Data Center Evolution

A data center is a physical facility that organizations use to house their critical applications and data. A data center's design is based on a network of computing and storage resources that enable the delivery of shared applications and data.

What defines a modern data center?

Infrastructure has shifted from traditional on-premises physical servers to virtualized infrastructure that supports applications and workloads across pools of physical infrastructure and into a multicloud environment.

The modern data center is wherever its data and applications are. It stretches across multiple public and private clouds to the edge of the network via mobile devices and embedded computing.

Why are data centers important to business?

In the world of enterprise IT, data centers are designed to support business applications and activities that include:

1. Email and file sharing,
2. Productivity applications
3. Customer relationship management (CRM) and enterprise resource planning (ERP)
4. Big data, artificial intelligence, and machine learning
5. Communications and collaboration services

What are the core components of a data center?

- They are routers, switches, firewalls, storage systems, servers, and application delivery controllers.
- Because these components store and manage business-critical data and applications, data center security is critical in data center design.
- Together, they provide Network infrastructure, Storage infrastructure, and Computing resources.

Data centers are at the center of modern software technology and they enable businesses to do more with less.

What is in a data center facility?

Data center components require significant infrastructure to support the center's hardware and software. These include power subsystems, uninterruptible power supplies (UPS), ventilation, cooling systems, fire suppression, backup generators, and connections to external networks.

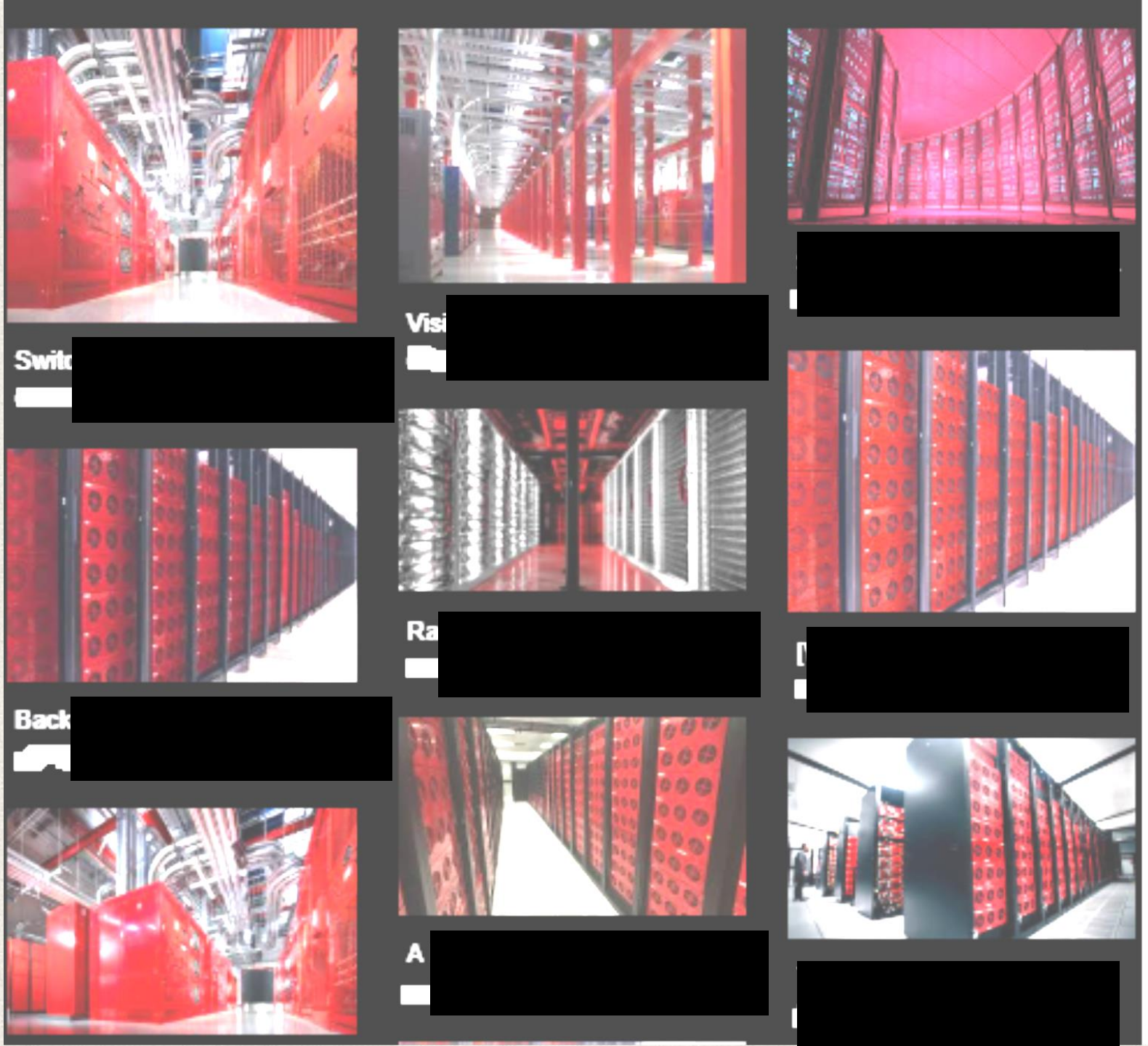
Types of Data Centers

Many types of data centers and service models are available.

Their classification depends on whether they are owned by one or many organizations, how they fit (if they fit) into the topology of other data centers, what technologies they use for computing and storage, and even their energy efficiency.

Types of Data Centers

- Enterprise data centers
- Managed services data centers
- Colocation data centers
- Cloud data centers



Top-Tier Data Center Players (Mega Clouds)

1.Amazon

With Amazon Web Services, users can "launch virtual machines or apps within minutes," a drastic reduction in provisioning time compared to traditional models.

Through a collection of remote computing services, Amazon Web Services creates and offers a cloud computing platform to subscribers.

2. Google Cloud Platform

An obvious contender as one of the largest data center players, Google offers options via Google Cloud Platform and Google Compute Engine.

Google's Cloud Platform enables developers to build, test and deploy applications on the same infrastructure that powers Google's vast capabilities. Google Compute Engine offers 99.95 percent monthly SLAs with 24-7 support.

3. Microsoft – Windows Azure

Microsoft rounds out the big three when it comes to major players in the cloud computing space with Windows Azure, offering the services of Microsoft-managed data centers in 13 regions around the world.

With per-minute billing and built-in auto-scaling, it proves to be a cost-efficient platform to build and deploy applications of any size.

Other Data Center Players: Rackspace, VMWare, Facebook etc.

A History of the Modern Data Center

Data centers play a critical role in the expanding capabilities for enterprises.”

The concept of “data centers” has been around since the late 1950s when American Airlines and IBM partnered to create a passenger reservations system offered by Sabre, automating one of its key business areas.

The idea of a data processing system that could create and manage airline seat reservations and instantly make that data available electronically to any agent at any location became a reality in 1960, opening the door to enterprise-scale data centers.

Since then, physical and technological changes in computing and data storage have led us down a winding road to where we are today.

Let’s take a brief look at the evolution of the data center, from the mainframe of yesterday, to today’s cloud-centric evolution, and some impacts they’ve had on IT decision-making.

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

1946

The Electronic Numerical Integrator and Computer (ENIAC) was built in 1946 for the U.S. Army to store artillery firing codes and was dubbed as the first general-purpose electronic digital computer.

Early 1960s

The first transistorized computer (TRADIC) was introduced in 1954 and was the first machine to use all transistors and diodes and no vacuum tubes. The innovation responsible for the data center as we know it today was the transistorized, integrated circuit based microprocessor.

Maturity in this technology eventually led to Intel’s 8086 chip, and all of its Successors.

1971

Intel introduced its 4004 processor, becoming the first general-purpose programmable processor on the market. It served as a “building block” that engineers could purchase and then customize with software to perform different functions in a wide variety of electronic devices.

1973

The Xerox Alto becomes the first desktop computer to use a graphical UI and included a bit-mapped high-resolution screen, large internal memory storage, and special software.

1977

ARCnet is introduced as the first LAN.

1980s

Personal computers (PCs) were introduced in 1981, leading to a boom in the microcomputer industry.
Sun Microsystems developed the network file system protocol, allowing a user on a client computer to access files over a network in a manner similar to how local storage is accessed.

Early 1990s

Microcomputers began filling old mainframe computer rooms as “servers,” and the rooms became known as **data centers.** Companies then began assembling these banks of servers within their own walls.

Mid 1990s

The enterprise construction of server rooms, lead to much larger facilities (hundreds and thousands of servers). The **data center as a service model** became popular at this time.

2002

Amazon Web Services begins development of a suite of cloud-based services, which included storage, computation and some human intelligence through "Amazon Mechanical Turk."

2006

Amazon Web Services begins offering IT infrastructure services to businesses in the form of web services, now commonly known as cloud computing.

2007

Sun Microsystems introduces the modular data center, transforming the fundamental economics of corporate computing.

2011

Facebook launches Open Compute Project, an industry-wide initiative to share specifications and best practices for creating the most energy efficient and economical data centers. About 72% of organizations said their data centers were at least 25 percent virtual.

2013

Telcordia introduces generic requirements for telecommunications data center equipment and spaces.

Google invested a massive $7.35 billion in capital expenditures in its Internet infrastructure during 2013. The spending was driven by a massive expansion of Google's global data center network, which represented perhaps the largest construction effort in the history of the data center industry.

Today and Beyond

Today's datacenters are shifting from an infrastructure, hardware and software ownership model, towards a subscription and capacity on demand model.

Two constantly developing technologies — the microprocessor/x86 architecture and disk-based storage medium — form the foundation for the modern data center. The modern data center provides Software Defined Storage to serve I/O to the applications, thereby offering IT organizations greatest storage flexibility and ease of use. It is also possible to develop projects that once seemed like pure science fiction.

More with Less

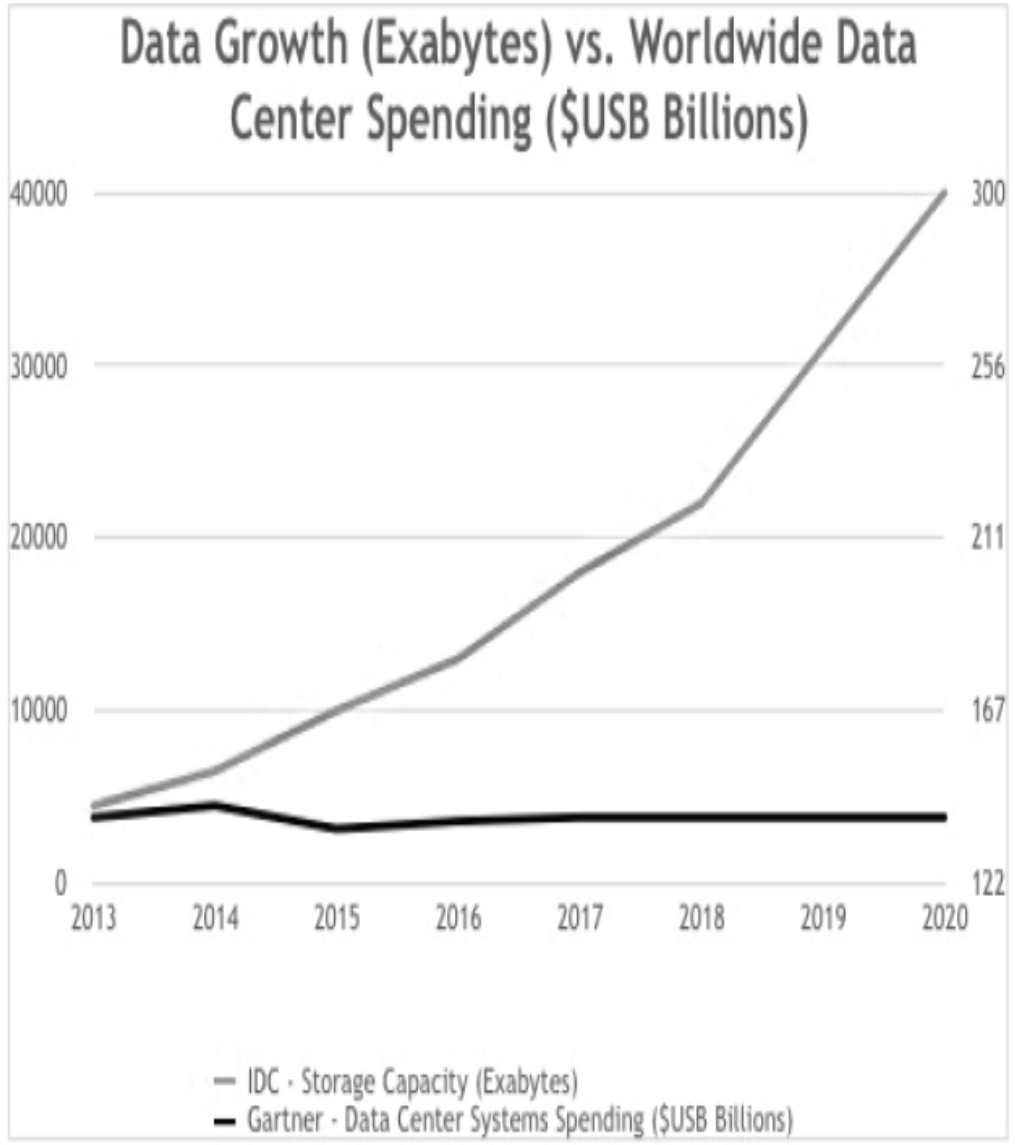


Figure Data Growth vs. Data Center Spending

Technology in the data center tends to be Cyclical and the technology improves with each new Cycle.

In the 1990s, the prevailing data center design had each application running on a server, or a set of servers, with locally attached storage media.

As the quantity and criticality of line-of-business applications supported by the data center grew, this architecture began to show some dramatic inefficiency when deployed at scale.

Plus, the process of addressing that inefficiency has characterized the modern data center for the past two decades.

The Rise of the Monolithic Storage Array

The inefficiency at scale actually had two components:

1. Servers very commonly used only a fraction of the computing power. Typically the server ran at 10% CPU utilization, thus wasting massive amounts of resources.

2. The available storage was under utilized. There were islands of storage created by placing direct attached storage with every server which could not be fully utilized. There came a great inefficiency caused by the need to allow room for growth.

For ex: If an enterprise had 800 servers in their data center. If each of those servers had 60 GB of unused storage capacity to allow for growth. That would mean there was 48 TB of unused capacity across the organization.

Paying for 48 TB of capacity to just sit on the shelf seems absurd, but until this problem could be solved, that was the accepted design.

Solution to this problem: Storage Area Network

- In a storage area network (SAN), rather than providing direct-attached storage for each server, disks are pooled and made accessible via the network.
- This allows many devices to draw from one capacity pool and increase utilization across the enterprise dramatically.
- It also decreased the management overhead of storage systems, because it meant that rather than managing 800 storage silos, perhaps there were only 5 or 10.
- These arrays of disks (“storage arrays”) were connected on a network segregated from the local area network.
- The SAN made use of a different network protocol more suited for storage networking called Fibre Channel Protocol.
- It was more suited for delivering storage because of its “lossless” and high-speed nature.
- The purpose of the SAN is to direct and store data, and therefore the loss of transmissions is unacceptable.

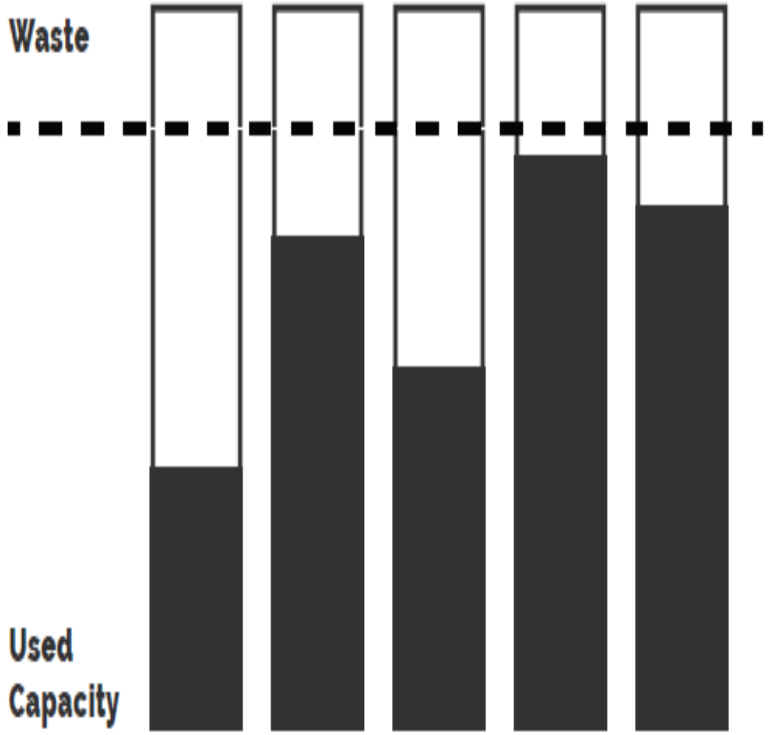


Figure Inefficiency of the microprocessor/x86 architecture & Disk-based storage

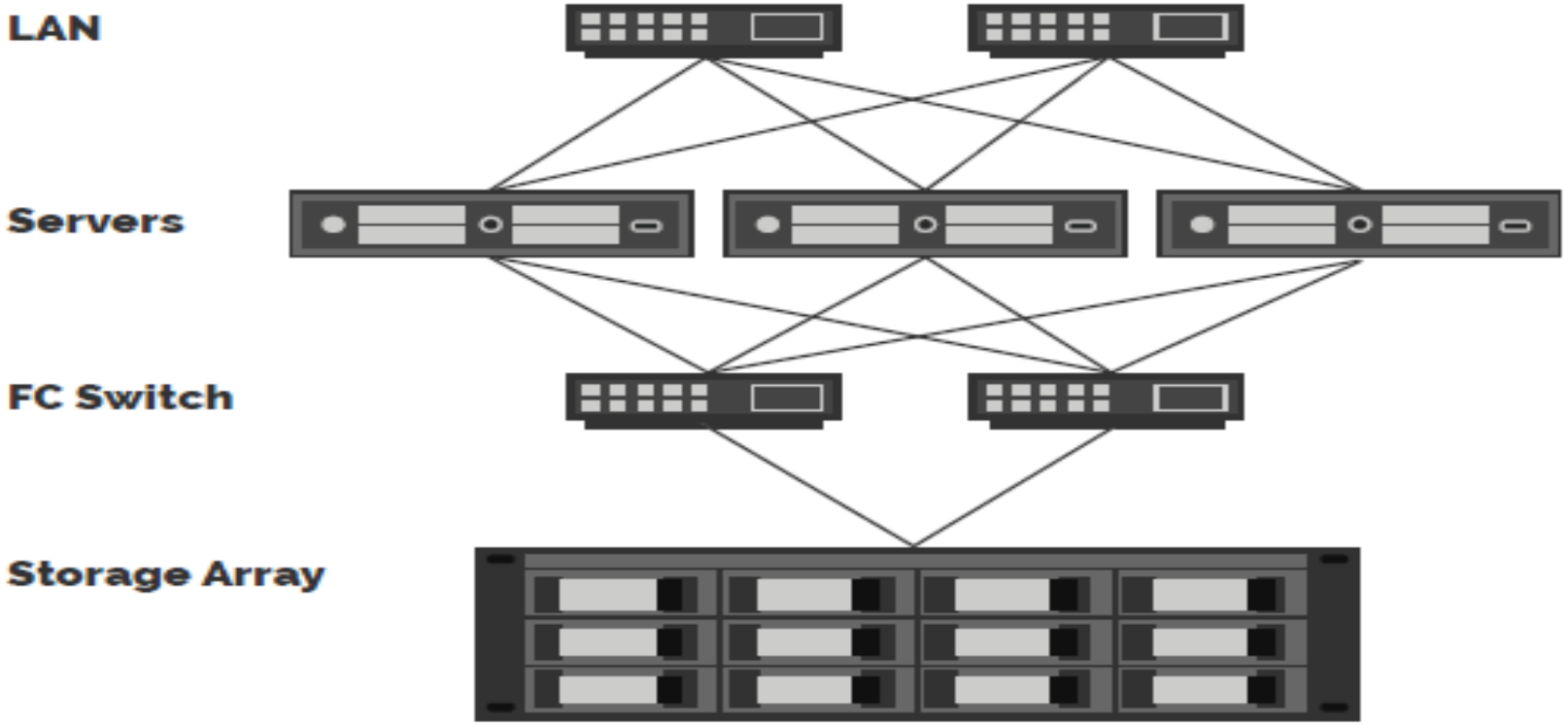


Figure : A Simple SAN

File vs. Block Storage

- Data stored on a shared storage device is typically accessed in one of two ways: at the block level or at the file level.
- In file level access the granularity of access is a full file. Protocols used are NFS, and SMB. File-based protocols may be used where an end user or application will be accessing the files directly —for example a network share.

For block storage the protocols used are: Fibre Channel (block), iSCSI (block). Block level access sends SCSI commands directly from the initiator (client side) to the target (storage array side).

Block-based protocols are more likely to be used when an operating system or hypervisor is accessing the storage, as direct access to the disk is preferred.

Data Services: Allow the administrator to manipulate and protect the stored data.

Types:

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

b. Replication: A storage feature that allows an administrator to copy a duplicate of a data set to another system. Replication is the most commonly used data protection method; copies of data are replicated off site and are available for restore in the event of a disaster.

Replication can also have other uses, however, like replicating production data to a testing environment.

c. Data Reduction: There is a large amount of duplicate data in enterprise environments. Virtualization compounds this issue.

Many storage platforms are capable of compression and deduplication, which both involve removing duplicate bits of data.

Compression happens to a single file or object, whereas deduplication happens across an entire data set. By removing duplicate data, often only a fraction of the initial data will be stored.

As the industry matured and more organizations adopted a shared storage model, the value of the architecture continued to increase.

In addition to file system snapshots, administrators could make use of volume-level snapshots also. This created new possibilities for backup and recovery solutions to complete backups faster and more efficiently.

Storage systems also contained mechanisms for replicating data from one storage array to another. This meant that a second copy of the data could be kept up-to-date in a safe location, as opposed to backing up and restoring data all the time.

Perhaps one of the greatest efficiencies achieved by adopting the shared storage model was the potential for global deduplication of data across the enterprise.

By the mid-2000s, average data centers had the efficiency of using shared storage across servers and applications, combined with the added efficiency of being able to globally deduplicate that data.

Performance of the shared storage systems grew as manufacturers continued to improve the networking protocols, the physical disk media, and the file systems that governed the storage array.

Using shared storage allowed more agility and flexibility with servers than was known with direct-attached storage.

Many organizations chose to provision the operating system disk for a server on the storage array and use a “boot from SAN” model.

The benefit of deploying operating systems this way was this: if one physical server failed, a new server could replace it, be mapped to the same boot volume, and the same operating system instance and applications could be back up and running in no time.

All these efforts brought down the cost of data centers. However there was still the problem of compute resources.

CPU resources were still generally configured far above the actual utilization of the application the server was built for.

Eliminating this problem was the second frontier in solving inefficiency in the modern data center.

The Virtualization of Compute — Software Defined Servers

Virtualization as a concept is not a new development. Virtualization has been around since the 1960s when the technology was developed to allow multiple jobs to run simultaneously on a mainframe.

Virtualization allowed for multiple workloads to run in tandem on shared hardware, yet be isolated from one another.

But the true power of modern virtualization came in 2001, when VMware released ESX, a bare-metal hypervisor capable of virtualizing server workloads in the data center. Later Microsoft came up with Hyper-V.

The **hypervisor**, a software that abstracts physical resources like CPU and memory from the virtual machines, is capable of running multiple workloads simultaneously and effectively isolated from one another.

It was estimated that the data center industry in 2006 consumed 61 billion kilowatt hours of electricity due to the large no. of physical servers. There was a great need to cut down the numbers without affecting the businesses. Virtualization was the answer.

Rather than having 10 physical servers running at 10% utilization, there were now two servers running at 50% or higher utilization. This brought down administrative as well as networking(cable) costs.

As hypervisor and virtual machine performance increased the demands on related infrastructure components also increased. There was a need for higher bandwidth, higher disk performance and lower latency. Spinning disks have served as primary storage, and tape-based storage systems have served higher capacity longer term storage needs.

The speed by which data on a spinning disk can be accessed cannot be increased beyond a certain limit as it will lead to the damage of the disk. There's also the issue of latency. Due to the mechanical nature of a spinning disk drive, latency (the time it takes to retrieve or write the data in question) can't be pushed below a certain threshold.

Tiny bits of latency added together across many drives becomes an issue at scale. This led to the replacement of spinning disk by flash storage in the data centers so as to increase the speed of data access.

Because flash storage is not mechanical in nature, it doesn't suffer from the same limitations as spinning disks. Flash storage is capable of latency on the order of microseconds as opposed to spinning disk's multiple milliseconds. It's also capable of far more I/O operations per second than a handful of spinning disks. Also flash storage durability has improved over time.

Lastly, because of the non-mechanical (or “solid state”) nature of flash storage, it requires much less power to operate when compared to spinning disk. As data center power bills move northwards any way to reduce power consumption is welcome.

The Fall of the Monolithic Storage Array

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

One of the primary challenges here is “mixed workload.” By the nature of virtualization, many different applications and operating systems share the same physical disk infrastructure on the back end.

The challenge with this architecture is that operating systems, & especially applications, have widely varying workload requirements and characteristics. For example, attempting to deploy virtual desktop infrastructure (VDI) on the same storage platform as the server virtualization has been the downfall of many VDI projects.

Due to the drastically different I/O characteristics of a desktop operating system versus a server operating system and the applications running on them, they require almost completely opposite things. An average Windows server might require 80% reads and 20% writes, whereas on the exact same storage array, with the same disk layout, same cache, and so on, a virtual desktop might require 20% reads and 80% writes.

As application performance requirements go up, it has also become increasingly important to provide very low latency. So which storage model is likely to have lower latency: the one where storage is accessed across a network and shared with all other workloads, or the one where storage is actually inside the server.

The answer is the model where the storage is local to the workload. Some new ideas have started popping up in the data center storage market over the past few years. Figure 2-5 shows the progression of storage design over time.

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

Figure 2-5: Storage design timeline

The data center of the future looks (physically) a lot more like the data center of the past, in which a number of servers all contain their own direct attached storage.

The difference is that all of this locally attached storage is pooled, controlled, accelerated, and protected by a storage management platform running on the hypervisor. The performance and scale implications of this model are massive: because each node added to the cluster with local storage contributes to the pool, this means that the storage pool can grow to virtually limitless heights.

Each server that is added has its own storage controller, meaning that throughput never becomes an issue. Increasing capacity of the pool is as easy as adding disks to existing servers or adding more servers overall. The control of all of this is done by either virtual machines (VSAs) or by kernel-level software, and the administrator typically manages it from the hypervisor's existing management interface.

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

The Emergence of Convergence

As the challenges for IT have grown in equal proportions with the ever-increasing scope of their responsibilities, IT decision makers have often looked to outsource parts of their operation.

A notable trend for data center “outsourcing” of sorts is now referred to as convergence.

Convergence is multiple pieces of the infrastructure assembled prior to delivery to the customer. Convergence saves time and frustration during the deployment phase and provides decreased time-to-value after procurement.

An example of a common form of convergence might look like this: a rack is delivered to the data center already containing a storage array, a blade chassis populated with blades, and a few top-of-rack switches. Everything is cabled up, and all the configuration of the switching and storage has been done prior to delivery.

At the moment the converged stack is delivered, the data center team can roll it into place, deliver power and upstream network connectivity, and the pod will be up and running.

This model of growing the infrastructure is substantially faster than the traditional model of having parts delivered, assembling them, hiring consultants, troubleshooting, and so on.

The value in convergence comes not only from the fact that the solution comes pre-assembled, but also from the fact that it includes all the pieces necessary. Half the challenge in traditional piecemeal solution-building is getting all the right parts and ensuring interoperability.

Convergence guarantees that with the purchase of a certain SKU, all the components contained within it will be compatible with one another, and all the necessary parts will be included. This has helped many organizations realize project objectives faster, and has saved a multitude of headaches over time.

But if a little convergence was good, does that mean a lot of convergence is great?

The successor to convergence is known as “hyperconvergence,” and it takes the idea of simplicity to the customer to new heights.

Hyperconvergence is so called because of the scope of what is being converged. In a converged infrastructure, many infrastructure components are brought together into one rack (or a few racks).

In a hyperconverged infrastructure (HCI), those same components are brought together within a single server node. Hyperconvergence is born from cloud data centers that pioneered and leveraged this technology to operate at the massive scale they require.

Converged v Hyperconverged Infrastructure

Component	Traditional Converged Infrastructure	Hyperconverged Infrastructure
Storage	Tiered storage area network (SAN)	Software defined storage
Management Software	Vertical stacks	Horizontal compute, storage and global file system
Scalability	Scale-up, using primarily proprietary components	Scale-out using mostly commodity components, including compute and storage
Workload Support	Core enterprise	Virtualization, AnyCloud
Integration	Hardware-defined, vendor defined	Software-defined, hypervisor-integrated
Architecture	Vertical	Horizontal, Symmetric scale-out architecture
Vendors / Solutions	Cisco-NetApp, VCE, Oracle, HP, IBM, Dell, Huawei, etc...	Atlantis, Nutanix, Simplivity, Scale Computing, Pivot3, Maxta, EVO:Rail (OEMs)

Figure 2-6: Converged vs. Hyperconverged Infrastructure

The Role of Cloud

Cloud technology is being leveraged all over the world by even small companies. Cloud computing is a model of delivering infrastructure or application resources in a flexible, rapid, and on-demand manner. This is why purchasing infrastructure from Amazon Web Services (AWS), for example, would be classified as cloud. It's on-demand, takes about two minutes to provision, & has tons of options. Because cloud is a model and not a thing, there are a number of different ways in which cloud infrastructure can be implemented.

Cloud Types

A. Based on Deployment

Different cloud deployment models fit different organizations. There are certain cases where an application has been developed to run in a cloud. In this case, it may make sense to use the Cloud. Cloud is simply a method of offering & provisioning on-demand services.

A ***private cloud*** deployment is simply an on-premises deployment of a tool like OpenStack that allows for rapid, on-demand provisioning of resources that can easily be created & destroyed. It can also be remotely managed. Private cloud offers higher control and security.

A ***public cloud*** model is one where all resources are provisioned in a third party data center provided by the likes of AWS, Microsoft, VMware, Google, or a friendly neighborhood cloud provider. Especially for some small businesses, being entirely public-cloud-based allows for an extremely light IT footprint in the office or storefront, resulting in less overhead.

Public cloud can be very affordable. It also offloads risk and overhead in terms of compliance, patching, equipment failure, hardware refreshes, and so on. Public cloud offers less or no control and security.

The next possible choice is a combination of on-premises cloud and public cloud; it's known as ***hybrid cloud***. Using this model, IT resources run in the corporate data center as usual, but an extension to a public cloud data center is in place. This means that based on certain requirements, constraints, or other design decisions, a workload can be provisioned either to the private data center or to the public one.
Ex1: An example of how hybrid cloud might work is that of a retailer. If Black Friday is coming up, the retailer may be able to spin up an extra 20 instances of their website and shopping cart application in the public cloud. The back end databases still exist in the on-premises data center and need not be migrated. This is commonly referred to as "bursting" to the cloud.

Ex2: A hybrid cloud model could work out well is in an organization that has a heavy in-house development workload. If developers are constantly creating and destroying test environments, it can require lots of horsepower to keep things running fast enough that developers are happy, and project scopes can change with a moment's notice. A much easier way to handle this situation would be to run production workloads in the on-premises data center, but have development and testing workloads provision to the public cloud. This can also save on cost as opposed to running the resources locally.

Ex3: In one of the surveys roughly 10% of all survey respondents said that they're managing over a petabyte of storage at their primary site.

Community Cloud – A community cloud is shared between organizations with a common goal or that fit into a specific community (professional community, geographic community, etc.).

It is a multi-tenant platform which allows several companies work on the same platform, given that they have similar needs and concerns.

Ex: Test-drive some high-end security products or even test out some features of a public cloud environment.

B. Based on Service

- Infrastructure-as-a-service (IaaS)
- Platform as a service (PaaS)
- Software as a service (SaaS)
- FaaS (functions as a service): Is an entirely new cloud model and a game changer to many. This model is a way to achieve a **"serverless"** architecture and is mostly used for building microservices.

EX:

App Engine - Platform as a Service to deploy Java, PHP, Node.js, Python, C#, .Net, Ruby and Go applications.

Compute Engine - Infrastructure as a Service to run Microsoft Windows and Linux virtual machines.

Cloud Functions - Functions as a Service to run event-driven code written in Node.js or Python.

Cloud Drivers

The first driver for cloud models (public, hybrid, or private alike) is agility.

By nature, any sort of cloud model will dramatically increase the speed of the software development lifecycle.

The second driver is cost.

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

A third driver is scale.

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

Uses of Cloud Computing

Although you do not realize you are probably using **cloud computing**, most of us use an online service to send email, edit documents, watch movies, etc. It is likely that cloud computing is making all this possible behind the scenes.

Today a variety of organizations ranging from tiny startups to government agencies are embracing this technology for the following:

- Create new apps and services as well as store, back up and recover data
- Host websites and blogs
- Stream audio and video
- Deliver on demand software services
- Analyze data for patterns
- Make predictions

Emerging Data Center Trends

The Emergence of SDDC

As the data center continues to evolve, there's an emerging need for flexibility, agility, and control. With web scale comes challenges that require new ways of approaching the data center. The current approach to address these issues is the software defined approach which refers to the idea of abstracting a physical data center resource from the underlying hardware and managing it with software.

The ability to virtualize the compute resources or the ability to create a new "server" with a few clicks or migrate a running workload between physical servers is the essence of the software defined approach.

The software defined approach is now starting to encompass all areas of the data center, which has led to the term software defined data center (SDDC).

In SDDC as many pieces as possible are abstracted into software. Compared to a legacy data center, the SDDC is more secure, more agile, and moves more rapidly.

The fallout of abstracting physical resources across the data center is that all of a sudden, the hardware is substantially less important to the big picture.

Commoditization of Hardware

Historically, computing has been enhanced by the creation of specialized hardware that is created to serve a specific purpose. ASICs are developed to serve one specific purpose. They have one primary application. While this model of computing can lead to increased performance, lower latency, or use of any number of desirable metrics as compared to commodity hardware, it also comes with substantial costs that must be weighed.

Some notable costs of ASIC-based hardware are:

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

Which is actually better? ASIC-based(custom) or commodity hardware?

The cost of custom hardware in terms of capital is generally more.

What is the cost (or risk) to an organization of becoming tied to the one particular vendor that makes the custom silicon?

What if the manufacturer goes out of business?

What if there's a blizzard and the parts depot can't get a replacement delivered for six days?

If it was commodity hardware, it could be supplied by a different vendor who is closer or not impacted by the severe weather.

Also commodity hardware is inexpensive and widely available, which are both significant advantages to an IT organization.

As the SDDC's goal is to abstract as much physical function into software as possible, the physical equipment becomes less important. This means that platforms which would previously have required special hardware can now be emulated or replaced with software and run on commodity hardware.

Commoditization allows for standardization. When many players in the market make products to serve the same purpose, there often becomes a need to create standards for everyone to follow so that all the products are interoperable. This is a win-win situation because the customer experience is good and the manufacturers learn from each other and develop a better product.

However ASIC-based computing isn't devoid of standards.

Shift to Software Defined Compute

Compute is a data center jargon meaning “CPU and memory resources.” In a post-virtualization data center, CPU and memory are grouped together as “compute,” and networking and storage are handled separately.

Software defined compute is the practice of controlling and automating abstracted compute resources.

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

The advantages businesses have seen from deploying software defined compute (which is just a different way of saying “server virtualization”) are broad and numerous.

Some of them include massive consolidation of physical resources, increased IT agility, increased performance and utilization.

Features of modern hypervisors allow for compute workloads to be migrated between physical servers without any downtime, and software intelligence places workloads on the most appropriate physical servers based on utilization.

Shift to Software Defined Storage

It is based on the idea of abstracting and controlling storage resources in the same way that had already been done with compute. **Software defined storage (SDS) could be thought of as storage virtualization.**

Similar to virtual server hypervisors we have storage hypervisors. The storage hypervisor manages, virtualizes and controls all storage resources, allocating and providing the needed attributes (performance, availability) and services (automated provisioning, snapshots, replication), either directly or over a storage network, as required to serve the needs of each individual environment.

SDS takes groups of physical storage medium, pools them, and abstracts them so that they can be consumed programmatically via the SDS platform instead of accessing each resource independently. So it is not only about abstraction, but it's also about control.

SDS will also include data services that allow administrators to optimize and protect the data stored. Some possible data services to include are:

- Thin provisioning
- Compression
- Replication
- Deduplication
- Cloning
- Snapshotting

One of the primary benefits of SDS: Heterogeneous storage platforms provide the look and feel of homogenous platforms to the administrator and to the applications.

In fact, SDS commonly moves the granularity of storage management from the storage aggregate and volume level to the virtual machine or virtual disk level. This allows far greater control and flexibility in the environment.

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

Whereas IT organizations used to spend large sums of money on proprietary monolithic storage arrays, SDS allows them to use alternate storage architectures that aren’t bound to the physical storage array.

Ex: One can place a handful of disks in each server (direct-attached storage) and logically aggregate those resources via software. Not being bound to the underlying hardware affords previously unknown levels of flexibility.

Shift to Software Defined Networking

It wouldn’t be a data center without networking! Networking is the backbone of everything that happens within and outside the data center. Network abstraction also unlocks new levels of scale and flexibility with technologies like VXLAN. Software defined networking enables control of the infrastructure/hardware via software.

On-demand cloud servers may be the biggest beneficiary of SDN. Because a tenant’s network environment can be created entirely programmatically without requiring any access to or modification of the underlying hardware, a button click can fire off a series of API calls that creates whole new networks in a matter of seconds or minutes.

Shift to Software Defined Security

If data in the data center is not secured it may result in large-scale data loss due to malicious breaches, thereby causing huge financial loss to the organizations.

Interestingly a report published by IBM in 2014 shows that 95% of the incidents that their security team responds to indicate that “human error” is partially or wholly to blame. Human error could mean anything from using weak passwords to misconfiguring network ACLs.

Either way, it would seem that if humans could be partially or entirely removed from the situation, a substantial number of security incidents might be avoided.

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

Software defined security is characterized by the abstraction of security management and policy from the devices and platforms that are providing security, and being secured. It allows for the automation of security policies and changes to said policies. Automating changes allows for higher precision which, in turn, leads to fewer security incidents due to human error.

Ex: Automatically deploy a software-based firewall for a new tenant and configure some default firewall rules that deny all traffic except for outbound traffic and inbound traffic on port 20.

The rules are not hard-coded in the automation, but are the result of policy applied to the tenant/application.

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

Creating advanced security via an abstraction layer is the security model of the future.

Parallel Paths of SDS and Hyperconvergence

By allowing commodity storage to be pooled across commodity servers while providing enterprise-class storage services, SDS also opens the door to a new data center architecture altogether. This data center Philosophy is called hyperconvergence.

It's the evolution of converged infrastructure, in which many disparate solutions are connected at the factory and sold as one package.

In hyperconvergence, those disparate solutions provided actually become one solution. That one solution provides compute virtualization, networking, storage, data services, and so on.

It's really many different layers of the SDDC that make hyperconvergence possible. Without SDS, the flexibility that makes hyperconvergence what it is would be impossible. Software defined storage (SDS) is characterized by Abstraction, Programmability, and Scalability, and is Policy-based.

SDS Features

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

Abstraction

SDS is an abstraction from the physical storage. It includes a type of storage virtualization akin to the way compute virtualization makes virtual machines independent of the underlying physical hardware. The strength of SDS is its flexibility which is made possible by abstraction. An SDS layer can provide the method for managing, automating, and scaling an already specialized storage solution.

Policy-Based

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

A practical example of policy-based management may be a policy that applies to a virtual machine. The policy could mandate that the virtual machine data is striped across a specific number of disks or nodes. It could also say that the virtual machine is snapshotted every 6 hours and snapshots are kept for 3 days onsite and are replicated offsite to keep for 7 days.

It might say that the workload must reside on Tier 2 storage. Imagine applying these specific settings to one virtual machine a single time. The task is not incredibly daunting, given the right software. However, imagine applying these same settings to 1,000 virtual machines in an environment where six new virtual machines are provisioned each week.

It’s only a matter of time before mistakes are made, and with each new virtual machine an administrator will burn time setting it up. With policy-driven SDS, simply by having applied the policy (created once), the virtual machines will be treated exactly as desired with accuracy and consistency over time.

Programmability

Management automation is the hallmark of the SDDC. For helpful and approachable automation to take place, the functions of a system must be accessible to third parties via the use of APIs. An API is a developer friendly way of exposing resources in such a way that another program can query them or manipulate them. EX: APIs like SOAP, and REST.

APIs uses some sort of orchestration engine to make all the pieces work together. That orchestration engine needs a way to interface with each of the individual components, and APIs provide that integration point.

The Programmability aims to allow anything and everything to be accessible via API.

Hyperconverged Infrastructure

Hyperconvergence is an evolution in the data center that's only just beginning to take hold. The past couple of years have seen hyperconverged solutions developing at an incredibly rapid pace and taking hold in data centers of all sizes. **Hyperconverged infrastructure (HCI) is the practice of combining multiple data center components into a single platform to increase Simplicity.**

Hyperconvergence is a data center architecture, not any one specific product. At its core, hyperconvergence is a quest for simplicity and efficiency. Every vendor with a hyperconverged platform approaches this slightly differently, but the end goal is always the same: combine resources and platforms that are currently disparate, wrap a management layer around the resulting system, and make it simple.

Simplicity is, perhaps, the most sought after factor in systems going into data centers today.

Hyperconverged infrastructure (HCI) aims to bring as many platforms as possible under one umbrella, and storage is just one of them. This generally includes compute, networking, storage, and management.

Hyperconvergence encompasses a good portion of what makes up the SDDC.

One Platform, Many Services

Convergence took many platforms and made them into one combined solution. Hyperconvergence is a further iteration of this mindset in which the manufacturer turns many platforms into one single platform.

Owning the whole stack allows the hyperconvergence vendor to make components of the platform aware of each other and interoperable in a way that is just not possible when two different platforms are integrated.

What characterizes hyperconvergence is the building-block approach to scale. Each of the infrastructure components and services that the hyperconverged platform offers is broken up and distributed into nodes or blocks such that the entire infrastructure can be scaled simply by adding a node.

Each node contains compute, storage, and networking; the essential physical components of the data center. From there, the hyperconvergence platform pools and abstracts all of those resources so that they can be manipulated from the management layer.

Simplicity

Makers of hyperconverged systems place extreme amounts of focus on making the platform simple to manage. Most effective hyperconvergence platforms take great care to mask back-end complexity with a clean, intuitive user interface or management plugin for the administrator.

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

A task like adding physical resources to the infrastructure is generally as simple as sliding the node into place in the chassis and notifying the management system that it's there. Discovery will commence and intelligence built in to the system will configure the node and integrate it with the existing environment.

Also, other things like protecting a workload are as simple as right-clicking and telling the management interface to protect it. The platform has the intelligence to go and make the necessary changes to carry out the request.

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

The short answer is that it's both.

Depending on the hyperconvergence vendor, the platform may exist entirely in software and run on any sort of commodity hardware. Or the platform may use specialized hardware to provide the best reliability or performance.

Neither is necessarily better, it's just important to know the tradeoffs that come with each option. If special hardware is included, it dramatically limits your choice with regards to what equipment can be used to run the platform. But it likely increases stability, performance, and capacity on a node (all else being equal).

The opposite view is that leveraging a software solution without custom hardware opens up the solution to a wide variety of hardware possibilities. While flexible, the downside of this approach is that it consumes resources from the hypervisor which would have served virtual machine workloads in a traditional design. This can add up to a considerable amount of overhead.

Which direction ends up being the best choice is dependent on myriad variables and is unique to each environment.

The Relationship Between SDS and HCI

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

One of the main advantages of pooled local storage is a highlight of the hyperconvergence model in general: the ability to scale the infrastructure with building blocks that each deliver predictable capacity and performance.

Hyperconvergence has SDS to thank for the fact that as this infrastructure grows over time, the storage provided to workloads is a single distributed system (an aggregation of local storage) as opposed to an ever-growing stack of storage silos.

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

The Role of Flash in Hyperconvergence

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

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

In a legacy monolithic storage array, there was one way of achieving additional performance for quite some time: add more disks. Each disk in a storage array can serve a certain amount of data at a time. This disk performance is measured in I/O Operations per Second (IOPS).

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

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

Flash storage is orders of magnitude faster than magnetic disk due to its solid state (non-mechanical) nature. A single solid-state drive (SSD) could easily deliver the IOPS performance of all 24 spinning disks. Because of this dramatic performance benefit, ***flash storage is critical to hyperconvergence.***

Physical limitations would not allow for the creation of a high performing hyperconverged system without the performance boost that flash can provide. Raw performance aside, SSDs can also provide high performing cache which can front-end a large amount of capacity.

Using SSDs as cache allows hyperconverged platforms to get high performance and great capacity numbers at the same time.

There are a number of different disk configurations that are used in hyperconvergence as shown below:

- **DRAM for metadata, SSD for cache**
- **SSD for metadata and cache, disk for capacity**
- **SSD for all tiers (“all flash”)**

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

Disk Configurations of Hyperconvergence

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

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

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

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

Where Are We Now?

The data center industry is at a major transition point right now. Flash storage coupled with the hyperconvergence of systems is completely changing the face of the data center.

The SDDC is already a reality in some environments, and is quickly on its way to becoming a reality in the rest.

The last few years have seen flash storage and software defined infrastructure growing in maturity, and both are finally getting to the point of being ready for the mainstream IT organization.

The cost of flash storage has been a challenge historically, but within the next year or two the cost of flash storage will drop to the point where it’s affordable for most situations.

Hyperconvergence will become more mature from a development standpoint but even more simple from an administration standpoint.

Thank You