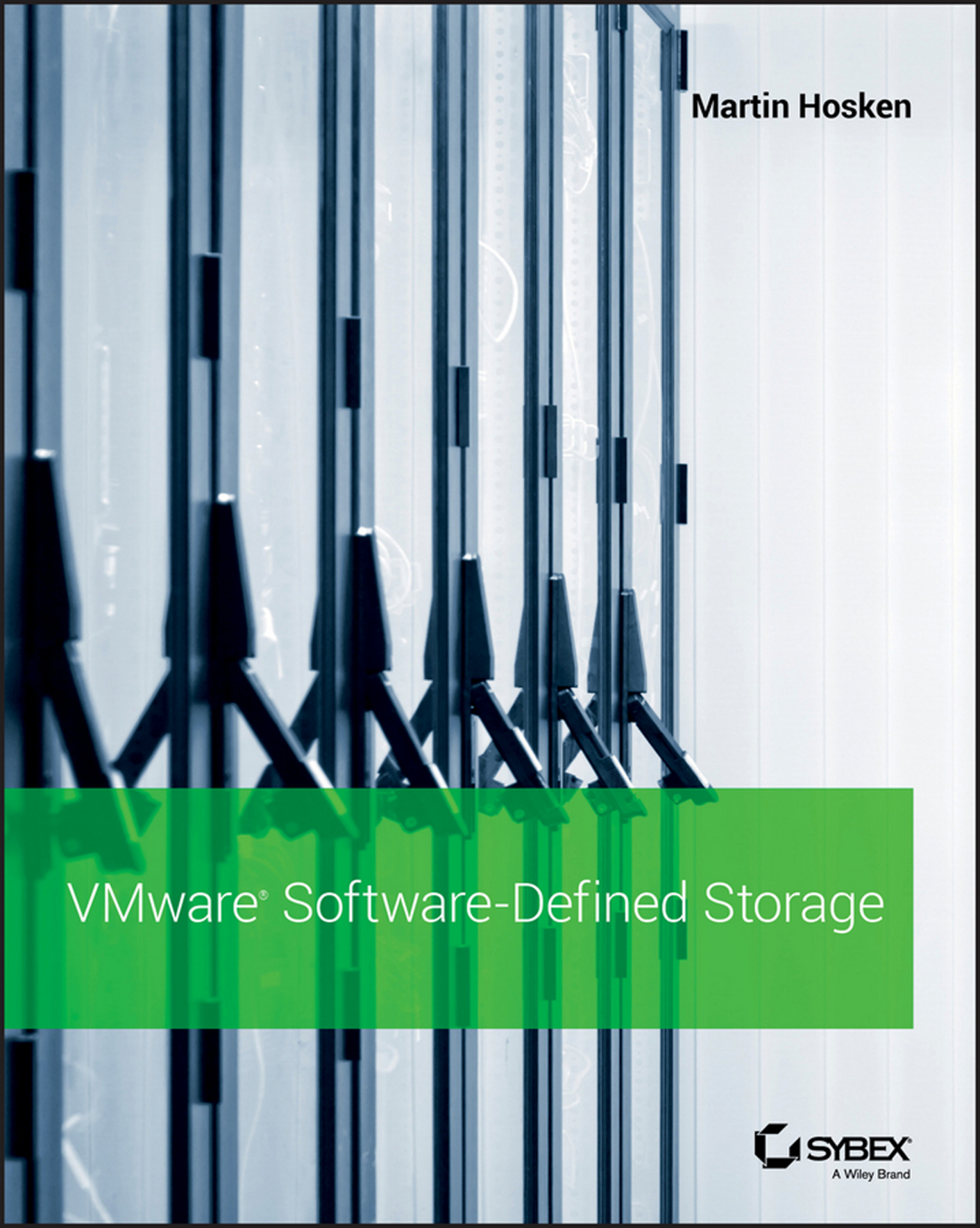
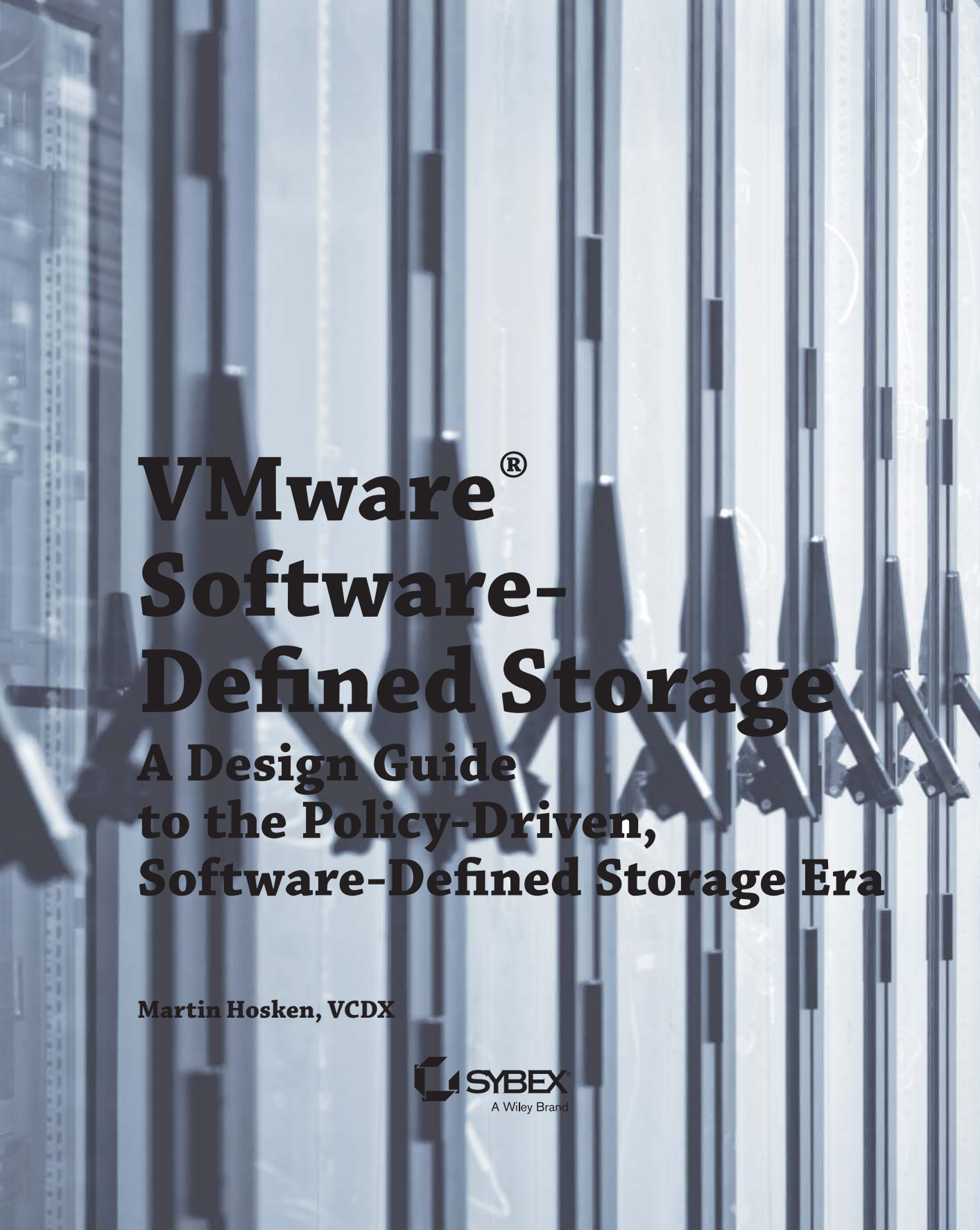


Martin Hosken

VMware® Software-Defined Storage

SYBEX
A Wiley Brand


VMware®
Software-
Defined Storage



VMware® Software-Defined Storage

A Design Guide to the Policy-Driven, Software-Defined Storage Era

Martin Hosken, VCDX

SYBEX
A Wiley Brand

Executive Editor: Jody Lefevere
Development Editor: David Clark
Technical Editor: Ray Heffer
Production Editor: Barath Kumar Rajasekaran
Copy Editor: Sharon Wilkey
Editorial Manager: Mary Beth Wakefield
Production Manager: Kathleen Wisor
Proofreader: Nancy Bell
Indexer: Nancy Guenther
Project Coordinator, Cover: Brent Savage
Cover Designer: Wiley
Cover Image: ©Mikhail hoboton Popov/Shutterstock

Published simultaneously in Canada

ISBN: 978-1-119-29277-7
ISBN: 978-1-119-29279-1 (ebk.)
ISBN: 978-1-119-29278-4 (ebk.)

Manufactured in the United States of America

Library of Congress Control Number: 2016944021

10 9 8 7 6 5 4 3 2 1

About the Author

Martin Hosken is employed as a global cloud architect within the VMware Global Cloud Practice, which is part of its Cloud Provider Software Business Unit.

He has extensive experience architecting and consulting with international customers and designing the transition of organizations' legacy infrastructure onto VMware cloud-based platforms. His broad and deep knowledge of physical and virtualized services, platforms, and cloud infrastructure solutions is based on involvement and leadership in the global architecture, design, development, and implementation of large-scale, complex, multitechnology projects for enterprises and cloud service providers. He is a specialist in designing, implementing, and integrating best-of-breed, fully redundant Cisco, EMC, IBM, HP, Dell, and VMware systems into enterprise environments and cloud service providers' infrastructure.

In addition, Martin is a double VMware Certified Design Expert (VCDX #117) in Data Center Virtualization and Cloud Management and Automation. (See the Official VCDX directory available at `http://vcdx.vmware.com`.) Martin also holds a range of industry certifications from other vendors such as EMC, Cisco, and Microsoft, including MCITP and MCSE in Windows Server and Messaging.

He has been awarded the annual *VMware vExpert* title for a number of years for his significant contribution to the community of VMware users. (See the VMware Community vExpert Directory available at `https://communities.vmware.com/vexpert.jspa`.) This title is awarded to individuals for their commitment to the sharing of knowledge and their passion for VMware technology beyond their job requirements. Martin is also a part of the CTO Ambassador Program, and as such is responsible for connecting the R&D team at VMware with customers, partners, and field employees.

Follow Martin on Twitter: `@hoskenm`.

About the Technical Reviewer

Ray Heffer is employed as a global cloud architect for VMware's Cloud Provider Software Business Unit. He is also a double VCDX #122 (Desktop and Datacenter). In his previous roles with End User Computing (EUC), Technical Marketing, and Professional Services at VMware, he has led many large-scale platform designs for service providers, manufacturing, and government organizations.

Since 1997 Ray has specialized in administering, designing, and implementing solutions ranging from Microsoft Exchange, Linux, Citrix, and VMware. He deployed his first VMware environment in 2004 while working at a hosting company in the United Kingdom.

Ray is also a regular presenter at VMworld and VMUG events, covering topics such as Linux desktops and VMware Horizon design best practices.

Contents at a Glance

Contents

Foreword by Duncan Epping

I had just completed the final chapter of the Virtual SAN book I was working on when Martin reached out and asked if I wanted to write a foreword for his book. You can imagine I was surprised to find out that there was another person writing a book on software-defined storage, and pleasantly surprised to find out that VSAN is one of the major topics in this book. Not just surprised, but also very pleased. The world is changing rapidly, and administrators and architects need guidance along this journey, the journey toward a software-defined data center.

When talking to customers and partners on the subject of the software-defined data center, a couple of concerns typically arise. Two parts of the data center have always been historically challenging and/or problematic—namely, networking and storage. Networking problems and concerns (and those related to security, for that matter) have been largely addressed with VMware NSX, which allows virtualization and networking administrators to work closely together on providing a flexible yet very secure foundation for the workloads they manage. This is done by adding an abstraction layer on top of the physical environment and moving specific services closer to the workloads (for instance, firewalling and routing), where they belong.

Over 30 years ago, RAID was invented, which allowed you to create logical devices formed out of multiple hard disk drives. This allowed for more capacity, higher availability, and of course, depending on the type of RAID used, better performance. It is fair to say, however, that the RAID construct was created as a result of the many constraints at the time. Over time, all of these constraints have been lifted, and the hardware evolution started the (software-defined) storage revolution. SSDs, PCIe-based flash, NVMe, 10GbE, 25GbE (and higher), RDMA, 12 Gbps SAS, and many other technologies allowed storage vendors to innovate again and to make life simpler. No longer do we need to wide-stripe across many disks to meet performance expectations, as that single SSD device can now easily serve 50,000 IOPS. And although some of the abstraction layers, such as traditional RAID or disk groups, may have been removed, most storage systems today are not what I would consider admin/user friendly.

There are different protocols (iSCSI, FCoE, NFS, FC), different storage systems (spindles, hybrid, all flash), and many different data services and capabilities these systems provide. As a result, we cannot simply place an abstraction layer on top as we have done for networking with NSX. We still need to abstract the resources in some shape or form and most definitely present them in a different, simpler manner. Preferably, we leverage a common framework across the different types of solutions, whether that is a hyper-converged software solution like Virtual SAN or a more traditional iSCSI-based storage system with a combination of flash and spindles.

Storage policy–based management is this framework. If there is anything you need to take away from this book, then it is where your journey to software-defined storage should start, and that is the SPBM framework that comes as part of vSphere. SPBM is that abstraction layer that allows you to consume storage resources across many different types of storage (with different protocols) in a simple and uniform way by allowing you to create policies that are passed down to the respective storage system through the VMware APIs for Storage Awareness.

In order to be able to create an infrastructure that caters to the needs of your customers (application owners/users), it is essential that you, the administrator or architect, have a good understanding of all the capabilities of the different storage platforms, the requirements of the application, and how architectural decisions can impact availability, recoverability, and performance of your workloads.

But before you even get there, this book will provide you with a good foundational understanding of storage concepts including thin LUNs, protocols, RAID, and much more. This will be quickly followed by the software-defined storage options available in a VMware-based infrastructure, with a big focus on Virtual Volumes and Virtual SAN.

Many have written on the subject of software-defined storage, but not many are as qualified as Martin. Martin is one of the few folks who have managed to accrue two VCDX certifications, and as a global cloud architect has a wealth of experience in this field. He is going to take you on a journey through the world of software-defined storage in a VMware-based infrastructure and teach you the art of architecture along the way.

I hope you will enjoy reading this book as much as I have.

Duncan Epping
Chief Technologist, Storage and Availability, VMware

Introduction

Storage is typically the most important element of any virtual data center. It is the key component in system performance, availability, scalability, and manageability. It has also traditionally been the most expensive component from a capital and operational cost perspective.

The storage infrastructure must meet not only today's requirements, but also the business needs for years to come, because of the capital expenditure costs historically associated with the hardware. Storage and vSphere architects must therefore make the most informed choices possible, designing solutions that take into account multiple complex and contradictory business requirements, technical goals, forecasted data growth, constraints, and of course, budget.

In order for you to be confident about undertaking a vSphere storage design that can meet the needs of a whole range of business and organization types, you must understand the capabilities of the platform. Designing a solution that can meet the requirements and constraints set out by the customer requires calling on your experience and knowledge, as well as keeping up with advances in the IT industry. A successful design entails collecting information, correlating it into a solid design approach, and understanding the design trade-offs and design decisions.

The primary content of this book addresses various aspects of the VMware vSphere software-defined storage model, which includes separate components. Before you continue reading, you should ensure that you are already well acquainted with the core vSphere products, such as VMware vCenter Server and ESXi, the type 1 hypervisor on which the infrastructure's virtual machines and guest operating systems reside.

It is also assumed that you have a good understanding of shared storage technologies and networking, along with the wider infrastructure required to support the virtual environment, such as physical switches, firewalls, server hardware, array hardware, and the protocols associated with this type of equipment, which include, but are not limited to, Fibre Channel, iSCSI, NFS, Ethernet, and FCoE.

Who Should Read This Book?

This book will be most useful to infrastructure architects and consultants involved in designing new vSphere environments, and administrators charged with maintaining existing vSphere deployments who want to further optimize their infrastructure or gain additional knowledge about storage design. In addition, this book will be helpful for anyone with a VCA, VCP, or a good foundational knowledge who wants an in-depth understanding of the design process for new vSphere storage architectures. Prospective VCAP, VCIX, or VCDX candidates who already have a range of vSphere expertise but are searching for that extra bit of detailed knowledge will also benefit.

What Is Covered in This Book?

VMware-based storage infrastructure has changed a lot in recent years, with new technologies and new storage vendors stepping all over the established industry giants, such as EMC, IBM, and NetApp. However, life-cycle management of the storage platform remains an ongoing challenge for enterprise IT organizations and service providers, with hardware renewals occurring on an ongoing basis for many of VMware's global customer base.

This book aims to help vSphere architects, storage architects, and administrators alike understand and design for this new generation of VMware-focused software-defined storage, and to drive efficiency through simple, less complex technologies that do not require large numbers of highly trained storage administrators to maintain.

In addition, this book aims to help you understand the design factors associated with these new vSphere storage options. You will see how VMware is addressing these data-center challenges through its software-defined storage offerings, Virtual SAN and Virtual Volumes, as well as developing cloud automation approaches to these next-generation storage solutions to further simplify operations.

This book offers you deep knowledge and understanding of these new storage solutions by

- Providing unique insight into Virtual SAN and Virtual Volumes storage technologies and design
- Providing a detailed knowledge transfer of these technologies and an understanding of the design factors associated with the architecture of this next generation of VMware-based storage platform
- Providing guidance over delivering storage as a service (STaaS) and enabling enterprise IT organizations and service providers to deploy and maintain storage resources via a fully automated cloud platform
- Providing detailed and unique guidance in the design and implementation of a stretched Virtual SAN architecture, including an example solution
- Providing a detailed knowledge transfer of legacy storage and protocol concepts, in order to help provide context to the VMware software-defined storage model

Finally, in writing this book, I hope to help you understand all of the design factors associated with these new vSphere storage options, and to provide a complete guide for solution architects and operational teams to maximize quality storage design for this new generation of technologies.

The following provides a brief summary of the content in each of the 10 chapters:

Chapter 1: Software-Defined Storage Design This chapter provides an overview of where vSphere storage technology is today, and how we've reached this point. This chapter also introduces software-defined storage, the economics of storage resources, and enabling storage as a service.

Chapter 2: Classic Storage Models and Constructs This chapter covers the legacy and classic storage technologies that have been used in the VMware infrastructure for the last decade. This chapter provides the background required for you to understand the focus of this book, VMware vSphere's next-generation storage technology design.

Chapter 3: Fabric Connectivity and Storage I/O Architecture This chapter presents storage connectivity and fabric architecture, which is relevant for legacy storage technologies as well as next-generation solutions including Virtual Volumes.

Chapter 4: Policy-Driven Storage Design with Virtual SAN This chapter addresses all of the design considerations associated with VMware's Virtual SAN storage technology. The chapter provides detailed coverage of Virtual SAN functionality, design factors, and architectural considerations.

Chapter 5: Virtual SAN Stretched Cluster Design This chapter focuses on one type of Virtual SAN solution, stretched cluster design. This type of solution has specific design and implementation considerations that are addressed in depth. This chapter also provides an example Virtual SAN stretched architecture design as a reference.

Chapter 6: Designing for Web-Scale Virtual SAN Platforms This chapter addresses specific considerations associated with large-scale deployments of Virtual SAN hyper-converged infrastructure, commonly referred to as *web-scale*.

Chapter 7 Virtual SAN Use Case Library This chapter provides an overview of Virtual SAN use cases. It also provides a detailed solution architecture for a cloud management platform that you can use as a reference.

Chapter 8: Policy-Driven Storage Design with Virtual Volumes This chapter provides detailed coverage of VMware's Virtual Volumes technology and its associated policy-driven storage concepts. This chapter also provides a low-level knowledge transfer, as well as addressing in detail the design factors and architectural concepts associated with implementing Virtual Volumes.

Chapter 9: Delivering a Storage-as-a-Service Design This chapter explains how IT organizations and service providers can design and deliver storage as a service in a cloud-enabled data center by using VMware's cloud management platform technologies.

Chapter 10: Monitoring and Storage Operations Design To ensure that a storage design can deliver an operationally efficient storage platform end to end, this final chapter covers storage monitoring and alerting design in the software-defined storage data center.

Chapter 1

Software-Defined Storage Design

VMware is the global leader in providing virtualization solutions. The VMware ESXi software provides a hypervisor platform that abstracts CPU, memory, and storage resources to run multiple virtual machines concurrently on the same physical server.

To successfully design a virtual infrastructure, other products are required in addition to the hypervisor, in order to manage, monitor, automate, and secure the environment. Fortunately, VMware also provides many of the products required to design an end-to-end solution, and to develop an infrastructure that is software driven, as opposed to hardware driven. This is commonly described as the *software-defined data center* (SDDC), illustrated in Figure 1.1.

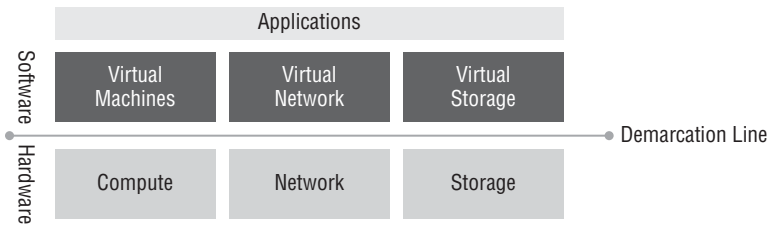


Figure 1.1 Software-defined data center conceptual model

The SDDC is not a single product sold by VMware or anyone else. It is an approach whereby management and orchestration tools are configured to manage, monitor, and operationalize the entire infrastructure. This might include products such as vSphere, NSX, vRealize Automation, vRealize Operations Manager, and Virtual SAN from VMware, but it could also include solutions such as VMware Integrated OpenStack, CloudStack, or any custom cloud-management solution that can deliver the required platform management and orchestration capabilities.

The primary aim of the SDDC is to decouple the infrastructure from its underlying hardware, in order to allow software to take advantage of the physical network, server, and storage. This makes the SDDC location-independent, and as such, it may be housed in a single physical data center, span multiple private data centers, or even extend into hybrid and public cloud facilities.

From the end user's perspective, applications that are delivered from an SDDC are consumed in exactly the same way as they otherwise would be—through mobile, desktop, and virtual desktop interfaces—from anywhere, any time, with any device.

However, with the SDDC infrastructure decoupled from the physical hardware, the operational model of a virtual machine—with on-demand provisioning, isolation, mobility, speed, and agility—can be replicated for the entire data-center environment (including networking and storage), with complete visibility, security, and scale.

The overall aim is that an SDDC can be achieved with the customer's existing physical infrastructure, and also provide the flexibility for added capacity and new deployments.

Software-Defined Compute

In this book, *software-defined compute* refers to the compute virtualization of the *x*86 architecture. What is *virtualization*? If you don't know the answer to this question, you're probably reading the wrong book, but in any case, let's make sure we're on the same page.

In the IT industry, the term *virtualization* can refer to various technologies. However, from a VMware perspective, virtualization is the technique used for abstracting the physical hardware away from the operating system. This technique allows multiple guest operating systems (logical servers or desktops) to run concurrently on a single physical server. This allows these logical servers to become a portable virtual compute resource, called *virtual machines*. Each virtual machine runs its own guest operating system and applications in an isolated manner.

Compute virtualization is achieved by a hypervisor layer, which exists between the hardware of the physical server and the virtual machines. The hypervisor is used to provide hardware resources, such as CPU, memory, and network to all the virtual machines running on that physical host. A physical server can run numerous virtual machines, depending on the hardware resources available.

Although a virtual machine is a logical entity, to its operating system and end users, it seems like a physical host with its own CPU, memory, network controller, and disks. However, all virtual machines running on a host share the same underlying physical hardware, but each taking its own share in an isolated manner. From the hypervisor's perspective, each virtual machine is simply a discrete set of files, which include a configuration file, virtual disk files, log files, and so on.

It is VMware's ESXi software that provides the hypervisor platform, which is designed from the ground up to run multiple virtual machines concurrently, on the same physical server hardware.

Software-Defined Networking

Traditional physical network architectures can no longer scale sufficiently to meet the requirements of large enterprises and cloud service providers. This has come about as the daily operational management of networks is typically the most time-consuming aspect in the process of provisioning new virtual workloads. *Software-defined networking* helps to overcome this problem by providing networking to virtual environments, which allows network administrators to manage network services through an abstracted higher-level functionality.

As with all of the components that make up the SDDC model, the primary aim is to provide a simplified and more efficient mechanism to operationalize the virtual data-center platform. Through the use of software-defined networking, the majority of the time spent provisioning and configuring individual network components in the infrastructure can be performed programmatically, in a virtualized network environment. This approach allows network administrators to get around this inflexibility of having to pre-provision and configure physical networks, which has proved to be a major constraint to the development of cloud platforms.

In a software-defined networking architecture, the control and data planes are decoupled from one another, and the underlying physical network infrastructure is abstracted from

the applications. As a result, enterprises and cloud service providers obtain unprecedented programmability, automation, and network control. This enables them to build highly scalable, flexible networks with cloud agility, which can easily adapt to changing business needs by

- Providing centralized management and control of networking devices from multiple vendors.
- Improving automation and management agility by employing common application program interfaces (APIs) to abstract the underlying networking from the orchestration and provisioning processes, without the need to configure individual devices.
- Increasing network reliability and security as a result of centralized and automated management of the network devices, which provides this unified security policy enforcement model, which in turn reduces configuration errors.
- Providing more-granular network control, with the ability to apply a wide range of policies at the session, user, device, or application level.

NSX is VMware's software-defined networking platform, which enables this approach to be taken through an integrated stack of technologies. These include the NSX Controller, NSX vSwitch, NSX API, vCenter Server, and NSX Manager. By using these components, NSX can create layer 2 logical switches, which are associated with logical routers, both north/south and east/west firewalling, load balancers, security policies, VPNs, and much more.

Software-Defined Storage

Where the data lives! That is the description used by the marketing department of a large financial services organization that I worked at several years ago. The marketing team regularly used this term in an endearing way when trying to describe the business-critical storage systems that maintained customer data, its availability, performance level, and compliance status.

Since then, we have seen a monumental shift in the technologies available to vSphere for virtual machine and application storage, with more and more storage vendors trying to catch up, and for some, steam ahead. The way modern data centers operate to store data has been changing, and this is set to continue over the coming years with the continuing shift toward the next-generation data center, and what is commonly described as *software-defined storage*.

VMware has undoubtedly brought about massive change to enterprise IT organizations and service-provider data centers across the world, and has also significantly improved the operational management and fundamental economics of running IT infrastructure. However, as application workloads have become more demanding, storage devices have failed to keep up with IT organizations' requirements for far more flexibility from their storage solutions, with greater scalability, performance, and availability. These design challenges have become an everyday conversation for operational teams and IT managers.

The primary challenge is that many of the most common storage systems we see in data centers all over the world are based on outdated technology, are complex to manage, and are highly proprietary. This ties organizations into long-term support deals with hardware vendors.

This approach is not how the biggest cloud providers have become so successful at scaling their storage operations. The likes of Amazon, Microsoft, and Google have scaled their cloud storage platforms by trading their traditional storage systems for low-cost commodity hardware,

and employed the use of powerful software around it to achieve their goals, such as availability, data protection, operational simplification, and performance. With this approach, and through the economies of scale, these large public cloud providers have achieved their supremacy at a significantly lower cost than deploying traditional monolithic centralized storage systems. This methodology, known as web-scale, is addressed further in Chapter 6, "Designing for Web-Scale Virtual SAN Platforms (10,000 VMS+)."

The aim of this book is to help you understand the new vSphere storage options, and how VMware is addressing these data-center challenges through its software-defined storage offerings, Virtual SAN and Virtual Volumes. The primary aim of these two next-generation storage solutions is to drive efficiency through simple, less complex technologies that do not require large numbers of highly trained storage administrators to maintain. It is these software-defined data-center concepts that are going to completely transform all aspects of vSphere data-center storage, allowing these hypervisor-driven concepts to bind together the compute, networking, and software-defined storage layers.

The goal of software-defined storage is to separate the physical storage hardware from the logic that determines *where the data lives*, and what storage services are applied to the virtual machines and data during read and write operations.

As a result of VMware's next-generation storage offerings, a storage layer can be achieved that is more flexible and that can easily be adjusted based on changing application requirements. In addition, the aim is to move away from complex proprietary vendor systems, to a virtual data center made up of a coherent data fabric that provides full visibility of each virtual machine through a single management toolset, the so-called *single pane of glass*. These features, along with lowered costs, automation, and application-centric services, are the primary drivers for enterprise IT organizations and cloud service providers to begin to rethink their entire storage architectural approach.

The next point to address is what software-defined storage isn't, as it can sometimes be hard to wade through all the marketing hype typically generated by storage vendors. Just because a hardware vendor sells or bundles management software with their products, doesn't make it a software-defined solution. Likewise, a data center full of different storage systems from a multitude of vendors, managed by a single common software platform, does not equate to a software-defined storage solution. As each of the underlining storage systems still has its legacy constructs, such as disk pools and LUNs, this is referred to as a *federated storage solution* and not software-defined. These two approaches are sometimes confused by storage vendors, as understandably, manufacturers always want to use the latest buzzwords in their marketing material.

Despite everything that has been said up until now, software-defined storage isn't just about software. At some point, you have to consider the underlying disk system that provides the storage capacity and performance. If you go out and purchase a lot of preused 5,400 RPM hard drives from eBay, you can't then expect solid-state flash-like performance just because you've put a smart layer of software on top of it.

Designing VMware Storage Environments

Gathering requirements and documenting driving factors is a key objective for you, the architect. Understanding the customer's business objectives, challenges, and requirements should always be the first task you undertake, before any design can be produced. From this activity, you can translate the outcomes into design factors, requirements, constraints, risks, and assumptions, which are all critical to the success of the vSphere storage design.

Architects use many approaches and methodologies to provide customers with a meaningful design that meets their current and future needs. Figure 1.2 illustrates one such method, which provides an elastic sequence of activities that can typically fulfill all stages of the design process. However, many organizations have their own approach, which may dictate this process and mandate specific deliverables and project methodologies.

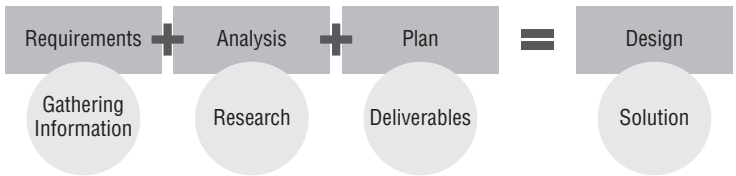


Figure 1.2
Example of a design sequence methodology

Technical Assessment and Requirements Gathering

The first step toward any design engagement is discovery, and the process of gathering the requirements for the environment in which the vSphere-based storage will be deployed. Many practices are available for gathering requirements, with each having value in different customer scenarios. As the architect, you must use the best technique to gain a complete picture from various stakeholders. This may include one-to-one meetings with IT organizational leaders and sponsors, facilitated sessions or workshops with the team responsible for managing the storage operations, and review of existing documents. Table 1.1 lists key questions that you need to ask stakeholder and operational teams.

Table 1.1: Requirements gathering

Architect Question	Architectural Objective
What will it be used for?	Focus on applications and systems
Who will be using it?	Users and stakeholders
What is the purpose?	Objectives and goals
What will it do? When? How?	Help create a scenario
What if something goes wrong with it?	Availability and recoverability
What quality? How fast? How reliable? How secure? How many?	Scaling, security, and performance

After all design factors and business drivers have been reviewed and analyzed, it is essential to take into account the integration of all components into the design, before beginning the qualification effort needed to sort through the available products and determine which solution will meet the customer's objectives. The integration of all components within a design can take place only if factors such as data architecture, business drivers, application architecture, and technologies are put together.

The overall aim of all the questions is to quantify the objectives and business goals. For instance, these objectives and goals might include the following:

Performance User numbers and application demands: Does the organization wish to implement a storage environment capable of handling an increase in user numbers and application storage demands, without sacrificing end-user experience?

Total Cost of Ownership Does the organization wish to provide separate business units with a storage environment that provides significant cost relief?

Scalability Does the organization wish to ensure capability and sustainability of the storage infrastructure for business continuity and future growth?

Management Does the organization wish to provide a solution that simplifies the management of storage resources, and therefore requires improved tools to support this new approach?

Business Continuity and Disaster Recovery Does the organization wish to provide a solution that can facilitate high levels of availability, disaster avoidance, and quick and reliable recovery from incidents?

In addition to focusing on these goals, you need to collect information relating to the existing infrastructure and any new technical requirements that might exist. These technical requirements will come about as a result of the business objectives and the current state analysis of the environment. However, these are likely to include the following:

- Application classification
- Physical and virtual network constraints
- Host server options
- Virtual machines and workload deployment methodology
- Network-attached storage (NAS) systems
- Storage area network (SAN) systems

Understanding the customer's business goals is critical, but what makes it such a challenge is that no two projects are ever the same. Whether it is different hardware, operating systems, maintenance levels, physical or virtual servers, or number of volumes, the new design must be validated for each component within each customer's specific infrastructure. In addition, just as every environment is different, no two workloads are the same either. For instance, peak times can vary from site to site and from customer to customer. These individual differentiators must be validated one by one, in order to determine the configuration required to meet the customer's design objectives.

Establishing Storage Design Factors

Establishing storage design factors is key to any architecture. However, as previously stated, the elements will vary from one engagement to another. Nevertheless, and this is important, the design should focus on the business drivers and design factors, and not the product features or latest technology specification from the customer's preferred storage hardware vendor.

A customer-preferred storage device could well be the best product ever, but may not align with the customer use cases, regardless of what they're being told by their supplier. Therefore, creating an architecture that focuses on the hardware specification and not the business goals is likely to introduce significant risks and ultimately fail as a design.

Although the business drivers and design factors for each customer will be different, with all having their own priorities and goals that need to be factored into the design, you likely will see many common design qualities, illustrated in Figure 1.3, time and time again.

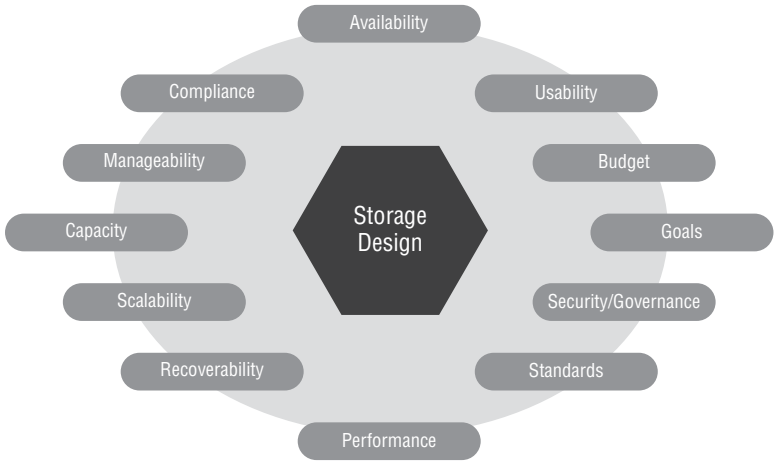


Figure 1.3
Storage architecture business drivers and design factors

Availability

The *availability* of the storage infrastructure is typically dictated by a service-level agreement (SLA) of some sort, and is often represented as a percentage of possible uptime (such as four nines, 99.99 percent). Availability is achieved through techniques such as redundant hardware, RAID technologies, array mirroring, or eliminating single points of failure. Additionally, high levels of availability can be provided by using technologies such as storage replication, vSphere anti-affinity rules, or Virtual SAN Stretched Clusters. An available design is reliable and implements multiple mechanisms to restore services within the IT organization's agreed-upon service-level agreement.

Compliance

Compliance means conforming to a specification, policy, standard, or law. Regulatory compliance is now a part of everyday life for an information technology architect. Having a strong understanding of the requirements that the customers must comply with will help significantly in producing a design that meets the needs of the organization you're working with. Compliance goals also differ for different countries. For instance, in the United States, architects may be familiar with the Sarbanes–Oxley Act of 2002 or the Health Insurance Portability and Accountability Act of 1996 (HIPAA). In addition, global compliance standards, such as the Payment Card Industry Data Security Standard (PCI DSS), cross geographical boundaries.

Usability

Usability is the ease of use and learnability of the day-to-day operations associated with the storage platform. As the architect, one of your tasks will be to ensure that the customer's operational team or administrators are able to manage the environment after you leave and move on to the next project. This, of course, links into manageability, and you may be required to provide operational documentation, or partake in knowledge transfer and training as defined in the scope of work.

Budget

Unfortunately, few projects have unlimited budgets. Cost is always at the forefront of stakeholders' minds, and, as the architect, you will probably find that justifying costs associated with the design will often come down to you. I can assure you from personal experience that CFOs and their representatives can be scary and love to ask difficult and challenging questions. (To be fair, all they are trying to do is justify costs, so let's not be too hard on them.) Your goal is to meet the organization's business needs, while remaining within budget. If this is not possible, you must be able to explain and justify the best course of action to the organization's key stakeholders, who hold the purse strings.

The budget will depend on multiple factors. It might be too small a number, and you can think of it as a design constraint. In an ideal world, the design should focus only on system readiness, performance, and capacity, with an aim to provide a world-class solution with the future in mind, regardless of the cost. However, this is rarely the case; typically, the task of an architect is to take in all of the requirements and provide the best solution with the lowest conceivable budget. Even if, as the architect, you are not accountable for the financial aspects of the design, it's typically useful to have an understanding of budgetary constraints and to be able to demonstrate value for money, as and when required.

Manageability

For this design factor, you should keep in mind KISS: *keep it standardized and simple*. Making a design unnecessarily complex has a serious impact on the manageability of the environment. Also, a design that is unnecessarily complex can easily contribute to failure, because the operational team might not understand the design, and making a change to one component can have implications on another. Instead, your aim should be to keep the design as simple as possible, while still meeting the business goals. The objective should be to keep the design easy to deploy, easy to administer and maintain for the operational teams, and easy to update and upgrade when the time comes.

Goals

The key goals for the design will be different for each project . However, in general, a good design is not unnecessarily complex, provides detailed documentation (which includes rationales for design decisions), balances the organization's requirements with technical best

practices, and involves key stakeholders and the customer's subject matter experts in every aspect of the design, delivery, testing, and hand-over of the storage platform.

Security and Governance

Needless to say, in today's world security is a key deliverable in every enterprise IT or cloud service provider project. On some of the projects involving government agencies and financial institutions that I've worked on almost every aspect of the design is governed by security considerations and requirements. This can have a significant impact on both operational considerations and budget.

Standards

An enterprise organization or cloud service provider typically has standards that must be met for every project. Hopefully, these standards include a clear methodology for identifying stakeholders, identifying the most relevant business drivers, and providing transparency and traceability for all decisions. Standards might also include a defined and repeatable approach to design, delivery, testing and verification, and hand-over to operational teams.

Performance

Like availability, performance is often governed by a service-level agreement. The design must meet the performance requirements set out by the customer. Performance is typically measured by achievable throughput, latency, I/O per second, or other defined metrics the customer deems appropriate. Storage performance is probably less understood than capacity or availability. However, in a virtualized infrastructure, not much has a greater impact on the overall performance of the environment than the storage platform.

Recoverability

Like availability and performance, recoverability is typically governed by a service-level agreement. The design should document how the infrastructure can be recovered from any kind of outage. Typically, two metrics are used to define recoverability: *recovery time objective* (RTO), which is the amount of time it takes to restore the service after the disruption began; and *recovery point objective* (RPO), which is the point in time at which data must be recovered to, after the disruption began.

Scalability

The design should be scalable—able to grow as the customer's data requirements change and the storage platform is required to expand. As part of the project, it is important to determine the business growth plans for data capacity, and any future performance requirements. This information is typically provided as a percentage of growth per year, and the design should take these factors into account. Later we address a building-block approach to storage design, but for

now, it's of key importance that the customer is able to provide clear expectations on the growth of their environment, as this will almost certainly impact the design.

Capacity

The design's capacity requirements can typically be achieved as a business grows or shrinks. Capacity is generally predictable and can be provisioned on demand, as it is typically a relatively easy procedure to add disks and/or enclosures to most storage arrays or hosts without experiencing downtime. As a result, capacity can be managed relatively easily, but it is still an important aspect of storage design.

The Economics of Storage

At first glance, storage technologies, much like compute resource, should be priced based on a commodity hardware model; however, this is typically not the case. As illustrated in Figure 1.4, each year the cost of raw physical disk storage, on a per gigabyte basis, continues to fall, and has being doing so since the mid-1980s.

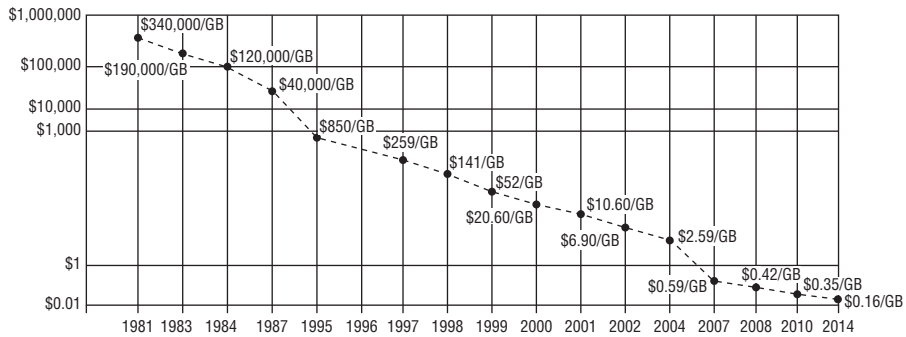


Figure 1.4 Hard disk drive cost per gigabyte

Alongside the falling cost of storage, as you might expect, in terms of raw disk capacity per drive, this has aligned with the falling cost per gigabyte charged by cloud service providers. This is illustrated in Figure 1.5, where the increasing capacity available on physical disks pretty much aligns with that falling cost.

Despite these falling costs in raw disk storage capacity, the chassis, the disk shelves used to create disk arrays, and the storage controllers tasked with organizing disks into large RAID (redundant array of independent disks) or JBOD (just a bunch of disks) sets, vendor prices for their technologies continue to increase year after year, regardless of this growing commoditization of the components used by them.

The reason for this is the ongoing development and sophistication of vendor software. For instance, an array made up of commoditized components, including 300 2 TB disks stacked in commodity shelves, may have a hardware cost totaling approximately $4,000. However, the end array vendor might assign a manufacturer's suggested retail price tag of $400,000. This price is based on the vendor adding their *secret source* software, enabling the commodity hardware to include features such as manageability and availability and to provide the performance aspects required by its customers, while also allowing the vendor to differentiate their product from that of their competitors. It is this aspect of storage that often adds the most significant cost

component to storage technologies, regardless of the actual value added by the vendor's software, or which of those added features are actually used by their customers.

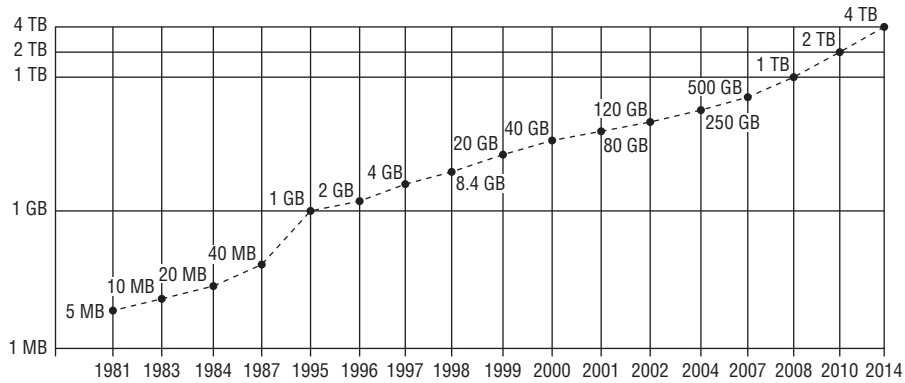


Figure 1.5 Hard disk drive capacity improvements

So whether you are buying or leasing, storage costs and other factors all contribute to the acquisition of storage resources, which is why IT organizations are increasingly trying to extend the useful life expectancy of their storage hardware. A decade ago, IT organizations were purchasing hardware with an expected life expectancy of three years. Today the same IT organizations are routinely acquiring hardware with the aim of achieving a five-to-seven-year useful life expectancy. One of the challenges is that most hardware and software ships with a three-year support contract and warranty, and renewing that agreement when it reaches end-of-life can sometimes cost as much as purchasing an entirely new array.

The next significant aspect of storage ownership to consider is that hardware acquisition accounts for approximately only one-fifth of the estimated annual total cost of ownership (TCO). This clearly outweighs the cost to acquire or capital expenditures (CapEx), and makes operational and management costs (OpEx) a far greater factor than many IT organizations account for in their initial design and planning cost estimations.

Calculating the Total Cost of Ownership for Storage Resources

As illustrated in Figure 1.6, the operational management, disaster recovery, and environmental costs are the real drivers behind the total cost of ownership calculations for storage devices.

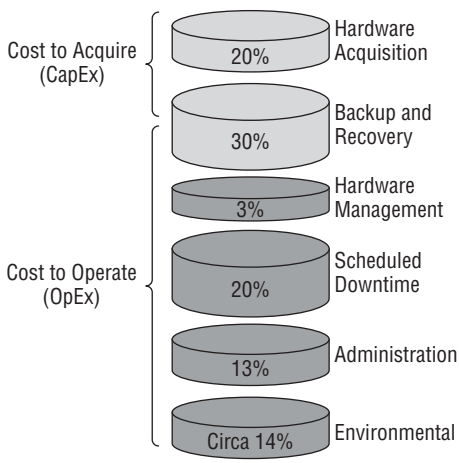


Figure 1.6 Breakdown of total cost of ownership of storage hardware

One of the factors that contributes to these operational costs is the heterogeneity of enterprise storage infrastructure. This significantly increases the challenges associated with providing a unified management approach, and as such, increases costs. Some IT organizations use this as a driver for the replacement of their heterogeneous storage platform, in favor of a more homogeneous approach. But typically, the replacement environment results from a deliberate attempt to procure the latest, best-of-breed technologies, or an attempt to facilitate storage tiering through a combination of hardware from a variety of vendors. Storage vendors often are unable to offer a varying portfolio of products for different types of workloads and data use cases. Furthermore, this problem is exacerbated by vendors who offer a wide range of products but can't typically offer a common management platform across all storage offerings. This is especially true when vendors have acquired the technology through a business acquisition.

The simplified formula in Figure 1.7 can be used to estimate the annual total cost of ownership of storage resources over the hardware's life expectancy.

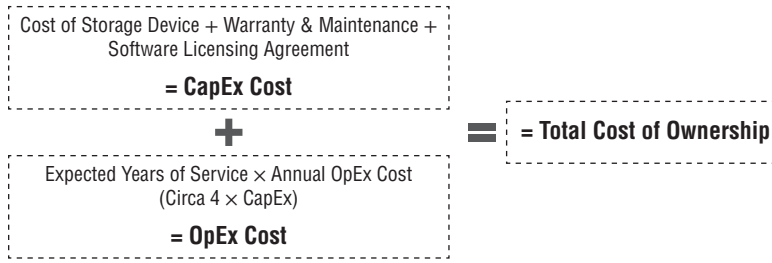


Figure 1.7 Simplified annual total cost of ownership

The next aspect of calculating storage costs relates to how efficiently storage capacity is allocated to the appropriate storage tier. *Utilization efficiency*, provides a measure of how effectively storage capacity is allocated to the correct storage type, based on factors such as frequency of access, availability requirements of the data, or required response time.

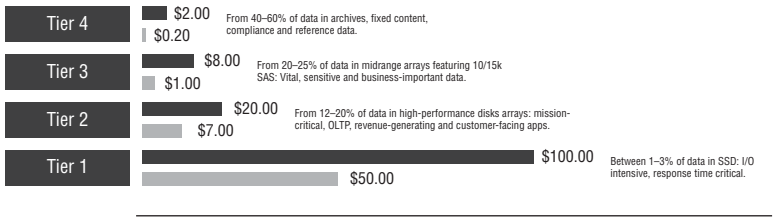
IT organizations often do not use storage capacity efficiently. They often use *tier-1* storage to host data for workloads and applications that do not require the expensive, high-performance attributes that the hardware is capable of delivering. Tiered storage is supposed to enable the placement of data based on cost-appropriate requirements for performance and capacity, as defined by the business. However, a growing movement to *flatten* storage via strategies such as those offered up by Hadoop (among others) in leveraging the *hyper-converged* storage model is, in itself, eliminating the requirement for tiered storage altogether.

IT organizations are typically charged with identifying the tiers of storage, the storage technologies employed, and the optimal percentage of business data that should represent each category of storage. Failing to do so undoubtedly leads to a significant increase in the per gigabyte cost of storage, which in turn results in an inflated total cost of ownership for the storage platform.

Figure 1.8 shows a tiered storage example. A business's IT organization uses this cost per gigabyte model to determine cost-appropriate storage for a specific workload type. For instance, if the IT organization requires a 100 TB storage estate, employing only two tiers of storage (tier 1 and tier 2 in this example), the total disk cost would be approximately $765,000. However, meeting the same storage requirements through the four tiers shown, segregated using the ratios

illustrated, would cost approximately $482,250, and therefore represent a savings of $282,750, or a 37 percent reduction to the original cost.[1]

Figure 1.8
Storage cost per gigabyte example



Features	Tier 1	Tier 2	Tier 3	Tier 4
Response Time SLA	3 ms	10 ms	20 ms	Best Effort (avg. 40 ms)
Availability SLA	99.999%	99.999%	99.999%	99.999%
Replication Availability	Yes	Yes	Yes	Yes
Use Cases	VDI, In-Mem DBs	High Transactional DBs, vHA HA	Standard DB & Application, OS	Low Activity (e.g., User Files) or Archive Data
Performance Characteristics	SSD	SSD + 15k HDD	15k or 10k HDD	SATA or NL-SAS
Approximate % of Storage Total Capacity	1–3%	12–20%	20–25%	40–60%

As you can see, an enterprise IT organization that fails to use this type of tiered storage strategy— which moves data across the storage estate based on its access frequency and other criteria—will suffer from poor utilization efficiency, as well as a significantly increased total cost of ownership of storage resources, within their storage platform.

Information Lifecycle Management

Information Lifecycle Management (ILM) is the primary approach used by businesses to ensure availability, capacity, and performance of data throughout its existence. When designing a storage solution for business systems, one of the key business requirements that you must understand is the ILM strategy being used by the customer for their business data.

Modern businesses and organizations must address the challenges associated with information management and its ever increasing growth, because business data and the way that it is used is playing a growing role in determining business success. For instance, companies such as Amazon and Rakuten are using their business data to gain strategic advantage over their competitors. The use of customer profiling and identifying what a customer may wish to purchase, based on their purchase history, provides a serious competitive advantage. In addition, understanding each customer's purchase habits (such as typically making all orders within the same few days each month, after payday) enables these businesses to target specific products at

[1] This calculation is based on 100 TB of storage, deployed at a cost of $50–$100 per gigabyte for flash (1–3 percent as tier 1), plus $7–$20 per gigabyte for fast disk (12–20 percent as tier1), plus $1 to $8 per gigabyte for capacity disk (20–25 percent as tier 3), plus $0.20–$2 per gigabyte for low-performance, high-capacity storage (40–60 percent as tier 4) totals approximately $482,250. Splitting the same capacity requirement between only tier 2 and tier 3 at the estimated cost range per gigabyte for each type of storage provides an estimated storage infrastructure cost of $765,000.

specific customers at a precise time, via a customized email based on the individual's purchase history and purchasing profile.

Another key consideration is how the value of the data changes over time. For instance, if a customer stops making purchases or closes their account, legislation might require that data to be deleted after a set period. Therefore, information that is stored may have a different value to the business, depending on its age. Understanding how an organization uses its data, and the value of its information throughout its life cycle, can be at the heart of storage design for many businesses (see Figure 1.9).

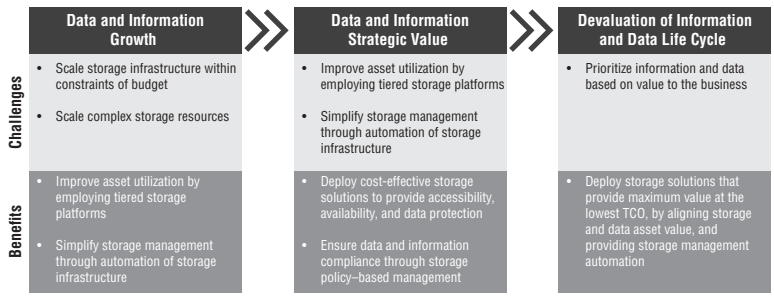


FIGURE 1.9 Information Lifecycle Management key challenges

It is also important to recognize that ILM is a *strategy* adopted by a business or organization, and not a product or service. This strategy must be proactive and dynamic, in order to help plan for storage system growth, and also must reflect the value of the information to the business.

Implementing an ILM strategy throughout a large organization can take a significant period of time, but can deliver key benefits that directly address business challenges and information management and utilization. The key design considerations that relate ILM strategy to the architecture of a storage platform include the following:

- Improving utilization by employing tiered storage platforms, and providing increased visibility into all enterprise information, alongside archiving capabilities
- Providing simplified storage management tools and increasing the use of automation for daily storage operational processes
- Implementing a wide range of backup, data protection, and recovery options to balance the need for business continuity with the cost of losing data
- Simplifying compliance and regulatory requirements by providing control over data placement, and knowing what data needs to be secured and for how long
- Lowering the total cost of ownership while continuing to meet the required service levels demanded by the business, and aligning the storage management costs with the value of the data, so that storage resources are not wasted, and unnecessarily complex environments are not introduced
- Providing a tiered storage solution that ensures that low-value data is not stored at the same cost per gigabyte as high-value data

Implementing a Software-Defined Storage Strategy

As a consequence of the ever-increasing cost of enterprise business storage, as outlined previously, more IT industry attention than ever before is focused on new storage architectures and technologies designed to drive down the total cost of ownership associated with storage. This approach aims to reduce both CapEx and OpEx costs by reducing hardware to its bare commodity components, and removing *secret source* software from the controllers, in favor of placing it onto a common storage software layer provided by either the hypervisor or a software-defined storage model.

In the past, several attempts have been made to develop a common management system that can transcend storage hardware and software vendors. For example, the Storage Networking Industry Association (SNIA) developed the Storage Management Initiative Specification (SMI-S), and the World Wide Web Consortium has Representational State Transfer (REST). However, these have seen only limited adoption by the storage industry. To achieve even limited interoperability and provide a sense of single point of management and support, the only real option for large enterprise IT organizations and cloud service providers has been to deploy homogeneous storage islands from a single hardware vendor in an attempt to manage operational overhead and therefore reduce OpEx costs.

The theory behind the software-defined storage model is to facilitate management across a common plane, by breaking down the barriers to interoperability that exist with proprietary vendor storage hardware. For most IT organizations, storage from different vendors, or even different models of storage array hardware from the same vendor, create isolated storage islands. It can be difficult to interoperate, share resources, or even manage across these islands from a single pane of glass.

The software-defined storage model aims to provide OpEx cost savings by driving efficient capacity utilization and platform management in a more agile way, typically by providing automation and a common management interface for all of the storage infrastructure. Therefore, the challenge for enterprise IT organizations and cloud service providers is to find the right software-defined storage solution, one that can apply the right centralized software services to the entire infrastructure by using simple, unified operational procedures within a common user interface.

The software-defined storage model also aims to reduce CapEx costs by moving away from proprietary storage hardware, and toward technology that facilitates unified management across all components of the storage infrastructure. When considering hardware solutions to deliver a software-defined storage-based environment, IT executives may be focused on reducing the total cost of ownership of storage resources. The following list provides a buyer's guide that IT organizations can use when working with their respective storage vendors to establish core storage requirements:

- Which storage solutions can work with the applications, hypervisors, and data that we currently have and are predicting to have going forward?
- Which storage solutions can enhance application performance?
- Which storage solutions best provide the required data availability?

- Which storage solutions can be deployed, configured, and managed quickly and effectively using currently available skills?
- Which storage solutions can provide greater, and if possible, optimal, storage capacity?
- Which storage solutions can best facilitate flexibility (provide the ability to add capacity or performance in the future without impacting the applications)?
- Which storage solutions provide automation and centralized management capabilities?
- Which storage technology will meet the preceding requirements within the available budget?

The approach often taken by IT organizations is to follow the lead of a trusted storage vendor. However, a key challenge for IT decision makers is to see beyond current trends in the industry and to arrive at a strategy that will provide a solution meeting not only today's storage requirements at an acceptable level of cost, but also next year's requirements for the various lines of business, and even the next decade's. This requires a subjective and clear-headed evaluation of the options, their costs, and the alternative approaches that could deliver the required storage functionality that optimizes both CapEx and OpEx budgets.

An additional challenge, which you also shouldn't overlook, is the complication associated with educating decision makers about the intricacies of storage technologies, in order to obtain budgetary approval. Enterprise IT executives rarely question the requirement to store and retain their ever-growing volume of business data. However, explaining the differences between various storage products, and their advantages and drawbacks, often requires a transfer of technical knowledge in order for the decision makers to grasp the concepts and challenges faced by the architect, and how they relate to their storage platform design.

When finances are stretched, as they so often are, a high storage infrastructure expenditure can significantly stand out on an IT executive's annual budget spreadsheet. By examining the storage environment and calculating the total cost of ownership of storage resources, IT organizations can seek to identify new and innovative ways to address CapEx and OpEx expenditures through the software-defined storage model, without compromising application performance, capacity, availability or other data-related services.

Software-Defined Storage Summary

Just as VMware introduced *x*86 server virtualization to improve the cost metrics and utilization efficiencies of the compute platform, so too can the software-defined storage model be used to make the most efficient use of storage infrastructure, thereby reducing the total cost of ownership through storage acquisition and operational cost savings.

In the software-defined storage data center, all storage—whether it is directly attached hyper-converged Virtual SAN, or is SAN attached and leveraging Virtual Volumes–enabled arrays—can be used as part of a storage resource pool. This eliminates the requirement to *rip and replace* all of the storage infrastructure in order to adopt a fully hyper-converged unified storage model as part of a single migration project, and allows the IT organization to spread the costs associated with a full storage infrastructure refresh over a number of years.

This is only one storage strategy. Equally valid is the mixed hybrid approach of employing Virtual Volumes and Virtual SAN as a long-term design, effectively using both solutions for specific use cases and workloads, as illustrated in Figure 1.10.

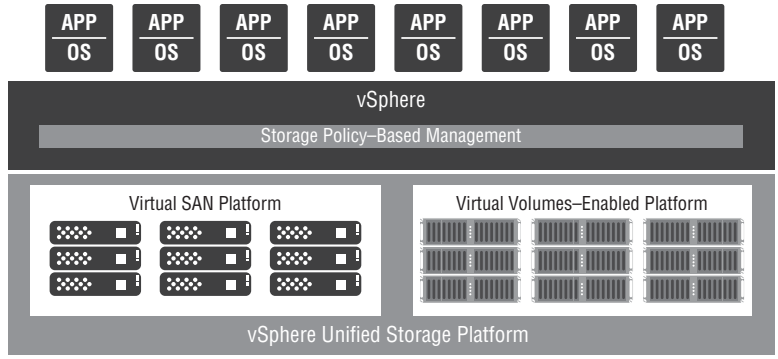


Figure 1.10 Hybrid Virtual Volumes and Virtual SAN platform

Just like the classic storage model, large enterprise customers and cloud service providers that are adopting software-defined storage typically should configure resources into pools. Each pool is composed of a different set of characteristics and services.

For instance, a Virtual SAN tier 1 pool may be optimized for performance and business-critical workloads, while a tier 0 pool may comprise all-flash disk groups and provide storage resources to specific I/O-intensive workloads. Following a similar model, high-capacity, low-cost, low-performance disks may be fashioned into a pool intended for the data that is infrequently accessed or updated. With this type of approach to storage provisioning, the software-defined storage model will continue to enable the implementation of a tiered storage strategy in order to provide improved capacity utilization and resource efficiency.

Furthermore, the implementation of a software-defined storage model allows technologies such as thin provisioning, compression, and de-duplication to be applied across an entire storage platform, rather than isolating these features behind specific hardware controllers. This helps to ensure that storage capacity can be used more efficiently, via a global storage policy.

These technologies can help slow the rate at which new capacity must be added to the infrastructure, and help ensure that where appropriate, less-expensive hardware can be deployed. In addition, centralizing this functionality through a single control plane enhances ease of administration, which in turn can also help reduce operational costs and the efforts associated with software maintenance.

The software-defined storage model is not an industry standard, and various approaches exist for the design, implementation, and function of the solution stack. Both VMware and independent software vendors (ISVs) have in recent years developed the concepts and product architecture of the software-defined storage platform for its integration into the market's leading hypervisor, to ensure that software-defined storage can operate within a robust and affordable model. These initiatives, which are the focus of much of this book, include the following:

- The introduction of the hyper-converged infrastructure product Virtual SAN, a bare-bones, hardware-agnostic model with a direct-attached storage configuration. This reduces or removes altogether the requirement for a switched fabric or LAN-attached storage infrastructure to manage, with no more proprietary storage hardware to support.
- The abstraction of advanced storage functions away from the storage vendor, and instead placed in the hypervisor software and management control plane. This approach

simplifies operations, with no more proprietary software licenses and firmware levels to manage, and enables storage services to be applied to all capacity, not just specific hardware.

- The introduction of a single storage service management plane, via a unified user interface. This removes the requirement for third-party tools and specific array element managers to monitor and administer a heterogeneous storage infrastructure.

All of these attributes provide a significant improvement over the ongoing challenges associated with classic storage infrastructures, although they do not address all the problems that make proprietary storage systems expensive to own and operate.

Hyper-Converged Infrastructure and Virtual SAN

The hyper-converged infrastructure (HCI) hardware architecture model uses the hypervisor to deliver compute, networking, and shared storage from a single *x*86 server platform. This software-driven architecture enables physical storage resources to become part of commodity *x*86 servers, enabling a building-block approach with a web-scale level of scalability. Also, by adopting this commodity *x*86 server hardware approach, and combining both storage and compute hardware into a single entity, IT organizations and cloud service provider data centers can operate with agility, on a highly scalable, cost-effective, fully converged platform.

Virtual SAN is VMware's HCI platform, which enables this approach to be taken through the VMware integrated stack of technologies. Virtual SAN aggregates local storage into a unified data plane, which virtual machines can then use. Virtual SAN also uses a fully integrated policy-driven management layer, which allows virtual machines to be managed centrally, through a policy-driven storage mechanism that is integrated into the virtual machines' own settings. These policies can define reliability, redundancy, and performance characteristics that must be obeyed, independently of all other virtual machines that may reside on the same storage platform.

Virtual SAN is the foundational component of VMware's hyper-converged infrastructure solution. This model allows the convergence of compute, storage, and networking onto a single integrated layer of software that can run on any commodity *x*86 infrastructure aligned with the requirements set out on VMware's hardware compatibility list (HCL). While vSphere abstracts and aggregates compute resources into logical pools, Virtual SAN, embedded into the hypervisor's VMkernel, can pool together server-attached disk devices to create a high-performance distributed datastore.

This approach can easily meet the storage requirements of the most demanding IT organization or cloud service provider, at a lower cost than legacy monolithic SAN or NAS storage devices. Virtual SAN also allows vSphere and vSphere storage administrators to ignore concepts such as RAID sets and LUNs, and instead focus on the specific storage needs of applications. In addition, Virtual SAN can simplify capacity planning by scaling both storage and compute concurrently, allowing for the nondisruptive addition of new nodes, without the purchase of costly storage frames or disk shelves. Virtual SAN is addressed in more detail in Chapters 4–7.

Virtual Volumes

While they are not part of an HCI architecture strategy, Virtual Volumes is nevertheless an important component in VMware's software-defined storage model. Virtual Volumes uses

shared storage devices in a new way, and transforms storage management by enabling full virtual machine awareness from the storage array. Based on a T10 industry standard, Virtual Volumes provides a unique level of integration between vSphere and third-party vendors' storage hardware, which significantly improves the efficiency and manageability of virtual workloads.

Virtual Volumes virtualizes shared SAN and NAS storage devices, which are then presented to vSphere hosts, providing logical pools of raw disk capacity, called a *virtual datastore*. Then, Virtual Volume objects, which represent virtual disks and other virtual machine entities, natively reside on the underlining storage, making the object, or virtual disk, the primary unit of data management at the array level, instead of a LUN. As a result, it becomes possible to execute storage operations with virtual-machine, or even virtual-disk, granularity on the underlining storage system, and therefore provide native array-based data services, such as snapshots or replication, to individual virtual machines.

To facilitate a simplified and unified approach to management, all this is done with a common storage-policy-driven mechanism, which encompasses both Virtual SAN storage resources and Virtual Volumes external storage, into a single management plane. Virtual Volumes is covered in more detail in Chapter 8, "Policy-Driven Storage Design with Virtual Volumes."

Classic and Next-Generation Storage Models

This book refers to storage technologies as either *classic* or *next-generation*. Because these terms can have multiple meanings, this section provides an overview of each to clarify.

This book uses *classic storage model* to describe the traditional shared storage model used by vSphere. This typically includes LUNs, VMFS-based volumes and datastores, or NFS mount points, with a shared storage protocol providing I/O connectivity. Despite its constraints, this model has been successfully employed for years, and will continue to be used for some time by IT organizations and cloud service providers across the industry.

The *next-generation* storage model refers to VMware's software-defined solutions, Virtual SAN and Virtual Volumes, which bring about a new era in storage design, implementation, and management.

As addressed earlier in this chapter, the primary aim of VMware's software-defined storage model is to bring about simplicity, efficiency, and cost savings to storage resources. The model does this by abstracting the underlining storage in order to make the application the fundamental unit of management across a heterogeneous storage platform. With both Virtual SAN and Virtual Volumes, VMware moves away from the rigid constraints of the classic LUNs and volumes, and provides a new way to manage storage on a per virtual machine basis, through its more flexible policy-driven approach.

However, before addressing these *next-generation* storage technologies, you first need to understand the approach taken to storage over the last generation of vSphere-based virtualization platforms, and see how the VMware stack itself interacts with storage resources to provide a flexible, modern virtual data center.

This first chapter has addressed the VMware storage landscape, processes associated with storage design, and challenges faced by vSphere storage administration teams when maintaining complex, heterogeneous storage platforms on a daily basis for enterprise IT organizations and cloud service providers. The next chapter presents many of the essential design considerations based on the classic storage model previously outlined.

Chapter 2

Classic Storage Models and Constructs

This chapter covers the design considerations for deploying classic storage technologies in a VMware-based virtual data center, and addresses the primary storage concepts that impact the platform design of the storage layer.

Classic Storage Concepts

Storage infrastructure is made up of a multitude of complex components and technologies, all of which need to interact seamlessly to provide high performance, continuous availability, and low latency across the environment. For students of vSphere storage, understanding the design and implementation complexities of mixed, multiplatform, multivendor enterprise or service provider–based storage can at first be overwhelming. Gaining the required understanding of all the components, technologies, and vendor-specific proprietary hardware takes time.

This chapter addresses each of these storage components and technologies, and their interactions in the classic storage environment. Upcoming chapters then move on to next-generation VMware storage solutions and the software-defined storage model.

This classic storage model employs intelligent but highly proprietary storage systems to group disks together and then partition and present those physical disks as discrete logical units. Because of the proprietary nature of these storage systems, my intention here is not to address the specific configuration of, for instance, HP, IBM, or EMC storage, but to demonstrate how the vSphere platform can use these types of classic storage devices.

In the classic storage model, the logical units, or storage devices, are assigned a logical unit number (LUN) before being presented to vSphere host clusters as physical storage devices. These LUNs are backed by a back-end physical disk array on the storage system, which is typically served by RAID (redundant array of independent disks) technology; depending on the hardware type, this technology can be applied at either the physical or logical disk layer, as shown in Figure 2.1.

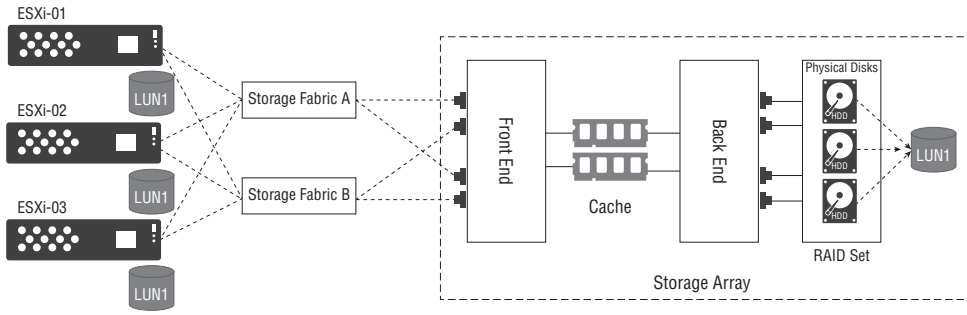


Figure 2.1
Classic storage model

The LUN, or storage device, is a virtual representation of a portion of physical disk space within the storage array. The LUN aggregates a portion of disk space across the physical disks that make up the back-end system. However, as illustrated in the previous figure, the data is not written to a single physical device, but is instead spread across the drives. It is this mechanism that allows storage systems to provide fault tolerance and performance improvements over writing to a single physical disk.

This classic storage model has several limitations. To start with, all virtual disks (VMDKs) on a single LUN are treated the same, regardless of the LUN's capabilities. For instance, you cannot replicate just a single virtual disk at the storage level; it is the whole LUN or nothing. Also, even though vSphere now supports LUNs of up to 64 terabytes, LUNs are still restricted in size, and you cannot attach more than 256 LUNs to a vSphere host or cluster.

In addition, with this classic storage approach, when a SCSI LUN is presented to the vSphere host or cluster, the underlying storage system has no knowledge of the hypervisor, filesystem, guest operating system, or application. It is left to the hypervisor and vCenter, or other management tools, to map objects and files (such as VMDKs) to the corresponding extents, pages, and logical block address (LBA) understood by the storage system. In the case of a NAS-based NFS solution, there is also a layer of abstraction placed over the underlying block storage to handle file management and the associated file-to-LBA mapping activity.

Other classic storage architecture challenges include the following:

- Proprietary technologies and not commodity hardware
- Low utilization of raw storage resources
- Frequent overprovisioning of storage resources
- Static, nonflexible classes of service
- Rigid provisioning methodologies
- Lack of granular control, at the virtual disk level
- Frequent data migrations required, due to changing workload requirements
- Time-consuming operational processes
- Lack of automation and common API-driven provisioning
- Slow storage-related requests requiring manual human interaction to perform maintenance and provisioning operations

Most storage systems have two basic categories of LUN: the traditional model and disk pools. The traditional model has been the standard mechanism for many years in legacy storage systems. Disk pools have recently provided compatible systems with additional flexibility and scalability, for the provisioning of virtual storage resources.

In the traditional model, when a LUN is created, the number and choice of disks directly corresponds to the RAID type and disk device configured. This traditional model has limitations, especially in virtual environments, which is why it was superseded by the more modern disk pool concept. The traditional model would often have a fixed maximum number of physical disks, which could be combined to form the logical disk. This maximum disk limitation was imposed by storage array systems as a hard limit, but was also linked to the practical considerations around availability and performance.

With this traditional disk-grouping method, it was often possible to expand a logical disk beyond its imposed physical limits by creating some sort of MetaLUN. However, this increased operational complexity and could often be difficult and time-consuming.

An additional consideration with this approach was that the amount of storage provisioned was often far greater than what was required, because of the tightly imposed array constraints. Provisioning too much storage was also done by storage administrators to prevent application outages often required to expand storage, or to cover potential workload requirements or growth patterns that were unknown. Either way, this typically resulted in expensive disk storage lying unutilized for a majority of the time.

On the plus side, this traditional approach to provisioning LUNs provided fixed, predictable performance, based on the RAID and disk type employed. For this reason, this method of disk provisioning is still sometimes a good choice when storage requirements do not have large amounts of expected growth, or have fixed service-level agreements (SLAs) based on strict application I/O requirements.

In more recent years, storage vendors have moved almost uniformly to disk pools. Pools can use far larger groups of disks, from which LUNs can be provisioned. While the disk pool concept still comprises physical disks employing a RAID mechanism to stripe or mirror data, with a LUN carved out from the pool, this device type can be built across a far greater number of disks. As a result of this approach, storage administrators can provision significantly larger LUNs without sacrificing levels of availability.

However, the sacrifice made by employing this more flexible approach is the small level of variability in performance that results. This is due to both the number of applications that are likely to share the storage of this single disk pool, which will inevitably increase over time, and the potential heterogeneous nature of disk pools, which have no requirement for uniformity, as it relates to the speed and capacity of individual physical disks (see Figure 2.2).

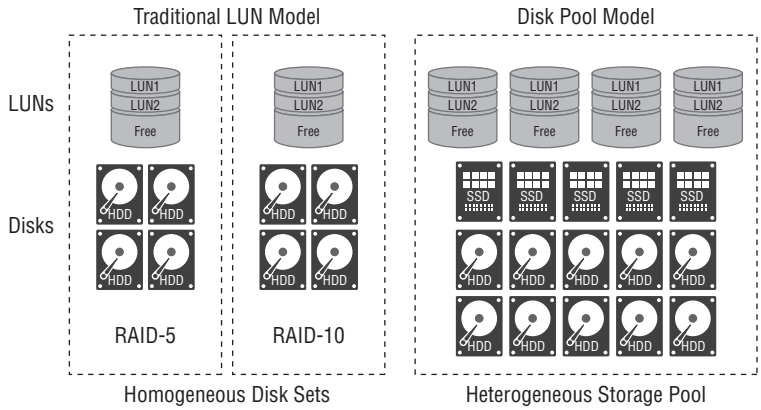


Figure 2.2 Storage LUN provisioning mechanisms

Also relevant from a classic storage design perspective are the trade-offs associated with choosing between provisioning a single disk pool or multiple disk pools. If choosing multiple pools, what criteria should a design use to define those pools?

We address tiering and autotiering in more detail later in this chapter, but this is one of the key design factors when considering whether to provision a single pool, with all the disk

resources, or to deploy multiple storage pools on the array and to split storage resources accordingly.

Choosing a single pool provides simpler operational and capacity management of the environment. In addition, it allows LUNs or filesystems to be striped across a larger number of physical disks, which improves overall performance of the array system. However, it is also likely that a larger number of hosts and clusters will share the same underlying back-end disk system. Therefore, there is an increased possibility for resource contention and also an increased risk of specific applications not using an optimal RAID configuration, and maximizing I/O, which is likely to result in a degraded performance for those workloads.

Using multiple disk pools offers the flexibility to customize storage resources to meet specific application I/O requirements, and also allows operational teams to isolate specific workloads to specific physical drives, reducing the risk of disk contention. However, as the pools are inevitably smaller in this type of architecture, some systems may experience lower levels of performance than with a single larger pool. In addition, with multiple smaller pools, capacity planning becomes more complex, as growth across disk pools may not be consistent, and there is likely to be an increase in overall disk resources not being used.

Neither of these options is without its advantages and drawbacks, and there is no one perfect solution. However, designing a solution that uses multiple smaller pools over one universal disk pool will likely come down to one or more of the following key design factors:

- Disk pools based on function, such as development, QA, production, and so on. This option may be preferred if you are concerned with performance for specific environments, and want to isolate them from impacting the production system.
- In multitenanted environments, whether public or based on internal business units, each tenant can be allocated its own pool. However, depending on the environment and SLAs, each tenant might end up with multiple pools in order to address specific I/O characteristics of various applications.
- Application-based pools, such as database or email systems. This can provide optimum performance as applications of similar type often have similar I/O characteristics. For this reason, it may be worth considering designing pools based on application type. However, this also carries the risk of some databases, for instance, generating very high volumes of I/O and potentially impacting other databases residing on the same disk pool.
- Drive technology and RAID type. This allows you to place data on the storage type that best matches the application I/O characteristics, such as reads versus writes versus sequential. However, this approach can also increase costs and does not address any specific application I/O intensity requirement.
- Storage tier–based pools (such as Gold, Silver, and Bronze) could allow you to mix drive technologies and/or RAID types within each pool, therefore reducing the number of pools required to support most application types, configurations, and SLAs.

RAID Sets

The term *RAID* has already been used multiple times in different contexts, so let's address this technology next.

RAID (redundant array of independent disks) combines two or more disk drives into a logical grouping, typically known as a RAID set. Under the control of a RAID controller (or in the case of a storage system, the storage processors or controllers), the RAID set appears to the connected hosts as a single logical disk drive, even though it is made up of multiple physical disks. RAID sets provide four primary advantages to a storage system:

- Higher data availability
- Increased capacity
- Improved I/O performance
- Streamlined management of storage devices

Typically, the storage array management software handles the following aspects of RAID technology:

- Management and control of disk aggregation
- Translation of I/O requests between the logical and the physical entities
- Error correction if disk failures occur

The physical disks that make up a RAID set can be either traditional mechanical disks or solid-state flash drives (SSDs). RAID sets have various levels, each optimized for specific use cases. Unlike many other common technologies, RAID levels are not standardized by an industry group or standardization committee. As a result, some storage vendors provide their own unique implementation of RAID technology. However, the following common RAID levels are covered in this chapter:

- RAID 0–striping
- RAID 1–mirroring
- RAID 5–striping with parity
- RAID 6–striping with double parity
- RAID 10–combining mirroring and striping

Determining which type of RAID to use when building a storage solution largely depends on three factors: capacity, availability, and performance. This section addresses the basic concepts that provide a foundation for understanding disk arrays, and how RAID can enable increased capacity by combining physical disks, provide higher availability in case of a drive failure, and increase performance through parallel drive access.

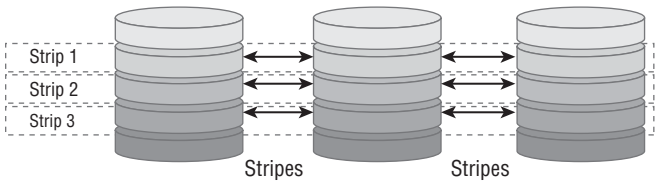
A key element in RAID is redundancy, in order to improve fault tolerance. This can be achieved through two mechanisms, *mirroring* and *striping*, depending on the RAID set level configured. Before addressing the RAID set capabilities typically used in storage array systems, we must first explain these two terms and what they mean for availability, capacity, performance, and manageability.

NOTE Some storage systems also provide a JBOD configuration, which is an acronym for *just a bunch of disks*. In this configuration, the disks do not use any specific RAID level, and instead act as stand-alone drives. This type of disk arrangement is most typically employed for storage devices that contain swap files or spooling data, where redundancy is not paramount.

Striping in RAID Sets

As highlighted previously, RAID sets are made up of multiple physical disks. Within each disk are groups of continuously addressed blocks, called *strips*. The set of aligned strips that spans across all disks within the RAID set is called the *stripe* (see Figure 2.3).

Figure 2.3
Strips and stripes



Striping improves performance by distributing data across the disks in the RAID set (see Figure 2.4). This use of multiple independent disks allows multiple reads and writes to take place concurrently, providing one of the main advantages of disk striping: improved performance. For instance, striping data across three hard disks would provide three times the bandwidth of a single drive. Therefore, if each drive runs at 175 input/output operations per second (IOPS), disk striping would make available up to 525 IOPS for data reads and writes from that RAID set.

Striping also provides performance and availability benefits by doing the following:

- Managing large amounts of data as it is being written; the first piece is sent to the first drive, the second piece to the second drive, and so on. These data pieces are then put back together again when the data is read.
- Increasing the number of physical disks in the RAID set increases performance, as more data can be read or written simultaneously.
- Using a higher stripe width indicates a higher number of drives and therefore better performance.
- Striping is managed through storage controllers, and is therefore transparent to the vSphere platform.

As part of the same mechanism, *parity* is provided as a redundancy check, to ensure that the data is protected without having to have a full set of duplicate drives, as illustrated in Figure 2.5. Parity is critical to striping, and provides the following functionality to a striped RAID set:

- If a single disk in the array fails, the other disks have enough redundant data so that the data from the failed disk can be recovered.
- Like striping, parity is generally a function of the RAID controller or storage controller, and is therefore fully transparent to the vSphere platform.
- Parity information can be
  - Stored on a separate, dedicated drive
  - Distributed across all the drives in the RAID set

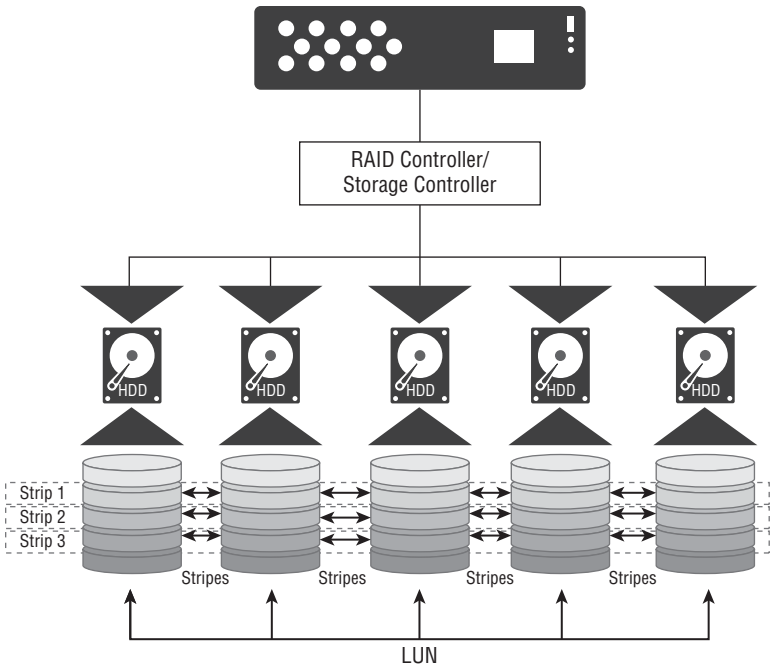


Figure 2.4
Performance in striping

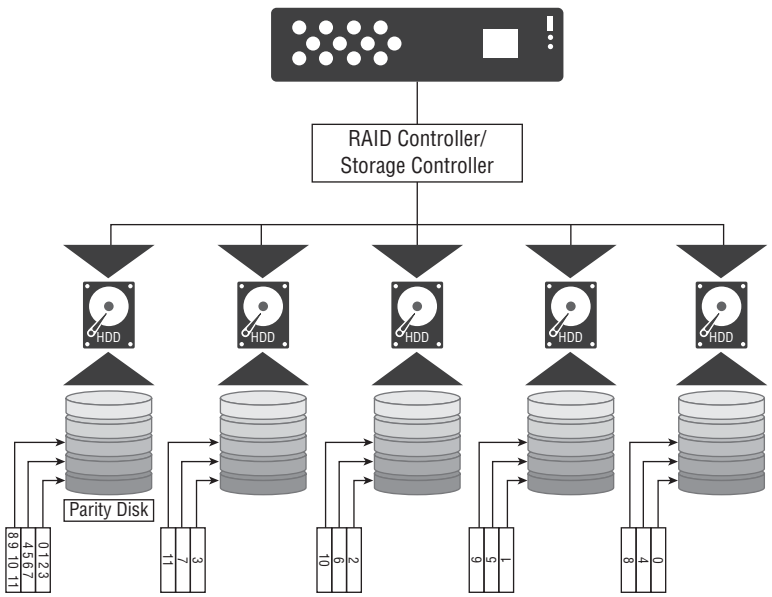


Figure 2.5
Redundancy through parity

Mirroring in RAID Sets

Mirroring uses a mechanism that enables multiple physical disks to hold identical copies of the data, typically on two drives. Every write of data to a disk is also a write to the mirrored disk, meaning that both physical disks contain exactly the same information at all times. This mechanism is once again fully transparent to the vSphere platform and is managed by the RAID controller or storage controller. If a disk fails, the RAID controller uses the mirrored drive for data recovery, but continues I/O operations simultaneously, with data on the replaced drive being rebuilt from the mirrored drive in the background.

The primary benefits of mirroring are that it provides fast recovery from disk failure and improved read performance (see Figure 2.6). However, the main drawbacks include the following:

- Degraded write performance, as each block of data is written to multiple disks simultaneously
- A high financial cost for data protection, in that disk mirroring requires a 100 percent cost increase per gigabyte of data

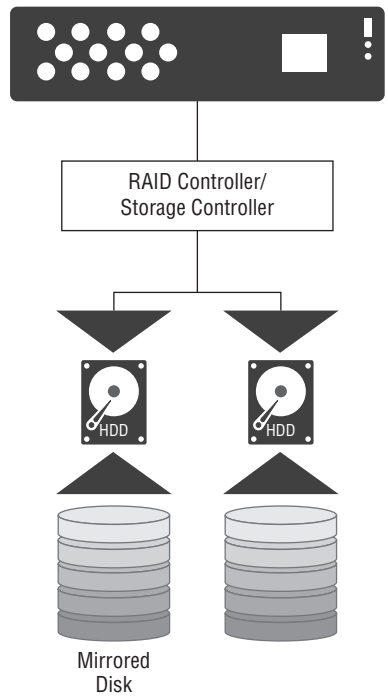


Figure 2.6
Redundancy in disk mirroring

Enterprise storage systems typically support multiple RAID levels, and these levels can be mixed within a single storage array. However, once a RAID type is assigned to a set of physical disks, all LUNs carved from that RAID set will be assigned that RAID type.

Nested RAID

Some RAID levels are referred to as *nested RAID*, as they are based on a combination of RAID levels. Examples of nested RAID levels include RAID 03 (RAID 0+3, also known as RAID 53, or RAID 5+3) and RAID 50 (RAID 5+0). However, the only two commonly implemented nested RAID levels are RAID 1+0, also commonly known as RAID 10, and RAID 01 (RAID 0+1). These two are similar, except the data organization methods are slightly different; rather than creating a mirror and then striping the mirror, as in RAID 1+0, RAID 0+1 creates a stripe set and then mirrors it.

Calculating I/O per Second RAID Penalty

One of the primary ways to measure disk performance is input/output per second, also referred to as I/O per second or, more commonly, IOPS. This formula is simple: one read request or one write request is equal to one I/O.

Each physical disk in the storage is capable of providing a fixed number of I/O. Disk manufacturers calculate this based on the rotational speed, average latency, and seek time. Table 2.1 shows examples of typical physical drive IOPS specifications for the most common drive types.

Table 2.1: Typical average I/O per second (per physical disk)

Drive Speed	Typical Average IOPS/Drive
Solid-State Disk (SSD)	6,000
15,000 RPM	175
10,000 RPM	125
7,200 RPM	75
5,400 RPM	50

A storage device's IOPS capability is calculated as an aggregate of the sum of disks that make up the device. For instance, when considering a JBOD configuration, three disks rotating at 10,000 RPMs provide the JBOD with a total of 375 IOPS. However, with the exception of RAID 0 (which is simply a set of disks aggregated together to create a larger storage device), all RAID set configurations are based on the fact that write operations result in multiple writes to the RAID set, in order to provide the targeted level of availability and performance.

In a RAID 5 disk set, for example, for each random write request, the storage controller is required to perform multiple disk operations, which has a significant impact on the raw IOPS calculation. Typically, that RAID 5 disk set requires four IOPS per write operation. In addition, RAID 6, which provides a higher level of protection through double fault tolerance, also provides a significantly worse *I/O penalty* of six operations per write. Therefore, as the architect of such a solution, you must also plan for any I/O penalty associated with the RAID type being used in the design.

Table 2.2 summarizes the read and write RAID penalties for the most common RAID levels. Notice that you don't have to calculate parity for a read operation, and no penalty is associated

with this type of I/O. The I/O penalty relates specifically to writes, and there is no negative performance or IOPS impact when calculating read operations. It is only when you have writes to disk that you will see the RAID penalty come into play in RAID calculations and formulas. This is true even though in a parity-based RAID-type write operation, reads are performed as part of that write. For instance, writes in a RAID 5 disk set, where data is being written with a size that is less than that of a single block, require the following actions to be performed:

1. Read the old data block.
2. Read the old parity block.
3. Compare data in the old block with the newly arrived data. For every changed bit, change the corresponding bit in parity.
4. Write the new data block.
5. Write the new parity block.

As noted previously, a RAID 0 stripe has no write penalty associated with it because there is no parity to be calculated. In Table 2.2, a no RAID penalty is expressed as a 1.

Table 2.2: RAID I/O penalty impact

RAID Level	Read	Write Penalty	Example of Write IOPS for a 15K Disk
RAID 0–Striping	1	1	175
RAID 1–Mirroring	1	2	85
RAID 3–Parallel transfer with parity	1	3	65
RAID 5–Striping with parity	1	4	40
RAID 6–Striping with double parity	1	6	30
RAID 10–Combining mirroring and striping	1	2	85

Parity-based RAID sets introduce additional processing overhead on the storage controllers, which results from the additional calculations required to determine the parity data. The higher the level of parity protection you provide to a RAID set, the more processing overhead you incur on the controllers, although, as you would expect, the actual overhead incurred is highly dependent on the workload's read/write balance.

In calculating the number of IOPS incurred by the RAID penalty, the following formula provides a good starting point, assuming you have derived the customer's workload balance between read and write operations from a current state analysis. However, you must also take into account peak and average workloads, to ensure that the storage device can deliver the required IOPS.

(total workload IOPS) × (% of workload that is read operations) + (total workload IOPS × % of workload that is read operations × RAID I/O penalty)

In this example calculation, the customer has provided the following workload I/O values:

- Total IOPS required: 250 IOPS
- Read workload: 50 percent
- Write workload: 50 percent
- RAID level required: 6 (I/O penalty of 6)

You would require a RAID 6 disk set that could support 875 IOPS, in order to meet the customer's requirement for 250 IOPS on a RAID 6 disk set, where the workload has 50 percent write operations.

As this example makes clear, the number of disks is far more important than the disk capacity. Based on the information provided by the customer, you would require twelve 7,200 RPM disks, seven 10K RPM disks, or five 15K RPM disks to support the required IOPS.

As you can see, determining the correct RAID type for a specific workload is key, and will come down to various design factors and compromises between cost, availability, and performance.

RAID Levels Explained

The RAID type chosen for a specific LUN determines the level of redundancy and data integrity that the LUN provides to the applications running on it. However, not all storage array vendors support all RAID types, and some have even developed their own. As part of ensuring that your storage design meets the customer's needs, you should establish the types of RAID available for the hardware vendor's storage devices. Tables 2.3 through 2.8 provide insight into the types of RAID that are most commonly employed in storage arrays, with illustrations in Figures 2.7 through 2.12.

Table 2.3: RAID 0—striped disk array without fault tolerance

Design Factor	Description
Data protection	None. RAID 0 stripes the information across the drives in the array without generating redundant data. By providing no parity or mirroring, there is no fault tolerance, making it extremely difficult to recover data.
Advantages	RAID 0 offers great performance, both in read and write operations. No overhead is created by parity controls, which also allows all storage capacity to be used. This technology is also easy to implement.
Drawbacks	RAID 0 is not fault-tolerant. If one drive fails, all data in the RAID 0 disk set will be lost. This RAID type should not be employed for business-critical systems.
Performance characteristics	RAID 0 is superior to a JBOD configuration, as it uses striping. All the data is spread out in chunks across all the disks in the RAID set. The I/O rate, or throughput, can be good when I/O sizes are small; however, larger I/Os will produce high bandwidth (data moved per second) with this RAID type. Performance can be further improved when data is striped across multiple controllers, with only one drive per controller.