractive to the data center manager — and the CFO! In some countries, governments offer substantial incentives for making environmentally friendly changes like reducing power consumption. In some cases, purchasing boatloads of flash storage to reduce power consumption may be cheaper than the cost of the fine for failure to comply.

From the Field: Survey Results

According to survey results, 38.5% of respondents have 10% or more of their data center storage provided by solid state disks. Another 40% serve between 1% and 10% of their storage needs with SSD. This means that only 21.6% of respondents aren't making use of at least some flash storage.

Flash storage becoming widely available has been a huge win for the data center industry. It allows much higher performance with substantially less power, and in the future flash storage will be available at a cost per gigabyte comparable to that of spinning disk. Exactly how long this will take is anyone's guess, but industry experts predict it could take until at least 2020. This maturing of flash storage has led data center architects to reconsider the way storage is accessed in the data center yet again. Just as the utilization and management issues of direct-attached storage gave birth to the monolithic storage array, performance issues and power/environmental concerns have birthed a new storage design. The data center of the future will likely see less of the monolithic storage array in favor of a return to direct attached storage . . . but with a twist.

The Fall of the Monolithic Storage Array

Monolithic storage arrays solved many of the data center's problems and allowed IT to achieve greater efficiencies and scale. Unfortunately, the things that made this architecture so attractive also eventually became its downfall. The virtualization of compute led to densities and performance requirements that storage arrays have struggled to keep up with ever since.

One of the primary challenges that manufacturers of monolithic storage arrays have been trying to solve for a number of years is the challenge of the "mixed workload." By the nature of virtualization, many different applications and operating systems share the same physical disk infrastructure on the back end. The challenge with this architecture is that operating systems, and especially applications, have widely varying workload requirements and characteristics. For example, attempting to deploy virtual desktop infrastructure (VDI) on the same storage platform as the server virtualization has been the downfall of many VDI projects. Due to the drastically different I/O characteristics of a desktop operating system versus a server operating system and the applications running on them, they require almost completely opposite things. An average Windows server might require 80% reads and 20% writes, whereas on the exact same storage array, with the same disk layout, same cache, and so on, a virtual desktop might require 20% reads and 80% writes. Couple this problem with the fact that hundreds — perhaps thousands — of virtual machines are trying to perform these operations all at the same time and you have what the industry has dubbed "the I/O Blender." This is a comical metaphor, but quite accurate at describing the randomness of I/O operations coming into the array.

As application performance requirements go up, it has also become increasingly important to provide very low latency. So which storage model is likely to have lower latency: the one where storage is accessed across a network and shared with all other workloads, or the one where storage is actually inside the server doing the processing on the SATA/SAS or PCIe bus, or even in memory? Of course, the answer is the model where the storage is local to the workload. Bus speeds versus network speeds are on totally different orders of magnitude. With that in mind, some new ideas have started popping up in the data center storage market over the past few years. **Figure 2-5** shows the progression of storage design over time.

Storage Array: Disk	Storage Array: Hybrid disk/flash	Storage Array: All flash	Hyperconverged Disk based architecture	Hyperconverged flash based architecture
1990-2010	2007-2015	2009-2015	2012-2015	2015+
Resilient, complex to manage, expensive & really slow	Complex to manage, better performance, better $/IOPS, performance issues	Complex to manage, expensive (even with dedup) great performance	Simple, quick time to value, sexy UI, easy to use & fast, limited storage feature set	Simple, quick time to value, easy to use & ultra fast, limited storage feature set
Disk based architecture	Disk based architecture	Flash based architecture	Disk based architecture	Flash based architecture
Bottleneck is disk array, Scalability: very limited	Bottleneck is controller, Scalability: very limited	Bottleneck is controller, Scalability: very limited	Bottleneck none, Webscale / HyperScale	Bottleneck none, Webscale / HyperScale

Figure 2-5: Storage design timeline

One idea is the concept of server-side caching. This design is less radical than others, because it continues to make use of existing shared storage. One of the first really well done implementations of this technology in the enterprise was a solution that used a virtual machine to consume local DRAM and use it as a high-speed cache. Another early option was a very expensive but high-performing PCIe SSD that accelerated remote storage with local caching. These designs solved common problems like boot storms in VDI environments, because the virtual machines on each host were able to retrieve hot blocks from local DRAM before ever traversing the network. This technology was mimicked and improved on, and today a number of options exist for caching with local SSDs and DRAM in front of shared storage arrays.

A different, more radical architecture is becoming more common, however, and will allow IT organizations to replace and/or pool the monolithic storage arrays for general purpose workloads in the future. This design, which will be discussed in-depth in a later chapter, is enabled by software defined storage (SDS). The data center of the future looks (physically) a lot more like the data center of the past, in which a number of servers all contain their own direct attached storage. The difference is that all of this locally attached storage is pooled, controlled, accelerated, and protected by a storage management platform running on the hypervisor. Local storage is just a bunch of SSD disks rather than being configured in a RAID group, and fault tolerance is controlled at the node level rather than at the storage controller level.

The resilience could be thought of like a network RAID, although it's more complex than that. The performance and scale implications of this model are massive: because each node added to the cluster with local storage contributes to the pool, this means that the storage pool can grow to virtually limitless heights. Each server that is added has its own storage controller, meaning that throughput never becomes an issue. Increasing capacity of the pool is as easy as adding disks to existing servers or adding more servers overall. The control of all of this is done by either virtual machines (VSAs) or by kernel-level software, and the administrator typically manages it from the hypervisor's existing management interface (like vCenter or SCVMM).

SDS is changing the data center in tangible ways, and as more organizations begin to adopt this architecture, vendors of monolithic storage arrays will have to innovate or pivot in order to stay relevant and survive.

A deep dive on SDS is coming in a later chapter, but the stage isn't fully set yet. There's a lot of moving pieces in the data center and it can get a bit overwhelming at times. Plus, it moves so fast. Wouldn't it be nice if the design and building of it was left to someone else, and what showed up on the loading dock was ready to use rather than ready to begin a 6-month project?

The Emergence of Convergence

As the challenges for IT have grown in equal proportions with the ever-increasing scope of their responsibilities, IT decision makers have often looked to outsource parts of their operation. A notable trend for data center "outsourcing" of sorts is now referred to as *convergence*. Put simply, convergence is multiple pieces of the infrastructure assembled prior to delivery to the customer. Convergence saves time and frustration during the deployment phase and provides decreased time-to-value after procurement.

An example of a common form of convergence might look like this: a rack is delivered to the data center already containing a storage array, a blade chassis populated with blades, and a few top-of-rack switches.

Everything is cabled up, and all the configuration of the switching and storage has been done prior to delivery. At the moment the converged stack is delivered, the data center team can roll into place, deliver power and upstream network connectivity, and the pod will be up and running. This model of growing the infrastructure is substantially faster than the traditional model of having parts delivered, assembling them, hiring consultants, troubleshooting, and so on.

Speaking of troubleshooting, there's another important facet to this approach: the pod that comes pre-built is based on a tested and validated reference architecture. This means that the customer doesn't need to tinker with exactly which configuration of the parts available will work for them; that design work has already been done. Also, when the pod is built at the factory, the technician building it actually makes sure that the connections are good and the infrastructure is highly available and performing as designed.

The value in convergence comes not only from the fact that the solution comes pre-assembled, but also from the fact that it includes all the pieces necessary. Half the challenge in traditional piecemeal solution-building is getting all the right parts and ensuring interoperability. Convergence guarantees that with the purchase of a certain SKU, all the components contained within it will be compatible with one another, and all the necessary parts will be included.

Convergence has helped many organizations realize project objectives faster, and has saved a multitude of headaches over time. But if a little convergence was good, does that mean a lot of convergence is great? A design methodology that will be discussed at length in following chapters is now taking the place of convergence in the data center. The successor to convergence is known as "hyperconvergence," and it takes the idea of simplicity to the customer to new heights.

Hyperconvergence is so called because of the scope of what is being converged (**Figure 2-6**). In a converged infrastructure, many infrastructure components are brought together into one rack (or a few racks). In a hyperconverged infrastructure (HCI), those same components are brought together within a single server node. Hyperconvergence is born from cloud data centers that pioneered and leveraged this technology to operate at the massive scale they require.

Converged v Hyperconverged Infrastructure

Component	Traditional Converged Infrastructure	Hyperconverged Infrastructure
Server	Hardware managed, proprietary	Flexible
Compute	Scale-up, large memory capacity, in-memory computing, flash memory	Basic nodes, OEM provided
Network	Datacenter with high performance and bandwidth	Simple, multinode
Storage	Tiered storage area network (SAN)	Software defined storage
Management Software	Vertical stacks	Horizontal compute, storage and global file system
Scalability	Scale-up, using primarily proprietary components	Scale-out using mostly commodity components, including compute and storage
Workload Support	Core enterprise	Virtualization, AnyCloud
Integration	Hardware-defined, vendor defined	Software-defined, hypervisor-integrated
Architecture	Vertical	Horizontal, Symmetric scale-out architecture
Vendors / Solutions	Cisco-NetApp, VCE, Oracle, HP, IBM, Dell, Huawei, etc...	Atlantis, Nutanix, Simplivity, Scale Computing, Pivot3, Maxta, EVO:Rail (OEMs)

Figure 2-6: Converged vs. Hyperconverged Infrastructure

Now, what's a technology designed for cloud data centers like Facebook or Amazon doing in a small corporate data center? It turns out that the cloud isn't just for the top 1% anymore. Cloud technology is being leveraged all over the world by even small companies.

The Role of Cloud

The term *cloud* has always been a bit confusing and hard to nail down. Unfortunately, many misconceptions exist about exactly what "the cloud" is, but in the most general sense, the cloud is pretty easy to grasp.

Cloud computing is a model of delivering infrastructure or application resources in a way that is flexible, rapid, and on-demand. This is why purchasing infrastructure from Amazon Web Services (AWS), for example, would be classified as cloud. It's on-demand, takes about two minutes to provision, and has tons of options. Because cloud is a model and not a thing, there are a number of different ways cloud infrastructure can be implemented.

The Cloud

Keep in mind that the cloud is simply a method of offering and provisioning on-demand services. With this definition in mind, it's easy to see that a private cloud deployment is simply an on-premises deployment of a tool like OpenStack that allows for rapid, on-demand provisioning of resources that can easily be created and destroyed.

But why does anyone do this anyway? What is the value in cloud-based solutions as opposed to the way virtual infrastructure has been deployed for the previous decade? The Cloud Drivers section answers these questions.

Cloud Types

Different cloud deployment models fit different organizations. There are certain cases where an application has been developed from the ground up to run in a cloud. In this case, it may make sense to use a public cloud model, where all resources are provisioned in a third party data center provided by the likes of AWS, Microsoft, VMware, Google, or your friendly neighborhood cloud provider. Especially for some small businesses, being entirely public-cloud-based allows for an extremely light IT footprint in the office or storefront, resulting in less overhead.

Public cloud can be very affordable. It also offloads risk and overhead in terms of compliance, patching, equipment failure, hardware refreshes, and so on. When purchasing cloud resources, you're purchasing a service and do not care what happens on the back end. If it works for the organization, an exclusively public cloud model is great — but it doesn't work for everyone.

The next possible choice is a combination of on-premises cloud and public cloud; it's known as *hybrid cloud*. Using this model, IT resources run in the corporate data center as usual, but an extension to a public cloud data center is in place. This means that based on certain requirements, constraints, or other design decisions, a workload can be provisioned either to the private data center or to the public one.

An example of how hybrid cloud might work is that of a retailer. If Black Friday is coming up, the retailer may be able to spin up an extra 20 instances of their website and shopping cart application in the public cloud. The back end databases still exist in the on-premises data center and need not be migrated. This is commonly referred to as "bursting" to the cloud.

Another example where a hybrid cloud model could work out well is in an organization that has a heavy in-house development workload. If developers are constantly creating and destroying test environments, it can require lots of horsepower to keep things running fast enough that developers are happy, and project scopes can change with a moment's notice. A much easier way to handle this situation would be to run production workloads in the on-premises data center, but have development and testing workloads provision to the public cloud. This can also save on cost as opposed to running the resources locally.

The third option for cloud deployment models is a private cloud. This phrase can be rather confusing if one thinks of "cloud" as a third party selling services on the Internet, or worse yet, if one thinks the Internet itself is a cloud.

Cloud Drivers

Although virtualization revolutionized the data center, expectations for performance, availability, and cost never cease to change the way things must be done. Eventually, the increased speed and reduced cost of virtualization wasn't enough anymore. There are a few things driving the adoption of cloud models, currently; as with most decisions, it ultimately comes back to the business and the bottom line.

Many organizations develop software. Some develop it to sell, and software is their product; others develop software for their own internal use. Either way, developers are the resources driving change in the data center. Because the software development lifecycle has become so important to so many business, any technology or deployment model that will allow that cycle to iterate faster will be of benefit to the business.

Therefore, the first driver for cloud models (public, hybrid, or private alike) is agility. By nature, any sort of cloud model will dramatically increase the speed of the software development lifecycle.

The second driver is cost. Many IT organizations are required to accomplish more projects than last year with less budget than last year, and they have to look at all available options. In the case of public and hybrid cloud, the cost of running workloads (especially ephemeral ones) in that way can cost significantly less than purchasing and configuring the hardware to accomplish the same goal on-premises. In the case of on-premises, private cloud, cost savings can be found in the fact that fewer developers iterating faster will accomplish the same work as more developers iterating slowly. Providing the tools needed to iterate quickly could allow the paring back of developer resources.

A third driver is scale. By leveraging a public cloud provider, the scale to which an organization's systems can grow is practically limitless. Where physical space and limitations on power, cooling, and other factors make scale a challenge when hosting the entire infrastructure on-premises, the public cloud makes scaling a breeze.

From the Field: Survey Results

Roughly 10% of all survey respondents said that they're managing over a petabyte of storage at their primary site. Storage at this scale alone is enough reason for some organizations to look for cloud-based alternatives.

Finally, the cloud infrastructure model is now important because the subscription-based fee is looked on favorably in budget meetings as opposed to large capital expenditures. The shift to operational expense as opposed to capital expenditure allows for much more flexibility year to year and even month to month. Because of that flexibility and control, many organizations choose cloud-based models to help move their IT projects forward and stay within budget. It's safe to say that whether public, private, or a mix of both, the use of cloud infrastructure has changed the way the modern data center does business.

Chapter Recap

In this chapter, you took a stroll down memory lane and learned about the history of the data center. You learned how things came to be the way they are today, and a bit about where things are headed. Following are a few key concepts to keep in mind as you learn more.

- The capacity and rotational speed of spinning disk grew quickly for many years. Today, capacity is still increasing, but rotational speed hasn't increased in the mainstream market since the year 2000.

- The monolithic shared storage array was the foundation of the data center architecture of choice for the past decade.

- **Convergence** made data center resources easier to provision and consume.

- Today, thanks to virtualization and the growing performance capabilities of **flash storage**, pooled direct-attached storage is becoming a popular option for data center storage.

- Despite misconceptions remaining from an era past, flash storage medium is quite reliable. Enterprise-grade drives are commonly warrantied for 10 full writes per day over the course of five years.

- **Agility, cost,** and **scale** are common reasons businesses today are adopting cloud-focused architectures.

Now that you have a good feel for the past and the present, and you've begun to see what the future holds, Chapter 3 will take a deeper look at what's on the horizon for the modern data center. The software defined data center (SDDC) vision is all the rage today, and for good reason. You will see how the software defined model is helping enable hyperconvergence and transforming the data center as we know it.

3

Emerging Data Center Trends

The Emergence of SDDC

As the data center continues to evolve, there's an emerging need for flexibility, agility, and control. With web scale comes challenges that aren't found in legacy or smaller infrastructures, and require new ways of approaching the data center. The current approach to address these issues is the software defined approach which refers to the idea of abstracting a physical data center resource from the underlying hardware and managing it with software. An example most IT professionals would be familiar with is the virtualization of compute resources. No longer allowing physical servers to be the container for data center systems, while providing and manipulating their resources with software, is the new normal. The ability to create a new "server" with a few clicks or migrate a running workload between physical servers is the essence of the software defined approach.

The software defined approach took hold with compute, but is now starting to encompass all areas of the data center, which has led to the term *software defined data center* (SDDC). The SDDC isn't any one thing specifically, but rather a way of describing a data center where as many pieces as possible are abstracted into software. The SDDC is characterized by automation, orchestration, and abstraction of resources into software and code. By nature, code is more reliable than humans, which means that compared to a legacy data center, the SDDC is more secure, more agile, and moves more rapidly. The fallout of abstracting physical resources across the data center is that all of a sudden, the hardware is substantially less important to the big picture.

Commoditization of Hardware

Historically, computing has been enhanced by the creation of specialized hardware that is created to serve a specific purpose. Application-specific integrated circuits (ASICs) are developed, as the name suggests, to serve one specific purpose. In other words, they have one primary application. While this model of computing can lead to increased performance, lower latency, or any number of desirable metrics as compared to commodity hardware, it also comes with substantial costs that must be weighed.

Some notable costs of ASIC-based hardware are:

- Increased manufacturing cost.
- Dependence on specific manufacturers.
- Inability to recycle hardware for dissimilar projects.
- Incompatibility across systems.

Which is actually better? ASIC-based or commodity hardware?

Examining the cost is more of a business exercise than it is a mathematical one. The cost of custom hardware in terms of capital is generally more; that piece is simple. But what is the cost (or risk) to an organization of becoming tied to the one particular vendor that makes the custom silicon? What if the manufacturer goes out of business? What if there's a blizzard and the parts depot can't get a replacement delivered for six days? If it was commodity hardware, it could be supplied by a different vendor who is closer or not impacted by the severe weather. Commodity hardware is inexpensive and widely available, which are both significant advantages to an IT organization.

How does this relate to the software defined data center? Well, because the SDDC's goal is to abstract as much physical function into software as possible, the physical equipment becomes less important. This means that platforms which would previously have required special hardware can now be emulated or replaced with software and run on commodity hardware. We'll discuss some more specific examples of this later.

Commoditization allows for standardization. When many players in the market make products to server the same purpose, there often becomes a need to create standards for everyone to follow so that all the products are interoperable. This is a win-win situation because the customer experience is good and the manufacturers learn from each other and develop a better product. In the IT industry, standards are almost always a good thing. ASIC-based computing isn't devoid of standards, as electrical engineering in general has many, many standards. However, when only one vendor is creating a product, they have free reign to do as they please with the product.

All of this is not to say that there isn't a case for specialized hardware. There are times when it makes sense based on the application to use custom silicon created just for that task. Hardware being a commodity also doesn't mean certain vendors can't set themselves apart. One vendor may beat another to integrating a new technology like NVMe, and that might make them the best choice for a project. But again, something like NVMe is a standard; it is meant to replace proprietary manufacturer-specific ways of doing things.

When hardware resources are abstracted into software and allowing commodity hardware to run the workload, IT is afforded more flexibility, choice, longevity of hardware, and likely a lower cost as well.

Shift to Software Defined Compute

Compute is data center jargon meaning "CPU and memory resources." The term is necessary since the mainstream adoption of server virtualization, which abstracted the resources that formerly defined a server — those resources being CPU, memory, networking, and storage. In a post-virtualization data center, CPU and memory are grouped together as "compute," and networking and storage are handled separately.

Software defined compute, then, is the practice of controlling and automating abstracted compute resources. In most cases, that would mean manipulating virtual machine workloads. Server virtualization could be looked at as the father of the SDDC, because compute was the first to mature and garner mainstream adoption. The IT industry as a whole is already very familiar with this practice, and it is widely accepted as the standard for deploying generic, mixed-workload server infrastructures.

The advantages businesses have seen from deploying software defined compute (which is just a different way of saying "server virtualization") are broad and numerous. But some of them include massive consolidation of physical resources, increased IT agility, and increased performance and utilization. Features of modern hypervisors allow for compute workloads to be migrated between physical servers without any downtime, and software intelligence places workloads on the most appropriate physical servers based on utilization.

The challenge for other SDDC resources like storage and networking is to get adoption and trust of the technologies to become mainstream. Software defined compute has already reached that point, so the challenge for it today is to add more control and intelligence to the platform. Software defined is all about the ability to manipulate the system from outside the system (typically via Application Programming Interfaces, or APIs), so expanding on the features and robustness of the compute platform's APIs will be key moving forward.

Shift to Software Defined Storage

One the heels of server virtualization is the idea of abstracting and controlling storage resources in the same way that had already been done with compute. Software defined storage (SDS) could be thought of as storage virtualization. The purpose is to take disparate groups of physical storage medium, pool them, and abstract them so that they can be consumed programmatically via the SDS platform instead of accessing each resource independently.

Software defined is about abstraction, but it's also about control. So, SDS will also include data services that allow administrators to optimize and protect the data stored. Some possible data services to include are:

- Thin provisioning
- Deduplication
- Compression

- Cloning
- Replication
- Snapshotting

One of the primary benefits of SDS is the fact that the abstraction of underlying storage allows the use of heterogeneous storage platforms in a way that looks and feels homogenous to the administrator and to the applications. In fact, SDS commonly moves the granularity of storage management from the storage aggregate and volume level to the virtual machine or virtual disk level. This allows far greater control and flexibility in the environment.

By decoupling underlying storage hardware from the SDS platform, one of the biggest IT pains of the last decade — a storage refresh — is more or less eliminated. Where mixed-generation hardware configurations would not be allowed on a legacy platform, SDS typically makes the transition smooth by allowing various types of underlying hardware to coexist.

SDS is one of the primary enablers for the commoditization of hardware. Whereas IT organizations used to spend large sums of money on proprietary monolithic storage arrays, SDS allows them to use alternate storage architectures that aren't bound to the physical storage array. One example of this is to place a handful of disks in each server (direct-attached storage) and logically aggregate those resources via software. Not being bound to the underlying hardware affords previously unknown levels of flexibility.

Shift to Software Defined Networking

It wouldn't be a data center without networking! Networking is the backbone of everything that happens within and outside the data center. Because connectivity it so integral to the function of a data center, abstracting network functions and topologies into software will substantially change the way that data centers are designed and operated. Network abstraction also unlocks new levels of scale and flexibility with technologies like VXLAN.

Similar to how SDS decouples the physical storage from storage management, software defined networking (SDN) decouples the control plane from the data plane and allows programmatic interaction with the control plane. This change allows the network to be redesigned on the fly or scaled to meet demand. Traditional networking has relied heavily on link oversubscription, and as virtualization has changed the game, SDN allows network architects to create dynamic networks that can be adapted to meet changing requirements.

SDN, as with storage and compute, also removes the dependence on specific manufacturers and vendors as the physical hardware is abstracted. This means that IT is no longer bound by the product life cycle and development iterations of a given vendor. Rather, they can move hardware in and out at will underneath their abstraction layer. This is made possible by projects like OpenFlow, which has become almost synonymous with SDN.

On-demand cloud servers may be the biggest beneficiary of SDN. Because a tenant's network environment can be created entirely programmatically without requiring any access to or modification of the underlying hardware, a button click can fire off a series of API calls that creates whole new networks in a matter of seconds or minutes. This enables the agility that cloud consumers are becoming accustomed to, and SDN can deliver the same thing to the enterprise.

Shift to Software Defined Security

More than ever, security is a concern in the modern data center. Large-scale data loss due to malicious breaches has caused millions of trusting customers to have their personal information exposed over the last few years. Interestingly, a report published by IBM* in 2014 shows that 95% of the incidents that their security team responds to indicate that "human error" is partially or wholly to blame. Human error could mean anything from using weak passwords to misconfiguring network ACLs. Either way, it would seem that if humans could be partially or entirely removed from the situation, a substantial number of security incidents might be avoided.

The answer to this security problem in the SDDC could be called *software defined security*.

As with the other software defined systems that have been discussed, software defined security is characterized by the abstraction of security management and policy from the devices and platforms that are providing security, and being secured. Software defined security allows the automation of security policies and changes to said policies. Plus, automating changes allows for higher precision which, in turn, leads to fewer security incidents due to human error.

An example of software defined security would be to automatically deploy a software-based firewall for a new tenant and configure some default firewall rules that deny all traffic except for outbound traffic and inbound traffic on port 443. The rules are not hard-coded in the automation, but rather the result of policy applied to the tenant/application.

* www-03.ibm.com/security/services/2014-cyber-security-intelligence-index-infographic/index.html

A similar scenario could exist for east-west traffic on an internal network. Policies applied to services allow communication between different applications or different tiers of multi-tiered applications and everything else is denied. These are security configurations that are made all the time without software defined security, but they are prone to human error, dependent on underlying security platforms, and not policy-driven. Creating advanced security via an abstraction layer is the security model of the future.

The Parallel Paths of SDS and Hyperconvergence

It was mentioned that software defined storage (SDS) is one of the primary enablers of hardware commoditization in the data center. By allowing commodity storage to be pooled across commodity servers while providing enterprise-class storage services, SDS also opens the door to a new data center architecture altogether. This data center philosophy, which was mentioned in Chapter 2, is called *hyperconvergence*. It's the evolution of converged infrastructure, in which many disparate solutions are connected at the factory and sold as one package. In hyperconvergence, the services those disparate solutions provided actually become one solution. That one solution provides compute virtualization, networking, storage, data services, and so on.

It's really many different layers of the SDDC that make hyperconvergence possible. To really understand hyperconvergence, a deeper level of understanding of software defined storage is especially critical. Without SDS, the flexibility that makes hyperconvergence what it is would be impossible.

The Details of SDS

Software defined storage is a really tricky subject to nail down and understand. This is because, similar to Cloud and DevOps, *software defined* is a philosophy and not any one specific thing. Thus, the categorization of what is and is not SDS can be a bit challenging.

There are a few broad features that characterize SDS which should apply to any solution or technology purporting to be SDS:

- It should provide abstraction from the underlying physical hardware.
- It should apply services and protection to data based on policy.
- It should be accessible and programmable via standard interfaces.
- It should have the ability to scale as the business requires.

Abstraction

First and foremost, SDS is an abstraction from the physical storage that is being managed. It includes a type of storage virtualization akin to the way compute virtualization makes virtual machines independent of the underlying physical hardware. This is very important, because the strength of SDS is its flexibility. That flexibility is made possible by abstraction.

The requirement to provide abstraction does not mean that SDS can't be a way of consuming storage from a more traditional, monolithic storage array. SDS is commonly associated with hyperconvergence; however, that's only one of many ways that SDS can be leveraged. An SDS layer can provide the method for managing, automating, and scaling an already specialized storage solution.

That abstraction is typically found in one of two implementation types. The first type is a virtual appliance deployed in the infrastructure. This virtual appliance contains the software that provides and manages the SDS platform and abstracts the storage behind it from the workloads in front of it. The other method is kernel-level storage virtualization. Rather than running in a virtual machine, software runs on the hypervisor itself to provide the storage features of the SDS platform.

Policy-Based

The application of policy rather than specific settings reduces administrative burden, eliminates opportunity for administrator error, and introduces a method of ensuring consistency over time in the environment. In an SDS environment, policy may dictate any number of settings related to the storage devices themselves or the how the workloads are placed, protected, or served.

A practical example of policy-based management may be a policy that applies to a virtual machine. The policy could mandate that the virtual machine data is striped across a specific number of disks or nodes. It could also say that the virtual machine is snapshotted every 6 hours and snapshots are kept for 3 days onsite and are replicated offsite to keep for 7 days. It might say that the workload must reside on Tier 2 storage (the qualifications for Tier 2 having been previously defined by the administrator).

Imagine applying these specific settings to one virtual machine a single time. The task is not incredibly daunting, given the right software. However, imagine applying these same settings to 1,000 virtual machines in an environment where six new virtual machines are provisioned each week. It's only a matter of time before mistakes are made, and with each new virtual machine an administrator will burn time setting it up. With policy-driven SDS, simply by having applied the policy (created once), the virtual machines will be treated exactly as desired with accuracy and consistency over time.

Programmability

Management automation is the hallmark of the SDDC. For helpful and approachable automation to take place, the functions of a system must be accessible to third parties via the use of APIs. An API is a developer friendly way of exposing resources in such a way that another program can query them (get data about a resource) or manipulate them (initiate actions or change properties). Some examples of API implementations are SOAP, which is becoming less common, or REST, which is becoming more common.

APIs are necessarily present in SDS because the SDDC as a whole uses some sort of orchestration engine to make all the pieces work together. That orchestration engine needs a way to interface with each of the individual components, and APIs provide that integration point. The Programmable Data Center is a subset of the overall software defined data center vision, which aims to allow anything and everything to be accessible via API.

Scalability

Finally, SDS is highly scalable in nature. This characteristic works in conjunction with the abstraction; in part, it is the abstraction that *provides* the scalability. By seamlessly allowing different physical hardware to be added and removed underneath the abstraction layer, changes to the scale of the system can be completed without the workloads every being aware. This gives organizations leveraging SDS a distinct advantage over the prior method of scaling storage. Historically, storage was scaled by buying a bigger unit and painfully, methodically migrating data to it. SDS allows the addition of more storage, or a shift to a new platform to take place, that is totally transparent to the workload.

What Is Hyperconverged Infrastructure?

Hyperconvergence is an evolution in the data center that's only just beginning to take hold. The past couple of years have seen hyperconverged solutions developing at an incredibly rapid pace and taking hold in data centers of all sizes. Hyperconvergence is a data center architecture, not any one specific product. At its core, hyperconvergence is a quest for simplicity and efficiency. Every vendor with a hyperconverged platform approaches this slightly differently, but the end goal is always the same: combine resources and platforms that are currently disparate, wrap a management layer around the resulting system, and make it *simple*. Simplicity is, perhaps, the most sought after factor in systems going into data centers today.

A common misconception is that *hyperconvergence* means "servers and storage in the same box." Pooling locally attached storage is a good example of the power of SDS, which itself is a part of hyperconvergence, but it is not the whole picture. Hyperconverged infrastructure (HCI) aims to bring as many platforms as possible under one umbrella, and storage is just one of them. This generally includes compute, networking, storage, and management. Hyperconvergence encompasses a good portion of what makes up the SDDC.

One Platform, Many Services

Convergence, which was discussed in Chapter 2, took many platforms and made them one combined solution. Hyperconvergence is a further iteration of this mindset in which the manufacturer turns many platforms into one single platform. Owning the whole stack allows the hyperconvergence vendor to make components of the platform aware of each other and interoperable in a way that is just not possible when two different platforms are integrated. For instance, the workload optimization engine might be aware of network congestion; this allows more intelligent decision to be made on behalf of the administrator. As IT organizations seek to turn over more control to automation by way of software, the ability to make intelligent decisions is critical, and tighter integration with other parts of the infrastructure makes this possible.

What characterizes hyperconvergence is the building-block approach to scale. Each of the infrastructure components and services that the hyperconverged platform offers is broken up and distributed into nodes or blocks such that the entire infrastructure can be scaled simply by adding a node. Each node contains compute, storage, and networking; the essential physical components of the data center. From there, the hyperconvergence platform pools and abstracts all of those resources so that they can be manipulated from the management layer.

Simplicity

Makers of hyperconverged systems place extreme amounts of focus on making the platform simple to manage. If managing compute, storage, and networking was complicated when they were separate, imagine trying to manage them at the same complexity but when they're all in one system. It would be a challenge to say the least. This is why the most effective hyperconvergence platforms take great care to mask back-end complexity with a clean, intuitive user interface or management plugin for the administrator.

Chapter 8 is entirely dedicated to hyperconvergence. If you're itching to learn more about this architecture, don't worry! There's plenty more to come.

By nature, hyperconvergence is actually *more* complex than traditional architecture in many ways. The key difference between the two is the care taken to ensure that the administrator does not have to deal with that complexity.

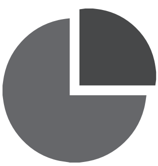
To that end, a task like adding physical resources to the infrastructure is generally as simple as sliding the node into place in the chassis and notifying the management system that it's there. Discovery will commence and intelligence built in to the system will configure the node and integrate it with the existing environment. Because the whole platform is working in tandem, other things like protecting a workload are as simple as right-clicking and telling the management interface to protect it. The platform has the intelligence to go and make the necessary changes to carry out the request.

Software or Hardware?

Because hyperconvergence involves both software and the physical resources required to power the software, it's often confusing to administrators who are learning about hyperconvergence.

Is hyperconvergence a special piece of hardware, or is it software that makes all the pieces work together?

The short answer is that it's both.



From the Field: Survey Results

57% of survey respondents say they are looking at both appliance-based and software-only options for hyperconvergence. Of those who have a preference one way or the other, 25% prefer the appliance-based approach, and 18% are more interested in software-only solutions.

Depending on the hyperconvergence vendor, the platform may exist entirely in software and run on any sort of commodity hardware. Or the platform may use specialized hardware to provide the best reliability or performance. Neither is necessarily better, it's just important to know the tradeoffs that come with each option.

If special hardware is included, it dramatically limits your choice with regards to what equipment can be used to run the platform. But it likely increases stability, performance, and capacity on a node (all else being equal).

The opposite view is that leveraging a VSA and no custom hardware opens up the solution to a wide variety of hardware possibilities. While flexible, the downside of this approach is that it consumes resources from the hypervisor which would have served virtual machine workloads in a traditional design. This can add up to a considerable amount of overhead. Which direction ends up being the best choice is dependent on myriad variables and is unique to each environment.

The Relationship Between SDS and HCI

It's important to realize how much software defined storage (SDS) technology makes the concept of hyperconvergence infrastructure (HCI) possible. If SDS didn't exist to abstract the physical storage resource from the storage consumer, the options left would be the architectures that have already been shown to be broken. Namely, those architectures are silos of direct attached storage and shared storage in a monolithic storage array. Pooled local storage has advantages over both of those designs, but would not be possible without the help of SDS which performs the abstraction and pooling.

One of the main advantages of pooled local storage is a highlight of the hyperconvergence model in general: the ability to scale the infrastructure with building blocks that each deliver predictable capacity and performance. Hyperconvergence has SDS to thank for the fact that as this infrastructure grows over time, the storage provided to workloads is a single distributed system (an aggregation of local storage) as opposed to an ever-growing stack of storage silos.

Most hyperconverged platforms offer the ability to apply data protection and performance policies at a virtual machine granularity. This capability is also a function of the SDS component of the hyperconverged system. Policy from the management engine interacts with the SDS interface to apply specific changes to only the correct data. This granularity, again, would not be possible without software defined storage.

The Role of Flash in Hyperconvergence

There are many things that go into making a hyperconverged model successful, but one component that hyperconvergence absolutely could not be successful without is flash storage.

The performance capabilities of modern flash storage are the only reason it's possible to attain acceptable performance from a hyperconverged platform.

In a legacy monolithic storage array, there was one way of achieving additional performance for quite some time: add more disks. Each disk in a storage array can serve a certain amount of data at a time. This disk performance is measured in I/O Operations per Second (IOPS). In other words, how many individual I/O requests (reads or writes) can the disk complete in one second.

As spinning disks have ceased to increase in rotational speed, the fastest spinning disks topped out somewhere between 160 and 180 IOPS. The implication, then, is that regardless of the storage capacity being used, if *performance* was depleted (meaning a workload needed more than 180 IOPS) then another disk was required to meet that need.

In a massive monolithic array, this was no problem. Add another shelf of disk, and you're on your way. In the land of hyperconvergence, this becomes a serious problem however. You can't just go on adding disks in perpetuity. A disk-focused 2U server using 2.5 inch disks can usually only fit 24 of them. So what happens if the workload requires more IOPS per node than 24 spinning disks are capable of providing?

Flash storage is orders of magnitude faster than magnetic disk due to its solid state (non-mechanical) nature. A single solid-state drive (SSD) could easily deliver the IOPS performance of all 24 spinning disks. Because of this dramatic performance benefit, flash storage is critical to hyperconvergence. Physical limitations would not allow for the creation of a high performing hyperconverged system without the performance boost that flash can provide. Raw performance aside, SSDs can also provide high performing cache which can front-end a large amount of capacity. Using SSDs as cache allows hyperconverged platforms to get high performance and great capacity numbers at the same time.

Using flash to provide caching for a group of higher capacity, slower spinning disks is commonly referred to as *hybrid*.

There are a number of different disk configurations that you might see used in hyperconvergence (**Figure 3-1**):

- DRAM for metadata, SSD for cache
- SSD for metadata and cache, disk for capacity
- SSD for all tiers ("all flash")

	Some Hybrid Configurations	Some Hybrid Configurations	All Flash Configurations
DRAM	Metadata	--	--
SSD	Cache	Metadata Cache	Metadata Cache Data
HDD	Data	Data	--

Figure 3-1: Disk Configurations of Hyperconvergence

Choosing a hyperconvergence platform that uses the right storage optimization method for a given workload has a big impact on the cost of achieving acceptable performance without overpaying.

Where Are We Now?

The data center industry is at a major transition point right now. Flash storage coupled with the hyperconvergence of systems is completely changing the face of the data center. The SDDC is already a reality in some of the more leading edge environments, and is quickly on its way to becoming a reality in the rest.

The last few years have seen flash storage and software defined infrastructure growing in maturity, and both are finally getting to the point of being ready for the mainstream IT organization. The cost of flash storage has been a challenge historically, but within the next year or two the cost of flash storage will drop to the point where it's affordable for most situations. Hyperconvergence will become more mature from a development standpoint but even more simple from an administration standpoint.

Is there anything holding the industry back? Well, hyperconvergence and the SDDC are entirely based on the prospect of a virtualized infrastructure. Therefore, customers who are only lightly virtualized (or haven't virtualized at all) have a long way to go before

Debunking Flash Storage Myths

Myth: Flash storage fails quickly and is unreliable.

Truth: Modern enterprise flash drives are commonly capable of withstanding 10 full drive writes per day over the course of five years. That durability is covered under warranty and the drive can be easily replaced if it doesn't live up to that promise.

Myth: Flash storage is too expensive to be economically viable.

Truth: Depending on workload requirements, flash storage likely has a lower cost per IOPS than spinning magnetic disks. As the cost of flash capacity comes down, SSDs will likely be cheaper in terms of both capacity and performance.

Use the following expression to determine whether flash is an economically superior choice for a given workload:

IOPS required / GB required < cost per GB (SSD) / cost per GB (HDD)

If the expression is true for the given workload, flash storage is a good choice.

If the expression evaluates to false (meaning the left side of the expression is NOT less than the right side) perhaps a hybrid or spinning disk approach would be preferred.

This consideration doesn't just apply to raw capacity, either. When applying data reduction techniques like compression and deduplication, the cost per gigabyte of flash in "effective capacity" is even lower than when evaluating the raw capacity.

hyperconvergence can be a reality in their data center. There are also decisions for them to make regarding how and where the workloads will move to as they phase out physical servers. Virtualization and hyperconvergence in an on-premises data center is one option, but they could just as easily choose to move those workloads to a public cloud provider and remove all physical equipment from the facility. In one way or another, it's safe to say that the forward progress of the SDDC ideal is totally dependent on virtualization.

As flash storage prices do fall even more in the near future and the software defined data center model becomes easier to implement, it's quite likely that there will be an exponential increase in the number of IT organizations who decide they're ready to make the leap into a new kind of data center architecture. The consideration for them will be whether they're going to build a new kind of on-premises data center with software defined compute, networking, and storage or whether they're going to shift toward a public or hybrid cloud consumption model. Rightscale's 2015 State of the Cloud report showed that 82 percent of IT organizations surveyed have a hybrid cloud strategy, so clearly this will be a major part of the coming years in the data center. The SDDC will be able to abstract the cloud provider so the SDDC

From the Field: Survey Results

Over 55% of survey respondents are more than 70% virtualization. This is great, since the realization of the SDDC vision depends on being virtualized.

Also, 15% are currently using hyperconverged infrastructure, and another 35% plan to add some in the next 12 to 36 months. Hyperconvergence will help virtualization gain traction inside the organization due to its simplicity and efficiency.

management platform can provision workloads in an on-premises data center or a public cloud data center based on policy.

Chapter Recap

In this chapter, you saw how the SDDC approach is revolutionizing the way the modern data center operates. You also briefly got a look at the characteristics of software defined storage and hyperconverged infrastructure. Recall that hyperconverged infrastructure is an architecture that is enabled by software defined storage (among other things). Following are a few key terms and concepts to take away from this chapter.

- The SDDC isn't any one thing specifically, but rather a way of describing a data center where as many pieces as possible are abstracted into software. The SDDC is characterized by *Automation*, *Orchestration*, and *Abstraction* of resources into software and code.

- Data center functions being abstracted into software allows for the commoditization of hardware. This drives cost down and decreases potential for vendor lock-in.

- **Software defined storage (SDS)** is characterized by *Abstraction, Programmability*, and *Scalability*, and is *Policy-based*.

- **Hyperconverged infrastructure (HCI)** is the practice of combining multiple data center components into a single platform to increase *Simplicity*.

- Hyperconvergence is *not* simply servers with pooled, direct-attached storage. That's one component, but hyperconvergence encompasses the platform services as well; data protection is one example.

- Adequate storage performance in hyperconvergence is made possible by the use of **flash storage**. The superior IOPS delivery compared to spinning disk allows the hyperconverged platform to meet performance needs when spinning disk alone could not perform adequately within the physical space constraints.

- The Flash Storage Myths callout explained that flash storage today is both reliable and economically viable.

In the next chapter, you will learn about defining modern business requirements so that IT can be successful. Namely, this means accurately addressing concerns like Risk, Cost, and Agility. Defining these requirements is critical to IT's success as their responsibility to the business grows. Look at this alarming statistic: according to research done by Oracle, "By 2020, IT departments will need to manage... 10-times the servers, 50-times the data, 75-times the files — All with only 1.5-times the people." Defined, accurate business requirements are the only way this will be possible.

4

Modern IT Business Requirements

We've covered the software defined data center (SDDC), how it relates to software defined storage (SDS), and the adoption growth of the SDDC model in data centers around the world. It's the flexibility and agility that the SDDC model brings which can help enterprises become vastly more efficient and agile. Only through this model can enterprises hope to meet the needs of the business, today and tomorrow.

The Business Requirement Challenge

Infrastructure is required to run applications that meet the business' needs. Businesses, though, have been notoriously bad at defining requirements for a projects and sticking to them. This has caused many headaches and wasted IT investments, in both time and dollars.

In many cases, business requirements change because the business didn't really know what they were getting into to begin with (they had never implemented such a project or they didn't know to ask the right questions in order to outline the requirements accurately). At other times, those requirements change because the business has to react quickly to new information, for example finding out that their competition is about to implement something on a larger scale, or unexpectedly getting an offer to acquire the competition in a certain area.

In the realm of IT, we don't always take the time to understand the business or connect with the people who run the business (who are the customers / consumers of the IT organization). In many cases, IT practitioners get frustrated when the requirements for a project involving IT are defined, then unexpectedly change. In other cases, there are IT pros who simply don't like change — at all. Those IT pros are frustrated with the change in requirements because they know that those changes will require that they invest unexpected and unplanned time into getting the infrastructure ready to meet the business requirements. Additional time investment, for full-time employees might require reprioritizing tasks or extra work hours during nights or weekends. With full-time employees those additional hours invested are soft-dollar costs, meaning they don't cost the company any extra real money. However, if the person working on the project is a contractor who is billing the company on an hourly basis, then those additional hours required to meet the new business requirements are hard-dollar costs, meaning they affect the company's bottom line.

Even worse, in many cases, a change in business requirements requires not only unexpected change in time investment but also unexpected changes in infrastructure requirements, and additional infrastructure (or even reconfiguring new infrastructure). These are all hard-dollar costs, and they are the costs that really get the attention of executives.

We in IT can try to educate business decision-makers on how unplanned and unexpected changes in business requirements cost the company time and money but, in many cases, the need to meet the business requirements (which affect the infrastructure) far outweigh the additional time and costs associated with changing the infrastructure.

Thus, the best option for IT infrastructure groups is to ensure that the infrastructure design and solutions are as efficient, scalable, and agile as possible. With such an infrastructure design, IT infrastructure groups can more easily adapt to eminent changes in infrastructure requirements due to changes in the business.

When analyzing IT infrastructure designs and solutions for the modern data center, there are 5 overarching challenges that such designs and solutions must be able to solve.

They are:

- Risk
- Cost
- Complexity
- Agility and Performance
- Resiliency and Availability

Let's analyze each of these modern data center challenges.

Risk

Risk is defined as "the potential of losing something of value." In the case of enterprise IT, you are entrusted with the keys to the company's data kingdom. Your company's data is by far the most valuable asset the company has, and IT is expected to do everything possible to ensure its security and integrity. But let's face it, that isn't always easy.

Just like the old business saying, "everyone is in sales," a similar saying applies to IT: "everyone is in security."

The two challenges of protecting the company's data and ensuring that it is available to everyone who needs it, anytime they need it, can easily consume every person who works in IT. After all, these two business requirements are in conflict. Acting as gatekeeper in between this conflict is the CIA triad as shown in **Figure 4-1**: confidentiality, integrity, and availability. When applied correctly (which isn't always easy to do), the CIA triad ensures that we are able to achieve both goals of protecting the data but making it available to those who need it, when they need it.

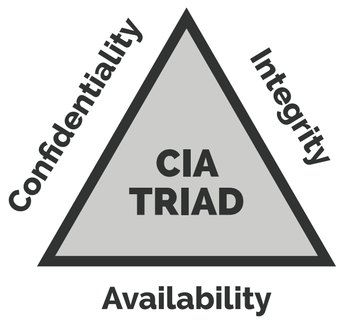


Figure 4-1: The CIA Triad

The question for you is, can you ensure that your data center is designed in such a way that you can meet the ever-changing business requirements without unnecessary risk?

For example, modern SDCC solutions utilize policy-based systems such that policies can be placed on a virtual machine or application data to ensure that it remains encrypted and highly available no matter where it moves across the enterprise data center or into the cloud.

- **Confidentiality.** Similar to privacy, confidentiality is typically ensured by implementing authentication, authorization, and accounting (AAA). Authentication ensures that everyone accessing the data is who they say they are; authorization defines what data they can access; accounting logs what data is accessed (or is unsuccessfully accessed), by whom, and when.

- **Integrity.** This ensures that the data hasn't been modified by unauthorized individuals, is consistent, and accurate. Integrity is ensured with AAA as well as data encryption. Because data today constantly moves from place to place and over different types of networks, ensuring data integrity is a tougher job than ever before.

- **Availability.** Availability ensures that the company's data is available to be used by end users and applications whenever and wherever needed. We'll talk more about the business requirements related to availability later in this chapter.

Other types of business risks that must be taken into account are as follows.

Assurance, Business Continuity, and Disaster Recovery

One of the requirements that the business needs from IT is assurance that company data is secure and that applications will be available even if the unexpected should happen. IT can only provide this type of assurance if they have first worked with the business to understand the business priorities, processes, and operations to create resiliency in the infrastructure and plan for business continuity and disaster recovery.

Resiliency and availability is typically related to ensuring that the infrastructure and associated applications continue to function even though some sort of failure occurred. We'll cover resiliency and availability later in this chapter.

When it comes to business continuity (BC) and disaster recovery (DR), IT groups all too often perform BC and DR planning in the bubble of the IT infrastructure without working with or considering the business priorities, processes, and applications.

BC planning is typically related to higher level business processes and staff, whereas DR planning and execution are typically relegated to lower level staff who take care of the more technical steps that would be taken in the event of a disaster.

To truly ensure success, companies should obtain top management commitment (both written and financial) to the development and ongoing maintenance of a DR plan before taking the next steps.

These steps include:

1. Performing risk assessments.
2. Understanding business applications.
3. Learning the priority of those applications to the business.

Next, a plan must be documented and then periodically tested to ensure that the business has the assurances needed that critical data and applications will be available in the event of unexpected events.

The question for you now is, how can your future infrastructure design and selection help you better provide the necessary business assurances and more easily support BC and DR processes if the unexpected does occur? After all, everyone in the business expected zero downtime of the infrastructure or applications until they are presented the cost required to make that possible. By leveraging more modern SDDC solutions that offer software defined disaster recovery or replication functionality, enterprises are able to dynamically adapt to infrastructure outages and recover critical applications at a much lower cost than traditional DR solutions.

The Regulatory Landscape

Increasingly, businesses are affected by the regulatory landscape. Governments and regulatory agencies create regulations to ensure that businesses act in such a way that protects the safety and interests of their customers and employees. These regulations can put tremendous pressure on IT groups to maintain data security, retention, and accounting in very specific ways to comply with regulations.

With today's constantly changing and growing data, ensuring compliance with such regulations while using traditional data center infrastructure can be next to impossible. What companies need is the ability to know what data they have, where it is, and whether it's compliant with the regulations.

Just as important, businesses also need to be able to ensure that that data maintains its own policy compliance as the data moves throughout the company's infrastructure, mobile devices, and the public cloud.

Examples of U.S. regulations that have a tremendous effect on businesses and their data centers include:

- **Sarbanes-Oxley (SOX)**. Affects all publicly-traded companies
- **Payment card industry (PCI)**. Affects all companies that accept credit cards

- **Health Insurance Portability and Accountability Act (HIPAA)**. Affects all companies that utilize medical records

Avoiding Lock-In (Hypervisor, Storage, Server)

Another risk that affects businesses of all shapes and sizes is being locked in to a particular brand of hypervisor, storage, or server in the data center.

For example, virtual machines may be stored in a format that makes them difficult to convert and move to another hypervisor, a cloud storage provider might charge you an exorbitant fee to move your data out of their cloud, or a hyperconverged server vendor might sell you a solution that requires you to continue purchasing their equipment as you need more capacity in the data center.

Certainly there are different levels of lock-in. On one hand, some lock-in may just be an inconvenience that requires you to go through additional required steps to leave that vendor's solution. On the other hand, other levels of lock-in may have massive financial impacts that require the company to continue paying a large monthly payment, whether they use the solution or not. For example, cloud or hosting providers might require you to continue paying a monthly fee or lose your data in a service that has a large barrier to exit.

That being said, most modern data center solutions are becoming so open that workloads and data can be dynamically moved from one solution to another, even without downtime. For example, if a solution uses vSphere you can usually utilize vSphere's Storage vMotion (svMotion) to move from one storage solution to another but keep in mind that deduplicated data will have to be rehydrated.

Still, you must be ever vigilant for solutions that try to lock you in, and evaluate your level of comfort with lock-in. Remember, every solution you choose has some level of lock-in, even if it is a minor one.

With on-premises solutions, the enterprise maintains ownership and, thus, ultimate flexibility and control. For this reason, on-premises software defined solutions are your best chance at avoiding lock-in.

Changing the Perception of IT

Another challenge that IT must overcome is the perception that most businesspeople carry of the value of IT to the business. Due to past inefficiencies and delays in delivering applications and services to the business (mostly because of limitations of hardware and software at that time), IT is typically seen as a slow and costly burden.

In some companies, the IT department is known as the "NO" department because they have turned down requests for applications or infrastructure so many times in the past. The result, at most companies, is that the IT group is no longer welcomed to participate when business decisions are made. In some companies, executives and power users have even found ways to bypass IT altogether (assuming they have the budget to do so), and they have either gone straight to the public cloud or they have acquired their own IT services (called *Shadow IT*).

The risk here is that if IT is not an active participant in the business processes, it's unlikely that they will be seen as relevant to the business. It's also unlikely that they will be able to help the company leverage technology to their advantage, and it's unlikely that they will receive accurate business requirements and project scopes for new initiatives.

What IT must do to be seen as a business-enabler and become a valuable part of the business process again is to show that they:

- Want to be part of the process.
- Are willing to take the time to understand the business and its technology needs.
- Have technology that can help the business to be agile, efficient, and scalable.

In other words, IT must not only "talk the talk" (of being a business-enabler) but also "walk the walk," or have the technology to back it up.

At many companies, IT organizations are starting to view themselves as an internal services provider (some call it IT as a Service, or ITaaS). The idea here is to treat IT as a separate company that must compete with external service providers in terms of cost and agility. After all, with the advent of cloud computing and software as a service (SaaS) applications, IT groups have real competition and it can all be bought using a company credit card and accessed in minutes. For companies that have pushed themselves into this competitive paradigm, they are being driven to become more competitive, more agile, and more cost-conscious than ever before.

This concept of IT becoming a "trusted partner" in the business may be one of the most important challenges that IT faces. After all, if IT is not perceived with respect and value in the organization, it's unlikely that they will even be able to gain the support of the business in modernizing their data center infrastructure in order to be prepare the business for the technology needs of the future.

Cost

With the exception of non-profit organizations, businesses are in the "business" of making money. Anything that IT can do to reduce their costs, effectively adds greater profit to the bottom line of the company's financial reports and provides greater shareholder or owner value to the company.

When put that way, IT is one of those ugly overhead expenses that it doesn't want to be. So, what can we in IT do to restructure our budgets, our technology, and our staff to achieve the position of "business-enabler" that we so desperately desire?

Here are some solutions:

- **Innovate with a new technology solution and allows the company to introduce a new product or service.** You would become a revenue generator and business enabler.

- **Help to drive down the cost of manufacturing below the cost of its competitors.** You then create a competitive advantage and are a business enabler.

- **Promote the business objective and benefits of its technology projects.** This would show IT's business value.

Changing Budgetary Landscape

When company executives read about public cloud services like Amazon Web Services (AWS) or Microsoft Azure in places like the *Wall Street Journal*, what gets them excited is the operational expense-based (OpEx) cost model.

What the OpEx model brings is the ability to simply pay for the infrastructure that they need to run the applications they require, solely based on resource consumption — even down to a per-minute consumption basis.

Compared to the traditional capital expense (CapEx) purchasing model where the company buys enough infrastructure to last for the next five years the OpEx purchasing model seems like a blessing to the business and to those in finance, keeping track of the bottom line. Not only does the traditional CapEx model require companies to purchase infrastructure based on future expectations, but most companies purchase enough infrastructure to last them three to five years which means they are over-buying infrastructure capacity that's going to sit idle for the next two or more years. Additionally, as you might know, technology infrastructure immediately begins a rapid depreciation process that typically happens faster than companies are able to pay it off with a traditional business loan. From a financial perspective, the traditional CapEx purchasing model is the worst possible way to provide the resources the company needs to run applications and store data.

Executives and financial people in enterprises around the world are asking IT to adopt an OpEx, pay-as-you-go pricing model to provide the infrastructure needed to run the company's applications and store their data. These pricing models certainly apply to public cloud infrastructure but they can also apply to on-premises infrastructure, either through the vendor who supplies the technology or through creative financing sources.

When it comes to selecting modern infrastructure for the data center, selecting infrastructure that is designed in such a way that it can easily scale up or down with the greatest efficiency and lowest cost possible may make the difference between being able to provide a reasonable

OpEx pricing model or not. Modern software defined infrastructure and virtualization allow companies to maximize their infrastructure purchases and more easily scale their infrastructure when needed.

The Changing Career Landscape

As the infrastructure in the data center changes to become more intelligent and efficient, and as the business expects more out of IT practitioners, the job duties and roles in IT organizations are changing as well.

No longer will servers, storage, and virtual infrastructures be as complex to configure as they have been in the past. Not only will the software be smarter and more efficient, but the physical hardware will be able to do much more than ever before. For example, servers will be able to run many more virtual machines, with some of those virtual machines providing distributed storage and software defined networking (SDN) services for the data center. The end result is that there will be far less hardware in the data center to monitor, configure, and troubleshoot. What this means is that IT practitioners will spend less time performing complex deployments, less time performing complex upgrades, and less time troubleshooting complex infrastructure issues.

To some in IT, their immediate concern is that less complexity in the data center will lead to less demand for their skills. All IT professionals should be assured that technology skills will always be in demand. However, as you know, the challenge with technology skills is to always keep them up to date.

But, when have you ever met an IT professional who felt absolutely up to date on all the latest technology?

From the Field: Survey Results

In our recent survey, 70% of respondents said that they had only “some knowledge” of SDS and 64% had “some knowledge” of hyperconvergence. Thus, the vast majority of IT Pros need to improve their knowledge of these technologies to be prepared to manage the datacenter of the future.

The reason that there are no IT professionals who absolutely, always feel current on all the latest technology is because there’s always something new in technology and the landscape is so wide. In other words, think of the mantra “evolve or die.” If you don’t adapt your skill set to the new and more efficient technology in the data center, then you will undoubtedly become obsolete. For example, there aren’t many fax server administrators or mainframe architects around these days because those technologies are becoming more and more obsolete.

With that being said, if data centers adopt SDS, many administrators question whether the role of the storage administrator will still be necessary. If someone is thinking about the SAN administrator who sits around provisioning LUNs all day or monitoring Fibre channel switch ports, then yes, that type of storage administrator may become obsolete. However, if the storage administrator was to elevate him- or herself to a higher level and think of themselves as being responsible for all data across the company, then that type of storage administrator (who administers the SDS infrastructure, cloud storage, and data security/protection infrastructure) would be very valuable to any company — today and in the future.

Taking that to a higher level, the architect who leads the company from traditional storage to SDS could help the company to achieve dramatic cost savings and gain business efficiencies (and earn a large bonus in the process).

Another way to think of future data center roles is that all roles need to be elevated to a higher level. For example, no longer do you have server administrators, storage administrators, network administrators, and virtualization administrators. Instead, with smarter infrastructure requiring less deep knowledge, the traditional data center silos can be broken down and the traditional data center positions/roles go along with them. Instead of those specialized administrative roles who had deep product knowledge, all of those roles in the future (as long as newer, more efficient infrastructure is used) could be consolidated into a role simply called "infrastructure administrator" or "infrastructure architect." People in this new role would have a greater breadth of infrastructure knowledge but less depth because it's no longer needed. With their time freed from constantly having to maintain deep product knowledge, they could be a much more valuable asset to the company by spending their time finding ways to help the company use technology to its advantage to become more competitive or increase bottom-line profitability.

By combining greater infrastructure efficiency and intelligence, IT infrastructure groups will be able to become more agile and turn into the business-enablers that they must become (as discussed earlier in this chapter).

Complexity

If you think about it, there is no business requirement or desire for complexity in the data center. In fact, it's quite the opposite. The business is in the business of earning profit and any complexity in the data center simply creates hurdles for technologists to overcome (and those technologists come with a high hourly price tag).

Most would agree, that with the state of today's technology we have finally gotten to a point where, when given enough hardware, software, and IT experts, we can meet the needs of the business in terms of agility, performance, availability, and scalability. However, what's lacking for most companies is that they simply can't afford or can't justify the costs required for the infrastructure and personnel in order to achieve their utopian vision of IT services. It's complexity, in most cases, that is the barrier we must overcome in order to meet or exceed the business requirements.

More modern SDDC solutions are able to eliminate the complexities that we've become accustomed to from more proprietary hardware-defined systems. These hardware-defined systems were only created originally, because commodity servers with a software layer simply couldn't provide the resources or performance required to run the services needed by the business. For example, storage area network solutions were only created when it became obvious that servers with local storage simply couldn't provide the performance or capacity necessary to store all of the company's files.

Today, commodity hardware and software defined solutions are able to provide all of the performance and capacity necessary for just about any enterprise. Plus, by running all infrastructure services in software so that many of the complex server, storage, and network complexities can be eliminated, the daily life of the infrastructure administrator can be made far easier (and make the role of the infrastructure administrator even possible, as we discussed above).

However, software defined solutions don't stop with simply making the life of the administrator easier. Modern software defined systems with less complexity and greater efficiency go a long way to making that utopian vision of IT services possible by making it easier (and less costly) for IT to provide the agility, performance, availability, and scalability that the business truly requires.

Agility and Performance

To truly compete in today's business environment, companies must be agile. What that means is they must be able to very quickly adapt to the ever-changing business and competitive environment. In order to do that, companies must be able to perform, or execute, what needs to be done exactly when it needs to be done, and exactly in the way that it needs to be done. In most cases, the company's agility and performance are the determining factor between success or being put out of business by the competition.

In terms of the modern infrastructure, agility and performance can be tied (at a high level) to being able to deploy new applications and application functionality very quickly, or being able to scale-up existing applications very quickly.

At a lower level, you could tie the term *performance* to the actual performance of an application. This, in turn, ties back to the actual performance of the servers, storage, and network that make that application possible. In order to achieve high performance at the application level, there is an abundance of hardware resources available, at lower costs than ever before (faster CPUs, faster data paths, flash storage, and greater densities).

No longer do large companies always have the edge over smaller companies. Today, it's the agile companies, performing what's needed and when needed, which are the most successful in the business world.

From the Field: Survey Results

In our recent survey, the single most important decision criteria (at 72%) when considering software defined storage was the performance of the that could be gained in their infrastructure, once implemented. For those who had implemented SDS, half of all respondents said that the performance was higher than they anticipated.

Automation and Orchestration

One way that companies are becoming more agile is by automating and orchestrating current business processes and the deployment of new IT services. There's always great confusion on the difference between *automation* and *orchestration* (see **Figure 4-2**).

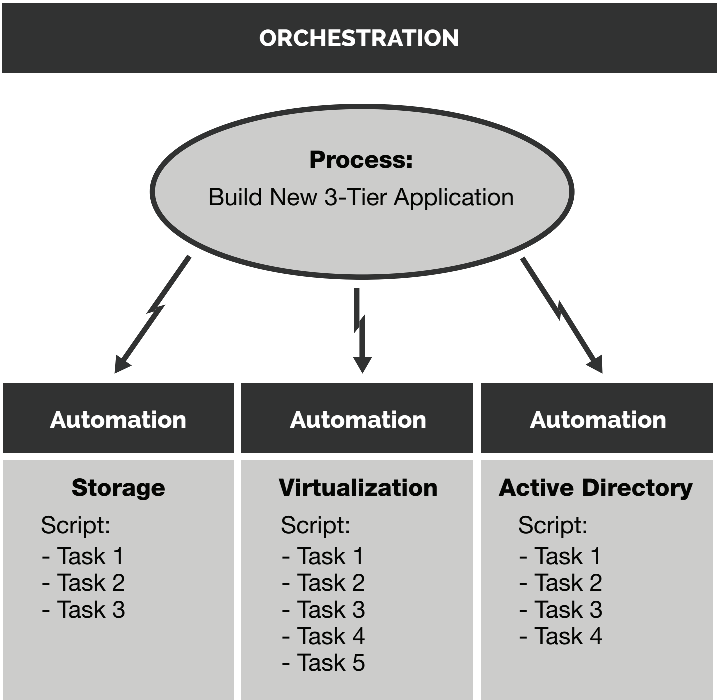


Figure 4-2: Orchestration vs. Automation

Automation

Automation is used to take common tasks and script them so that they can be done, for example, in a single command. One example is deploying a new Windows server virtual machine. Instead of manually having to take a template from a library and deploy it with a handful of customizations (for example a new server name and IP address) over a 10-minute period (with many clicks of the mouse), automation would take this entire process and allow you to perform it using just a single command. This single command might have positional parameters for the server name and IP address. If you think of a large symphony orchestra, automation is represented by the sheet music that the different musicians use to tell them exactly what to play, for every given song, such that the songs can be performed accurately, at any time, whenever they are directed to play them by the orchestra's conductor.

Orchestration

Orchestration, on the other hand, works at a higher level and leverages automated tasks. One example of orchestration is deploying a new 3-tiered application environment with a Web server, middleware server, and database server. To accomplish this level of orchestration, the orchestration engine would leverage prebuilt automated tasks (which you can think of as building blocks). One of those tasks would be the same example of automation previously discussed where we have scripted the deployment of a new Windows server. The orchestration engine would use that automated task to deploy three new Window servers, all from the same automation scripts, and then use other automation scripts that perform the configuration of these three different types of servers. The entire process that is pulled together by multiple automated task, is orchestration.

Orchestration is about automating *processes*, whereas automation is about automating *tasks*. If you think back to that large symphony orchestra, the conductor represents the orchestration engine. It's the job of the orchestra to perform their pre-written sheet music, on command of the conductor, who tells them when to stop, whether to play faster, slower, louder, or softer.

Software defined systems make automation and orchestration far easier to accomplish. Software defined systems utilize application programming interfaces (APIs) which make automation and orchestration possible by allowing administrators to script common tasks and to utilize orchestration engines to execute those automated tasks in order to accomplish the necessary business processes. It is by being able to efficiently and quickly perform business processes that a company can achieve the greater agility and performance that they so greatly desire, and thus become, and stay, competitive in today's business environment.

Self-Service

With business processes orchestrated and common tasks automated, companies can then start to implement self-service (**Figure 4-3**). You can think of self-service just like you think of a soda or candy vending machine, where you select what you need from the menu of choices and your selection is dispensed to immediately fulfill your desire.

When it comes to IT infrastructure, application self-service works similarly to the vending machine concept where when a company employee needs an IT application/service or a developer needs infrastructure and those applications or infrastructure are almost immediately dispensed to fulfill the business requirements.

Many in traditional IT organizations are opposed to self-service primarily because they don't want to lose control of their company's IT resources and data. This is a very valid concern. IT self-service

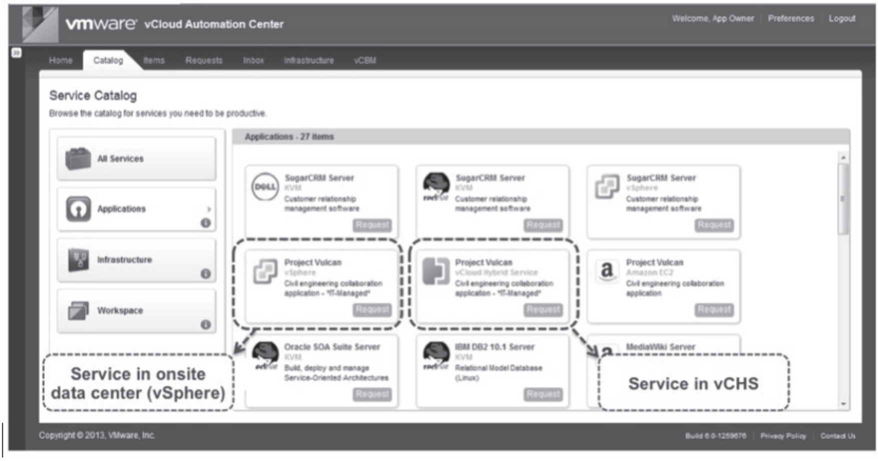


Figure 4-3: Self-Service Catalog in vRealize Automation (Credit: VMware.com)

must be done in a way such that IT doesn't lose control. Properly designed self-service solutions allow IT to define the IT services that are available, who would have the ability to utilize those resources, how long they can use those resources, if approval is required, and how chargeback or showback will be done to ensure that the department or business unit from which the employee or end user belongs is held accountable for the resources that they consume.

Keep in mind that self-service doesn't necessarily mean that every employee in the company has access to an application catalog and can deploy whatever applications they fancy or whatever virtual machines their heart desires. For example, self-service can be performed in such a way that only trained developers or trusted power users in the company have access to perform the deployment of fully orchestrated environments.

Self-service combined with automation and orchestration also brings together IT services from multiple internal and external sources into a single interface, making deployment and reconfiguration easier for everyone.

Even if some in IT are apprehensive about the concept of self-service, once automation and orchestration are implemented, self-service is what modern IT organizations should push for in order to become a business enabler by giving the business the agility that they need to be competitive.

The Data Growth Challenge

Have you ever heard of a company who doesn't have a problem with ever-growing storage capacity demands? The answer is "no," because honestly there is no company whose data is shrinking. Every company in the world is facing a challenge simply to store the amount of data that their company creates on a daily, weekly, or monthly basis.

Here are two statistics that anyone planning for the future of enterprise data center storage should be alarmed about:

- According to the United Nations Economic Council, global data will more than double in the next 5 years to over 40 Zettabytes (equivalent to about 250 Billion DVDs) by the year 2020.[*]

- According to research done by Oracle, "By 2020, IT departments will need to manage… 10X the servers, 50X the data, 75X the files; All with only 1.5X the people."[#]

When it comes to finding a solution for the data growth challenge utilizing SDS is the best place to start. Most SDS solutions include data reduction services in the form of deduplication and compression. You'll want to fully understand the details of how the SDS solution performs these data reduction services because not all deduplication and compression are created equal.

* UNECE Stat: www1.unece.org/stat/platform/display/msis/Big+Data

Oracle Stat: www.slideshare.net/aitoribanez/bilbao-oracle12c-keynote

To further exasperate the data growth challenge, IT organizations are required to create multiple copies of the data in order to achieve application resiliency and availability (which we'll talk more about later).

Increasingly, IT organizations are able to gain greater storage efficiencies than ever before through software defined data reduction functionality.

Resiliency and Availability

It is not an exaggeration to say that in the world of on-demand everything, application end-users and company executives (both of whom are the customers for most IT organizations) expect and assume that their enterprise applications will never go down and data will never be unavailable. After all this is their expectation of the Internet-based software as a service (SaaS) applications (that utilize SDS), such as Gmail, Dropbox, and Facebook.

So, what can enterprise IT do to ensure that applications and data are both resilient to failure and completely available in order to meet business requirements and customer expectations?

Traditional storage systems had zero application awareness and local high-availability for such storage systems required fully redundant, costly, and complex designs. Additionally, high-availability between data centers for the storage infrastructure not only required a duplication in cost and complexity of the storage but also expensive licenses and plentiful bandwidth between the sites.

By utilizing SDS, IT organizations can provide vast improvements in resiliency and availability. Today's SDS systems are able to, for example, fully distribute application data across multiple hosts and multiple locations and do so more intelligently and efficiently than ever before.

SDS systems also work in a policy-based manner such that they can increase the number of replicas (on the fly, without downtime) for the most critical applications, and use the minimum number of replicas for less critical applications. They can also replicate only the changed blocks of the most critical applications to remote sites to ensure the data is protected and that applications can remain highly available in the event of the unexpected (while maintaining data efficiency). By utilizing application oriented policies, SDS systems are able to provide application resiliency and availability where it's most needed and when it's most needed.

After all, if the Internet-based SaaS applications have set the bar for your end-users in terms of high availability, shouldn't enterprise IT organizations also use the same type of advanced software defined intelligence used in that "HyperScale" world?

Chapter Recap

In this chapter, you learned that modern IT business requirements are more demanding than ever. IT organizations must "adapt or die" in order to remain relevant to the business. You also learned about the numerous facets of business that IT is a part of — risk, cost, complexity, agility, performance, resiliency, and availability — and how modern software defined solutions can help IT to rise to the challenge of modern business requirements.

Only by modernizing infrastructure and elevating yourself to a higher level in the IT organization can IT technologists become business-enablers and help the organization achieve success. The following are a few key terms and concepts to take away from this chapter.

- *Risk* is defined as “the potential of losing something of value.” In IT, this is specifically the company’s data.

- The challenge of avoiding risk while maintaining usability is known as the CIA triad: confidentiality, integrity, and availability.

- Anything that IT can do to reduce their costs effectively adds greater profit to the bottom line of the company’s financial reports and provides greater shareholder or owner value to the company.

- To truly compete in today’s business environment companies must be agile. What that means is they must be able to very quickly adapt to the ever-changing business and competitive environment.

Up next, you’ll learn some principles that can help you make your own data center more agile. Get ready to learn how to make a big impact on the business.

5

Making Your Data Center Agile: Principles & Strategies

The modern enterprise is a very different beast than the enterprise of days past. In the modern enterprise, IT and the data center play pivotal roles in the operation and success of the business. In fact, the role that IT and the data center plays in helping the business achieve its goals is becoming increasingly critical. IT is not just an expense that impacts the bottom line. For many businesses, IT is a key top-line revenue driver.

However, the old ways of managing these critical resources are no longer sufficient. Rather than build complex conglomerations of components that are carefully crafted into crazy combinations, the emerging enterprise must create environments that are agile and that imbue the business with ferocious flexibility.

There are three key principles around which agile IT revolves:

- Thinking big
- Starting small
- Moving fast

Each of these three principles is discussed in this chapter, along with strategies that can help you forge your own path through the modern data center's forest of options.

Think Big

Small thinkers need not apply. Today's technology environments demand big thinking from people who have deep understanding of both the business and technology.

Be bold.

In practical terms, here's what this means: it's time to look at the whole technology environment and re-evaluate everything.

Does the data center lend itself to emerging constructs, such as pay-as-you-go adoption? Data centers of days past required massive initial investments, which were neither economical, nor particularly practical. Organizations then spent years attempting to recoup these large investments, only to find that they may have never really realized a maximum return.

New thinking may require that you jettison your old ideas around how IT systems are procured, deployed and supported. Your efforts, though, will result in a streamlined IT function that is laser-focused on the needs of the business.

Are you still providing break/fix services for devices like desktop computers and printers? *Anyone* can do that. Outsource it. You will likely save money and end up getting better service.

It's easy to say that you should outsource something, but it's important to understand the reason why you should outsource these kinds of commodity services. That reason is summed up by two words: *opportunity cost.*

The future of IT revolves around "value added" services, not commodity services that are easily outsourced. For every commodity service that you keep in-house, and to which you dedicate internal staff, there is less time that can be devoted to those value-add projects that can propel the business forward. This time you take away from potential value-add projects is the *opportunity cost* of the commodity services.

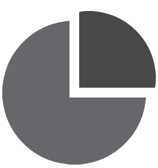
When you're making the decision to keep a particular service in-house or outsource it, the actual cost of the service is only a part of the decision process. Even in the unlikely scenario that you can perform certain services for less money than an outside party that specializes in those services, you are probably paying a steep opportunity cost to do so.

Although agile IT principles demand that you consider all aspects of the IT organization, since the focus of software defined storage (SDS) and hyperconverged infrastructure (HCI) is on the data center, let's start there.

Do you have automation and orchestration tools in place that enable a streamlined and more efficient data center? If not, rethink how everything works in your organization. If you have people performing manual, repetitive tasks, look for ways to automate those tasks.

If you have users who would be better served through the implementation of self-service tools, implement them. Just remember, doing so requires that you have the aforementioned automation and orchestration tools in place.

Agile IT also demands that the underlying infrastructure be bereft of complexity in all ways. Traditional infrastructure solutions introduced a great deal of complexity. SDS and HCI solutions bring simplicity to the data center. By bringing ease-of-scale and ease-of-management to what have been very difficult to manage, size, and scale resources, these approaches are perfect infrastructure fits for environments that embrace agile IT principles.



From the Field: Survey Results

36% of survey respondents indicated that their organization will add more software defined storage systems in the next 12 to 36 months. 35% of the same group indicated that they plan to deploy more hyperconverged infrastructure in the next 12 to 36 months.

Start Small

With an understanding for what agile IT demands, the next question you may have is, "How do I do it?"

Even the creators of the most ambitious plans need to start somewhere. It's not generally feasible to simply throw away everything that already exists and replace it with something new. At the very least, new initiatives need to be staged for implementation in such a way that they minimize impact to production.

So, start small. Find something that needs to be fixed and, well, fix it. Perhaps the physical server environment is not well-managed and you have a desire to increase your use of virtualization. At the same time, part of your "big thinking" plan is to move the entire data center to an HCI solution or one that leverages SDS.

Unless you're a very small company, you can't do that all at once. So, use the server virtualization initiative as your stepping stone. Implement your brand new virtual desktop environment on an HCI or SDS solution. Then, during the replacement cycle for your server environment, transition that environment over to the new one. In this way, you'll maximize your investment in your legacy infrastructure while eventually gaining the benefits of the new one, such as easy scale, fast deployment, and simple administration.

Your projects should be use-case driven and tied to clear business outcomes. Even achieving a simpler, more cost effective data center is a business outcome if it means that the IT department as a whole can now be more responsive to business needs.

Beyond the infrastructure though, starting small may mean finding a place to begin automation. Many organizations, for example, continue to manually provision user accounts. You don't necessarily have to adopt an expensive and complex identity management solution, but with just a few scripts, you can probably automate a good chunk of the process. The role of the person who used to provision accounts then moves to one of oversight rather than action, freeing that person up to service other higher priority needs.

To be sure, it can be difficult to get started on such initiatives, particularly in IT organizations that are strapped for people resources. However, every automation that is accomplished clears a little of that challenge. The more you can automate, the faster you can automate even more.

Now, with that extra time that you gain in IT, what do you do with it? You improve the business. Meet with the business unit leaders in your company and help them discover and implement new services that increase customer satisfaction or revenues Help them find technology-driven ways to reduce costs. Figuring out how to provide analytics that helps a company market better to their customers. For example, if you're a bank, helping to implement the ability to scan checks to deposit them via smartphones is a great example of how technology can drive customer satisfaction. This service changed the retail banking industry and it's primarily a technology solution.

Move Fast

If you're asking yourself when you should get started, that's an easy answer: *Now!* Don't wait.

Here are six steps to accomplish your goals quickly:

1. **Talk to the Business.**
   Interview business unit leaders and senior management and identify business objectives and where technology support gaps may exist or where there may be opportunity to use technology to improve a process or introduce a new service.

2. **Assess the Technology Environment.**
   At this point, you know what the business needs. The next question is this: can the technology environment support it?

Tip from the Field by Scott D. Lowe

As a consultant, I am frequently brought in to organizations to determine the "state of IT." Rarely do I begin with discussions with the IT department, however. More often than not, my initial discussions are with executive leadership, the heads of business units, and end users. These initial discussions consist of conversations around current issues as well as desired direction and strategic priorities.

Only after I gain a broad understanding for current priorities, expectations, and perceptions for how well — or how poorly — IT is meeting those expectations do I engage with IT to continue investigatory efforts. This approach also served me well as a CIO.

— Scott D. Lowe

The technology environment (the hardware and software) itself is absolutely critical to the business. However, over time, as is the case with many things, such environments begin to take on lives of their own. What once was state-of-the-art is now a complex morass that has been extended to meet unanticipated needs. Of course, this is not always the case, but it's certainly not unusual, either.

The more that the environment has been customized and extended with a lot of different point solutions, the more complex it is to manage and the slower that IT can react in modifying it to meet new business demands.

Your job is to figure out which technology solutions need to be maintained in order to meet business needs and then find ways to simplify what remains so that it is manageable and cost effective. This need absolutely includes the data center, too.

Look at the data center environment both holistically and granularly. With all of the information you gather, create a list of every product and service and identify, where appropriate, the business need served by that product or service. For example, your marketing department might have its own application environment that supports their customer relationship management (CRM) system.

Your job at this point is not to eliminate services and products, but simply to identify.

3. **Create a Support Inventory.**
   Analyze every aspect of the IT organization and list every process that is currently supported and determine where there are support gaps based on what you learn in the previous step.

   For example, is that CRM environment being maintained in a way that truly meets the needs of the business? If not, why not? Is it because the staff has skill deficiencies? Is it because there are too few staff to support the existing operational environment? Is it because the existing staff do not have a focus on the business?

   This is your opportunity to determine where there may be deficiencies that would prevent IT from fully executing on the strategic priorities that were identified during your discussions outlined in the previous steps. In this step, focus on people and process.

4. **Identify Core Services and Gaps.**
   At this point, you should have a good understanding for how the various products and services present in the organization actually meet business needs. You should also have a good understanding for where gaps exist in the environment.

   You should be able to answer the following questions:

   - Which products and services are core to the mission and vision of the business?

   - What gaps exist in the current product portfolio that make it difficult for IT to meet business goals?

   - What gaps exist in the IT support structure that make it difficult for the business to meet its goals?

   The core services list may not mean that you need to maintain the status quo. It simply means that the particular service is required to be provided in some way. It's here that you begin taking concrete action to correct deficiencies and help propel IT forward.

5. **Decide What to Outsource**.
   Many IT departments continue to believe that *all* technology services must be provided by the in-house IT department. That is simply not the case anymore.

   People have been really wary of the word *outsource* because it can imply that people will lose their jobs. But you have to look beyond all of this and figure out what is best for the business. Outsourcing is not just about saving money; it's also about creating opportunity for the business. Every time you outsource a function, the internal people that were handling that function can redirect their efforts to more value-add projects.

As you review the list of services being supported by IT and the list of support gaps that you have identified, you need to look for the best way to address the core services. Sometimes, this will mean outsourcing a core service. For example, if Microsoft Exchange is a core service, does it make more sense to keep it in-house or should you move it to Office 365?

Other times, it may mean outsourcing a non-core service. For example, you might choose to outsource your SharePoint portal and redirect those staffing resources to managing your Exchange infrastructure.

With your lists in hand, figure out what you can "make someone else's problem" via outsourcing and then pick up the phone and start calling potential vendors.

6. **Decide What to Improve.**
   With what's left, you must now decide where you want to improve. The hard part can be figuring out where to begin. With that in mind, here's a tip:

   Look at the items on the list and, for each one, assign two metrics (a value of 1 to 5 for each):

   - **The difficulty level for addressing that need.** For the difficulty metric, a 1 indicates that it's low difficulty and a 5 means that it's a high level of difficulty.

   - **The potential payoff — in terms of both staff time and money — that will result if that item is addressed.** For the cost metric, a 1 means low payoff value while 5 means high payoff value.

Now, immediately attack the items that have a 1 or 2 difficulty level and a 4 or 5 payoff level. Get them done and done fast.

Obviously, you will need to coordinate with business leaders to really prioritize your efforts so that everything you do aligns with what the business needs.

But What About IT's Needs?

If it looks like this advice is 100% focused on the business rather than IT, that's because it is. However, as you're making your way through these six steps, look for opportunities to improve IT itself.

Would a move from a traditional data center environment to one based on hyperconverged infrastructure be able to address the complexity issue that we discussed?

Would assigning staff to begin automating routine tasks be of benefit to the business?

These kinds of initiatives can absolutely be beneficial to the business as they can free up scarce IT personnel to tackle those complex business challenges that were identified when you began your journey. Sometimes, by addressing internal IT needs, you're also making it easier to address the critical goals of the business.

Chapter Recap

In this chapter, you learned that time is not on your side. But by leveraging the tools and services available to you, it's possible to move beyond just "keeping the lights on." Following are a few key terms and concepts to take away from this chapter.

- **Agile IT** is a set of principles that require that people think *big*, *start small*, and *move fast*.
- There are 6 steps to help you move fast:
    1. Talk to the business
    2. Assess the technology environment
    3. Create a support inventory
    4. Identify core services and gaps
    5. Decide what to outsource
    6. Decide what to improve
- The key to success is to make sure that you can align your IT function with what the business needs.
- The ability to disrupt the complex and expensive legacy environment that pervades many data centers is important.
- By disrupting the cost and complexity equation, companies can begin to shift their focus to value-add services that drive the business.

Up next, it's time to talk about action. Chapter 6 will discuss how to begin the actual transformation of your data center.

6

Transforming Your Data Center

Jack Welch, most known for his role as CEO of General Electric for two decades, is quoted as saying, "An organization's ability to learn, and translate that learning into action *rapidly*, is the ultimate competitive advantage." Starting with the principles from Chapter 5, it's time begin the transformation of your data center. This must be done immediately!

Look at what Jack says: the competitive advantage comes from translating knowledge into action rapidly. Data center transformation sounds like a lofty goal, but it needn't be quite so scary. There are plenty of opportunities to begin effecting radical change without tearing everything down and starting from square one.

Once your IT organization is properly positioned to make this transformation valuable to the business as a whole, it can start picking off the low-hanging fruit. Some examples will be discussed shortly, but the possibilities are as varied as there are companies in the world in need of transformation. As work on the easy targets in the organization helps the software defined data center (SDDC) take shape, the transformation will gain momentum. By the end, the new data center will look nothing like the old.

Align the Data Center and Business Needs

Prior to beginning this transformation process, it's important to evaluate the motives for the transformation. It's easy to get caught up in the technical aspects of the transformation and in the exciting new tools and processes. But transforming a data center is only valuable for one reason, and it's the same reason that the data center exists in the first place: the data center makes the business money.

Tip from the Field by James Green

Whether the data center is the product, or the data center supports the people who sell the product, or some combination of both, the only reason that a data center exists is to make the business money. Keep this perspective when assessing opportunities for data center transformation. You're sure to get stakeholder buy-in when the transformation will have a measurable impact on the bottom line.

— James Green

With that in mind, the first step to transformation is to take a hard look at which transformation choices will affect the bottom line.

For example, a radical overhaul to turn all of the blue LEDs in the storage arrays to newer, sleeker green LEDs are not likely to be well received by the board. However, if these transformations lower operational expenses by reducing administrative complexity, they will be better received. Or do these transformations increase the accuracy of the final product, reducing the number of products that are discarded as faulty or returned? If so, that's another way to garner support. Perhaps the transformations make another business unit happy; that's always a good way to find support for a project! If the finance team needs updates to their portion of the website to be completed more quickly, changing the development workflow and using automation and orchestration to increase the speed of iteration on development projects will make them happy.

Regardless of what the benefit to the business is, you must have a clear goal in mind before beginning this transformation. Implementing a hyperconverged solution to aid in building an SDDC simply for the sake of having a software defined data center is missing the point and is liable to get someone fired.

On the other hand, clearly defining business transformation objectives and achieving business growth by meeting these objectives using the principles and knowledge within this book is a surefire way to garner yourself a pat on the back, a promotion, a raise, a lead assignment on a high visibility project, or what have you.

So, what's the best way to make sure that a project looks good from the start? Get some easy wins right out of the gate. This makes the project look good to stakeholders and increases support for the project moving forward. Let's look at some ways to get started on the right foot.

Where to Address the Low-Hanging Fruit

No business is exactly the same as any other, so there can be no conclusive blueprint for completing the transformation to a modern data center. However, there are a number of technology use cases that apply to a great number of businesses. It's quite likely that one of these use cases applies to your business in one way or another. Any one of these use cases can be the perfect opportunity to show the value of the software defined approach by taking a technology and business process that the organization is familiar with and streamlining it.

Typically, these types of transformations exhibit a bit of a snowball effect. As the transformation goes on, code can be reused, knowledge about the infrastructure gained from a previous phase can accelerate a different phase, and so on. That's why it's wise to begin the data center transformation with one of the technologies that is most familiar to the team, one that has specific value to the business, and one that is extensible into other areas of the data center — the low-hanging fruit. Because of the team's in-depth knowledge of the technology, the project will be easier to complete than the same transformation on a product or system they're unfamiliar with. In other words, they have

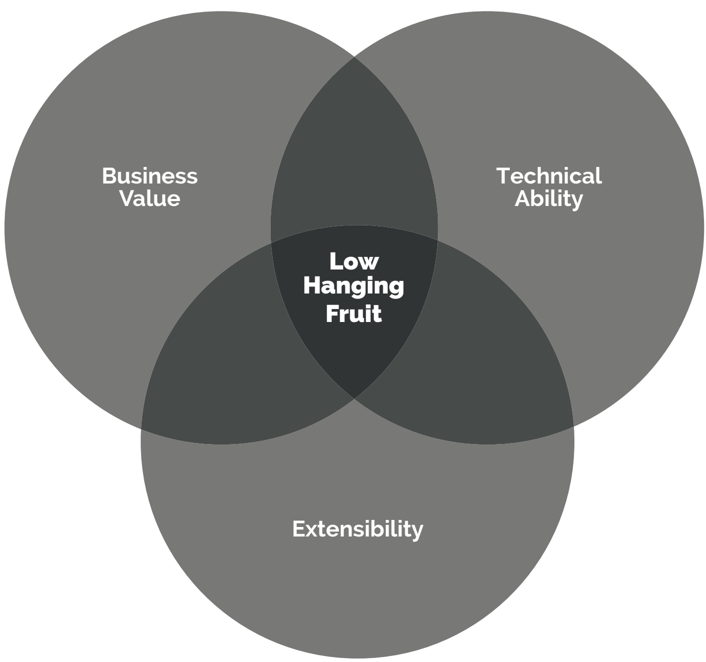


Figure 6-1: Address Low-Hanging Fruit First

a high ability to execute technically. Also, the business value will give immediate return on the investment in the project. And ensuring the work done can be reused and extended into other areas of the data center makes the project more efficient. The Venn diagram in **Figure 6-1** represents the factors in identifying low-hanging fruit for data center transformation.

Test/Dev

The software defined transformation may affect the testing and development (Test/Dev) environments of an organization in a bigger way than any other part of the business. Due to the purpose of the Test/Dev environment, the quicker fresh environments can be created and destroyed, the faster the developers can make progress on projects.

Plus, the more accurately the environments are reproduced each time, the less error prone the Test/Dev process, and finally, the more automated the creation and destruction of environments, the less time operations-and-development staff waste performing repetitive operations. Their attention can then be directed to other important tasks.

Test/Dev environments are low-hanging fruit for many due to the fact that development staff can immediately see the benefit of the work being done, and sometimes are even eager to help. Getting input from the developers can help create an agile infrastructure that caters perfectly to their needs.

Software Defined Networking

Software defined networking (SDN) can be a boon to the development process in that it can enable the rapid deployment of applications that are completely isolated from the rest of the environment. It is all too common in a legacy environment that development components get their wires crossed with production components or (more commonly) an identical development component that another developer is using.

SDN can allow the developers to sandbox their work with no additional effort required, which leads to less frustration and quicker, more accurate testing.

Software Defined Storage

Software defined storage (SDS) could be leveraged to automate the copying of production data to a testing platform in a timely, efficient way. The more quickly and accurately the production data can be replicated in the testing environment, the more quickly deployments can be validated and approved. Also, due to the fact that these types of environments typically contain many copies of the same data, SDS can provide deduplication mechanisms that reduce the capacity needed to store this data. As Test/Dev could be one of the most complicated environments to transform, the effort expended here will likely make other areas of transformation simpler down the road.

Remote/Branch Offices

A software defined approach in remote/branch office (ROBO) situations can really enable an IT organization to accomplish more with less. One of the challenges with remote offices is providing the level of functionality the users want and the level of availability the business needs without spending the money it takes to build a Tier 2 data center in the office. By leveraging software defined compute (SDC), SDS, and SDN, the remote office deployment can be more robust and agile at a substantially reduced price point.

Software Defined Compute

SDC leads the way. In a ROBO, physical space needs to be carefully considered. SDC will allow the creation of a fairly sizable compute

From the Field: Survey Results

82% of survey respondents indicated that they support at least one remote/branch office. A full 19% of them support 25 or more sites! A software defined approach to managing this expansive environment could dramatically cut costs in many environments.

deployment in a small (relatively speaking) physical footprint. Less physical servers also reduce the cooling challenge and consumes less power overall. To top it off, SDC makes it easier to manage all of the remote site's compute workloads from the IT office.

Software Defined Storage

SDS can also help to create similar advantages to SDC. When used to create a hyperconverged infrastructure (HCI), a disparate storage appliance can be avoided and storage can be included in the physical servers. This again reduces noise, heat, and power usage — all of which are important in a ROBO setting. The reduced footprint and increased simplicity also makes it less likely that a dedicated IT resource will be needed in the branch office. SDS also might allow storage workloads to be spread between storage local to the branch office and storage residing in the main company data center. This can increase resilience and control while maintaining performance and a good user experience.

Software Defined Networking

Lastly, SDN can allow the rapid creation of new networks for new offices. It can also enable things that have traditionally been more complex (like stretched Layer 2) in a matter of moments by leveraging technologies like VXLAN. SDN coupled with Network Function Virtualization would also allow the easy deployment of network equipment at the remote office like firewalls, distributed routers, and load balancers where deploying a physical appliance would have been challenging.

Server Virtualization

Server virtualization is the heart of most of today's data centers. Therefore, it makes sense that it's one of the most well understood use cases for SDDC transformation. Deep knowledge of the system will provide the needed leverage in the early stages of transformation. Also, because the production servers are one of the most visible aspects of the IT organization to end users, a successful transformation in this area will help create support for future projects to accomplish the same results in other areas.

The SDDC transformation has already begun in most data centers, but if it hasn't in yours, it must begin immediately. That value of intelligent, optimized, automated server virtualization is huge and provides operational and financial benefits in almost every case. If this process has begun with some basic server virtualization, automating deployments and configurations in the next big step. Leveraging configuration management tools like Puppet or Chef to ensure continuity throughout the environment and radically increase the speed of provisioning will pay dividends.

Software Defined Storage

SDS is likely the big frontier of many data centers today. The challenge is to abstract storage, whether it be monolithic or hyperconverged, and control it with policy and with code. SDS in the server virtualization arena is really a means to an end; creating the SDS platform exposes helpful interfaces to allow for orchestration, higher performance, higher efficiency, and potential space reduction.

To use physical space reduction as an example: an SDS solution that unifies a number of disparate storage arrays might be able to offer global deduplication. This ability alone can dramatically reduce the physical storage footprint and utility costs in the data center.

Software Defined Networking

SDN in the server virtualization world will allow an IT organization, especially service providers and high security environments, to leverage security measures like micro-segmentation to fully isolate east-west traffic. This level of firewall protection would be problematic to build and operate with physical firewalls, but SDN not only makes it possible but relatively easy. Besides distributed firewalls, SDN might also provide distributed routing and switching. This allows a server virtualization network to scale with much less complexity than when using a traditional architecture.

Big Data/Analytics

In the big data space, scale is king. Being able to scale up and down rapidly based on the job that's being run is critical, as idle resources are expensive resources.

Software Defined Compute

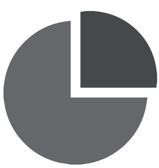
SDC is the key to achieving this. With physical compute nodes for job processing, their resources are wasted. With SDC, the orchestration platform can create and destroy job nodes on the fly to accommodate the size of the job and the availability of existing nodes.

While compute agility is important in big data, the name "big data" also implies there's a lot of it, and all of it must be stored. Storage agility is also critical.

Software Defined Storage

The nature of big data environments is that the data storage requirements are always growing. The storage needs to perform faster to allow for expedient analysis of that data, and the capacity needs to grow to allow more data to be stored. SDS is the only painless way to meet these needs. With silos of storage or even a monolithic, scale out storage array, managing the changing needs of big data storage without pain is unlikely.

Thanks to the abstraction that SDS can provide, however, scaling the performance and capacity requirements of the big data workloads is simple. SDS also allows the workloads to be controlled by policy, allowing the precise needs of the workload to be met automatically.



From the Field: Survey Results

53% of survey respondents indicated that "Big Data" was one of their primary uses cases for using Software Defined Storage or Hyperconverged Infrastructure. With the massive amount of data storage involved, it's no surprise that these organizations are looking to leverage SDS to keep a handle on things.

Software Defined Networking

SDN enables big data analytics by exposing APIs that can be used to automate the creation and destruction of job processing environments. Because big data environments often contain sensitive information, SDN can provide micro-segmentation for enhanced security as well as automated, policy-based access to the data.

Disaster Recovery

While disaster recovery (DR) is last on the list of low-hanging fruit use cases, it is certainly not the least. DR is important to environments of all sizes.

Software Defined Compute

SDC can allow the flexible creation of certain components of DR environments only in the event of an actual disaster. This saves on ongoing costs.

Software Defined Storage

SDS can be the backbone to a successful DR implementation. Leveraging the APIs provided by the SDS platform, IT organizations can create a robust while granular backup and replication strategy. SDS can enable an IT organization to do data protection based on policy. This removes the human element of remembering to add the server to a backup job, for example. Rather, policy dictates that all virtual machines in a specific container are replicated to the DR sites with a specific recovery point objective (RPO). SDS can also dramatically simplify the failover automation process. The APIs provided by the SDS platform can be used by a DR orchestration tool to fully control the failover.

Software Defined Networking

Finally, SDN can be used to programmatically create DR network infrastructure on-the-fly. It can also be used in tandem with processes like revision control to keep the DR site perfectly in sync with the production site. In the event of a DR scenario, SDN will provide the tools to allow a seamless failover from an infrastructure perspective that is all controlled by the DR orchestration tool.

Virtual Desktop Infrastructure

Depending on where you are in the process of creating your organization's unique version of the software defined data center, virtual desktop infrastructure (VDI) can be a big win.

Software Defined Compute

If your organization is not already standardized on virtual desktops, now might be a good time to explore desktop virtualization powered by SDC. This is the process of migrating, or replacing, physical desktops with virtual desktops in an elastic VDI farm.

From an administrator's standpoint, this simplifies their job and increases security due to the fact that the company data never leaves the data center, even if the endpoint devices are on the other side of the world. From an end-user's standpoint, VDI allows increased mobility because they can access resources from a variety of endpoint devices. VDI also breaks the tiring cycle of desktop refreshes by removing physical desktop computers and replacing them with virtual ones and small, cheap endpoints used to connect.

Software Defined Storage

If desktop virtualization is already deployed in your organization, the odds are that there are user-experience and cost challenges that could be easily addressed by implementing SDS. Employee desktops are extremely visible and if SDS can solve a serious pain points such as boot and login times, application slowness, or cutting the infrastructure cost by 50% for an existing VDI deployment, the data center transformation team can quickly become heroes.

Leveraging SDS to create a hyperconverged infrastructure (HCI) for desktop virtualization could potentially eliminate storage performance bottlenecks. Monolithic arrays have been notorious for being crippled by VDI workloads. While plenty of monolithic arrays exist today that can handle the workload, perhaps a scalable, highly performing hyperconverged infrastructure powered by software defined storage can enable more flexibility in the environment.

Software Defined Networking

Finally, depending on scope, perhaps a transition to SDN can be layered on to enable desktops to be created and destroyed in a self-service fashion based on the needs of a business unit. The agility that SDN enables would allow for the secure delivery of desktop services to different business units without sacrificing speed of provisioning new services.

How to Adopt SDDC

The low-hanging-fruit environments make it sound easy, but knowing *how* to get started is the hardest part of the entire data center transformation process. The modern data center can be extremely complicated in its quest for simplicity and hands-off automation.

When it comes to the SDDC, the first step in the transformation is to gain knowledge about specific products and technologies that will enable the transition. We'll explore a few ideas for specific technologies that you could dig into implementing SDC, SDS, and SDN.

Then, armed with some knowledge and excitement, the next step is to begin testing a very simple abstraction: a "Hello, World!" sort of experiment with software defined solutions. This will cement a better understanding of how the tools that enable the SDDC to work.

Tip from the Field by James Green

When adopting an SDDC mentality, there's a critical step that many organizations overlook: begin viewing everything through the lens of the SDDC vision.

If you miss this step, your SDDC vision will never be realized, despite doing all the other steps correctly. It's all too common for an organization to complete an SDN project and then attempt to retrofit the use of the SDN tools and infrastructure in their current operational model. At this point, the value of SDN is drastically diminished and this effort is doomed to fail. To successfully complete the SDDC portion of the transformation to the modern data center, SDDC must be the new operational lens moving forward.

— James Green

Software Defined Compute

Practically speaking, how do you go about adopting SDC technologies and practices into your business? The answer to that depends largely on which use case (VDI, Test/Dev, etc.) will be the target of the SDC initiative.

There are several questions you'll need to answer as shown in **Figure 6-2**.

Figure 6-2: Questions to Consider

Cloud or On-Premises?

Depending on the workload target, the first decision to be made is whether on-premises resources or public cloud resources will be used. If public cloud is the chosen direction, SDC is likely already provided. The task, then, is simply to integrate this resource with the management systems and workflows of the rest of the SDDC. If the compute platform is to be built on-premises, there's many more decisions to make and more implementation work to do.

Virtual Machines or Containers?

The basis of SDC will be some sort of virtualization platform, so it's best to start with choosing that. Immediately, there's a fork in the road: whether to implement machine-based virtualization (virtual machines) or operating system-based virtualization (containers). Because other intersections in the data center rely so heavily on the virtualization method, the choice here will impact the direction of other solutions, especially SDS and SDN. Take great care to explore all the differences when choosing the abstraction type.

At this point, the choice comes down to one of three basic options: virtual machines, containers, or both. For many organizations, the choice for now will be "both."

Hypervisor/OS of Choice?

Assuming your organization has chosen to move forward with both virtual machine and container-based virtualization, the first step of real action is to begin building hypervisors and/or host machines to perform the virtualization. For virtual machines, choosing and installing a hypervisor will be the first step. For containers, choosing and installing the Linux distribution for the host operating system will be the step there. Either way, a system now exists that has the following software defined features: it abstracts the compute workload from the physical resource running it, it provides a programmatic interface to allow for automation and orchestration, and the compute workloads can be controlled by policy. And just like that, the compute is software defined.

Cloud Management Platform?

For bonus points, deploy a cloud computing suite of tools to leverage the virtualization platform as a resource for self-service provisioning, multi-tenancy, metering and chargeback, and more robust automation. While a hypervisor cluster with a management platform technically qualifies as software defined, the real acceleration from a business standpoint comes when the cloud computing tools are utilized on top of the hypervisor or OS virtualization platform.

Software Defined Storage

Adopting SDS can be either dead simple or quite complex depending on the end goal and the tools or products used to get there. SDS also has a large number of variables that can make planning difficult. But it needn't be so complicated if you choose the right set of tools.

One of the primary objectives of SDS vendors is to make the process of implementing the technology easier and smoother. The ideal outcome would be to get the business outcomes SDS can facilitate without the technical complexity, and some vendors have succeeded in providing this.

What's the Objective?

A successful foray into SDS starts with determining the objective. Stephen Covey writes in his best-seller *The 7 Habits of Highly Effective People*, "Begin with the end in mind." Nowhere is this more fitting than in storage design. Is the goal for the organizations storage system(s) to be more scalable? To be able to automate storage provisioning? To allow policy-based workload placement and data protection? Or to allow for non-disruptive upgrades moving forward?

Type of Provisioning and Storage?

Once the goal is identified (and of course, it can be multiple goals) the next step is to choose a storage model. Today, the number of options for storage deployment models seems to grow by the day. At a high level, however, there are two primary paradigms for provisioning storage today: monolithic arrays or appliances, and hyperconverged.

SDS will allow one or both of these deployment models to be used, but knowing which model fits the application will help with making SDS decisions moving forward. Also at this stage, the type or storage medium (or the ratio in the case of mixed medium) needs to be determined. Will the performance and capacity requirements necessitate flash storage? Disk? DRAM?

What Can Be Done with What's Available?

SDS capabilities are commonly overlooked or not utilized due to a lack of understanding. Therefore, one of the steps an organization should take when implementing an SDS strategy is to look at what can be done with what is already available. For example, the organization's hypervisor of choice might already have the ability to place virtual machines on certain storage based on policy.

This is a facet of an overall SDS strategy that can be implemented immediately with technology that is already owned by the business. Showing this value immediately and without a capital purchase can help increase support for later phases of the transformation. However, you should consider whether using SDS mechanisms that are vendor-specific could be a decision that paints oneself into a corner.

Deploying SDS solutions that are tied to the hypervisor could increase the risk of lock-in.

Software Defined Networking

Adopting SDN might be the biggest challenge of the three. But it can also come with some incredible rewards.

What Is the Business Objective?

The first step in adopting SDN, as with the others, is to identify the business objective. For example, "We want to provide enhanced security of multi-tenant environments," or, "We want to enable complete application elasticity." Once a valid business objective is identified, it becomes much clearer which of the candidates for SDN platforms is a good fit.

Hardware or Software?

There's a dichotomy in SDN that also exists in SDS and SDC, but is not as prevalent. That is that some solutions, while enabling the software defined vision, actually require certain pieces of hardware, while other solutions are completely software. Neither model is right nor wrong in a universal sense; the challenge is to choose the one that is right for a given situation.

What Test Cases to Try?

After settling on the platform and deploying the SDN infrastructure, the organization can begin to implement different test cases. One of the easier options for just starting out is to begin to segment east-west traffic via policy. While this would be a somewhat involved process with physical infrastructure, it's really pretty straightforward when the access rules are defined in a policy and the SDN controller handles the programming of the virtual switches.

As your organization becomes more comfortable with the SDN paradigm, perhaps they'll feel prepared to take on a task like the automated provisioning of new network segments. In multi-tenant environments or large organizations that perform frequent acquisitions, being able to leverage SDN to provision these resources in an automated fashion can save the business boatloads of operating capital in the long run.

When adopting an SDDC approach to transforming your data center, you can't expect to change everything at once. The SDDC transformation is often a gradual process, starting with compute, then moving to storage, and finally networking and security. However, the most common process isn't the best for every organization, and there may well be organizations that should transform in the exact opposite order. No matter what, getting measurable improvements to business processes, in terms of reliability and speed, is the end goal. Always aim for that, and the project will likely be successful.

Simplification

As important of a transformation as any other in the journey to the modern data center, the simplification of data center systems and processes has the potential to revolutionize the way an IT organization operates and in the end, how it spends money. The simplification process needs to be a broad and sweeping one, yet inspected at the most granular level possible. Leave no stone unturned in the quest for removing complexity.

Complexity may, in fact, be the single costliest attribute of a data center. Think about the fallout from complexity in the data center: troubleshooting is a total disaster because the system has to be reverse engineered before troubleshooting can even begin; operating expenses increase as more staff is required to maintain the complex systems;

new systems take ages to implement because of the rigidity of the environment. It's plain to see that attention paid to the process of simplification in the data center can return immediate benefits.

A potential angle for approaching simplification in the data center that is applicable to many organizations is to look at the server and storage architecture. Is the management of these systems optimal? Are the systems scalable, agile, and efficient? In many cases, the answer to these questions is *no*. Because servers and storage are so foundational to the data center, beginning the simplification process with these systems can be a great starting point. While certainly not the only option, hyperconvergence is a great way for many organizations to achieve their goals for scalability, agility, and efficiency.

Eliminate Monolithic Storage Systems

It would be fair to speculate that migrating between storage platforms is on almost every IT administrator's "Top 5 Things I Hate to Do" list. Maintaining storage systems is one of the most complex chores that every IT organization has to deal with. The status quo in the traditional data center for many years has been the monolithic storage array.

Unfortunately, everything about monolithic storage systems is painful, from the upgrades to the scaling challenges to the scope of failure domains.

But there's good news!

In the modern data center, there is a better way to solve the storage problem. The solution is hyperconvergence. While hyperconverged infrastructure (HCI) is not a panacea, it's quite a good fit for solving many of the problems exhibited by traditional data center architectures. Chapter 8 will discuss hyperconvergence in great depth, but for the purposes of this section, just understand that *hyperconvergence* is the "pooling of direct attached storage, flash and potentially local DRAM to create a distributed storage system."

Rather than many servers pointing at a single storage target, the storage is spread throughout the servers. Software defined storage (SDS) tools allow that direct attached storage to be protected and managed as if it were one big array.

What does it look like to eliminate monolithic storage? The deployment of hyperconvergence for the sake of simplifying the data center overall is, not surprisingly, quite simple. Most manufacturers that offer a hyperconverged platform go out of their way to make the IT administrator's experience simple. That makes removing the complex monolithic storage array even more attractive. By implementing HCI for a small subset of workloads, data can begin to be moved off the primary storage arrays and into the hyperconverged storage solution. The simplest way to approach this is to leverage SDS to abstract the current storage; this makes swapping the underlying storage is the next phase transparent to the workload.

Alternatively, it seems that finding a certain project that's a good fit and deploying a hyperconverged system (rather than upgrading or scaling an existing legacy system) is a successful way for many IT organizations to begin the transformation. This strategy of finding a specific, well-fitting project and using it as a way to open the door for hyperconvergence can be referred to as *opportunistic hyperconvergence*. In other words, rather than a rip-and-replace transformation of the data center, you would do a new implementation or make the move to hyperconvergence as new systems are built and old systems that aren't supported any more need to be replaced.

Implement Opportunistic Hyperconvergence

Opportunistic hyperconvergence comes in a few different flavors. The first is the one previously discussed — leveraging hyperconverged infrastructure for a specific project to prove its potential. A very common example of this is VDI. Because the nature of VDI workloads is so different from that of server workloads, it is preferred that they run in segregated infrastructures so that they don't cause each other performance problems.

When an organization is looking to deploy a new VDI platform or do a refresh from a previous implementation, hyperconvergence can be a great fit because the workload is to be segregated anyway. Deploying a different infrastructure model for it doesn't cause problems with the design for the rest of the data center. Once the business sees value from the VDI project, then it's much easier to expand into other areas.

Keep in mind that VDI is only an example. Any project where the intention is already to deploy separate infrastructure is a perfect candidate for opportunistic hyperconvergence.

Another way you could practice opportunistic hyperconvergence is to place hyperconverged systems in new remote offices or at acquisitions. Since this infrastructure is outside of the main data center, it gives you the opportunity to evaluate hyperconvergence on a smaller scale. The potential challenge to this scenario is that some of the best benefits of hyperconvergence come with greater scale.

However, if the business is hesitant to try this new direction and a new remote office is being built, why not use that limited-risk opportunity to give hyperconvergence a shot? This outside-in approach is surprisingly easy to grow as internal support for the technology increases. Because of the way most HCI platforms are designed, adding systems in the main data center down the road and connecting them up with the nodes out in the remote office is a trivial process.

Management

It makes little sense to transform the details of the data center for the better if the big picture remains blurry. What will eventually make the SDDC shine in the eyes of the business is having a robust yet nimble grip on the entire data center by using a set of management tools that monitor and control the big picture. Insight is sought after more than gold in organizations today, but providing it is tricky. Taking appropriate action based on that insight is trickier still. The final component to transforming an old, tired data center into a modern data center is to bring new life to the management systems.

It's critical when managing a data center to be able to get a top-to-bottom view of the entire infrastructure. All aspects of operating a data center are made more difficult by not having complete visibility. Being able to manage all the way through the infrastructure stack makes troubleshooting, maintenance, and design more fluid. It's also important to begin to shift toward a hands-off approach where systems function without the need for IT's intervention. This means investing in automation, workflow orchestration, and self-service provisioning. The modern data center accomplishes far more than the data center of the past but with less manual work required of the IT administrators. This frees up staff resources to keep innovating and revitalizing the data center.

Full Stack Management

Because visibility through the stack is so important to the overall management picture, it's vital that your hyperconvergence vendor of choice for transforming the data center provides the tools needed to get this visibility. The more parts of the stack that are under their control, the more insight can be gained. This is the challenge with traditional data centers. The storage system is completely unaware of the network system which is completely unaware of the compute system. Making decisions without all the relevant information is nearly impossible. The only way to make truly beneficial decisions regarding workload optimization or failure protection is to have all the details.

Today, there seems to be two methods of full stack management, neither being more preferable than the other:

- **The vendor providing the infrastructure components also provides the full stack management insights.** This is something that can be delivered by hyperconvergence vendors due to the fact that all the components making up the infrastructure are a part of the HCI platform. However, this method risks some vendor lock in.

- **A third-party tool aggregates data from all the components involved to create the big picture.** In some cases, this may be the only option, and in those situations it's certainly better to have a third party managing the full stack than no one at all. A potential disadvantage to the third-party tool is that the insight may not be quite as deep (though this isn't always the case).

Key Features

The full stack management system (with all of its awareness of the subsystems) should provide, or at the very least enable, the implementation of three very important data center characteristics:

- Automation

- Orchestration

- Self-Provisioning

If it can accomplish these three things, the modern data center will have been realized. It would be preferable that the management system itself provides these functions, but if it doesn't, it's acceptable that it simply exposes APIs to allow other, better suited systems to interface with it in order to perform those roles in the environment.

Automation

Streamlined automation is the hallmark of the modern data center. There are two key reasons why automation is critical:

- **Accuracy.** Humans are notoriously inconsistent and fallible, while the opposite is true of computers. As developers say, sometimes it's maddening that computers don't make mistakes, because that means that if the code isn't working, it's probably your fault! Automation combats our ineptitude by performing repetitive tasks correctly every single time.

- **Speed.** Because a computer can execute code much faster than a human can interface with a computer, automation is also leveraged to complete tasks much faster than they could be completed by hand. Powerful automation is only possible (at least, without expending great amounts of effort) with a full stack management system that can monitor and control the big picture.

Orchestration

The second function the full stack management platform should provide or allow is orchestration.

Infrastructure automation (as mentioned previously) is a step in the right direction; however, the *real* value to the business comes from *fully automating a business process.* This is called *orchestration.*

An example of a business process would be on-boarding a new user with regard to all of the systems that IT manages. Creating an Active Directory account can be automated. Orchestration of this is the workflow which kicks off all of the different automation tasks for creating the Active Directory account, creating a mailbox, adding the user to security groups and distribution lists based on department, provisioning a phone extension, tying an instant messaging/presence identity to an account and phone, and the list could go on. It's easily conceivable that this process could take an hour to a couple of hours for an IT administrator to complete. This leaves the new user waiting and unproductive until the process is complete. In a large organization with thousands of employees, this single business process alone could add up to a few full-time employees' worth of work each year.

However, orchestrating the on-boarding process so that it can be completed without human input would create dramatic savings, as

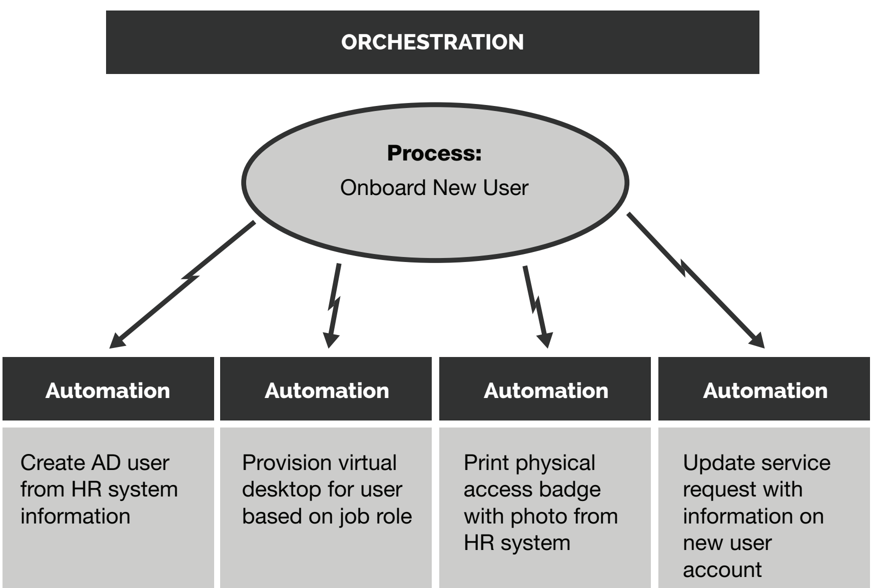


Figure 6-3: Automation versus Orchestration

well as allow the new user to get started almost immediately. **Figure 6-3** illustrates the difference between automation (a smaller task) and orchestration (multiple automated steps carrying out a business process).

Orchestration at the infrastructure layer, enabled by the full stack management system, allows this same level of reduction in effort. The creation and destruction of Test/Dev environments as discussed at the beginning of this chapter could be easily orchestrated so that the entire process is completed without the need for human intervention. However, this can only be done with the right management tools.

Self-Service

The final thing a full stack management system should provide or enable is a self-service provisioning model.

This may be an Infrastructure as a Service (IaaS) platform, or it may be something not quite so complex. However, allowing properly entitled administrators or users to request provisioning of their own resources and then have the system handle it and charge it back to them is the only way the modern data center will be able to keep up with demand. Self-service provisioning of resources will be a direct follow-on of the management system's ability to orchestrate, as fulfilling the request will likely involve a number of different processes and systems.

Chapter Recap

In this chapter, you learned about how to identify and address the low-hanging fruit in your organization and transform the data center. Some of the potential starting points for this transformation that were discussed are: Test/Dev environments, ROBO environments, server virtualization, big data, disaster recovery, and VDI. The following are a few key terms and concepts to take away from this chapter.

- Transforming a data center is only valuable for one reason, and it's the same reason why the data center exists in the first place: **the data center makes the business money.** With that in mind, the first step to transformation is to take a hard look at which transformation choices will affect the bottom line.

- Address low-hanging fruit first. It's wise to begin the data center transformation with a technology that is most familiar to the team, has specific value to the business, and is extensible into other areas of the data center.

- To successfully complete the SDDC portion of the transformation to the modern data center, the SDDC must be the new operational lens moving forward.

- When adopting an SDDC approach to transforming your data center, you can't expect to change everything at once. The SDDC transformation is often a gradual process.

- Complexity may, in fact, be the single costliest attribute of a data center. Thus, **simplification** is key.

- Leverage new projects to implement **opportunistic hyperconvergence,** where the architectural change meets the least friction.

- A successful data center transformation will include a shift in the paradigm of data center management to focus on: **automation, orchestration,** and **self-service.**

One of the major players in the data center transformation is software defined storage (SDS). The next chapter will take an in-depth look at what SDS is and how it works.

7

Software Defined Storage

It's undeniable that all electronic storage is accessed and managed through some type of computer software. So aren't all storage systems built on software? The answer is, yes, of course. All of the traditional storage systems that are in use in data centers today are built on software.

However, there is a difference between *storage* and *software defined storage*, or SDS.

To clear up the confusion about what storage is compared to software defined storage, this chapter will answer questions such as:

- What *is* software defined storage?
- What *isn't* software defined storage?
- How does SDS compare to traditional storage?
- What data services does SDS offer?
- What are the requirements to use SDS?

The SDS Landscape

In the previous chapter we talked about the "software defined everything" trend in the data center; however, here we'll focus on SDS specifically.

The SDS landscape is a bumpy one. Let's answer some of the most common questions about SDS to help clear things up.

What Is Software Defined Storage?

The exact definition of *SDS* is still evolving, but the generally accepted definition is that *software defined storage* is "where the management and intelligence of the storage system is decoupled from the underlying physical hardware."

This means that the SDS software is then able to provide policy-based management and provisioning of the data being stored, regardless of the storage hardware that's being used.

Most SDS systems create a file system overlay on top of the physical hardware. That filesystem is then utilized by virtualization hosts (to store virtual machines) or by physical servers — both of which provide application services.

The SDS storage system is typically distributed across the multiple physical hosts that provide the actual physical storage capacity. The distribution offers both high availability for the data (in the event of failure) as well as performance (by having multiple copies of the data available). That physical storage capacity can be a mix of traditional spinning disk or flash storage (which can be used for caching or for data storage, depending on the SDS design).

From the Field: Survey Results

Some might think that SDS and hyperconvergence are very new and not used in the field but in our recent survey we found that 36% of companies are already using SDS or hyperconvergence and 35% say that they will be increasing that usage.

Today's SDS systems include storage virtualization that provides this abstraction of the storage intelligence away from the underlying storage. SDS systems typically allow consumers great flexibility in the underlying storage hardware and advanced storage features. For example, they may provide deduplication, compression, thin provisioning, snapshots, replication, caching, and other advanced storage functionality.

More and more, SDS systems are fully accessible thought a RESTful API (**Figure 7-1**) so that they can participate in automation and orchestration processes which allow the storage to dynamically adapt, as the business demands change.

Control / Management Plane	Open REST API, unified storage management
Data / Data Plane	Resilence, high availability, data protection, data mobility, data reduction, performance acceleration
Storage / Physical Storage	Local storage, shared storage

Figure 7-1: Software Defined Storage

SDS may be sold separately from the underlying storage, or it may be included with the underlying storage. The most important thing is that the software that makes the storage possible can be separated from the storage hardware. In many cases, SDS runs on top of different operating systems and is compatible with multiple types of underlying storage hardware.

Along with software defined networking (SDN), SDS is a critical piece of the software defined data center (SDDC) ideal that most companies are striving to adopt.

What Isn't Software Defined Storage?

Traditional storage area networks (SAN) and network attached storage (NAS) systems that are packaged as a single solution where the storage intelligence and the hardware are coupled so tightly that neither of them can be interchanged, are not SDS.

Additionally, to be SDS, it must be able to both manage storage and be able to create and present usable storage to applications. Software solutions that consolidate management of storage arrays, or that can use an API to tell a storage system to create a logical unit number (LUN), are not SDS.

SDS Compared to Traditional Storage

So, if SDS provides so many benefits and such tremendous flexibility, why haven't storage solutions always been architected and delivered in this way?

The reason that these so-called "hardware-defined storage systems" were created in the first place was because, at the time, the server hardware that could have run this storage intelligence simply wasn't adequate to provide the processing and throughput that the applications required of it. The rest, dedicated SAN and NAS hardware, was created in order to tightly couple the software with specialized hardware to provide high-performance.

Today with server CPU, bus, and I/O channels being able to offer such high performance, there is no reason that intelligent software can't run on just about any X86 server in the modern data center and provide the storage services that are required by the applications.

If you were to compare SDS to traditional SAN/NAS solutions, you would find the following differences:

- **Hardware flexibility.** SDS runs on existing servers or on commodity servers, which are available from many sources and at more affordable costs.

- **Simplified administration.** SDS solutions are typically administered from a simplified web-based interface that can be understood by most IT professionals in short order.

- **Simplified configuration.** SDS typically sees storage as a single pool instead of managing storage through the construct of LUNs.

- **Advanced features included.** SDS typically includes advanced features in the product (such as deduplication, compression, replication, and more).

- **New features included.** Just like a smart phone, SDS typically includes any new features and functionality whenever a new software patch or update is released.

- **Greater integration.** Because SDS is software running on commodity hardware (just like other operating systems and applications), it stands a greater chance of being able to be integrated into existing data center systems. For example, SDS can be easily integrated with the virtualization hypervisor and virtualization management system through the use of application programmable interfaces (APIs).

- **Lower overall costs.** In most cases, due to the elimination of dedicated, complex, and costly hardware, SDS will almost always be a lower cost, overall. This is true because with SDS you can use commodity hardware since you aren't tied into a costly service contract, and there is lower complexity for administration and troubleshooting.

SDS Requirements

So let's say that, at this point, you are interested in using SDS. What does it take? What are the requirements?

Abstraction, Pooling, and Storage Virtualization

SDS, like server virtualization, is based on the abstraction (or virtualization) and the pooling of resources. Just like server virtualization where you can create a compute cluster, pooling the resources of CPU and memory, with SDS you can just as easily pool storage resources into a storage cluster where they can be managed and monitored as a single resource.

What Can You Pool?

The first step in software defined storage is to abstract away the underlying hardware resources and pool those resources into, typically, one giant resource pool so they can be shared by all servers and applications that need access to the storage.

SDS can pool numerous types of hardware (and even cloud) resources into a single pool that can then be managed using software-based policies which are created and applied based on the needs of the company and their applications.

The most common types of storage resources that can be pooled by software defined storage include:

- **SAN/NAS.** It may seem strange, but traditional SAN and NAS solutions can be abstracted and pooled into an SDS cluster, just like local storage and other resources. The benefits of doing so are that you can gain advanced functionality that your traditional storage likely doesn't offer, such as storage virtualization, tiering, global deduplication, and more. However, pooling of SAN/NAS resources into an SDS cluster is typically only done temporarily until lower cost commodity-based storage can be put in place, and the SAN/NAS can be eliminated.

- **DAS.** Directly attached storage (DAS) is the local storage inside a server and is the most common type of storage bits used in SDS systems today. The reason for this is that DAS storage is the lowest-cost storage available. However, since SDS doesn't require advanced features from the hardware itself and instead provides the advanced features itself, low-cost DAS storage is the ideal storage solution for SDS.

- **Flash Storage.** With the lower costs, higher capacities, and the incredible performance of flash memory, flash storage is being used more and more in just about every SDS solution. Flash storage is commonly used in SDS for performance acceleration through caching or, in some cases, as primary storage in all-flash storage solutions. There are multiple form-factors of flash that you have to select from including solid state drive (SSD), memory channel storage (MCS), and PCIe-based flash. Any of these flash storage options can be added to existing servers and, along with traditional spinning disks, can be used to provide advanced performance (though caching) or advanced efficiency (through deduplication and compression).

Types of Flash

Many may assume that there is but a single type of flash storage and that any variation in price is all a marketing-scheme. However, as of 2015, there are actually five types of flash storage that you need to be aware of. The technical differences cause not only variations in price but variations in life span, endurance, performance, and capacity.

The five types of flash are:

- **SLC.** Single level cell (SLC) flash is the highest performance and likely the highest cost flash available – typically used only in enterprise applications. It has the highest endurance at around 100,000 program erase cycles per cell as well as the highest performance.

- **eMLC.** Enterprise multi-level cell (eMLC) flash offers good performance and good endurance at 20,000 to 30,000 program erase cycles per cell. Used by many enterprises, eMLC many enterprises and higher-end consumer products, eMLC sits between SLC and MLC in price, performance, and endurance.

- **MLC.** Multi-level cell (MLC) flash is lower-grade than eMLC and its endurance reflects that at roughly 10,000 to 30,000 program erase cycles per cell. Used by many consumer-grade applications, it's not recommended for applications that perform many writes.

- **TLC.** Triple level cell (TLC) flash offers the maximum density possible but it's endurance is typically only between 1,000 to 5,000 program erase cycles per cell; however, it's going to be the lowest cost flash storage available. Used for low-end consumer-grade electronics, TLC flash is also the lowest performance flash storage.

- **3D NAND TLC.** A new type of flash storage, 3D NAND, is when TLC flash is configured to use larger bit cells. The result is a low cost flash alternative that provides similar performance and similar durability as MLC flash. The relatively new flash offering is reported to offer roughly the same number of P/E cycles as MLC but at a lower cost.

- **RAM.** Almost every virtual host, where SDS runs, has some excess RAM capacity. Many SDS solutions will leverage a portion of that RAM memory to improve data efficiency with the end result being improved storage performance (**Figure 7-2**).

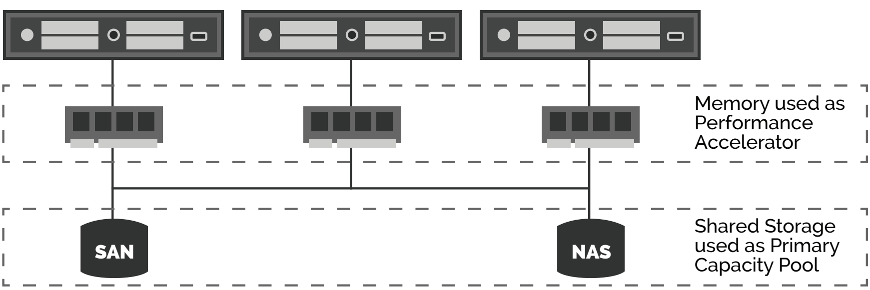


Figure 7-2: Memory Being Used as a Performance Accelerator in SDS

- **Public Cloud.** It may be surprising to see public cloud listed as a resource alongside HD and flash storage but public cloud storage is another of the resources that SDS can abstract and pool. Cloud storage is becoming more popular because of its extremely low cost per GB. However, the downside of public-cloud storage is that, depending on the connectivity you have to the cloud, the latency to access that storage can be high (and usually is much higher than local storage). For these reasons, public-cloud storage is often used for archival and data protection purposes instead of for immediate access. Still, SDS solutions can abstract and pool public-cloud storage as if it were local, but apply policies to it so that only low priority backup or archival data is stored there.

Presenting Storage to the Application

Once the underlying hardware-based storage is abstracted and pooled, that storage must then be presented to the servers and applications that need access. There are multiple methods of doing this and similar to traditional storage arrays, most SDS systems support more than one data presentation method.

Here are the most common storage presentation methods:

- **File.** File-based storage systems provide shared access to entire file systems down to the individual file. File-based storage is the easiest form of shared storage and is commonly accessed by protocols such as SMB (or server message block, used by the Windows OS) and NFS (network file system, used by the Linux OS). File-based storage systems can also be accessed by most virtualization hypervisors. Traditional NAS storage systems are known for offering file-based storage access.

- **Block.** Block-based storage systems provide shared access to SCSI LUNs presented by iSCSI (Internet SCSI) or Fibre Channel SAN protocols. Block-based storage provides efficient transportation of large amounts of data. With block-based storage, LUNs are formatted with file systems such as NTFS (for Windows servers) and VMFS (for vSphere servers). Traditional SAN storage systems are known for offering block-based storage access.

- **Object.** Unlike traditional files in block-based storage presentation methods, object-based storage is presented for the storage of individual, static data objects such as photos, videos, email, backups, and virtual machine images. Object-based storage is ideal for objects that are organized in a traditional hierarchy method. Object-based storage systems are also designed with underlying redundancy such that they have no single point of failure. Object-based storage systems are also among the easiest kinds of storage to scale to support many nodes. Cloud-based storage systems such as Amazon S3 and the Swift OpenStack object-storage project are known for offering object-based storage access.

Advanced Data Services

Being able to abstract, pool, and present storage to servers and applications is critical, and software defined storage provides additional functionality in the form of advanced data services.

You might be familiar with some of these advanced data services based on features that are already available in traditional storage systems. The difference with SDS is that these advanced data services are commonly included with the SDS solution at no additional cost.

Also, SDS makes certain that new types of data services possible, such as data mobility across the data center, thus opening up new opportunities for more agile data centers.

Advanced data services commonly offered by SDS solutions include the following.

Data Reduction

With data at enterprises and across the cloud growing exponentially, enterprises need every form of data reduction available. The two most common forms of data reduction are deduplication and compression.

Deduplication occurs when a storage system reduces the size of the data by eliminating the redundancies (copies of data) over a large data set.

For example, consider for a moment all of the copies of the Windows or Linux OS that are used in an enterprise data center. A very large percentage of the operating systems for servers and virtual machines stored in enterprise data centers is duplicate data. Deduplication allows you to store just a single instance of each block of data in your environment. By enabling deduplication in your storage environment, you can save tremendously on capacity. For example, imagine if you had 100 absolutely identical servers. With deduplication, you would in essence store just a single copy of that server and would not need to store the other 99.

Deduplication Extends the Life of Flash

You know that deduplication saves space. But even better, when well-implemented inline by the vendor, data deduplication can even help you get more life out of your flash storage.

As you probably know, flash storage devices have a finite lifespan, measured in the number of program erase (P/E) cycles that the device can withstand. Every time data is written to a device, the device must perform a P/E cycle on the target cells.

Now, imagine you have 100 identical servers and deduplication means that you only have to store one of them. That means that you can avoid having to write the other 99 copies, thereby forgoing the need to put the flash media through the P/E cycles necessary to write those other 99 copies. This technique, also called write avoidance, is one of the primary methods by which flash storage vendors are able to ensure that flash media can last for the long term.

As shown in **Figure 7-3**, there are different types of deduplication designs such as in-line, or pre-process deduplication (performed at the time the data is written to the storage system), and post-process deduplication (performed after the data has been written). Each type has its own set of pros and cons. See the sidebar entitled "Pros and Cons for Deduplication Types" to learn more.

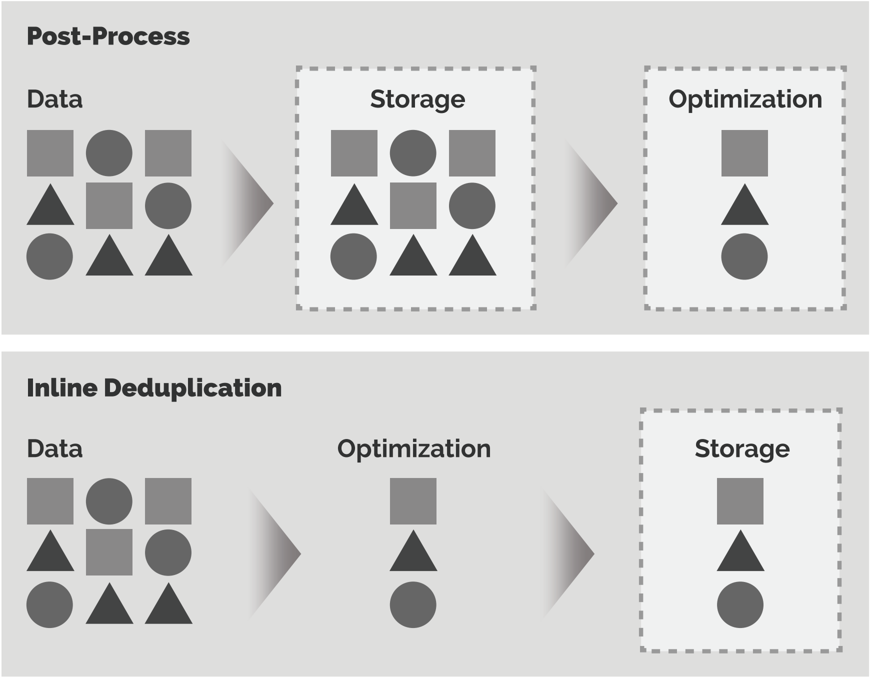


Figure 7-3: Post-Process vs. Inline Deduplication

Pros and Cons for Deduplication Types

There are numerous, highly technical, methods for performing data deduplication, but when viewed from a higher level, the two choices that most IT organizations choose are either pre-process or post-process deduplication. Each of these has its own pros and cons.

Inline Deduplication

With pre-process deduplication, data is deduplicated before it is written to disk (usually in memory or in a flash tier). The upside to this is that there is never any duplication of the data on disk so disk space is not consumed for duplicate data and an I/O is not consumed in the storage infrastructure.

The downside to pre-process deduplication is that there is may be a storage latency penalty and associated with performing deduplication of data before it is written. However, many deduplication solutions are able to eliminate the overhead of pre-process duplication with caching.

With any pre-process deduplication solution, a potential concern is whether or the solution can provide global pre-process deduplication (on all data) or if the pre-process deduplication system just works on a small subset of data and then the remaining data still has duplication (which could be eliminated by post-process deduplication).

Post-Process Deduplication

With post-process deduplication, data is deduplicated after it is written to the disk, by a separate process, after the fact. The upside to this is that there is no performance overhead for the deduplication process as it is typically done during non-peak hours when there are free I/O and CPU cycles anyway.

Another upside is that post-process deduplication has a better chance of performing global deduplication, across all data in the data center, than pre-process as it typically has access and resources to do so.

The downside to post-process deduplication is that it occurs after data is already written, and that means that it happens with the penalty of a write I/O and requires storage capacity on disk (until the post-process deduplication process happens).

Many advanced storage systems give enterprises flexibility by being able to perform both pre-process and post-process duplication.

Also of special consideration when considering deduplication systems are the block size used when the deduplication is performed as well as the type of media that the deduplicated blocks are stored upon. Both of these factors can be significant factors in the performance of the deduplicated data, whether it is pre- or post-process duplicated.

Compression, on the other hand, also eliminates duplicate data but does so in small data sets such as in a file or a block of data. Like deduplication, compression requires CPU processing cycles and doing real-time compression (comparable to pre-process deduplication) has its trade-offs between data reduction and performance throughput.

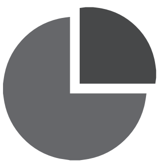
While both compression and deduplication can consume resources if done in real time, most modern SDS solutions will eliminate that performance penalty via the use of intelligent caching, providing both storage reduction and high performance.

I/O Acceleration

You previously learned about pooling resources and how RAM and flash storage can be part of those pools. SDS solutions often use high-speed RAM and flash storage, not just to mitigate the performance impact of software defined data services but also to accelerate storage throughput.

For example, in order to achieve the highest virtual-machine-to-host consolidation ratio possible or to run applications that have high I/O per second (IOPS) requirements, SDS solutions may use RAM or flash to temporarily write storage I/O (usually across the cache of multiple hosts for redundancy) before acknowledging the I/O and then writing it to permanent disk storage.

Because SDS allows you to manage your storage using policies applied per virtual machine or application, this type of I/O acceleration can also be applied on a per-virtual machine or per-application basis instead of across an entire array or an entire storage LUN.



From the Field: Survey Results

In our recent survey, respondents rated advanced data services as either important or very important when it came to selecting SDS or hyperconvergence solutions. For example, 90% said data reduction was either important or very important and 92% said that replication was important or very important.

Snapshots

As with virtualization hypervisors and traditional storage systems, you can take snapshots with SDS solutions.

The downside to hypervisor-based snapshots has always been the performance penalty for taking and retaining the snapshot.

The downside to traditional storage snapshots is that they usually require you to snapshot the entire storage LUN. This requires much more capacity than is needed.

Depending on how the storage is presented, SDS lets you take storage- or virtual machine-level snapshots that are virtualization-aware and, in some cases, deduplication-aware. Essentially, what this means is that you get the best of both worlds. You get to offload the resources necessary to take the snapshot onto the storage (where it belongs), but the preserved data only pertains to the virtual machine.

Cloning

Almost every virtualization administrator has used clones in their virtual infrastructure. Clones duplicate a virtual machine in the case of a virtualization hypervisor. Clones can save administrators a tremendous amount of time, because when they need to create new virtual machines, they can simply clone an existing virtual machine.

Clones are also used heavily in virtual desktop infrastructure (VDI) implementations where many virtual machines are based on a single "golden image" virtual machine that is then cloned to create all the end user desktops in the pool.

The concern that administrators have always had with clones is the performance impact that they will have depending on what type of clone is being performed. With SDS cloning, like snapshots, enterprises receive the best of both worlds where a clone of a virtual machine can be taken, with virtually no performance penalty, and be used immediately.

Replication

Data replication is used heavily by companies of all sizes for data protection in the event of a localized data center outage (such as failure of a SAN or NAS) or in the event of a large scale outage (e.g., the company headquarters and data center were destroyed by a tornado). No matter the issue, when data is not available, applications are going to be down — the company will start to lose money.

Replication essentially copies data on disk to another disk, and that other disk could be local to the same data center, or in another data center across the state or country. In the event of data loss, that data can be used to bring up, for example, the virtual machines running the company's point-of-sale system that were replicated to a backup data center.

With traditional storage, replication was typically only enabled when a separate license key was purchased from the storage vendor. The most granular data replication that could be done was on a per-LUN basis, which likely included many virtual machines.

With SDS, however, replication is yet another of the advanced data services that can be enabled on a per-virtual machine basis and it's included in the SDS software (no extra key or licenses are needed). SDS-enabled replication can be used to protect data both in a local data center cluster, across data centers, or across the cloud.

Data Mobility

Once SDS abstracts away the physical storage hardware, your data is now mobile and can be moved across any of the various forms of storage — flash, HD, SAN/NAS, cloud, and more (**Figure 7-4**). What this means is that if, for example, you replace your SAN and move to I/O-accelerated local storage, your data can move dynamically, from one form of storage hardware to another without any downtime for the applications using that data.

However, storage mobility isn't just for hardware replacement. For example, let's say that you want to provide a virtual machine with a higher tier of storage (from silver to gold). With storage mobility, you could change the storage policy on a virtual machine and, with no downtime for the virtual machine or its applications, the virtual machine storage could be migrated from one type of storage (perhaps

SATA “bronze tier” storage) to another type of storage (perhaps I/O-accelerated “gold tier” storage).

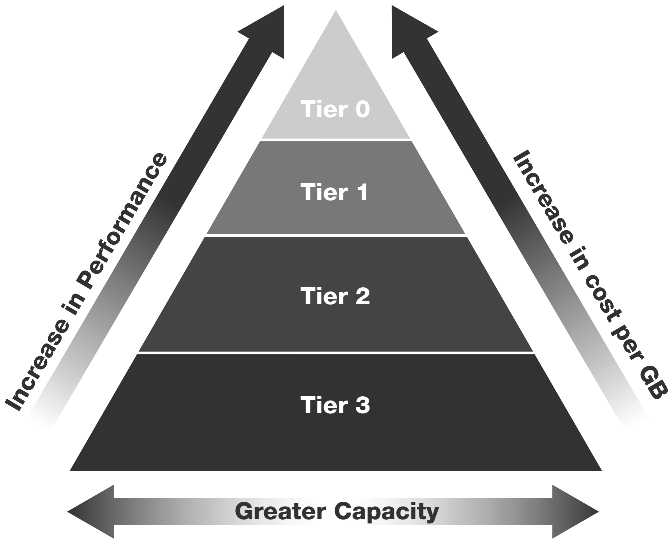


Figure 7-4: Tiered Storage Systems

Encryption

Just like other advanced data services enabled all in software, data encryption can be enabled to ensure that all data (or just an individual virtual machine, for example) is encrypted and secure when stored.

Thin Provisioning

When it comes to SAN LUN provisioning or virtual machine disk provisioning, LUNs and virtual disks are filled with whitespace in order to reserve their total capacity. However, in most cases, the storage provisioned is only fractionally used. Thin provisioning tells the device which requested the storage that the total capacity is available. However, in reality, the only storage capacity that has actually been used has been reserved (not the total).

As shown in **Figure 7-5**, with SDS, thin provisioning is enabled by default and is done across the entire storage pool, not just on individual LUNs.

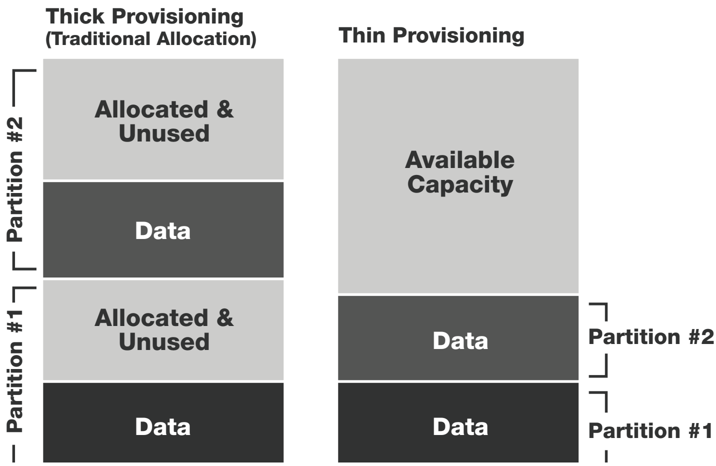


Figure 7-5: Traditional Allocation vs. Thin Provisioning

Resilience/Data Protection

Everyone knows Murphy's Law, "if something bad can happen, it will." Storage systems aren't immune to this "law." Thus, every storage system should ensure that it offers data resiliency and multiple levels of data protection. SDS is no different.

For example, what if you lose a disk? A group of disks? A caching disk? Or a host in the storage cluster?

When analyzing SDS solutions, there are several types of data resiliency and data protection capabilities to ask about.

SDS Checklist

When considering SDS options, there are many traits that SDS should display. They are:

- [ ] Software-only
- [ ] Data reduction such as deduplication & compression
- [ ] High availability
- [ ] Data protection
- [ ] Pooling of different types of storage media
- [ ] Automation (often through REST API)
- [ ] Simplified configuration
- [ ] Policy-based management
- [ ] Storage virtualization
- [ ] Hardware flexibility
- [ ] Integration with multiple hypervisors (or at least the ones you plan to use for the foreseeable future)
- [ ] Lower upfront and total cost
- [ ] I/O acceleration
- [ ] Snapshots
- [ ] Cloning
- [ ] Replication
- [ ] Data mobility
- [ ] Encryption
- [ ] Thin provisioning
- [ ] Stretched Cluster (AKA Metro Cluster)

Failure Scenarios

First, consider all the different ways that an SDS system could eventually fail. What will be the effect on the data and applications using that data should the storage system fail?

Avoiding Single Points of Failure

Most commonly, an outage for a storage system occurs when there is a failure of critical hardware. For example, failure of a node that contained shared data which was not redundant would cause that data to be unavailable. Thus, that node was a single point of failure in the storage infrastructure.

SDS solutions mitigate to prevent a single point of failure by automatically replicating data to other nodes in the storage cluster.

Recovery

When failure does occur in the storage system, storage administrators should be notified so that they can repair or replace the failed hardware, at which time the SDS system can rebuild the distributed data, across all nodes in the storage cluster.

High Availability

With traditional SAN/NAS storage, creating a fully redundant storage infrastructure is complex and expensive. To create true storage high availability, you must either purchase a second, redundant SAN/NAS or, at minimum, configure your SAN/NAS with fully redundant everything.

SDS systems must provide high availability both at the disk level and host level.

Complete Virtualization Implications

You should always consider the implications of virtualizing the storage infrastructure on top of the virtualization hypervisor. If there is a problem in the virtualization layer, it could also mean that the SDS infrastructure won't work, and, thus, neither will any applications or servers that require that storage. This is the classic "chicken or the egg" scenario.

However, with most companies moving toward 100% virtualization, if there is a problem in the virtualization layer where virtual machines won't start — and the storage layer is one of those virtual machines — then you have a critical issue.

Chapter Recap

In this chapter, you learned what software defined storage is, how it compares to traditional storage, the different approaches to SDS design (VSA versus hypervisor/kernel-integrated), and the requirements that you need to meet in order to use SDS. You also learned how SDS is used to abstract and pool both traditional and modern storage options (like flash) into a single shared storage resource pool. Finally, you learned about the advanced data services (such as compression, deduplication, replication, and I/O caching) that modern SDS systems offer to enterprises — all in software, all included, and without additional costs. The following are a few key terms and concepts to take away from this chapter.

- SDS is where the management and intelligence of the storage system is decoupled from the underlying physical hardware.

- Software solutions that consolidate management of storage arrays or that can use an API to tell a storage system to create a LUN **are not** SDS.

- SDS is characterized by **abstraction, pooling,** and **storage virtualization.**

- Advanced data services commonly offered by SDS solutions include the following: **Data Reduction, I/O Acceleration, Snapshots, Cloning, Replication, Data Mobility, Encryption,** and **Thin Provisioning.**

- When considering SDS platforms, look at how the solution handles **failure scenarios and recovery**, how it **avoids single points of failure,** and how it **provides high availability.**

With a solid understanding of what SDS is and how it works, the next step is to explore the topic of hyperconvergence and see how SDS enables hyperconverged infrastructure (HCI).

8

Hyperconverged Infrastructure

With the rise of server virtualization, companies are realizing the improved efficiency, productivity, and return on investment (ROI) that server virtualization can bring to the data center. Specifically, many virtual machines can be run on each host, administrators can "do more with less" by administering more servers per administrator, and companies can gain a greater financial return on investment in every single server once the virtualization hypervisor is installed.

From the Field: Survey Results

When we surveyed companies about the status of server consolidation using virtualization in their datacenter, we found that 73% of companies surveyed were 50% virtualized or greater in their datacenter, 42% were 80% virtualized or greater, and 8% were 100% virtualized.

When companies first started virtualizing servers, they realized how inefficient and costly the storage silo of the data center was. It slowed the provisioning of new virtual machines, it was complex to maintain, and required dedicated administrators and experts. Worst of all, it ate up the majority of infrastructure budgets while providing a much lower return than other data center infrastructure. But what could be done about the storage?

The initial answer came in the form of "converged infrastructure" where a company would pre-rack and pre-configure servers, storage, and network in order to eliminate the deployment step while offering a single phone number and contract for support of the entire stack. Converged infrastructure was also pre-architected such that specific configurations would support specific workloads. This idea was beneficial for some of the largest enterprises out there, but not right for most. In the end, converged infrastructure was seen as too large of a purchase for most companies, and it wasn't as scalable as they needed — after all, the way to add more capacity was to buy another very costly "block" of convergence.

The next solution to the storage problem came in the form of *hyperconvergence*. What is hyperconvergence, exactly?

Hyperconvergence

Hyperconvergence breaks down into hyper- (meaning "quickly, or with a hypervisor") and convergence (meaning "bring together"). Therefore, you can loosely translate hyperconvergence to mean "quickly bring together with a hypervisor, or software." In the case of today's hyperconvergence solutions, they bring the storage into the compute. In the future hyperconvergence may include the network, or more.

Hyperconvergence Defined

The term *hyperconvergence* has been thrown around a lot lately. Plus, there are other similar terms being used as well, *including Server SAN,* *converged infrastructure, hyper-convergence,* and more. So what do we really mean by hyperconvergence?

Figure 8-1 shows what hyperconvergence looks like:

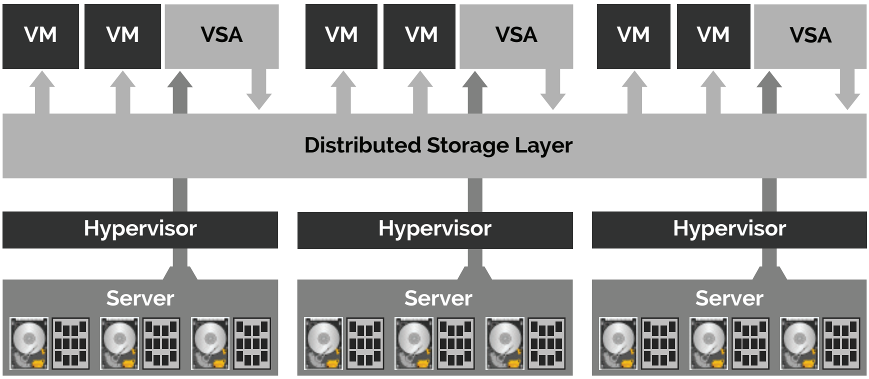


Figure 8-1: Hyperconvergence Architecture Utilizing a Virtual Storage Appliance Design

As you can see (**Figure 8-1**), the servers run a hypervisor, and that hypervisor either runs the virtual storage appliance (VSA) or uses a hypervisor-integrated storage management module to provide the distributed storage layer for the virtual machines. That distributed storage layer is spread across the servers that make up the storage and compute cluster.

Distributing the storage across the compute layer using software defined storage (SDS) and running your business workloads alongside the VSA is the minimum requirement to dub something "hyperconvergence." However, most modern hyperconvergence solutions also provide:

- **"Single pane of glass" management.** This means a single, integrated management interface is available for both the virtualization and storage resources, all in one place, for administration, performance/capacity monitoring, and troubleshooting.

- **Simplified packaging and support.** You get a single vendor from which to purchase the hardware and software (including virtualization and storage), as well as a single support contract to a single support group that will support the entire HCI (which includes server hardware, an SDS layer, and a virtualization layer).

- **Advanced data services.** If you recall the advanced data services covered in the previous chapter (e.g., deduplication, compression, I/ O acceleration, snapshots, cloning, and replication), many hyperconvergence solutions provide these types of advanced features all in the software along with the solution.

Hyperconvergence Implementation Options

When you consider the variety of hyperconvergence solutions available today, you'll find that besides the differences in what is offered, there are a number of differentiators in how hyperconvergence is implemented.

SDS in Hyperconvergence

Increasingly, hyperconvergence is being driven with greater and greater innovations in the area of software-based advanced data services. Think of this as, "What else can the SDS do, other than just distribute the storage across the compute?"

Just as server virtualization provided numerous management efficiencies, there are numerous ways that SDS could make the lives of data center administrators easier.

For example, when learning about and shopping for SDS solutions, you should ask, "How can the SDS solution..."

- **Reduce data being stored through** intelligent deduplication and compression, saving storage costs?

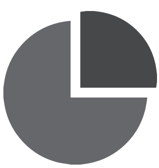
- **Accelerate I/O** to allow you to virtualize applications that weren't candidates before (due to their high I/O demands), and/or allow you to consolidate more virtual machines on the same compute or storage cluster in order to provide a greater ROI?

- **Provide integrated data protection** through offsite replication and cloud backup options?

- **Eliminate existing third-party solutions** to provide you greater management efficiency and greater return on investment?

The Hyperconvergence Design Model

When SDS is joined with a hypervisor and a unified management interface is offered, this is typically the minimal design requirement of any hyperconvergence solution.



From the Field: Survey Results

When we asked respondents to tell us what their future plans are regarding storage, the responses paint a bleak future for disk-based storage. A full 19% of respondents – almost 1 in 5 – say that they will fully decommission their disk-based storage systems over the next two to three years. The primary gainers in the same timeframe will be all flash arrays and hybrid storage arrays, but 35% each also say that they will expand their use of software defined storage and hyperconverged infrastructure.

This union of the two can be done in two different ways:

- Hypervisor/kernel-integrated
- Via a virtual storage appliance (VSA)

Hypervisor/Kernel-Integrated Storage

With hypervisors/kernel-integrated storage, the SDS solution runs inside either the kernel of a virtualization hypervisor or inside the kernel of another operating system.

The benefits of this type of SDS deployment are that the storage software has direct access to the operating system and thus, the underlying storage hardware. Some vendors claim that hypervisor/kernel-integrated storage offers a slight performance benefit over the VSA approach (discussed more later); however, the real performance difference in SDS is determined by a combination of the SDS software's speed, the storage media used for caching/acceleration, the storage media used as the primary storage tier, and the networking connecting all of these components.

When inline deduplication and compression is incorporated into the SDS platform, the closer the data reduction is to the application/workload, the faster the SDS system will perform. For example, doing deduplication or compression in DRAM on the hosts where the application is running will result in the lowest latency and best storage performance.

The drawback to using this type of SDS deployment is that the SDS solution is typically only compatible with the kernel of the virtualization hypervisor or operating system that it's integrated with. What that means to enterprises is that hypervisor/kernel-integrated storage solutions are version-locked, and in some cases feature-constrained, due to this level of deep integration.

Additionally, some hypervisor/kernel-integrated storage solutions only support local storage, which limits your storage options.

Another drawback is that these types of SDS deployments may require the purchase of specific and costly virtualization hypervisor licenses that really are required to run the SDS solution, and that don't offer immediate value to the enterprise. All of these drawbacks are typically summed up into the single term, *hypervisor lock-in*.

VMware's Virtual SAN (VSAN) is one example of a hypervisor/kernel-integrated SDS solution (as shown in **Figure 8-2**, below). VSAN is integrated into the vSphere hypervisor, is incompatible with other hypervisors, and requires that you license vSphere for each of your hosts in the SDS cluster. (Another example of a hypervisor/kernel-integrated storage solution is Microsoft Storage Spaces.)

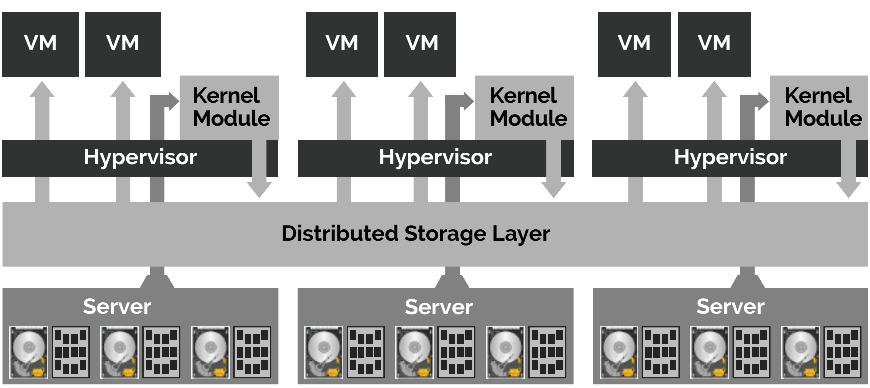


Figure 8-2: Hyperconvergence With a Hypervisor/Kernel-Integrated Storage Design

Virtual Storage Appliances

For hyperconvergence solutions that utilize a VSA, the storage runs inside the kernel of an operating system which, in turn, runs inside of a virtual machine. This virtual machine, with its operating system and storage running inside, is what is termed a *virtual storage appliance*, or VSA (as shown in Figure 8.1, earlier in the chapter).

Different VSA solutions have different requirements for the minimum number of nodes necessary to achieve high availability. Some can operate with as few as two nodes and, as such, are the most economic. Others require a minimum of three or four nodes.

The benefits of the VSA approach include:

- **Compatibility.** Compatibility with multiple virtualization hypervisors or operating systems gives the enterprise great flexibility on where they run the SDS and at what cost.

- **Ability to upgrade and/or change the underlying virtualization or hypervisor without affecting the storage system.** Of course, the VSA vendor must have the ability to support the hypervisor choice you may make.

- **Flexibility.** The VSA approach offers the flexibility to use the lowest cost virtualization hypervisor or operating systems to run the SDS on top of, as makes sense for the business.

With the VSA approach, the hypervisor is essentially commoditized, allowing the business to choose whatever solution is the lowest cost or makes the most sense.

No matter which hyperconvergence design you leverage (hypervisor/kernel-integrated or VSA) the high-level benefits still apply, and you'd be far better off than you would be leveraging traditional storage solutions.

In many cases, too much is made out of how the storage layer is designed — either with VSA or hypervisor-integrated storage. In the end, neither the applications nor administrators will think about this design choice on a daily basis.

What matters is the efficiency of the hyperconverged system in terms of how many compute and storage resources are available to workloads.

To be candid, there are more important hyperconvergence implementation options, such as the following.

Hybrid or All-Flash

Of particular importance when choosing a hyperconvergence solution are the types of storage media and how they are used in the HCI nodes. Hyperconverged solutions can use all hard drives, hybrid (mix of flash and hard drives), all-flash, or DRAM with flash.

The number of hard disks versus flash disks is going to have a number of business impacts, because it determines:

- **The number of virtual machines** that can be run on the hyperconverged infrastructure.
- **The performance that the hyperconverged infrastructure provides the applications** and the ability of the solution to run highly intensive applications such as virtual desktop infrastructure (VDI) or database analysis.
- **The cost of the hyperconverged infrastructure solution.** The more flash disks, the higher the cost of the solution in most cases. However, the more flash disks the nodes have, the greater the performance and the more virtual machines you'll be able to run. Thus, having more flash disks (or all flash disks) may actually more than pay off in the end, because that additional storage I/O

performance that you'll achieve can be used to run additional virtual machines. This, in turn, provides you a greater return from your hyperconvergence investment.

Keep in mind though, that the cost comparison isn't always this simple. Many modern hyperconvergence solutions utilize pre-process deduplication in order to significantly reduce the flash storage requirements and, thus, reduce the cost. In many cases, the cost of an all-flash HCI can be reduced to the point that it costs less than HCIs that utilize hybrid storage.

Additionally, modern hyperconvergence solutions can also use memory to eliminate the pre-process deduplication overhead and even accelerate storage I/O performance for the the data stored in the flash infrastructure.

Appliance vs. Software/Reference Architecture

When choosing a hyperconvergence solution you'll find that numerous HCI solutions are offered as integrated "single SKU" appliances (essentially a physical server with a disk, virtualization layer, and distributed storage layer on top). Or, they are offered simply as software-only solutions that you can choose to implement yourself, on whatever hardware you choose.

For the HCI solutions that are offered as software-only, many of those offer a reference architecture. Reference architectures tell enterprise architects and administrators how many virtual machines or what type of performance the enterprise will receive if they configure the hyperconvergence software with a specific configuration of CPU, memory, and storage.

Reference architectures might also be as specific as to state what you can achieve if you use a specific brand and model of server, with a specific configuration. Then, if the enterprise truly uses the architecture as specified and doesn't receive the expected outcome, the hyperconvergence vendor will support and troubleshoot the performance of the HCI.

In other words, you can think of reference architectures as blueprints for an HCI that are defined by the hyperconvergence vendor in cooperation with hardware server vendors.

Both of these approaches have their own sets of pros and cons.

Packaged HCI Appliance Approach

With the "single SKU" packaged HCI appliance approach, the enterprise doesn't have to think about server hardware, compatibility, or even sizing. Additionally, with the appliance approach, if there is a problem in the infrastructure, they will have a single group to call under a single support contract.

The downside with the packaged hyperconverged appliance approach is that they are essentially locked in to only specific, static, server configurations from their hyperconvergence vendor with few options for price negotiation.

Software-Only Approach

With the software-only, reference architecture approach, enterprises likely have the option to use their existing servers or at least obtain servers from their existing vendor/VAR at previously negotiated discount levels. This maintains the enterprises' comfort-level while gaining the best of both worlds: leveraging a pre-architected, pre-sized, and fully supported blueprint to build their own HCI, while using their own hardware.

Also with a software-only approach, reference architecture approach, enterprises have the flexibility to leverage whatever hardware that they choose and that gives them the option to use the latest and greatest, highest performance, densest server configurations possible. For example, with software-only, enterprises could run hyperconvergence on blade servers or on super-dense servers with massive amounts of memory and CPU.

Enterprises also have the option to either *scale up* or *scale out* by either adding more CPU and memory to existing hosts or by adding more hosts of varying sizes (discussed more in the "Question of Scale" section later).

Hypervisor Choice

Another aspect of hyperconvergence selection is whether the solution offers a choice in hypervisor support. Any solution that is hypervisor-integrated is only going to be offered on the hypervisor that the integration was done on.

It's the virtual storage appliance (VSA) design of many hyperconvergence solutions that support multiple hypervisors and offer the most hypervisor choice.

Understand that even hyperconvergence solutions which support multiple hypervisors typically don't support a heterogeneous hypervisor cluster (a single cluster made up of hosts utilizing different hypervisors).

While the hypervisor that you are using today may be supported by the hyperconvergence solution you are considering, it's important to remember support for multiple hypervisors in your selection criteria, because these hyperconvergence solutions will give you the greatest leverage when it comes time to negotiate software maintenance and support on your existing hypervisor (whether you choose to exercise that choice or not).

Server Hardware Choice

As mentioned previously with the comparison between "Appliance vs. Software/Reference Architecture," when considering hyperconvergence solutions, it's important to ensure that they also offer you hardware choice. You may have little choice in hyperconvergence solutions that are sold or packaged as an integrated appliance (some vendors offer you the choice between two brands of appliances), but will have ultimate flexibility with hyperconvergence solutions that are sold as software-only, reference architecture.

While some form of lock-in is inevitable when choosing hardware and software solutions, enterprises want to ensure that the solution that they choose offers them choice or the flexibility to easily move to the alternative (whether they ever choose to exercise that choice, or not). In other words, very rarely is any enterprise ever really "locked in" and unable to make a change in hardware or software. So the real question is, "What is the price to change?" The level of lock-in is very different if the price to move is $0 versus $10 million.

When considering choice in the context of server hardware; however, there are many organizations that want someone to tell them what to do. It's easier for these companies to simply buy a complete hardware and software bundle that they can put it a rack, turn on, and immediately begin using. While they may become locked in to that platform, that is *by choice*.

The Question of Scale

Before we go into details on the question of scale, let's define the difference, between scale up and scale out (**Figure 8-3**).

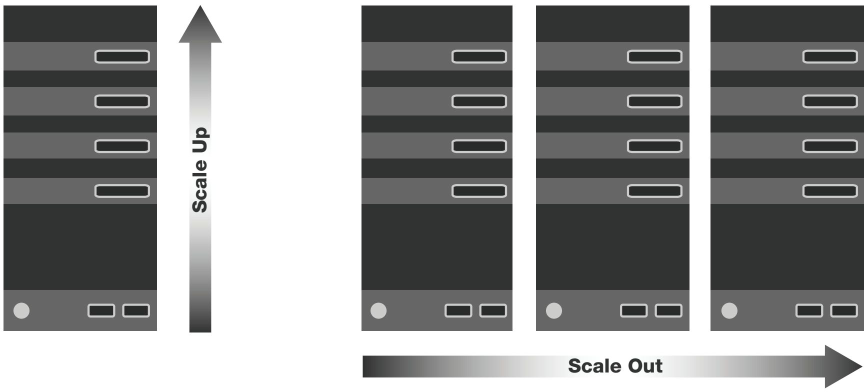


Figure 8-3: Scale Up vs. Scale Out Architecture

Scale Up Defined

With scale up architectures, when you need additional resources (whether it's CPU, memory, storage capacity, or storage throughput), you only have the option to scale up or add more of those resources.

For example, with traditional storage arrays that were scaled up, you had to add more disk shelves to the array to add more capacity and throughput (that is, until you hit bottlenecks with the storage controllers in the array head).

With scale up hyperconvergence solutions, you have the option to add more CPU, memory, storage capacity, or storage throughput to your existing servers in the hyperconvergence cluster.

Scale Out (Linear Scalability) Defined

With scale out solutions, or solutions that scale linearly (like a horizontal line), when you need to add more resources (such as CPU, memory, storage capacity, or storage throughput), you only have the option to "scale out."

What that means is that, when you need more capacity, you add another node complete with more CPU, memory, storage capacity, and storage throughput even if all you needed was, for example, just more memory.

Thus, with the scale out hyperconvergence design using the appliance-approach, you will always be over-provisioned.

Do You Have Scale Up and Scale Out Options?

Another consideration when selecting a hyperconvergence solution comes back to the question of how much scalability is available.

When we discussed "Appliance vs. Software/Reference Architecture" we covered how, with the appliance-based approach, there are concerns about the lack of scalability (similar to the older converged infrastructure approach where your scale is limited by the unit you can purchase). If a hyperconvergence solution is only available in a specific hardware appliance then, while the granularity of the scale is smaller, it is still a one-size-fits-all solution.

To counter this, appliance-based hyperconvergence solutions have started offering their hyperconvergence hardware appliances in multiple sizes; however, there are, and will always be, enterprises that require different configurations for unique applications and use cases.

Appliance-based hyperconvergence solutions are only able to answer the scale question with the option to scale out.

With software-only hyperconvergence solutions where you have the option to bring your own server, you have the option to either scale up or scale out. Plus, just as importantly, you have the option to scale compute (CPU and memory) independently of storage, in either direction.

Chapter Recap

In this chapter, you started off by learning what hyperconverged infrastructure (HCI) is with a basic definition of a storage layer that is distributed across the compute infrastructure. However, you also learned that there is much more to hyperconvergence than its most simplistic definition. You learned what makes hyperconvergence unique, options to consider when selecting a hyperconvergence solution, and the various hyperconvergence design models available to choose from. Finally, you learned how you need to make sure that your hyperconvergence solution can both scale out as well as scale up, to give you the ultimate flexibility to meet the needs of your applications while not forcing you to overprovision your resources. The following are a few key terms and concepts to take away from this chapter.

- In the case of today's hyperconvergence solutions, they bring the storage into the compute, although the hyperconvergence of the future may include the network, or more.

- In the end, neither the applications nor administrators will think about this design choice on a daily basis. What matters is the efficiency of the hyperconverged system in terms of how many compute and storage resources are available to workloads.

- HCI design choices include: hybrid or all flash, appliance or software, which hypervisors are supported, which hardware is supported, and will it scale up, out, or both?

In the next chapter, you'll read about some real world experiences with the software defined data center (SDDC), SDS, and HCI. Chapter 9 will focus on case studies and interviews detailing experiences with these technologies.

9

From the Field: Software Defined Storage & Hyperconverged Infrastructure 2016

IT budgets are shrinking. Demands on IT are increasing. Data center technology has become a quagmire of complexity. Traditional storage has struggled to keep pace with workload demands. With these challenges, CIOs, technical decision makers, and IT staff members are looking for ways to continue meeting critical business needs with solutions that stabilize data center costs while also being simpler to manage. Perhaps the biggest challenges facing the data center today revolve around storage. It's expensive. It's complex. And, until flash became more common, it suffered a great deal with regard to performance.

Both *software defined storage* and *hyperconverged infrastructure* have emerged as solutions intended to solve the storage problem. They have entered the market mainstream as forceful options for consideration. Both bring heretofore unheard of levels of simplicity while also helping to turn the data center economic picture on its head. Rather than buying three to five years' worth of storage, data center administrators can take more of a "just in time" approach to storage, thanks to the easy scalability opportunities that present themselves with these architectural options.

Much remains misunderstood around software defined storage and hyperconverged infrastructure, though. There is often confusion about what these terms even mean. In short, software defined storage is a direct replacement or enhancement for existing storage arrays and, as the name suggests, leverages a software layer to provide storage services that used to exist only in hardware. While it is possible to build a brand new software defined storage architecture, many organizations layer software defined storage tools atop existing storage devices in order to breathe new life into them and leverage the hardware investment that has already been made. To expand capacity in a software defined storage system, administrators can either add more nodes (scale out) or add more storage to existing nodes (scale up), making such systems easily scalable.

Hyperconverged infrastructure takes the data center to new levels by eliminating the array altogether and combines storage and compute into single nodes. In both cases, growth is achieved via *scale out* mechanisms. As more capacity is needed, administrators need only to add another node to the storage or hyperconvergence cluster.

With great interest in these technologies, we sought to understand what businesses think of each. To that end, we surveyed more than 1,200 IT professionals and business decision makers to get their thoughts around these technologies and how adopters are using them.

This chapter provides the main highlights around people's existing data center and storage environments with a focus on software defined storage and hyperconverged infrastructure. If you'd like to read the full results and learn even more about how your peers feel about these technologies, download the full report from www.atlantiscomputing.com/survey2016.

Technology Domain Knowledge

With that in mind, we begin our analysis with a look at how the IT pros that responded to our survey view their own knowledge of various data center elements. **Figure 9-1** shows the breakdown of survey results. As is very obvious, the only area in which respondents believe that they have expert level knowledge is server virtualization, with 55% responding as such. For two primary emerging technologies – software defined storage and hyperconverged infrastructure – 12% and 18%, respectively, of respondents feel that they have little to no knowledge of the subject matter. Only 18% of respondents feel that they have expert-level mastery of each these topics. Given the relative age of these technologies when compared to other data center technologies – server virtualization, datacenter networking, and enterprise storage – it's not that surprising that knowledge level is quite a bit lower. Over time, we expect to see people's comfort level with software defined storage and hyperconverged infrastructure approach that of enterprise storage,

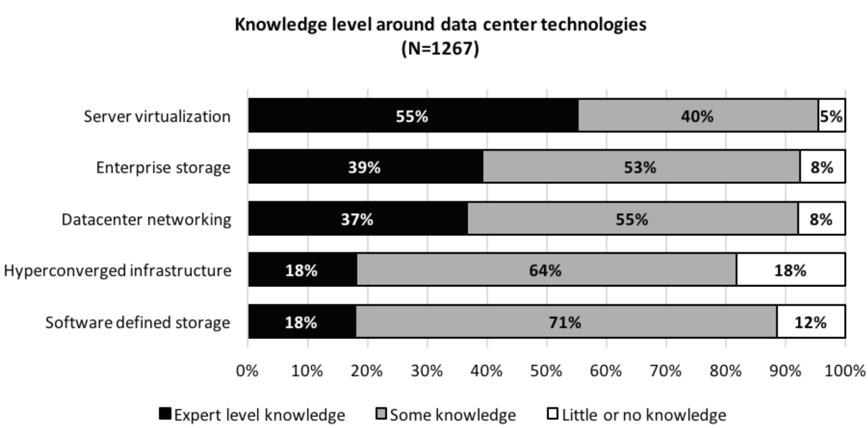


Figure 9-1: Knowledge levels of various data center technologies

which 39% of respondents have mastered. You will notice that, overall, people are more comfortable with software defined storage over hyperconverged infrastructure. 12% say that they have no knowledge of software defined storage while 18% say the same about hyperconverged infrastructure. This is likely due to the fact that many software defined storage systems more closely resemble traditional storage arrays whereas hyperconverged infrastructure is quite different.

Virtualization Penetration

Particularly with hyperconverged infrastructure, virtualization penetration is a key indicator for just how much of the existing environment can be migrated. Hyperconverged infrastructure deployments require that applications run virtualized. With that in mind, gaining an understanding for a respondent's level of virtualization is important to learn just how successful that deployment might be. We learned from respondents that most are at least 71% virtualized on the server front, but that desktop virtualization is truly still in its infancy or, at the very least, not of interest to many organizations. Only 19% of respondents are more than one-half virtualized on the desktop.

For those considering software defined storage rather than hyperconverged infrastructure, virtualization levels aren't really all that important except for the fact that virtualization is just another workload type to support.

In **Figure 9-2**, you can see the virtualization penetration rate for those that have deployed either software defined storage or hyperconverged infrastructure. The results aren't radically different, but you can see that 75% are at least half virtualized. The most interesting item here really revolves around the desktop. In the total population, a full 23% have done no VDI. For those that have deployed either software

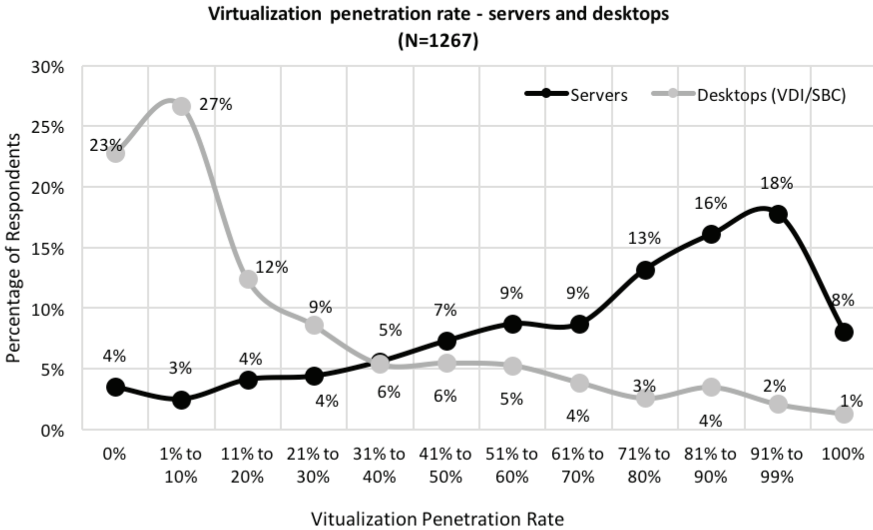


Figure 9-2: Virtualization penetration for servers and desktops

defined storage or hyperconverged infrastructure, only 10% have not deployed VDI. This suggest that virtual desktops are of more interest to SDS/HCI adopters. **Figure 9-3** shows this correlation.

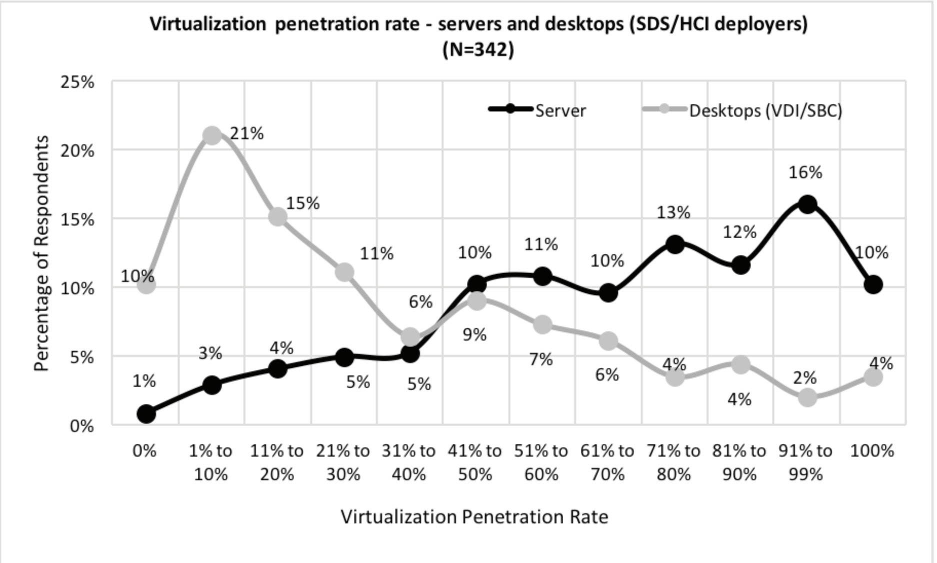


Figure 9-3: Virtualization penetration for servers and desktops – SDS/HCI deployers

Hypervisor Usage

Given the magnitude of virtualized applications – people are virtualizing more and bigger workloads all the time – hypervisor choice is a critical issue. Not every hyperconverged infrastructure solution is able to support every hypervisor available on the market. It's with that in mind that it comes as no surprise that VMware vSphere remains the dominant choice in the hypervisor market (see **Figure 9-4**). It's also no surprise to see that, over the next 24 to 36 months, many vSphere administrators intend to migrate to the latest version of VMware's hypervisor. Hyper-V will be the likely recipient for much of vSphere's loss. XenServer 6 looks to hold pretty steady as well. However, for those on XenServer 5, it looks like they will abandon the platform for other options.

We were surprised to see that KVM did not increase in perceived future market share. In fact, based on our results, KVM's share of the market will actually decrease a little. There are a number of hyperconverged infrastructure solutions on the market that use KVM as their core. With that in mind, we believe that, rather than a decrease, we will probably see KVM adoption *increase* over the next few years. Here's why: the hypervisor layer has achieved commodity status. For many, the actual hypervisor in use really doesn't matter as long as the solution meets all needs. With the KVM-based hyperconverged infrastructure options on the market, users may not focus as much on which hypervisor they're actually running. When we ask them to focus on the hypervisor, KVM doesn't stand out, but in practice, many may not really care, especially in smaller organizations.

We were not surprised to see Docker more than doubling in adoption in the next few years. Container technology is getting more attention and, much as was the case in the early days of virtualization, we expect to see container adoption start to become more interesting to people as they learn more about the technology and as it expands to support more workload types.

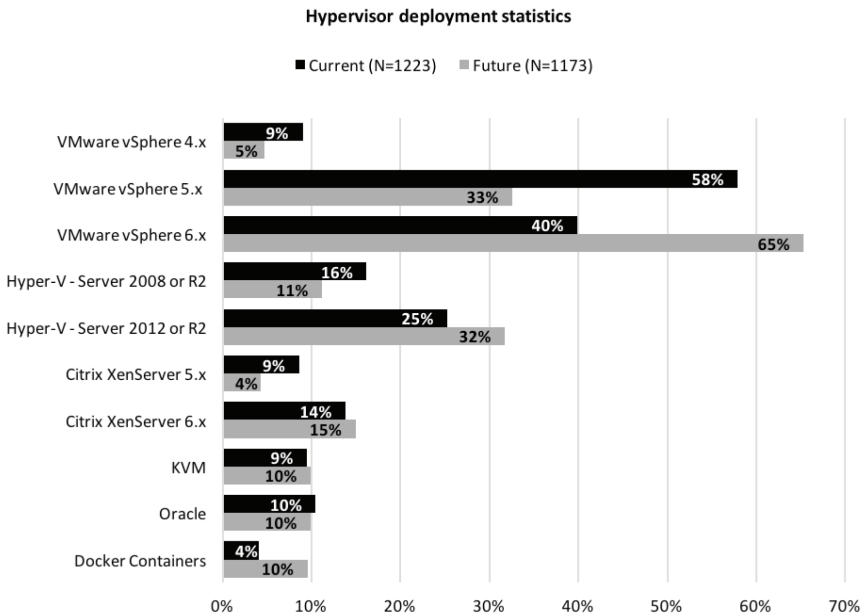


Figure 9-4: Hypervisor in use in respondent organizations

What is surprising is what happens when the hypervisors are grouped into their respective vendor categories and then analyzed. As you can see in **Figure 9-5**, VMware will maintain its market share. This may seem insignificant, but when considering the market as a whole, there is a story there, especially as the virtualization market is likely to continue to grow overall. What that means is that VMware is likely to simply maintain share in a growing market while those that are abandoning other platforms – such as Citrix – are more likely to jump to to Hyper-V rather than VMware.

Those companies providing what are considered "next generation" options – such as Docker – will also rise significantly in popularity in the next few years. For today – and likely for your next replacement cycle – VMware remains the clear leader in workload support, but over time, as more hypervisor and container options grow in usage, expect to see more hyperconverged solutions that provide comprehensive support for these products. While most people don't care whether or not a solution will support multiple hypervisors, they do care whether or not a solution supports the hypervisor or technology in use in *their* organization.

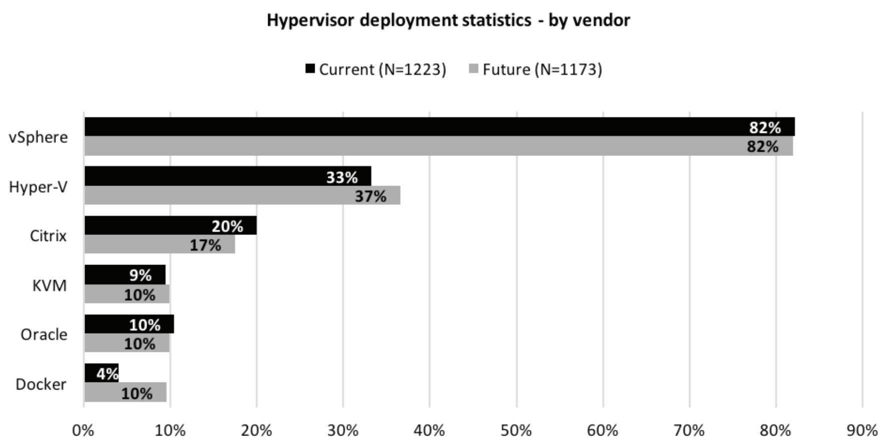


Figure 9-5: Current and future hypervisor/container breakdown by product

Now, let's look at the same information, but this time just for those that have already adopted software defined storage (see **Figure 9-6**). The information here suggests those that deploying software defined storage will do so at VMware's expense, with that company dropping from 81% of market share among SDS adopters to 76%. Likewise, Citrix will drop from 30% share to 24% and Oracle will lose 4% of its share. The gainers will be Microsoft, KVM, and Docker. Microsoft is poised to gain 6% share among SDS adopters, while KVM will see 3%, and Docker a large 7% increase, almost doubling its market penetration.

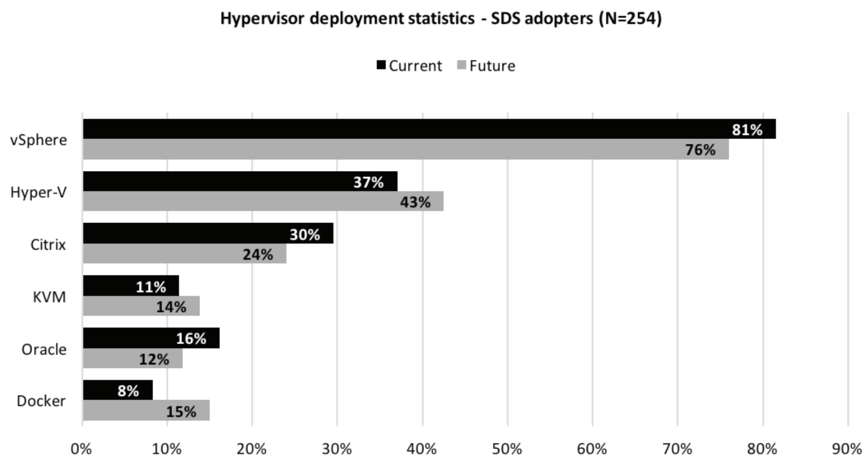


Figure 9-6: Current and future hypervisor/container breakdown by product – SDS adopters

Among hyperconverged infrastructure adopters, the trends are similar, but with a somewhat different magnitude. See **Figure 9-7** for the breakdown of this. Here, VMware's market drops from 85% to 77%, a full 8% drop, which is substantial. Microsoft's Hyper-V starts today at 42% and is expected to jump to 47% among our HCI adopter population. Citrix only loses a single point of their adopter base,

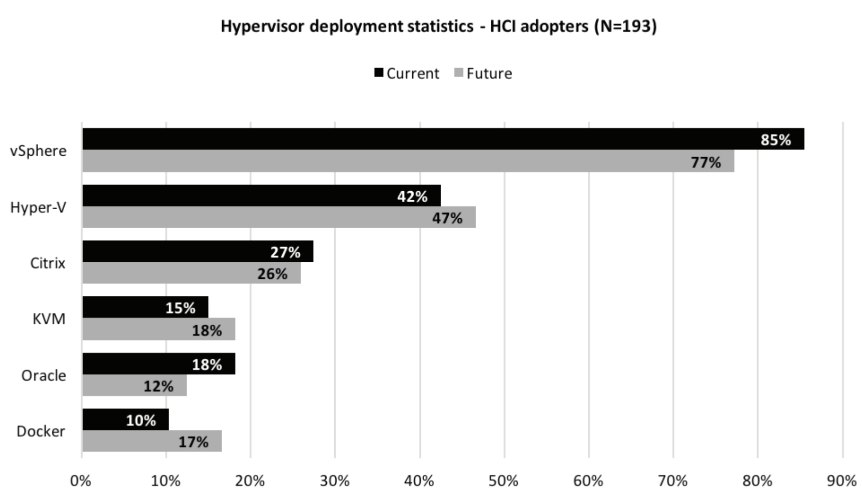


Figure 9-7: Current and future hypervisor/container breakdown by product – HCI adopters

and KVM jumps a full 3%, to 18%. We do expect to see an increase in KVM adoption among hyperconverged infrastructure users as the KVM-based HCI options continue to penetrate the market. Among HCI users, Oracle usage is poised to drop 6%, which is interesting since Oracle has their own converged infrastructure solution. And, again, Docker looks to gain significant followers as that product continues to improve.

Storage Capacity

Being able to support workloads means having sufficient storage capacity in your organization across both your primary location as well as any remote or secondary locations. Both hyperconverged infrastructure and software defined storage solutions have the capability to support both very small as well as very large deployment scenarios and either one can support centralized or distributed storage needs. As you can see in **Figure 9-8**, storage capacity varies widely and there are

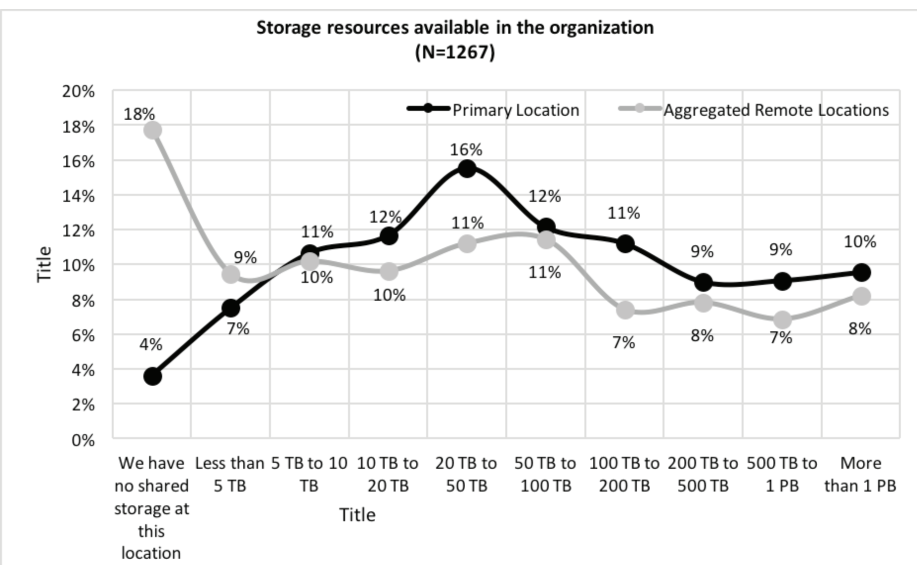


Figure 9-8: Storage capacity in primary and across remote locations

substantial storage resources housed at remote locations. From this chart, you can see that about 16% of respondents are running 20TB to 50TB of storage at their primary location. The most surprising piece of information here is just how much storage is present across remote and distributed sites. Only 18% of respondents indicate that they have no storage outside the headquarters location.

It probably comes as no big surprise to learn that overall primary location capacity changes with company size. In **Figure 9-9**, you can see that smaller organizations tend to have less overall storage while large companies tend to have much more. While this is common knowledge, our data absolutely reinforces it.

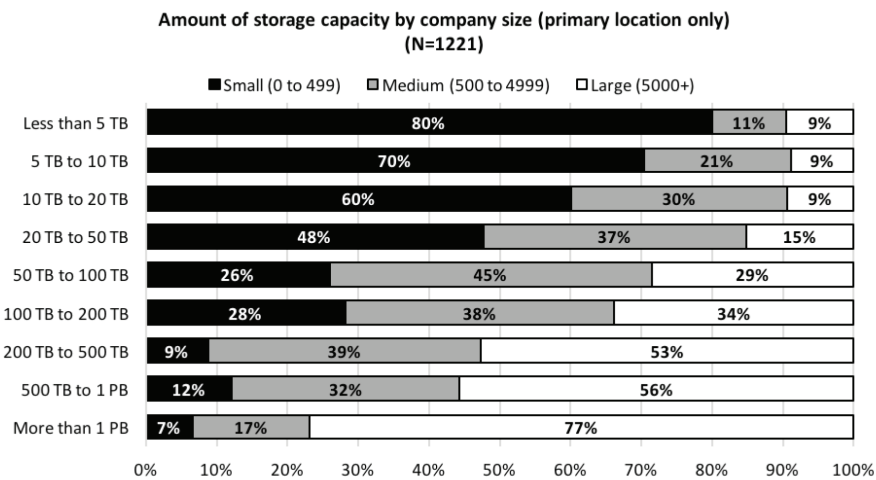


Figure 9-9: Storage capacity by company size (primary location only)

When breaking the data down by our four primary verticals, it's really easy to see that the 20 TB to 50 TB storage range is the sweet spot for our overall respondent group (see **Figure 9-10**). It's also easy to see that different verticals have very different average storage needs. For example, only 4% of those in the education vertical are running 200 TB to 500 TB of storage whereas 21% from finance have that level of capacity. Given the data-driven nature of financial companies, this comes as no big surprise, but is interesting nonetheless. By comparing the individual bar sizes in **Figure 9-10**, you can begin to see where each vertical ranks with regard to storage capacity. Here are the major ranges for each vertical (again, this is storage capacity at the primary location only):

- Education: 20 TB to 50 TB
- Finance: 200 TB to 500 TB
- Government: 20 TB to 50 TB
- Healthcare: 50 TB to 100 TB

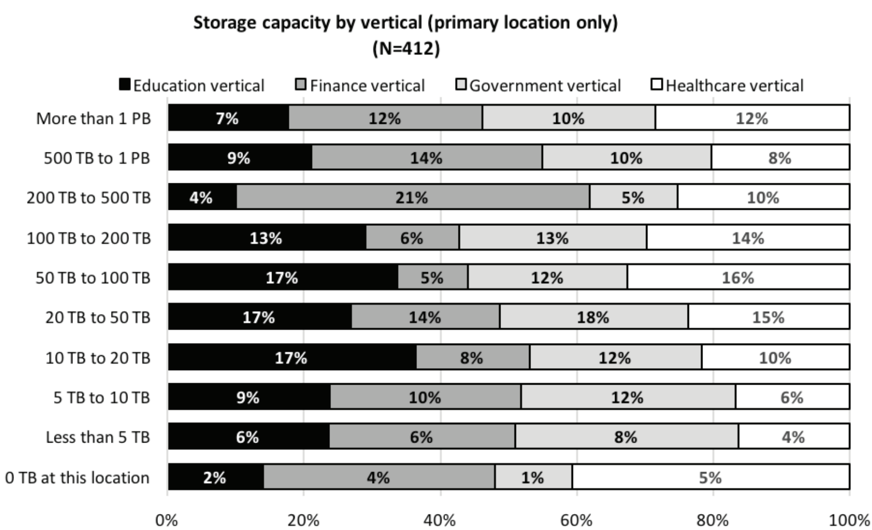


Figure 9-10: Storage capacity by vertical (primary location only)

Now, let's look at the storage capacity breakdown across the aggregate of all respondent remote sites. **Figure 9-11** excludes storage at the primary location. The data here is slightly more mixed than we see with capacity figures at the primary location, with a large number of respondents having no remote storage capacity. However, for those that do have storage resources in remote sites, the 20 TB to 50 TB range is once again the leader of the pack, but we also see a jump in the number of overall organizations that have more than 1 PB spread across remote storage systems. As mentioned earlier, this situation reinforces the need for hyperconverged infrastructure and software defined storage solutions that focus on ROBO use cases. Here are the major ranges for each vertical (this time, this is storage capacity at remote sites):

- Education: 20 TB to 50 TB
- Finance: 20 TB to 50 TB and More than 1 PB
- Government: 5 TB to 10 TB
- Healthcare: 50 TB to 100 TB

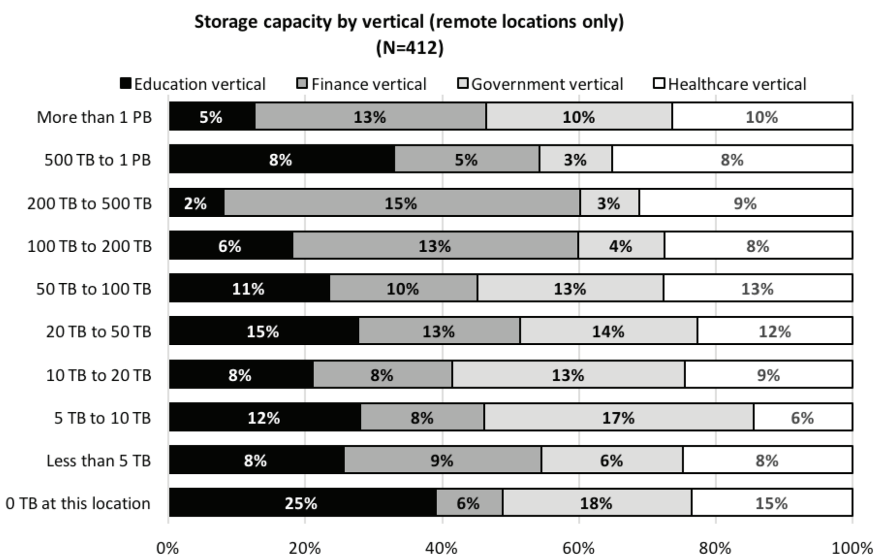


Figure 9-11: Storage capacity by vertical (remote locations only)

With ROBO being a key use case for hyperconverged infrastructure, we wanted to look at overall capacity at remote locations for organizations that deployed one of these technologies. There were a total of 342 respondents that have undertaken such deployments. In **Figure 9-12**, you can see the remote storage capacity breakdown for each technology. Earlier, we learned that storage capacity and company size are linked to one another; bigger companies have more storage. From **Figure 9-12**, it's clear that some very large companies have deployed both software defined storage and hyperconverged infrastructure since the choice "More than 1 PB" garnered the greatest number of respondents.

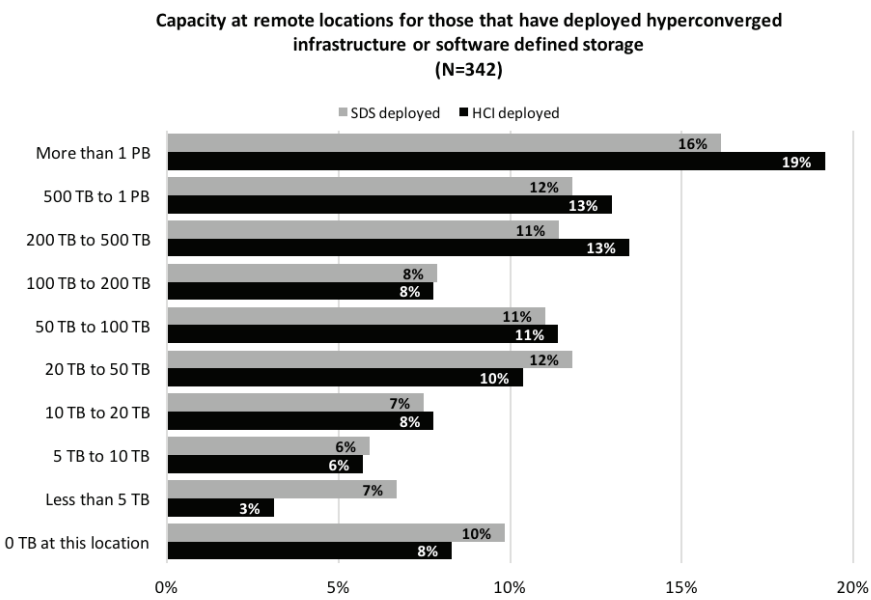


Figure 9-12: Storage capacity for SDS/HCI adopters (remote locations only)

Data Growth

Perhaps one of the most serious technology challenges facing organizations is keeping up with the sheer growth of data. **Figure 9-13** shows you that most organizations are seeing a 10% to 30% annual data growth rate. However, a number of companies see much higher rates, even 50% or 100%. For these respondents, finding a storage solution that can scale easily and inexpensively is absolutely critical to maintaining reasonable level of expense and application availability.

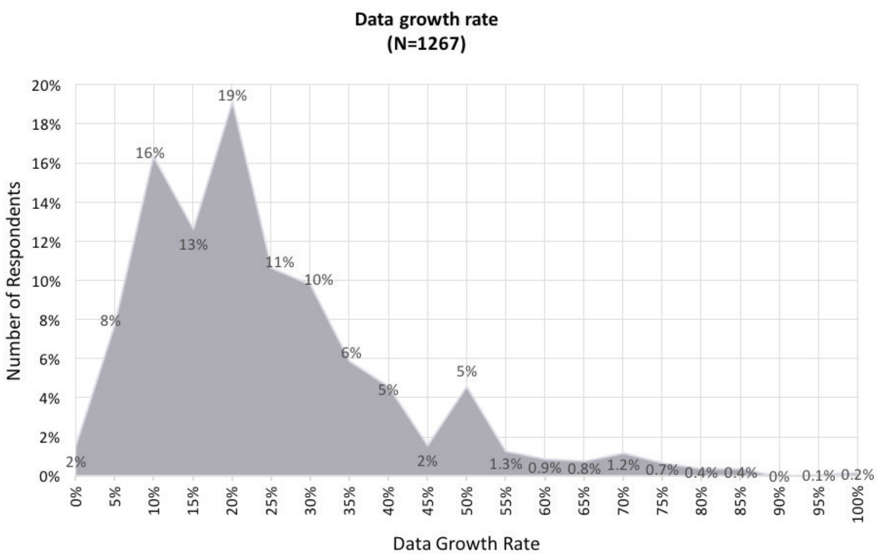
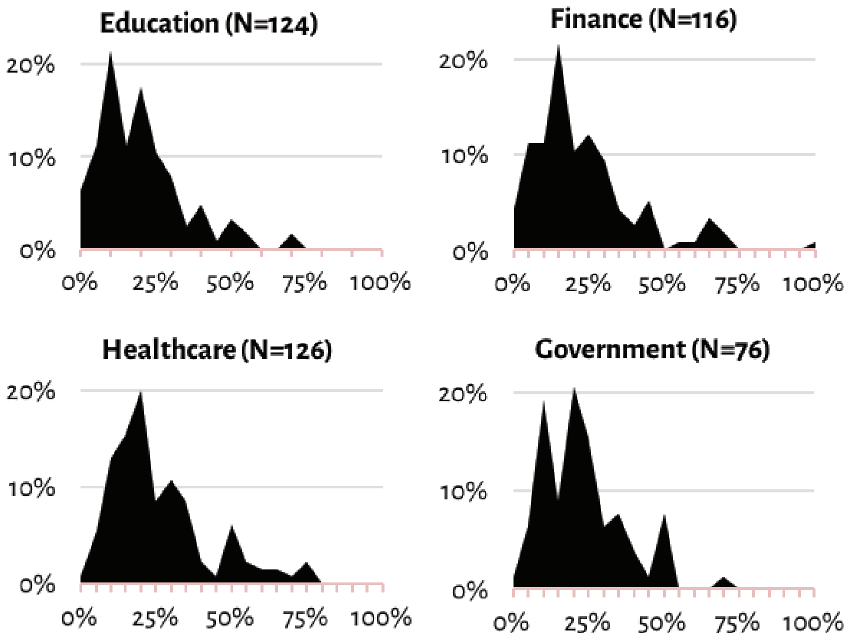
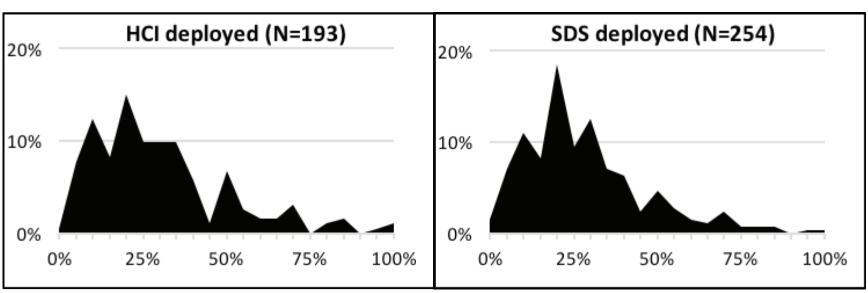


Figure 9-13: Respondent annual storage capacity growth rate

In the four charts below, we can get a look at the data growth patterns for the four primary verticals under scrutiny for our survey. As you can tell, in general, the data growth patterns are all pretty similar; most organizations, regardless of vertical, fall primary into the 10% to 30% data growth range and have some kind of peak around the 50% data growth rate. Here, though, finance is something of an outlier, with its "secondary peak" coming in at around 45% with a smaller third peak coming at 65%.



It’s a similar story when considering this information just for those that have deployed hyperconverged infrastructure or software defined storage. However, while the peaks are in similar places – in the 10% to 30% data growth range, fewer software defined storage users report these levels of growth.



Storage Performance

While storage capacity is absolutely critical to consider, storage performance is also a key success factor for workloads. Over the years, storage performance challenges have become severe, leading to the rise of flash-based storage solutions and a renaissance in the overall storage market. Software defined storage and hyperconverged infrastructure are two rising markets that have emerged as a part of this renaissance. But, just how well are these newer entries in the storage market meeting performance goals?

As it turns out, pretty well. **Figure 9-14** shows that only 16% of respondents have solutions that are slower than their disk-based storage systems. A full 50% say that their solutions are faster than their disk-based systems, with 14% saying that it's as fast as an all flash system. Overall, from a performance perspective, these newer storage options are holding their own.

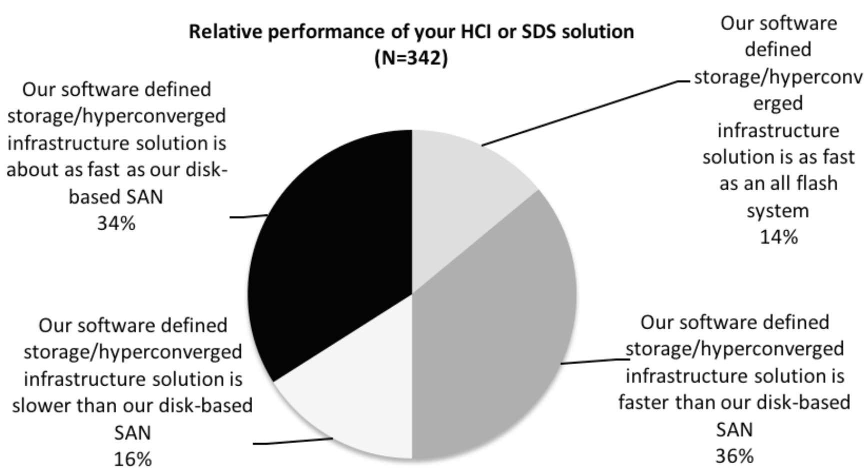


Figure 9-14: Performance of SDS/HCI solutions (adopters only)

Current Storage Systems

With 75% of respondents still running such systems, disk-based storage still rules the data center, although it is joined by hybrid storage (55%), all-flash storage (21%), software defined storage (21%), and hyperconverged infrastructure (16%) solutions. (see **Figure 9-15**)

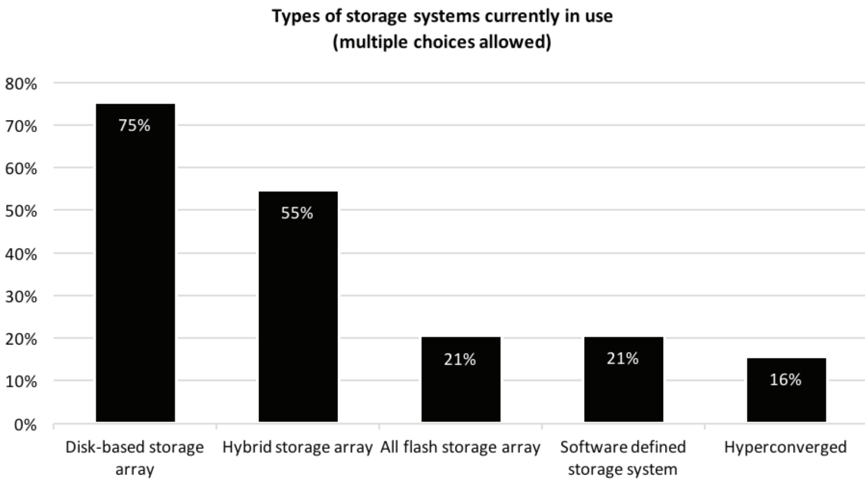


Figure 9-15: Types of storage systems currently in use

When we asked respondents to tell us what their future (future = 2-to-3 years out) plans are regarding storage, the responses paint a bleak future for disk-based storage. A full 19% of respondents – almost 1 in 5 – say that they will fully decommission their disk-based storage systems over the next two to three years. **Figure 9-16** shows that the primary gainers in the same timeframe will be all flash arrays and hybrid storage arrays, but 35% also say that they will expand their use of software defined storage and hyperconverged infrastructure.

None of this comes as a major surprise. Flash storage has been plummeting in price for quite some time and is expected to hit price parity with disk within the next few years. Once raw price parity is achieved, expect to see spinning disk quickly fall off in terms of usage.

Flash simply carries with it far too much performance potential when compared to disk.

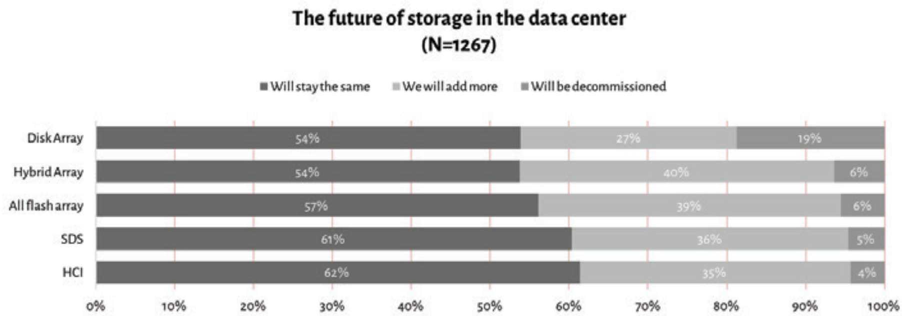


Figure 9-16: Respondent future storage plans

While software defined storage might be considered architecturally similar to traditional storage in that storage remains isolated from compute, it is hyperconverged infrastructure – in which compute and storage are combined – that is of more interest to those considering these technologies. 27% of respondents are most likely to adopt the former while 33% plan to adopt the latter. However, those considering more traditional approaches still outweigh those looking at emerging storage approaches. 39% are considering all flash arrays while 45% are considering traditional systems, which include hybrid storage arrays (see **Figure 9-17**).

50% of respondents, though, say that they intend to deploy cloud-based storage services. For the foreseeable future, we expect that most deployments will be of the hybrid variety in which organizations combine cloud-based storage with local storage. Over time, as more companies seek to further simplify their data center environments, many are turning to the public cloud, which eases deployment. However, because of security concerns, locality concerns, and even cost challenges, many companies are discovering that keeping things private makes more sense. We'll see how this plays out in the coming years, but for now, cloud is still a big plan for many.

This information would seemingly contradict what you just learned – that 19% of people currently using disk-based storage arrays intend to decommission them. However, bear in mind that, for those that intend to "remain the same" on disk-based storage, that means that they will ultimately need to replace them, which we believe is the reason that we see strong results for Traditional SAN/NAS devices in **Figure 9-17**. Also note that the response categories are slightly different, since we add cloud storage as an adoption option to this question.

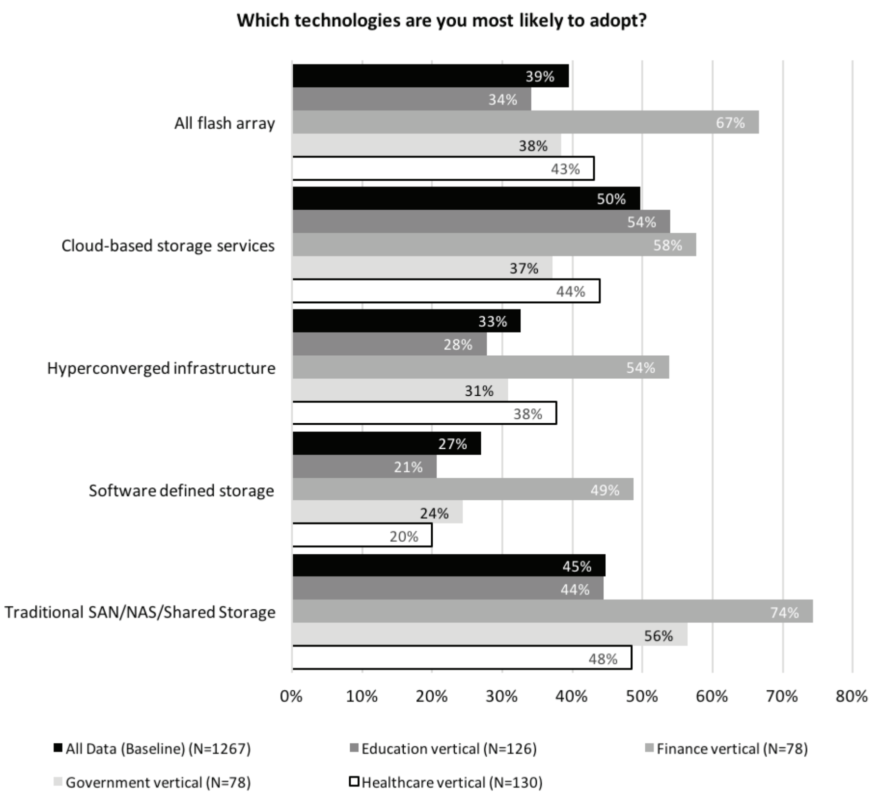


Figure 9-17: Storage adoption intent by vertical

You can also see that we have provided a breakdown of adoption intent by vertical. It's clear that those in finance have major plans when it comes to storage in the coming years, with 67% intending to deploy all-flash arrays. Finance also intends to add a lot of software defined storage (49%) and hyperconverged infrastructure (54%). We were somewhat surprised to see to relatively low software defined storage/ hyperconverged infrastructure uptake intent in the education and government sectors, however. Especially in education, technology is often seen as a cost center, with the benefits of these emerging technologies helping to drive down costs.

The Form Factor Debate

Hyperconverged infrastructure and software defined storage solutions are available as either just software deployments or as hardware appliances that include the software. There are different solutions available depending on customer needs. Software-only solutions provide more hardware flexibility since the customer can specifically size the individual nodes. Preconfigured hardware appliances offer a bit less individual

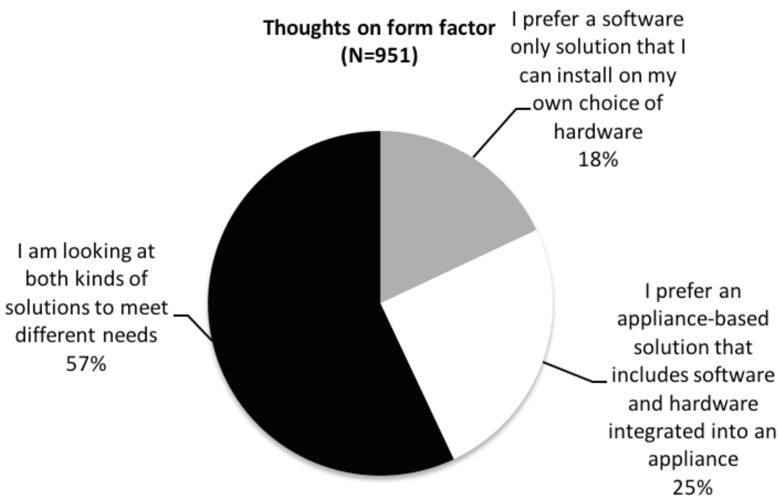


Figure 9-18: Respondent thoughts on form factor

resource flexibility, but do offer a simplified deployment experience. As you can see in **Figure 9-18**, for those that have an opinion, most prefer appliance-based solutions, but not by a wide margin. 57% of respondents are keeping their options open and considering both kinds of solutions.

Business Workload Support

Storage and data center infrastructure is deployed to support business workloads. We asked respondents to tell us what they want to accomplish with software defined storage and hyperconverged infrastructure. **Figure 9-19** provides you with a look at the top three use cases identified by each segment that we analyzed for this chapter. As becomes very apparent, Test and Development is a clear top use case for those that have deployed or have an interest in software defined storage while server

Data Segment	Type	1st Use Case	2nd Use Case	3rd Use Case
All Respondents	SDS	Test & Dev (43%)	File & Print (41%)	SV (40%)
	HCI	SV (45%)	VDI (39%)	Database (39%)
Education Vertical	SDS	Test & Dev (44%)	Private Cloud (37%)	File & Print (36%)
	HCI	SV (45%)	Database (41%)	VDI (40%)
Finance Vertical	SDS	Test & Dev (62%)	Database (59%)	Big Data (55%)
	HCI	SV (68%)	Test & Dev (65%)	DCC (64%)
Government Vertical	SDS	Database (46%)	SV (45%)	Test & Dev (45%)
	HCI	SV (45%)	Test & Dev (41%)	VDI (40%)
Healthcare Vertical	SDS	Test & Dev (42%)	Private Cloud (40%)	File & Print (39%)
	HCI	SV (47%)	Database (45%)	DCC (42%)
Those that have deployed SDS	SDS	File & Print (55%)	SV (55%)	Database (55%)
	HCI	Test & Dev (47%)	SV (46%)	DCC (44%)
Those that have deployed HCI	SDS	Test & Dev (45%)	Analytics (42%)	DCC (39%)
	HCI	SV (64%)	Private Cloud (56%)	Database (55%)

SV = Server Virtualization, VDI = Virtual Desktop Infrastructure, DCC = Datacenter Consolidation

Figure 9-19: Top use cases broken down by analysis segment virtualization is, in general, a top choice for those that have deployed or have an interest in hyperconverged infrastructure. Given the highly versatile nature of software defined storage, it's not a surprise that it has use for more than virtualization tasks. Hyperconvergence, on the other hand, assumes that virtualization is the standard, and virtualized server workloads are a must on these platforms, hence respondent interest in server virtualization for hyperconvergence. Other top use cases include database workloads, VDI, private cloud, file and print and data center consolidation.

Remote Office and Branch Office Support

One of the key use cases that has emerged for both software defined storage and hyperconverged infrastructure is supporting remote office and branch office (ROBO) environments. These technologies are very well-suited to ROBO needs and are emerging as a leading way to support ROBO environments. **Figure 9-20** indicates that 9% of

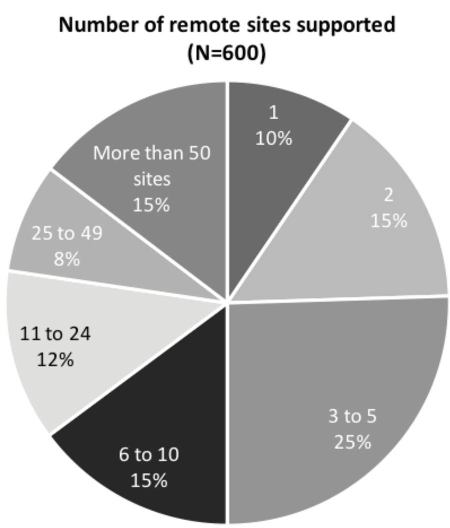


Figure 9-20: Number of remote sites supported

respondents have just one remote site. 15% of respondents have more than 50 sites to support.

Flash Storage Deployment

In recent years, flash storage has taken the market by storm and is poised to eventually mostly supplant disk as prices for flash continue to decrease. As of today, though, just 1% of respondent's data centers are all flash. Over 60% of respondent's data centers are less than one-tenth flash based, with 21% of respondents saying that they do not yet have *any* flash deployed. Just 6% of respondent's data centers are over one-half flash. **Figure 9-21** shows this flash penetration. For vendors that are able to provide affordable flash solutions, this is actually a good situation as there is significant upside in the flash market.

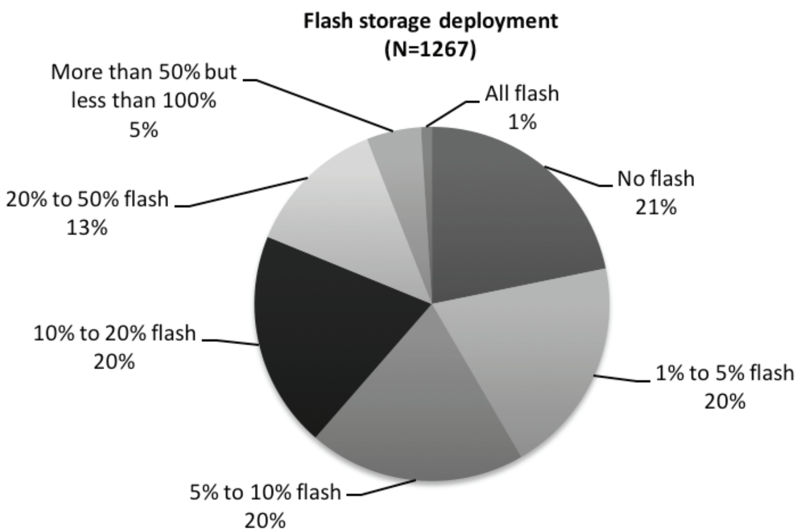


Figure 9-21: Flash storage deployment statistics

Software Defined Storage and Hyperconverged Infrastructure Deployment Intent

As mentioned, software defined storage and hyperconverged infrastructure solutions are somewhat new entrants into the storage market and are, for many, still being proven. As they continue to prove their capabilities, more organizations will consider them for implementation. According to our survey respondents, 15% are either very likely or definitely planning to deploy such services over the next two to three years. 53% say that it's a possibility, while 32% say that it's either not likely or there is no chance of deployment. In general, this is good news for vendors selling these solutions and is also a good indicator of interest in this technology for those considering the technology. **Figure 9-22** depicts respondents' deployment intentions.

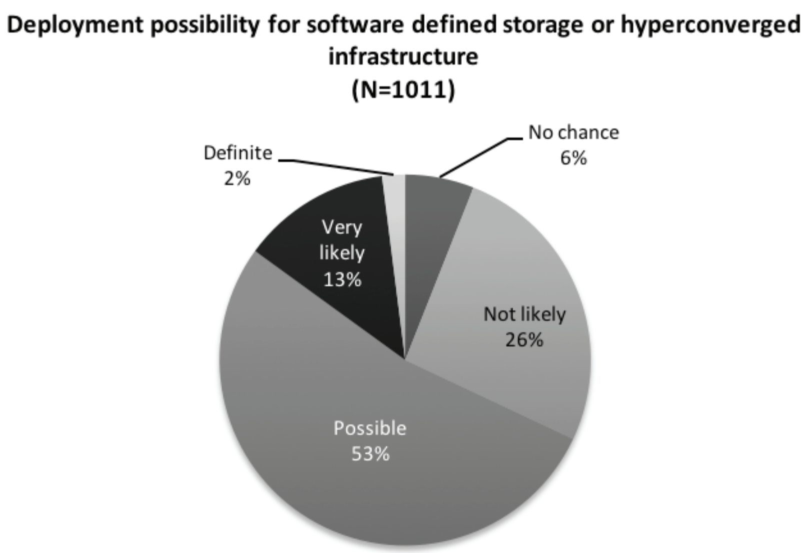


Figure 9-22: SDS/HCI deployment potential

Software Defined Storage and Hyperconverged Infrastructure Deployment Experience

Companies don't deploy technology for technology's sake. They deploy it in pursuit of a goal of some kind. Most often, new technologies are deployed because they either cost less or are more efficient in some way. This fact certainly holds true for software defined storage and hyperconverged infrastructure solutions. Given people's concerns around traditional storage costs and complexity, it would make sense that those that have adopted newer methodologies would do so to offset cost and complexity.

We asked respondents to tell us about their experiences with software defined storage and hyperconverged infrastructure as it relates to a number of different areas. **Figure 9-23** provides a look at the results. In almost every area, people have had a better experience – or at least a comparable one – with software defined storage and hyperconverged infrastructure to their experience with whatever they had before. The only exception is with personnel cost, which have increased for those that have deployed software defined storage.

In terms of systems performance, data center space, and power and cooling costs, there have been tremendous gains for implementers of software defined storage and hyperconvergence. On the performance front, it's more than likely that the gains have come from the fact that the previous storage was more disk-focused while the new solution is either hybrid or all flash. Data center space is much improved in hyperconverged infrastructure scenarios since compute and storage are integrated together into server nodes. Further, less equipment in the data center overall translates to lower power and cooling costs.

You will notice that direct costs – acquisition and support costs – stay relatively constant. About the same number of respondents experienced higher costs as lower costs. While still providing a lot of value, software defined storage and hyperconverged infrastructure solutions have not yet helped companies reduce initial expenditures on infrastructure, but *have* helped when considering the total cost of ownership. This leaves a major opportunity for companies in the emerging storage space that can reduce both acquisition cost and TCO.

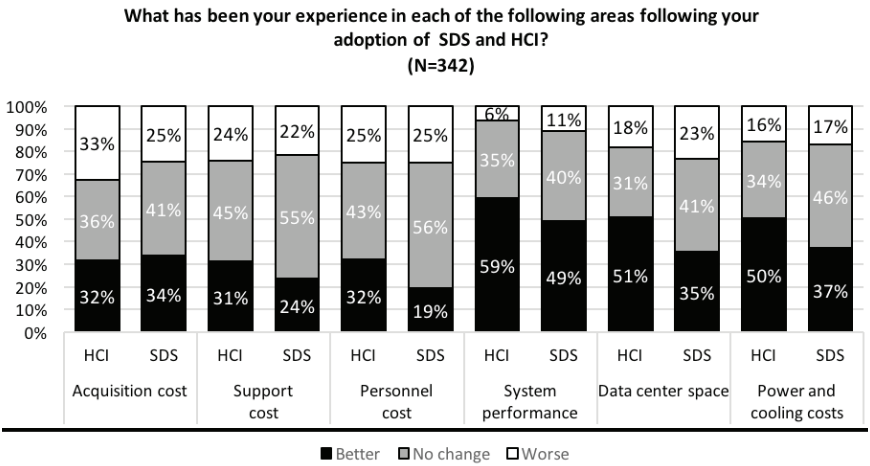


Figure 9-23: SDS and HCI deployment experiences

Hyperconverged Infrastructure or Software Defined Storage Deployment Timeframe

Deployment of software defined storage and hyperconverged infrastructure is happening in waves and is more than likely taking place based on existing hardware replacement cycles. **Figure 9-24** shows that over the next year or so, 17% of respondents say that they will undertake deployments. Over the next two years, that number jumps to a total of 62%. 27% of respondents say that they are uncertain as to their deployment plans, which could mean that they are still not sure whether they will definitely deploy or they truly don't know when they might plan a deployment.

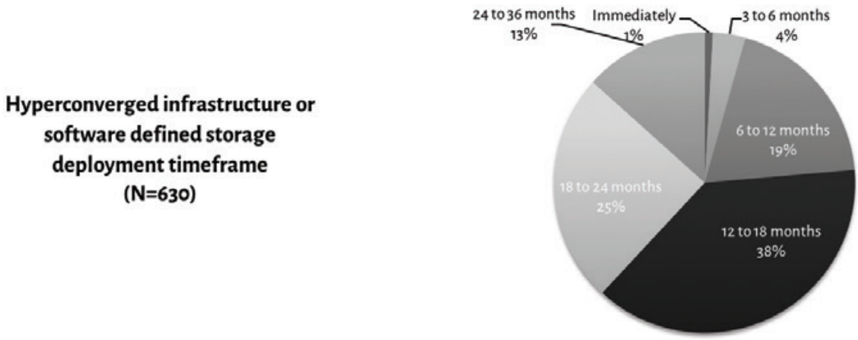


Figure 9-24: SDS/HCI deployment timeframe

Software Defined Storage and Hyperconverged Infrastructure Decision Criteria

Most people would likely assume that cost would be the primary decision point around any new technology, but, interestingly, that's not the case. Cost is actually tied for second in terms of decision criteria. Overall performance of a solution is the key issue for many people (72%), while cost is tied with availability as the second most important need (68%). **Figure 9-25** illustrates the various decision criteria.

Particularly noteworthy here is that respondents rated performance and things like cost as top criteria, but did not choose as top criteria the method by which those benefits are achieved (i.e. all flash configurations and server brands). The same holds true for high availability and stretched clustering abilities. Further, features such as data reduction, which can significant lower costs, were not rated as highly as direct cost savings. Of course, often when people think of "cost" in data center solutions, they often equate that to "price." With that thinking, it's not a surprise to see cost and data reduction considered separately. For many, features like data reduction don't change the price, but they do decrease the total cost of ownership (TCO), which is not something that is always considered when purchasing a new solution.

We mentioned that a lot of people – close to one-third – indicated that server brand is not important. In recent years, commoditization at the server level has led people to understand that the brand of the underlying hardware in many cases isn't all that significant. While there may still be compelling reasons for some to adopt server hardware solutions that may bring ancillary benefits, for many, they don't care about the brand as long as the hardware can adequately do its job.

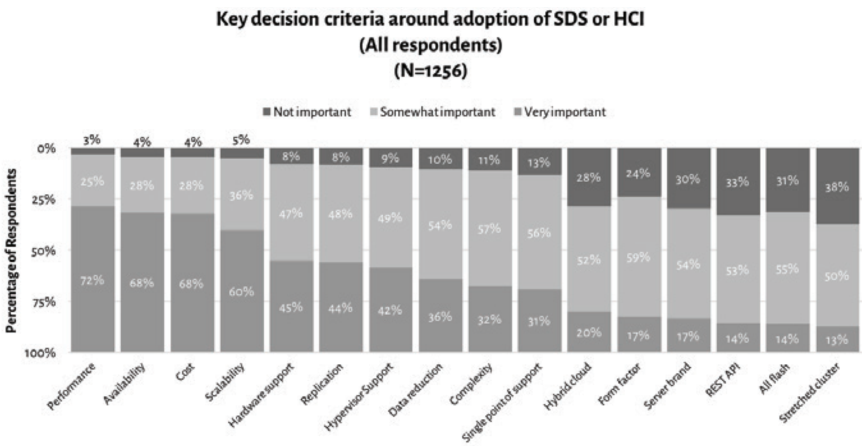


Figure 9-25: SDS and HCI adoption decision criteria

Chapter Recap

This results provided here are just a glimpse of the overall analysis provided in the broader research report. To view the full report, visit www.atlantiscomputing.com/survey2016.

10

IT Transformation

The world's largest automobile manufacturer, the Toyota Motor Company, is well known for producing an extraordinary amount of cars. Their high production capacity is thanks to the "Toyota Production System" which is studied worldwide and is regarded as an engineering marvel. In 1924, Toyota's founder Sakichi Toyoda invented the automatic loom which spun thread and weaved cloth automatically. The machine ran nonstop, unless it detected a problem, in which case it stopped to allow the operator to correct the issue. Because it detected the issue, defective products were not created. Of course, as one could guess, this methodology evolved to create the Toyota auto manufacturing line that is known and respected today.

As IT evolves in the coming years, the industry will be characterized by a greater focus on removing the human element from workflows. Just as Toyota is able to achieve great production value by utilizing automation and orchestration to enable their just-in-time manufacturing model, data centers are looking to automation to decrease their time to value and increase the scope of how much infrastructure a team member can be responsible for.

On the Toyota production line, one operator can be responsible for many machines because his or her job is to monitor and provide the "human touch," not to sit there and operate a single machine. In the same way, less engineers will be required to operate a world-class data center if many of the processes and operations are left to computers and the humans just provide oversight.

The Challenge

The number of disparate systems in the data center has grown to an unmanageable level. There's a product (or a few products) for nearly every problem. In many environments, the burden of the number of moving pieces causes serious inefficiency on the part of the IT staff.

The concept of *orchestration* is "the methodical linking of related systems in such a way that a higher level construct can be the point of management, and the orchestration system controls the linked systems."

A real-world example of orchestration would be provisioning servers without administrator intervention. The orchestration system might kick off a virtual machine deployment from a template, customize the guest OS with an IP address pulled from the IPAM system, register the machine with the monitoring system, update the virtual machine via the patch management system, and resolve the ticket, notifying the requestor that the build is complete and the server is available at a given address.

This is a simple example, but imagine how much time the staff saves managing the data center each year by not needing to dedicate time to this task. All they need to do at this point is monitor the orchestration system to be sure tasks are completed as expected, and fix it if something is not working correctly.

The Solutions

Due to the way that cloud-based IT infrastructure tends to be more suited for ephemeral workloads, automation and orchestration will become increasingly important. The amount of resource creation and destruction mandates this. Since the primary purpose of data center infrastructure is to serve the application that runs on it, a new culture is emerging that emphasizes the community and teamwork of the development staff and the operations staff as a single unit. These roles have been segregated in the past few years, but experience has shown that there's much more value in the teams working together. As such, the DevOps methodology will experience dramatic uptake in the coming years.

Let's take a look at each of these three concepts — Automation, Orchestration, and DevOps — in a bit more depth.

Automation & Orchestration Concepts

These two ideas were briefly covered in Chapter 6. Feel free to jump back to that section for a quick review before diving in deeper here. This section focuses on the future and potential long term impact of automation and orchestration as compared to the basic overview in Chapter 6.

Automation

Automation has always been attractive to the IT organization, because many tasks that IT administrators perform are iterations of the same task for different users, sites, applications, and so on. However, it wasn't until recently that this concept has started to become mandatory.

At a certain scale, it's no longer possible for humans to maintain the level of accuracy and speed needed to meet the requirements of the business. Driven by the cloud, automation of repetitive or broad modifications to the environment is becoming commonplace, and will become far more commonplace in the future.

Why Automate?

There are two main purposes for implementing IT automation: speed and accuracy. Both of these requirements ultimately line up to the needs of the business as it relates to provisioning time and downtime aversion.

The first purpose is speed. Computers can issue commands much faster than humans can manually enter them, thus even the most adept operator can be easily outdone by a machine. Therefore, as the scope of the change or implementation grows, so does the advantage to the machine over a human.

The simple example of changing a single switch port configuration on 300 network switches worldwide will show the value of automation. In a network with automation tools in place, an administrator might issue a command and the software makes the configuration changes in a matter of *seconds*. On the other hand, the administrator can log in to the switch, paste the commands to change the port, and save the configuration in 10 seconds each time, meaning that all 300 operations — back to back —would take about an *hour*.

The second purpose for IT automation is accuracy. Imagine that the IT administrator really could configure 300 switches at 10 seconds each, back to back. How accurate is that likely to be? Is it possible that a switch would be overlooked, or that a typo would be made in the configuration of one of them? These types of mistakes are actually quite likely; for 1 in 300 to have a mistake is easily conceivable. On the other hand, a computer can accurately make all the changes with no potential for typos or misconfigurations.

When even a slight misconfiguration can lead to downtime, this is a big deal. An IT organization would likely be interested in automation for either one of these purposes, but when both are at stake, automation is a no-brainer.

The Future of Automation

So where is automation in the data center headed? There are two parts to that question: where the software that enables automation is headed, and where the administrators who design and operate the data center are headed. The answer is that they're going to meet in the middle.

Automation software tends to be more friendly toward developers than operations staff. Infrastructure administrators that are used to GUIs and Command Line Interfaces may have some scripting background, but many of them aren't familiar with development-oriented ways of thinking. So asking the operations folks to provide JavaScript Object Notation (JSON) or YAML can be a bit intimidating. In this sense, the automation software will continue to get simpler and more intuitive so that staff doesn't necessarily need a development background to use the tool.

On the other hand, the data center is all about applications, and it's high time that infrastructure-focused staff spent a little more time focusing on development.

Operations staff will be encouraged over the next few years to hone their development chops and get more comfortable with code. By doing this, they'll not only become more efficient and effective, but have a better understanding of the needs and methodology of the development staff.

From an administrator's perspective, automation has more or less created what we know as the *cloud*. The characteristic self-service nature of cloud resources is provided by automation (and orchestration, which is discussed next).

In the data center over the next few years, users will be blessed with (and come to expect) self-service provisioning of resources in a way that streamlines their jobs and allows them to be more productive. This is a win for the internal customers as they don't have to wait on IT, and it's a win for IT because they won't be burdened by the request and can continue to focus on being proactive.

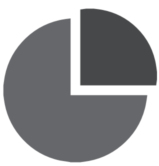
Justifying Automation

A key challenge with IT automation is justifying the front-end investment of time. It will be vital to every organization putting serious effort into automating IT processes that they quantify and measure the return on investment (ROI) that they're getting from automation projects. At the end of the day, automation is only valuable if the change it creates is directly reflected in the bottom line. Following an automation project, the business will need to be able to say that, "by implementing this automation project (in which we have invested 60 hours), the business saved $120,000 this year."

Orchestration

Orchestration can be seen as either an extension of or a complement to automation. The idea of orchestration is that a single platform would control many disparate systems to accomplish a defined business function.

Think of it this way: if the IT organization is a kitchen, automation is a kitchen appliance like a mixer or a blender. Orchestration then, is the chef, and the desired business process is the metaphorical tasty dinner! The chef could perform by hand the tasks that a mixer or blender accomplishes, but it would take substantially longer and be less consistent. The kitchen appliances accelerate the time to get the ingredients in the desired state. The chef is responsible for skillfully incorporating all the ingredients of the dish. A good chef will add the right ingredients in the right amounts, at the right time, and in the proper order. This is precisely the idea of orchestration in the data center.



From the Field: Survey Results

67% of survey respondents said that relative to SDS and HCI, the REST API was one of their key decision criteria. This clearly shows the importance current data center architects and CIOs are placing on orchestration and the ability for these systems to interact with other systems.

As an example, suppose that a production application needs to be upgraded. Automation would do the upgrading of code via a declarative statement such as, "the web servers should be running version 3.1.4," but orchestration would control the entire process. Orchestration might do something like this:

- Suspend monitoring of a subset of the application servers.
- Place them in maintenance mode on the load balancer so no traffic is directed to them.
- Instruct the configuration management tool to modify the code version of the application servers.

- Run tests to be sure the system is functional.

- Take the system out of maintenance mode on the load balancer.

- Resume monitoring of the system.

- Move on to do the same thing to the next subset of application servers until the whole farm is at the proper version.

Orchestration vs. Automation

Orchestration is a great value to the IT organization because of the efficiency and consistency it can offer. Although automation is incredibly helpful on its own, delivering end-to-end value without relying on a human operator is what really increases value and characterizes the data center of the future.

The end-user self-service model mentioned earlier in the automation section is really made possible by orchestration. A user-requested, self-service task is precisely the business process that an orchestration workflow would automate. In the case of cloud services, the user might say, "Give me a virtual machine with these unique-to-the-workload specifications, and place it on the same network segment as these other machines, and allow inbound access on these couple of ports." Orchestration would break up that outcome into subtasks and perform each one in the necessary order to achieve the goal. It would call bits of automation to do each one.

The Future of Orchestration

The future of orchestration is the same as the future of automation: manufacturers of orchestration tools will continue to strive to make them as simple and intuitive as possible without sacrificing power, and operations teams will learn to be more comfortable with tools that have traditionally been reserved for developers so that they can work more closely with those developers. In fact, many organizations are already adopting a development philosophy that more closely integrates operations staff and developers called *DevOps*.

DevOps

It's long overdue for the IT community as a whole to admit that the functionally segregated IT teams that we've built are a hindrance rather than a help. In many places, the operations staff sits together in one area, the QA team sits together in another area, and the developers sit or stand at their trendy motorized desk in a different area. When one team needs something from the other, they "throw it over the wall." One team hands off a task to another and is now at the mercy of the other team. The business process that is most obviously impacted by this team structure is the software development life cycle. As software moves from development to testing, and later to staging and production, the operations team is heavily involved in moving the code around and deploying it.

Software development teams have mostly transitioned to using workflows that utilize frameworks like Agile and Lean to accelerate and improve quality in the software development life cycle.

This is in contrast to the traditional waterfall development process where code is built and integrated sequentially.

Next-generation development frameworks encourage a constant re-evaluation of the project and short bursts of focus on specific goals. The end goal is to make development a more iterative process — a process that results in higher quality code that is closer to expectations and can evolve with the vision of the project.

Combining Development and Operations

It turns out that to iterate so quickly, the operations team must be heavily involved. This is where the idea of DevOps comes from.

DevOps is the idea of pulling the operations team into the development methodology to use their unique expertise to help the development process be more rapid. It also includes implementing tools and processes that leverage all the different skill sets available.

Because iteration is so frequent, DevOps culture has a heavy focus on automation. If administrators had to manually deploy new versions of code each time (as was likely done in the waterfall era) it would be an overwhelming task and would take away from the productivity of the team. Deployment is an example of a task that will likely be automated right away in a DevOps-focused organization.

The development process isn't the only way DevOps brings developers and operations staff together. The entire life cycle of a product is given the DevOps treatment; this means, for example, that operations and infrastructure staff are involved in the application design process as well. Their understanding of the infrastructure can lend a better perspective regarding server architecture, performance, high availability, and so on.

Benefits of DevOps

Developers making a decision regarding application architecture without understanding the underlying infrastructure is always a bad thing; and it is equally unfortunate when systems architects make infrastructure decisions without an understanding of the way an application functions. By placing the two groups together on one team, collaboration is dramatically enhanced, which leads to better decision making and project planning.

A byproduct of increased collaboration (and one of the main drivers for DevOps adoption) is the elimination of a phenomenon known as *technical debt*.

Technical debt is the unseen cost of:

- Cutting corners.
- Implementing temporary workarounds that never get permanently resolved.
- Not documenting code or infrastructure configurations.
- Generally, doing anything in a less-than-ideal manner.

The principle of technical debt says that these shortcuts only "borrow" time or efficiency. However, the price is paid elsewhere.

For example, as more of these shortcuts and workarounds show up in the code base or in the infrastructure, it becomes increasingly frustrating to work on. This means IT staff can become unmotivated at best and disgruntled and malicious at worst.

Technical debt is just like credit card debt; used responsibly, it can be a tool, but it is more frequently used irresponsibly, which leads to more cost than if it had not been used.

Placing the development, QA, operations, and other relevant team members in a collaborative environment leads to better planning and more accurate expectations. This, in turn, can and should directly lead to less technical debt over time. This alone is reason enough for an organization to move toward a full adoption of DevOps methodology.

Future IT organizations, once shifted to this methodology, will increase their rate of integration to an alarming pace. For an example of an organization that is already there, look no further than Amazon Web Services (AWS). They deploy multiple times per minute!

Compare this to organizations using the waterfall methodology who can deploy as rarely as every few weeks. When iterating at these dramatically different intervals, imagine the impacts of a bad deployment. On the AWS side, a very small section of the code base was impacted and the negative impact of the bad commit is likely very small. DevOps made this possible. On the legacy development side, a huge multi-week deployment going bad leaves a steaming crater in the data center and leaves everyone scrambling to put out fires. The advantage to the DevOps methodology in regard to this problem is pretty clear.

Industry Needs

As much as IT organizations are being forced to change by advancing technology, there are other variables. Business is the motivation that drives the change in the data center, but there are business requirements that are also challenging the status quo. Performance and capacity can only go so far in making a product a good fit for an organization. Beyond that, it needs to fit in to the existing ecosystem of the IT environment. It needs to be a sensible business decision from a budgetary perspective based on the cost of the product and the desired outcomes to the business.

Historically, a large amount of technology business has been conducted based more heavily on an executive's personal network than on the technical merits of a solution. Unfortunately, this will never go away completely, but the environment is changing and more transparency and justification is required these days. This change puts added pressure on manufacturers to show precisely how their solution helps the business and what specifications the solution will provide.

From a manufacturer's standpoint, the IT solutions market is growing more saturated all the time. Advances in technology will perpetually open up new markets, but even so, it's becoming nearly impossible to be a one-man show in the data center. As IT organizations adopt various products to meet their business needs, it's vital to the overall health of the IT ecosystem that the products from different vendors are interoperable.

With these needs in mind, there are a minimum of three objectives that the IT industry as a whole must meet to move toward.

- **Open standards and interoperability.** The networking segment of IT has really shown the rest of the industry how it should be done by contributing large amounts of knowledge to open standards bodies like the IEEE, IETF, and ISO. The rest of the industry should follow suit more closely.

- **Transparent costs.** Vendors need to be transparent regarding the cost of their product. The industry (channel included) is driven by meaningless list prices and deep discounts, leading to a real price that is unknown to the customer until they're being smothered by six sales people. The industry is also notorious for sneaking in licensing and add-on fees that aren't initially clear. The next generation of IT professionals has no time for these games and just want a reasonable number from the start. Companies that do this today (freely publishing the true cost of their product) are having great success in attracting customers who don't want to play games for an extra 3% off.

- **Performance benchmarking standards.** The industry needs to agree on performance benchmarking standards that will mean the same thing across all manufacturers and to all users. The enterprise storage industry has become infamous for publishing numbers regarding IOPS — a number which is easily manipulated to mean whatever the publisher wants them to mean when the context isn't included.

Open Standards and Interoperability

First of all, what does *open* mean exactly? And, in terms of standards, if something is the opposite of open, what does that make it? "Open" is basically a way of describing a collaborative work where the responsibility is shared between multiple entities and anyone can view the work.

Multiple businesses or individuals that have a common problem or goal work together to develop a product or standard. So in this sense, the opposite of *open* would be *proprietary*.

In the case of open standards, this means that a few collaborative entities work together to define a standard that all of their products will conform to. This common blueprint ensures compatibility of components in an environment, even if they're coming from different manufacturers.

There's more fuss in the IT industry now regarding open standards and interoperability, and there's a perfectly good reason for that. The various layers of abstraction present (and being developed) in the data center are making certain components a commodity. Decision makers have less loyalty to any one manufacturer than was previously common, and instead choose based on features and cost. In essence, vendors lose the ability to "own the whole stack" and bake-in their proprietary intellectual property.

The Challenge of Interoperability

While this is good for competition and the industry in general, it does provide a bit of a challenge. When it comes to interoperability, it's impossible for a manufacturer to say that their component will work with every other component you could buy.

What would be really nice, however, is if the manufacturer didn't have to validate all these different platforms to know that it would work. This is the part where open standards come in.

If storage companies with similar interests developed an open standard for a certain storage feature, then everyone could implement that feature while ensuring their products would be compatible with one another. A real example of this in the storage world will be seen later when NVMe is discussed.

A practical example of an open standard that network administrators would be familiar with is the IEEE 802.1AB standard; this is the standard that defines Link Layer Discovery Protocol (LLDP). LLDP is used to advertise and gather information about devices on a network. A network device will advertise the capabilities it has, as well as information regarding its manufacturer, model, management IP, and so on. Other devices receive this information on links connected to it.

This open standard is so valuable because a network made up of Cisco, Brocade, Juniper, and HP switches (which all run proprietary software) can be fully discovered with LLDP. Why? Because the different manufacturers all implemented an open standard which ensures compatibility with the others.

In sticking with the discovery theme, you could contrast an open standard like LLDP with a proprietary standard like Cisco Discovery Protocol (CDP).

A proprietary standard like CDP has a major advantage to the manufacturer, but a major disadvantage to the IT organization. The benefit of such a standard is that the manufacturer (Cisco) can do whatever they want to with the technology. This means they can develop it more rapidly, expand in any direction they want, and discontinue support for the protocol if they want to. That all sounds pretty good, so what's the problem? Unfortunately, in a mixed environment like the one above, the Brocade, Juniper, and HP switches may not be able to discover the Cisco switches. You can clearly see the problem here for the IT organization: vendor lock-in. Unless the organization buys all Cisco switches, the networking picture becomes disjointed.

The Future of Open Standards and Interoperability

In data centers of the future, open standards will be more vital than they've ever been as IT organizations flee from vendor lock-in. Being a part of an open standards committee or committing code to open source projects is going to be nearly a requirement for any major manufacturer.

One final word on open standards: trust is harder to come by now than ever, and in the future it will be more so. One of the benefits of open standards is that they're driven by a committee, anyone can participate, and anyone is free to review the specifications. This has a major security implication, because hiding malicious code or an "accidental" backdoor will be much harder when the whole world can peer at the specifications.

Proprietary software essentially has no guidelines, no checks and balances, and no reviews by objective third parties. For this reason, many security-conscious organizations have a strong affinity to open standards. Government agencies are particularly friendly to the open standards ecosystem for this reason.

Transparency on Pricing and Cost

In the IT industry, products have two main ways of finding their way to customers: either the manufacturer sells the product directly to the customer, or the product is distributed through channel partners, also known as "the channel." Both ways have been notoriously opaque and complex in recent history, and many IT practitioners have grown weary of the long sales cycles and complicated buying process. Through the use of meaningless list prices and hidden add-on costs and licenses, IT manufacturers have found a way to make higher margins on their product at the expense of seriously frustrating their customers.

There are a number of ways this happens in practice. When a customer purchases a product through a channel partner, there are multiple levels of the product being bought and sold before it reaches the customer. Figure 10.1 illustrates the breakdown of cost relative to list price in the simplified example that follows.

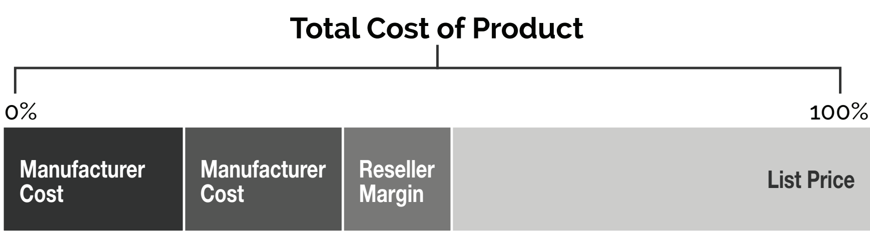


Figure 10-1: Costs relative to list price

Each level needs to make a profit, so they all purchase at a certain discount from list price before the product ultimately makes it to the customer. The channel partner ecosystem is immensely valuable to the IT industry, but this method of selling can sometimes cause unintended harm.

As an example, imagine that a manufacturer has a product with a list price of $100. A channel partner will then purchase the product from the manufacturer for $40. The salesperson interacting the customer tacks on a little for him- or herself (and the reseller), and sells the product to the customer for a final price of $55.

The list price ($100) is basically meaningless and confusing. A prospective customer looking for a $50 product may not consider this product as an option because the list price is so far out of range. Plus, the actual customer may be getting a great deal or an awful deal and never know the difference. You see, a product is worth what someone is willing to pay for it. So if the salesperson at the reseller can get the customer to pay $65, they could sell the product for that. If the customer really wants to do the project, but is considering walking away because of the price, the salesperson could sell it to them for $47 and still turn a profit. Throughout this process, the best interest of the customer winds up being fairly low on the list of priorities.

There are multiple reasons why hiding the true cost of products can seem to provide leverage. Many vendors don't even publish a meaningless list price; even worse, they don't publish any price at all! Why? Because then a customer has to ask for the price.

As soon as a customer asks, they're paired with a salesperson who can engage them and guide them through the buying process. Hiding list prices is a lead generation tactic. While that may have worked well in the past, in the future, it's going to frustrate a generation of people who are accustomed to having an overwhelming amount of information at their fingertips. Searching for information and being unable to find it is very frustrating to this new generation. Beyond list prices, the vendors who sneakily require certain important features to be purchased as an add-on, and charge for expansion or software upgrades, are expecting that each of these additions is an extra dose of revenue. This is a fundamentally short-sighted approach. In the same way as the list price issue, this practice will frustrate the customer (or potential customer) and ultimately lead to a loss of business over time, even if not immediately.

As an example, the current hyperconverged market contains a lot of marketing about the cost savings potential of a given platform. The problem is that there's no frame of reference provided. Their price isn't listed; the factors that make up the "40% savings" aren't listed, nor is what the platform is being compared to in order to make that judgment. Without an accurate frame of reference, saying a potential customer can expect to save a certain amount or percentage is meaningless.

This all seems pretty confusing, and that's because it is. The buying process is ultimately very complicated and slow due to all these details. Negotiating the discounts to be applied between the vendor and the partner, and then between the partner and the customer, can take time. The complicated ordering process can cause confusion because sorting through 65 cryptic line items on a quote proves to be too daunting for the potential customer. Future IT customers will not put up with the productivity and time lost playing this game. This way of doing business in IT needs to change, and a number of industry leaders are stepping up to blaze a new trail in this regard. They're posting true costs right on their website. They're creating an ordering process that involves two line items on a quote instead of two pages' worth. They're making it easy for the customer to buy the product.

Benefits of Transparency

What does the hardware or software manufacturer stand to gain by creating transparency and clarity in their buying process? A whole list of words than any company would love to be associated with like *trust*, *loyalty*, and *expeditiousness* on the part of the buyer to name a few.

Customers who don't feel like they're having the wool pulled over their eyes are likely to want to develop a healthy working relationship and are likely to purchase right away.

If the buying process is made clear by a quoting process that is fast and simple, the customer is likely to purchase quickly.

Manufacturers must start doing these two things in order to stay relevant:

- Show the true costs and remove the games from the buying process, while keeping in mind that each business needs to make money.

- Make the purchasing process (quotes) simple and clear. Manufacturers that are succeeding in this area commonly have a mere two or three line items on a quote. These quotes might include a product (the single line item which includes everything) and a couple of years of support, for instance. Without all the SKUs that look like hieroglyphs and vague descriptions, it's easy for the customer to see what they're getting into.

Doing this will change the industry in a few ways.

- **Customers will be more likely to purchase the proper product.** It's entirely likely that a customer might pass on a product that might be the right technical or operational fit based on the fact that they can't find out the true cost. Showing the true cost up front will make it easier for them to make the right decision.

- **Removing some of the opacity from the channel distribution model will change the way resellers make money.** Since margin in the products will likely go down, resellers will have to put more focus than ever before into their service offerings. The additional knowledge and skill a reseller has is what will set them apart, not just how deep of a discount they can give off the list price. Once the industry adjusts to this new mindset, it will be a win for everyone.

Performance Benchmarking Standards

In the business of IT (especially enterprise IT), recent years have seen a growing trend of what seems to be a disconnect between the marketing department and the product development teams.

While it's true that many products are being developed today that can achieve things we barely thought possible only a few short years ago, it's also still a fairly common occurrence that a product doesn't live up to the hype and promises made by marketing folks and sales people.

From the Field: Survey Results

61% of survey respondents indicated that they expect performance when deploying SDS and HCI to be higher than that of their legacy infrastructure. This expectation makes it critical that performance tests are performed in an acceptably unbiased and realistic way.

Unfortunately, as a result, many IT practitioners have become overly cautious and have formed a distrust for numbers and specifications. In IT sales presentations around the world, a question commonly heard today is, "Are these *real* numbers, or marketing numbers?" The root of this issue lies in the fact that the industry as a whole is a bit wishy-washy in terms of accepted methods for performance testing. Some areas have more established testing methodologies and tools than others, but overall the performance testing arena is a bit of a free-for-all.

Storage Benchmarking

One of the most obvious areas this problem shows up in is enterprise storage. The tools commonly used to benchmark storage systems are confusing and potentially ambiguous at best, and downright inaccurate if configured improperly.

Part of the problem here is that many of the benchmarking tools generate a synthetic workload pattern.

This means that in benchmarking, the storage platform will be serving a different I/O characteristic than it will serve when it's in production. This leads to benchmark numbers being unreliable unless the benchmarking tool can accurately simulate a production environment.

Other common pitfalls in storage benchmarking are: testing with a workload that fits entirely in cache and building systems specifically for benchmarking that aren't the same as the system a customer will buy, even though they have the same model number. The unfortunate fact is that many tools simply can't simulate a production environment with any degree of realism.

The Importance of Open Benchmarking

So in the future of IT, what can be done about this?

In the same way that open standards need to be adopted for the sake of interoperability and the development of platforms, open benchmarking standards need to be created and adopted. Some areas of IT already have these standards while others don't.

In the storage industry, manufacturers need to stop publishing numbers and showing demos with tools like Iometer that are widely misunderstood. While Iometer can be a useful tool, operating it and properly interpreting the results takes deeper knowledge than many individuals who operate it possess. Publishing numbers based on improperly conducted Iometer tests can be downright misleading.

Again, manufacturers should conform to an open, objective standard for publishing performance results.

In the case of storage, this would be something like the specifications developed by the Storage Performance Council (known as SPC-X where X represents a certain performance test). SPC benchmarks don't accurately test *all* types of workloads, but they do provide common ground upon which any manufacturer can test their product and compare it apples to apples with another vendor's competing product.

In the future of IT, publishing results based on standards like this will be a requirement. To combat all the distrust and cynicism the industry has developed in regards to performance in recent years, objective assessments are the only way to gain trust in a claim moving forward.

Chapter Recap

In this chapter, you learned about the way the data center industry is being transformed by a focus on automation, orchestration, and the DevOps methodology. You also learned about some of the changing needs in the IT industry. The following are a few key terms and concepts to take away from this chapter.

- **Automation** is becoming critical in the modern data center. It solves the problems of speed and accuracy by allowing computers to do the work that can be codified, allowing humans to focus on higher level objectives.

- **Orchestration** can be seen as either an extension of or a complement to automation, but extending across systems and platforms. Orchestration completes entire business processes without human intervention. We used the example of IT's onboarding of a new employee.

- **DevOps** is the idea of pulling the operations team into the development methodology to use their unique expertise to help the development process be more rapid. It also includes implementing tools and processes that leverage all the different skill sets available.

- The IT industry has changing needs, some of the most important of which are: **open standards and interoperability, transparency on pricing and cost,** and **performance benchmarking standards.**

In the next and final chapter, we'll discuss the future of the data center. We'll take a look at what technologies and business considerations are shaping the coming years in the data center.

11

The Future of the Data Center

This final chapter will expound upon the same principle that started this book in Chapter 1: that computing power doubles every few years. This principle, better known as Moore's Law, has, and will continue to, radically change the data center.

As computing power reaches new heights in the coming years, the most successful organizations will be empowered by taking advantage of this. Because of the exponential nature of Moore's Law, failing to stay current will more quickly result in obsolescence than ever before. That means that if the technologies a competitor uses are a year or two newer than yours, they will provide much greater performance and insight which could result in rapid loss of your market share.

Beyond rapidly changing technology, the IT industry is changing in regard to the preferred method of software development and consumption; customers also have different expectations from manufacturers than they did a few years ago. The evolution of cloud-based data centers will continue to change the way IT is done in the enterprise, whether on-premises or with a public cloud provider.

This chapter will explore the trends and technologies that are on the horizon and will revolutionize data center architecture.

Keep in mind that as the cost of NVMe-based drives and Non-Volatile RAM in general comes down, performance potential in the data center will skyrocket. Also, as Intel continues to perfect the Rack Scale Architecture model, new possibilities will emerge from a disaggregation and abstraction standpoint. This means the data center will look at whole lot different in a few short years.

Future Trends

The recent evolution of the data center and IT as a business unit has been more exciting than ever. However, the coming few years are likely to bring even more excitement. Some of the technologies that are currently in development (and could be in data centers within a few years) are simply mind boggling.

Some of the trends that will grow in the next few years include:

- The increased uptake of a container-based service provisioning approach.
- Improvements in storage capacity and performance, including ever-evolving open source tools and flash storage capabilities.
- Different ways to pool and abstracting physical resources.
- Advancements in the way IT organizations interact with cloud services.

These will make the data center 3 to 4 years from now look quite different than it does today.

For example, enabled by ever-increasing compute power, the amount of data that will be stored and acted upon in the next decade huge. New application architectures will be required, and dramatically enhanced hardware will be created to support the new requirements.

Plus, the types of storage that have changed the data center in the past few years (namely NAND flash) will likely be displaced by a successor even more suited to the performance and capacity requirements of future IT.

Migrating to the cloud has garnered a lot of lip service lately, but relatively few organizations are actually doing it compared to the number that are talking about it. Soon, most organizations will adopt a cloud-focused model for provisioning services. There will be abundant opportunities in the services industry for helping customers migrate to the cloud.

Let's explore these trends in depth.

Containers Instead of Virtual Machines

The last couple of years have seen the rise of software products that leverage Linux Containers (LXC) to deploy many instances of an application on one operating system. Running applications in LXC is an alternative to running applications in a virtual machine.

However, LXC is not a new technology; it was first released in 2008 and is viewed as relatively mature. In fact, *containerization* as a concept existed earlier in Solaris Zones (2005) and AIX Workload partitioning (2007).

The State of Virtual Machines Versus Containers

Past	Present	Future
- Virtual Machines	- Mostly Virutal Machines - Some Containers	- Mostly Containers - Some Virtual Machines

Containers are easy to deploy and consume very little resources as compared to a virtual machine. Where virtual machines abstract operating systems from the underlying hardware, containers abstract applications from the operating system.

Containers make it possible to run many copies of the same or similar applications on top of one operating system, thus using a single copy of all the operating system files and shared libraries. **Figure 11-1** illustrates the difference between VMs and containers.

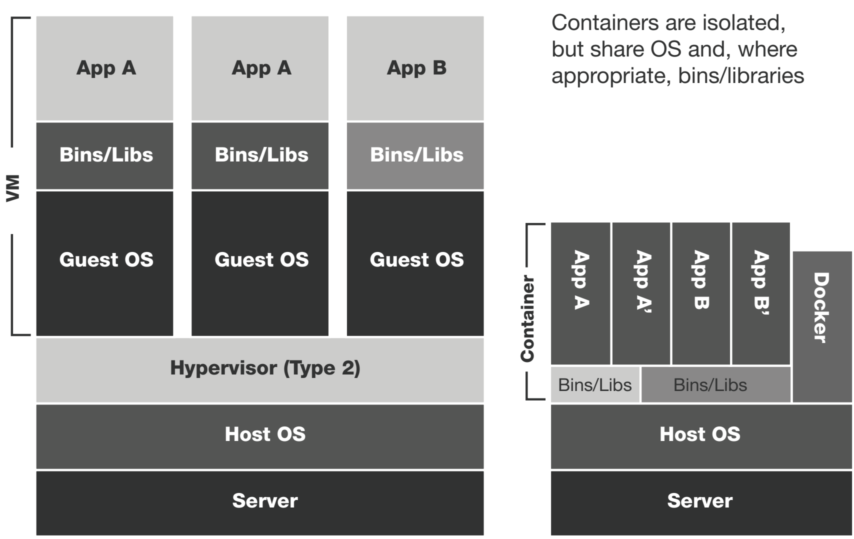


Figure 11-1: Virtual Machines versus Containers

From an efficiency standpoint, this is obviously a huge win; it also has the benefit of ensuring consistency across the applications since they're sharing the dependencies rather than each application having a potentially disparate version of the dependency.

LXC developers are now expanding on the platform with LXD, a new way of interacting with containers which exposes a REST API. This will allow much greater orchestration of containers moving forward.

As containers become more popular, the distribution of data center software will likely focus more on containers than on virtual machines.

Tip from the Field by James Green

LXC is respected as a great project, but also has the potential to be circumvented by projects like libcontainer due to their efficiency and potential for going cross-platform in the near future.

— James Green

Pre-Packaging of Applications as Containers

It's commonplace in today's data center for a new application to be downloaded from the vendor's website as an open virtualization archive (OVA) file. OVA is an open standard that defines virtual machine specifications; it can be imported into a hypervisor-based virtualization platform like vSphere, Hyper-V, or KVM.

The OVA file contains a pre-built virtual machine with the vendor's application pre-installed. Compared to downloading the binaries and installing them manually, downloading the full OVA carries the benefit of allowing the vendor to make additional modifications such as including dependencies and tuning the underlying operating system to suit the application.

Because of how lightweight and portable containers are, the future of the data center will contain a shift toward applications being shipped as containers rather than OVA/virtual machines like they are today.

Pre-packing applications as containers rather than virtual machines will allow IT organizations to continue deploying applications that have been pre-configured by the software manufacturers but without the overhead of a traditional virtual machine.

Some bleeding-edge environments may not even employ hypervisor-based virtualization at all in favor of container-based virtualization on bare metal with an operating system such as CoreOS. Pre-packaged containers will allow these customers to deploy the application in an environment where an OVA file would be next to useless.

From the vendor's standpoint, using a container as the method of delivery may also be preferable to the OVA/virtual machine, because it may integrate more tightly into the internal development process.

Enterprises are turning to container-based virtualization to accelerate the software development life cycle of their internal applications; however, companies whose business is creating software will benefit from this shift as well. Naturally, with the ability to iterate more quickly, the end product will be better and be delivered sooner.

Cross-Platform Container Support

Containers' primary advantage compared to virtual machines is also a disadvantage at times. When many containers share a single operating system, it allows for much greater efficiency than the 1:1 nature of a virtual machine. However, this can be problematic in the sense that not all applications are suited for the same guest OS. This is most clearly seen in that applications may be developed to run on a Linux OS or a Windows OS. Not only that, but applications may have a dependency on a particular kernel version of the operating system in question.

The first obvious solution to this problem is to have at least one of each type of guest OS required to run all the application. But this immediately becomes a management nightmare if each one is a management silo.

This is where cross-platform container support and solutions come in.

Container management platforms step in here and are beginning to release software that will manage these disparate operating systems as if they're one big pool of resources and allow the container to run on whichever host resource is most appropriate.

In this way, any given container can be run on a Linux machine where appropriate or on a Windows machine where appropriate, but all management is done from a single platform.

The next logical step in solving this problem is to actually develop containers that can run on multiple platforms. Meaning, a certain application could be run on a Window host or a Linux host interchangeably. This is far away from reality in common practice, but Microsoft and Docker have already demonstrated a .NET application being run on both a Linux and Windows machine.

This endeavor will characterize the next phase of containerization in the data center: allowing containers to be compatible across multiple OS platforms.

Open Source Tools

The pervasiveness of open source software in the modern data center is growing very rapidly, as of late. But if open source software is so great, why isn't it just used for everything?

Well, that's exactly what many of the web giants of the past decade have asked as well. Google, Facebook, and the like are responsible for some of the most popular open source projects being developed at present.

Open source software, due to its collaborative nature which spans verticals and geographies, is commonly produced at a much faster pace and with higher quality than a competitive proprietary project started around the same time.

For this reason, many IT organizations are choosing to focus their new initiative on open source first, and proprietary software second if no open source option fits the bill. Because of the highly competitive environment at the top of any industry, being on the bleeding edge with open source software can give a non-trivial advantage over a competitor who is at the mercy of a proprietary software developer.

Moving into the future, open source software will be the default mode of most organizations. It's easy to see from examples like Docker and Hadoop that the power of the community that can form around an open source project can extend well beyond the capabilities of any single organization. That community of contributors and committee members not only helps to develop the project with exceptionally high quality, but they act in the best interest of the user base.

Proprietary Software vs. Open Source

One of the primary problems users of proprietary software face is that they're at the mercy of the software vendor. With open source software, on the other hand, the direction of the project is directly influenced by the user base. Changes won't be made that aren't in the best interest of the users and new features will be focused on with priorities defined by the users. Try getting that level of input with most pieces of proprietary software and you'll be sorely disappointed.

The State of Open Source

Past	Present	Future
Enterprises primarily use proprietary software	Enterprises primarily use proprietary software, but leverage open source projects	Enterprises primarily use open source software and only choose proprietary software when there's no other option

The final differentiator in the open source versus proprietary software debate in many organizations turns out to be cost. No big surprise there. Community-supported editions of open source software are commonly available for free or for a nominal fee. At scale, this difference can really add up and make open source a no brainer.

Flash Capacity

NAND flash has set new expectations for performance in the data center, but has left something to be desired in terms of capacity. When NAND solid-state drives (SSDs) first became available to consumers, a 30 GB SSD wasn't small. However, that's far too small to be of much practical use to enterprise data centers storing (potentially) petabytes of data.

Over the last five years, the capacity issue has been addressed to some degree, but the breakthrough in flash capacity is just happening now.

Traditional vs. 3D NAND Structure

Flash memory-based drives store binary data in cells. The traditional approach to increasing flash capacity, given that the form factor of the drive cannot change, has been to decrease the size of those cells. Using smaller cells means that more of them can fit on a given chip.

Unfortunately, at this point the cells have become so small that creating smaller ones is becoming quite challenging.

With new technology referred to as 3D NAND, flash memory manufacturers have begun stacking multiple sets of cells on top of one another to achieve greater density in a single disk.

For example, Intel and Micron jointly developed a technology known as 3D XPoint. The technology stacks flash cells vertically, one on top of the other. This is the origin of the "3D" portion of the name 3D NAND.

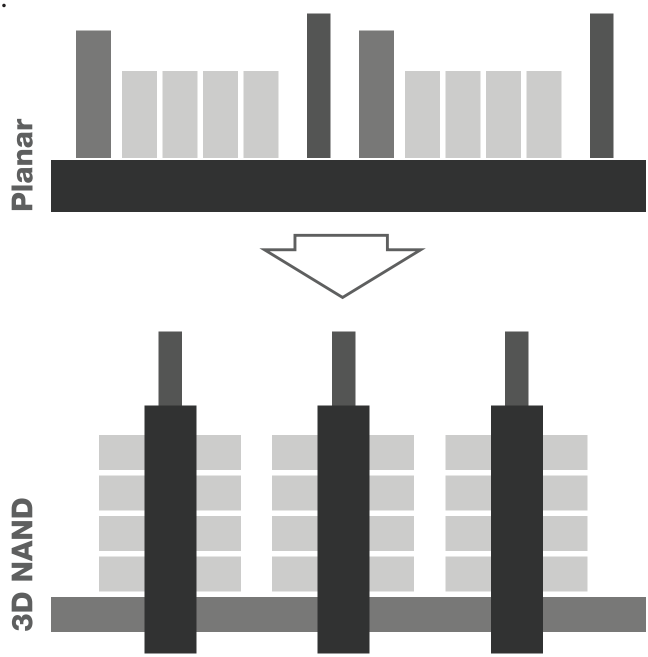


Figure 11-2: Planar vs. 3D NAND

Previous implementations of flash memory were planar, or two-dimensional. By stacking multiple layers, the third dimension is added. With this third dimension, the need to reduce the size of the cells is removed, and flash manufacturers can actually use slightly larger cells while still increasing capacity. **Figure 11-2** illustrates how the components are turned from horizontal to vertical in this type of memory.

The Future of 3D NAND Capacity

The capacity increases that have been possible since the development of this 3D NAND technology are substantial to say the least. Samsung announced a 16 TB SSD in 2015 that is powered by this technology and should be available to consumers in the near future.

In the future of the data center, technology advances will allow the size of flash cells to continue to decrease. This, in combination with the 3D design, will allow SSDs to have acceptable capacities to all but eliminate the need for spinning, magnetic disks.

Technology improvements in memory and storage seem to alternate between performance and capacity. Along with this breakthrough in capacity, IT organizations also need to ensure that all that capacity can perform at the highest level.

The State of Non-Volatile Memory

Past	Present	Future
-Planar NAND -Disk for capacity, flash for performance	-3D NAND in develoment -Flash for Tier1 -Disk for additional capacity & secondary storage	-3D NAND & other type of dense non-volatile memory for primary storage -NVMe leverage for performance -PCM, MRAM, and RRAM may be used

Non-Volatile Memory Express

Flash storage presents an interesting problem in terms of performance. It's too fast.

It's too fast in the sense that many SSDs can perform at substantially higher read/write speeds than the bus and access specification is capable of providing.

The Past

In the recent past, storage has typically used the advanced host controller interface (AHCI) standard for communicating with the rest of the system. The problem is that this standard for accessing storage was designed with mechanical hard drives in mind. Many SATA-based SSDs could perform higher than the maximum throughput limit of the bus if the bus and interface standard could handle it.

SSDs need faster access to the CPU and memory, and the best place to find that is on the PCI Express (PCIe) bus.

PCIe and NVMe

In the early years of enterprise SSD usage, vendors did create drives that utilized the PCIe bus. The problem was that a standard did not exist so that manufacturers could create hardware components, system drivers, and software that was all interoperable.

Realizing this, over 90 companies, led by Intel, collaborated as a part of the NVM Express Workgroup to define and create the Non-Volatile Memory Express standard, commonly referred to as NVMe.

The low latency and parallelism of the PCIe bus allow SSDs using NVMe to achieve orders of magnitude higher performance. While AHCI has a single command queue with a depth of 32 commands, NVMe has 216 (65,536) queues with a depth of 216 commands per queue.

Also, AHCI allows devices to allocate a single interrupt, where NVMe allows (via the MSI-X extension) devices up to 2,048 interrupts. Especially in the case of random reads, this parallelism allows IOPS performance approaching double that of the same drive using AHCI at certain queue depths.

The Future of NVMe

NVMe as a standard is still relatively new to the manufacturers working to implement it. Development of the specification has been taking place for some time; the project was underway by late 2009, the spec was released in early 2011, and the first commercially available drive to leverage NVMe was released in 2012.

The year 2016 will likely be the year where mainstream adoption of the technology takes place, and from there the price will start to come down to a place that is more approachable for the average data center. Because the economy of flash storage is largely related to cost per gigabyte, other technologies like 3D NAND will make devices leveraging NVMe a more viable option by increasing density.

Beyond Today's Flash

While NAND flash has been in development for decades and is very mature at this point, engineers are realizing that there are physical limitations to the density NAND flash can be driven to.

As a reminder, NAND flash capacity is increased by decreasing the size of the cells, so more can fit in the same amount of physical space. At this point, NAND flash cells really can't get smaller. The temporary bandage to this problem is 3D NAND (stacking NAND cells to achieve higher density). In time though, this density improvement will not be adequate either and an all together new technology will need to take the place of NAND flash.

There are several options that will likely emerge as the leader to replace NAND down the road. Each of these technologies may sound like they're straight from *Star Trek*, but they have the potential to be in enterprise data centers in only a few short years.

Phase Change Memory

One option is called Phase Change Memory (PCM). Rather than storing a bit (1 or 0, on or off) in a transistor which holds an electrical charge, PCM represents the bit's value by literally changing the atomic state of the material the cell is made out of. That material is chalcogenide glass; interestingly, this is the same material that makes up the core of fiber optic cables and the portion of rewritable discs (CDs/DVDs) that stores the data.

PCM has some of the same density challenges NAND does, but it has substantially better endurance.

Resistive RAM

Another future non-volatile memory technology that could replace NAND is called Resistive RAM (RRAM). This may also be referred to as ReRAM or Memristor technology.

This type of memory functions on the concept of resistive switching, wherein the resistance of a dielectric material changes based on electric current being applied.

RRAM is possibly the most likely candidate to replace NAND due to very low power consumption combined with high durability and performance.

Magnetroresistive RAM

A third contender for the future of flash is called magnetoresistive RAM (MRAM). This technology is similar to PCM and RRAM in that it measures resistance to determine the cell state (1 or 0).

What's different about MRAM is that, rather than using an electrical charge to represent state, MRAM uses a magnetic charge. The resistance of the material will change based on the presence or absence of a magnetic charge.

However, there are some potential scalability problems with this technology, so it's not the strongest candidate.

Other Options

PCM, RRAM, and MRAM are far from an exhaustive list of the technologies being explored to replace NAND flash in the future. It won't be useful to expound on all of the other options here, but research on developing non-volatile memory technologies might also include:

- Ferroelectric RAM
- Spin-Transfer Torque RAM.
- Solid Electrolyte Memory.
- Racetrack Memory.

Each has its merits and challenges, but there is certainly no shortage of options for the future of non-volatile memory.

Pooling of Resources

Non-volatile memory isn't the only place where massive change is on the horizon. Especially in large public cloud data centers, the standard container of computing power could change in the near future. The unit of computing (as far as a data center operator has been concerned) in recent years has been a server. As you know, a server contains a CPU or two, a bit of memory, some network interfaces, and perhaps some internal storage. A rack is loaded full of servers to provide the cumulative computing power that the data center needs. But what if the container used to contain the computing power was bigger than a server?

Rack Scale Architecture

In conjunction with some of its partners, Intel has been developing a new architecture called Rack Scale Architecture that abstracts compute resources to a set of hardware-level APIs. It can be used in tandem with a very high-speed, rack-scale fabric like one created with silicon photonics to provide disaggregated compute resources. This basically removes the physical server as the boundary for compute resources and moves to container out on the rack level.

In massive data centers, this will allow for more efficiency and agility in exactly the same way server abstraction via hypervisor did just a few years ago.

Imagine a standard data center rack. At the top are some very dense network switches (perhaps 64 ports each) using the small and high-speed MXC connector. This fabric uses silicon photonics technology to achieve multiple terabits of throughput in and between racks. Below that sits a number of trays of CPUs. Not whole servers, just the CPUs themselves. Below that, trays and trays of memory in the same way. And at the bottom of the rack, some NVMe-connected storage. **Figure 11-3** illustrates a disaggregated architecture like the one just described.

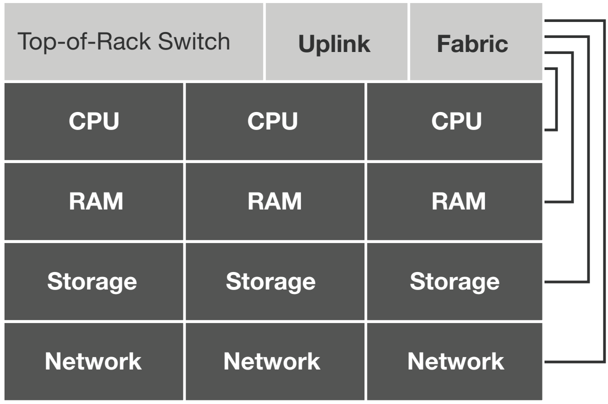


Figure 11-3: Disaggregated resources at the rack scale

This is what the future data center may look like. But why?

Benefits of Disaggregation

What is to be gained by the disaggregation and pooling of components that were previously contained to a server? Scalability and efficiency are certain to increase. Plus, this new architecture would change the way that upgrades are done.

Today, if an IT organization wants to use the latest and greatest CPUs, the server is replaced with a new one that supports them. In an architecture where CPUs were disaggregated from the rest of the system, the CPUs could be replaced without having to replace the motherboard, memory, storage controllers, network cards, and so on. This potentially makes upgrades quite a bit cheaper if only a certain component needs to be upgraded. It will also make it much easier for the IT staff doing the upgrade.

The Future

The key to achieving a mainstream uptake of the disaggregation of compute resources into a larger pool will be manufacturer adoption and interoperability. Just like with the other awesome technologies Intel develops, it only becomes commonplace once other ecosystem vendors invest the time and effort into creating compatible products. Since this architecture is vastly different than the architecture that has defined data centers for years, this one could take a while.

The State of Resource Pooling

Past	Present	Future
-Standalone physical servers and storage	-Virtualized servers & storage, pooling takes place at the hypervisor node level	-Virtualization and disaggregation takes place at the rack level, individual "servers" become obsolete

The Cloud

The concept of cloud computing has completely revolutionized the industry and changed the way many companies do business. It has enabled agility and growth that would have very likely been impossible in a legacy environment. However, cloud adoption is not complete.

There are still many phases of the migration to the cloud that businesses are still working through. This is likely to be ongoing for a number of years because, just like any major transition in architecture, it's not typically a lift-and-shift sort of move. It's a gradual, opportunistic process where the changes in infrastructure design begin to incorporate cloud components where it makes sense.

Hybrid Cloud

As a part of this gradual shift, a large number of data center infrastructures in the coming years will take a hybrid cloud stance. *Hybrid cloud* means that a portion of the IT resources exist in an on-premises data center and a portion exists in a third-party cloud data center. But even in that hybrid approach, there are questions yet to be answered.

How do you handle security in the cloud? How about making sure that data is protected and can be restored in the event of a loss? Answering questions like these will help you plan for and feel prepared to make the leap.

Blurring the Lines Between Public and Private Cloud

Since the first enterprises began adopting public cloud provisioning as a resource model, there has been a clear distinction between private cloud (a cloud-oriented infrastructure in their on-premises data center) and the public cloud (cloud-oriented infrastructure hosted and sold by a third party). As trust in public cloud offerings grows and feature sets become more rich, the lines the segregate public and private cloud will begin to blur. This is because managing two distinct infrastructures is hard, therefore vendors will create software to allow an organization to manage two infrastructure silos as if they are one.

One of the ways this will take place is by abstracting the cloud resources in such a way that the application or user doesn't know or care where they're coming from.

Another way it will take place is by implementing solutions that leverage cloud-based solutions to manage on-premises equipment. Some good examples of this are managed OpenStack tools where the control mechanism lives in a public cloud and manages on-premises equipment. Is that public or private cloud? Both, perhaps? Similarly, network equipment that exists physically on-premises by is managed entirely by a controller in a public cloud. This is neither an entirely public nor a private architecture.

The future growth in this area will allow more agility and flexibility than has been available in the past. By choosing the best location for any given infrastructure component without being constrained to a certain environment, more creative and available designs can take shape. In some multi-tiered application use cases, the most optimal deployment involves running some front-end components in a public cloud for scalability and availability purposes and running the database tier in an on-premises data center.

Running Workloads in the Data Center Using Public Cloud Storage

Multi-tiered applications aren't the only thing that can span public and private clouds. In fact, there are manufacturers and technologies proposed exactly the reverse of what a multi-tiered application would do.

Where a tiered application would likely run web servers in the cloud and keep the storage in the on-premises data center, there are also certain use cases where the path that makes the most sense is to run user-facing workloads in the data center while the storage back-end lives in the public cloud.

Maintaining storage systems is hard. Upgrading storage is one of the most frustrating experiences many IT administrators have had in the last decade. There are vendors who offer unique solutions that take advantage of caching, WAN acceleration, and a little bit of "special sauce" that allows an organization to keep primary data in a public cloud. For performance reasons alone, this would have been unheard of only a few years ago. However, because of networking connectivity advancements and a period of time in which lessons were learned, it's possible today.

Does hosting primary data in a public cloud make sense as an IT strategy moving forward?

Well, if managing storage on-premises is frustrating and costly, it certainly seems attractive to get rid of it. Moving primary storage to the cloud in a managed service approach allows an IT organization to ensure that they're always running on up-to-date hardware and software and have a well-trained set of eyes on their data without having to actually do that themselves. From an operational perspective, this move has the potential for quite a high return on investment (ROI). If all the hours spent upgrading and migrating storage in a three-year period could be redirected toward other efforts, that would likely make a huge impact on the bottom line.

Naturally, there are some concerns with this architecture. The primary concern would be security, of course.

More than ever, companies are sensitive about their data's security. Placing data in someone else's data center necessarily turns over control of security to them as well. With the ever-growing number of data security breaches in a year, organizations may be hesitant to adopt this sort of model.

What many of those organizations should consider is that the public cloud data centers where their data would reside are likely held to even higher standards than they are themselves. The public cloud may be safer than many perceive it to be, due to providers being held to higher standards for things like encryption, patch compliance, change control, and governance.

Moving Workloads Seamlessly Between Private and Public Cloud

Sometimes running workloads on-premises is the perfect decision, while sometimes running workloads in a public cloud is the perfect decision. Most times, however, the decision is multi-faceted and a case could be made either way. Also, the requirements for the service may change over time. With that in mind, what is really needed to move cloud adoption forward is the ability to seamlessly move workloads from an on-premises data center to a public cloud data center.

This choice and flexibility based on requirements at any one moment will be the push that many organizations need to begin moving critical workloads to the cloud.

There are various reasons why you might want to move a workload from an on-premises data center to the cloud and back. For example, many data centers charge based utilization and peak hours much like an energy company does (cloud resources are commonly viewed as a utility, just like energy). It may make sense to run a workload in an on-premises data center when peak hours make running in the public cloud expensive, but then migrate the workload to the public cloud during off-peak hours for better performance.

Another example may be an online retailer who has dramatic changes in seasonal traffic. They may choose to run their workloads on-premises during the spring and summer, but move their workloads to a public cloud for the holiday season to accommodate the additional scale required. This allows them to keep the costs of the on-premises equipment low while only paying for what they need during the busy season.

Making this seamless migration possible is challenged mostly by application design and connectivity constraints. It's relatively easy to migrate workloads between data centers without downtime from an infrastructure perspective. The problem is that the application being moved is not typically designed to handle this sort of move so it must be unaware. A shim of sorts must be put in place at the application layer to allow the workload to move without it knowing that it moved.

There are two ways to get past this in the future: either the applications will need to be redesigned to be disaggregated in such a way that which data center they're running in is irrelevant, or manufacturers will create better shims.

In reality, both of these options will probably be utilized in the future. Innovators will create applications that can move seamlessly; manufacturers will create ways of abstracting the move from the application so that IT organizations stuck on legacy software for various business reasons will still be able to access the operational and financial advantages of seamlessly moving workloads between the public and private cloud.

Brokering and Abstraction of Cloud Resources

As cloud adoption grows, many organizations will make use of multiple cloud-based infrastructures across their organization. That may be due to segmentation between different subsets of the business or business units. Or it may be due to a desire for the increased availability offered by diversifying across multiple data center providers and locations.

Whatever the reason, multiple cloud resources which providers use complicates things to say the least. For this reason, in the future, more attention will be paid to brokering and abstracting cloud resources. In the same way that data centers today run workload placement engines to determine the best hypervisor and storage volume to place a virtual machine on, data centers in the future will run advanced workload placement engines to determine which cloud among the organization's multiple options is the best one to run a given workload in.

This will come with a number of distinct advantages to managing them manually:

- **It will be more reliable than humans;** processes carried out by people are error prone, but processes carried out by policy and abstraction are more accurate.

- **It will also be likely to make better decisions.** Humans aren't very good at considering all the inputs and frequently overlook things. A brokering engine can make intelligent decisions by taking all the variables into account every time.

- **It will reduce the operational complexity of the data center.** This will occur from a day-to-day perspective and from the viewpoint of the operations staff.

One of the main objectives of the cloud brokering technology will be to provide and interface for controlling the whole environment from an automation and orchestration standpoint.

Rather than interacting with all the separate cloud options the organization has available, the cloud abstraction platform will expose a set of APIs that other infrastructure components can interface with to provide a single point of control across all of the cloud platforms. The high-level overview gained by designing the environment this way will also give organization insight into the health and performance of their entire enterprise as opposed to distinct silos of cloud resources.

From a planning and troubleshooting perspective, this view is critical to maintaining control of the environment as it grows to span multiple cloud environments. With an environment that spans multiple cloud providers, there's a few additional challenges to consider beyond how to manage it all. Namely, how is the data protected from failures and user error and how is it all secured?

Data Protection and the Cloud

Data protection in an on-premises data center is pretty well understood. From the beginning of data center history, organizations have recognized a need to protect the important data they're storing. Cloud adoption causes some challenges here because the unique architecture and paradigm shift changes the way that data protection must be done. For example, there are plenty of organizations around that are still backing up to tape. (About the only place that backing cloud resources up to tape makes sense is in a comic strip!)

However, besides presenting a totally different architecture to design data protection practices around, public cloud data protection allows the customer and the service provider each own different portions of the bigger data protection picture.

Determining who is responsible for what, and making sure that all the gaps in protection are covered between the two sides, is half the challenge of a successful cloud data protection strategy. Another part of the challenge is determining exactly what it is that must be protected.

As applications are designed or redesigned in a cloud-native fashion, the application infrastructure becomes much more ephemeral and transient. A web or an application server failing is almost inconsequential. The reaction will be to create a new one to takes its place (which may even happen automatically in code) and everything will be back to normal. With that in mind, it's possible that the data protection strategy required is more simple.

The future of data protection in the cloud will involve customers answering a very important set of questions. What infrastructure and application components must be backed up? Who is responsible for backing up each of those components? What is the method for backing them up, and where does the backup data reside?

Answering these questions, combined with the typical RTO/RPO discussion, will yield a healthy backup strategy with cloud resources included.

Technology developments will make it possible to back up on-premises resources to the cloud with ease, and from public cloud to public cloud with the same ease. This allows one provider to provide the primary infrastructure whole another cloud service provider can be leveraged to store the backup data. In this way, the IT organization isn't vulnerable to a failure of any one public cloud provider. Protecting yourself from vulnerabilities in the cloud is important. But surprisingly, an IT organization's biggest threat to cloud security might be the IT organization itself.

Security in the Cloud

Security of the data and services an organization has running in the cloud remain the number-one concern among the majority of companies who still avoid moving to public cloud models. While these reservations may have merit in some limited cases, you must consider whether these fears are actually legitimate and how that perception will evolve.

Obviously, data loss is never trivial. No matter how small the event is, losing control of company data has an impact on the business. So caution in certainly warranted. However, there are substantially more cases of data loss each year where the breach was due to improper configuration by the customer rather than a vulnerability that was the fault of the cloud provider.

The largest improvement in cloud security over the next couple of years will likely not come from technology, but rather the people using it.

What is required for an enterprise to secure the resources running in the public cloud is educating the teams that build, implement, and maintain the cloud systems. From a technology perspective, the systems in use today are actually quite secure. So, if the individuals and teams working on an IT organization's cloud resources understand the design considerations and technical configurations required to secure their data in the cloud, security is increased substantially.

Unfortunately, people are imperfect and mistakes will be made. Pride will cause people to configure things they don't know how to configure and leave gaping security holes. This is why education will be so important.

The IT industry analyst firm Gartner speculates that by the year 2020, a full 95% of cloud security incidents will be the fault of the customer and not the cloud provider. This means that hesitant organizations who would love to take advantage of a public cloud's operational and financial advantages should actually turn their focus inward. They'll need to consider the policies and practices that will make their teams successful at implementing their cloud adoption strategy securely.

This isn't to say that cloud security won't improve from a technology standpoint as well. One of the fastest growing security technologies in the coming years will be the cloud control and accounting tools that help a large organization audit and control access to their cloud resources. One of the challenges with software as a service (SaaS) and platform as a service (PaaS) services is accurately understanding who has access to what across the entire organization. These tools will help organizations understand that in order to stay secure.

Chapter Recap

In this chapter, you saw a glimpse of what the future holds for the data center. This is so important, because the changing data center will change the way the rest of the business operates. Fortunately, the future looks very exciting! The following are a few key terms and concepts to take away from this chapter.

- The future of the data center likely contains a heavy emphasis on **containers,** which are easy to deploy and consume very little overhead as compared to a virtual machine.

- There will be an increased adoption of **open-source tools** because of the increased visibility into the product and the communities that form around them.

- The performance of the future data center will be affected by **advanced flash storage technologies** such as NVMe, PCM, RRAM, and MRAM.

- For scale purposes, we're likely to see the **disaggregation** of compute resources to the rack scale (as opposed to the current server/node scale).

- No doubt, the blurring lines between public and private cloud will change the way IT does business in the near future. There will be new markets for cloud **brokering** and **abstraction**.

- Due to increased cloud adoption, there will be a growing need for **data protection** and **security** in the cloud.

Now that you're equipped to transform your own data center, don't wait to start. Remember what Jack Welch said, "An organization's ability to learn, and translate that learning into action rapidly, is the ultimate competitive advantage." That exact advantage can come to your organization, but only if you start today!

Building a Modern Data Center as a Strategic Advantage

A fundamental transformation in business and technology is forcing IT decision makers to rethink their entire business, and, in particular, the data center. Yesterday's tools and knowledge are no longer sufficient and companies must find modern options that can help them move into the future. This book will help IT professionals learn how to create business value from new approaches such as software-defined storage (SDS) and hyperconverged infrastructure (HCI).

"A clearly written guide about the realities of SDS and hyperconvergence that every vAdmin and CIO should read before making tough decisions about their IT budgets and operational models."

- David Marshall, Founder and Editor, VMblog.com

"An invaluable resource for both IT veterans and newcomers alike..."

- Dan Frith, Blogger at PenguinPunk.net

"Insights that actually help IT departments deliver more value to their organizations."

- Stu Miniman, Senior Analyst and Principal Research Contributor at Wikibon.com

"Highlights the important factors; financial, operational and technical that must be considered in evolving the delivery of IT for next-generation architectures. A must-read for anyone looking to keep ahead of the technology curve."

- Chris Evans, Founder of Blog.Architecting.IT and Langton Blue, Ltd

"Does a phenomenal job diving into the critical elements around cost, risk, complexity, and even business resiliency."

- Bill Kleyman, CBS Interactive and InformationWeek, and VP of Strategy and Innovation at MTM Technologies

"A fantastic resource for practitioners and thought leaders of enterprise organizations."

- Dane Young, Virtualization Practice Manager at Entisys Solutions

"Will take you on the journey of the past, present, and future to begin the thought process shift that is absolutely necessary for all businesses."

- Theresa Miller, Founder of 24x7ITConnection.com, Co-host of Current Status

ISBN 978-1-943952-07-6
50999>
9 781943 952076

US $9.99

Published by ActualTech Media
In Partnership with Atlantis Computing