NUTANIX

THE DEFINITIVE GUIDE TO

END USER COMPUTING

A Hybrid and Multicloud Approach to Virtual Desktops and Applications

Table of **Contents**

Author
Brian Suhr has over two decades of IT experience in enterprise infrastructure design, implementation, and administration. He has provided architectural and engineering expertise in a variety of virtualization, data center, and cloud-based engagements while working with high performance technical teams on global-reaching environments. As an independent author of the DataCenterZombie and VirtualizeTips blogs, Brian focuses on creating content that revolves around virtualization, automation, infrastructure, and evangelizing products and services that benefit the technology community. Find Brian on Twitter: @bsuhr

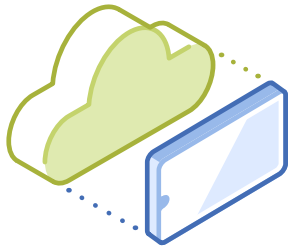
About this eBook
This book is focused on infrastructure design for VDI and End User Computing (EUC) environments. The content within this book has been adapted from the infrastructure-focused chapter of Architecting and Designing End User Computing Solutions (available on Amazon).

Introduction

After the selection of the right strategy and software vendor for delivering EUC services and applications, the deployment architecture and infrastructure choices make up the next big decisions for application and desktop virtualization projects.

The decision on where the control plane will run and how it will be operated is an increasingly important decision, as there are a growing array of options. Infrastructure is the foundation that one builds services on. Similar to electricity and water, we count on them, and they should just work when we turn the faucet or switch on. They can either be rented as a service or someone has to care for them.

Without a stable, highly available, and highly performant control plane and infrastructure, IT can face any number of challenges during the deployment and operational phases of your EUC project. While infrastructure plays an essential role, IT should not spend significant amounts of time deploying and maintaining it, which has too often been the case. The right infrastructure should just work, freeing architects and engineers to focus on providing the EUC services and applications.



EUC Deployment Architectures

Identifying the optimal deployment architecture depends on an organization's requirements, and this decision in turn influences an organization's choice of EUC control plane. Following are the different types of control planes and deployment options for EUC brokers. The control plane, as the name suggests, handles the provisioning, power, and brokering and is the primary interface for administrators. It usually functions as an API interface as well.

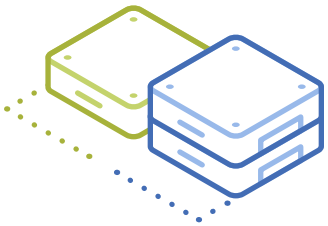
EUC control planes typically broker a user connection to an application or desktop. For applications, they broker through application presentation (typically RDSH based), and for desktops, they can use either hosted shared desktops (HSD) or a virtual desktop infrastructure (VDI) session.

Following are the three primary alternatives, along with some example use cases.

Traditional On-Prem Broker

With this EUC broker option, organizations typically buy licenses on a per-user basis and install in their datacenter. While several vendors and products fall into this category, Citrix Virtual Apps and Desktops (CVAD) and VMware Horizon are by far the most widely used.

When building an on-prem EUC deployment, the responsibility for the architecture and deployment falls to the organization or to outsourcing, including determining all the components of the broker software to be deployed, such as controller servers, database servers, licensing servers, edge security servers or devices, and any other required supporting services. As part of deploying these items, you also determine what high availability (HA) options are available for each service and select one that makes sense for your design requirements.



There is also typically a load-and-scale aspect to most of these services. You will need to properly size and architect how many connections each service can handle to determine the proper number of controller servers you need for, say, 5,000 connections with HA. You can then determine whether you're going to deploy all the controller servers for 5,000 connections up-front or add them as you scale your deployment. If you plan to scale past this number, you can use these details to continue to scale the different services and stay in compliance with best practices and supported maximums.

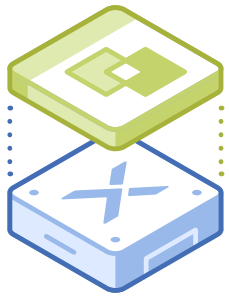
The next consideration is operations, where it's necessary to understand how to patch and upgrade all the services in the deployed EUC broker. Beyond the services we've already discussed (typically contained in all server VMs deployed), there are also agents and clients you need to maintain. The agents are contained in the images used to deploy the pools of application servers and desktops users connect to. How you update the agents depends on how you provision these pools. The clients are located on the endpoints users use to connect to these services, which vary greatly depending on the style of endpoint.

Despite the name, you can actually deploy a traditional on-prem broker in a public cloud. It nearly always makes more sense to deploy it on-prem, hence the name.

Cloud Broker

Cloud brokers are offered as-a-service in the typical Software-as-a-Service (SaaS) model and are most commonly referred to as a Desktop-as-a-Service (DaaS) offering. Several vendors and products fall into this category. Public cloud providers have their offerings, however since Citrix is the clear leader in the EUC market, Citrix Virtual Apps and Desktops Service (CVADS) offering is a logical place to start evaluating the benefits of DaaS. Most of the offerings provide a set of features on par with those provided by their traditional on-prem counterparts, but that's not to say they don't have their own benefits and limitations.

There are many different DaaS offerings available from software vendors, cloud providers, and service providers. They vary widely in what they offer when you look beyond the big features like VDI and application presentation, which is why it is important to understand your use cases and their requirements when making a decision.



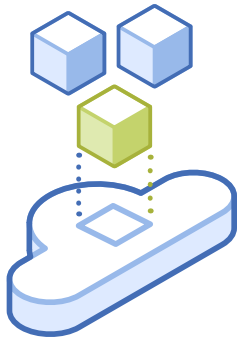
The requirements are your true north; if you like an option that doesn't meet some of the requirements, you have to determine if you can live without them until that gap is closed, if it ever is.

As a service that you subscribe to, they have several interesting characteristics. First, you typically pay for the service per-user per-month, which makes the cost easy to calculate and track. The cost also comes from the OpEx budget, which is a benefit for some as they're forced to move to the cloud consumption model. You can sign long-term deals with these DaaS offerings that probably offer better pricing in turn for a longer commitment; if you subscribe, though, you have the option of exiting if your demand decreases or disappears. With the traditional license approach, organizations typically buy licensing and then pay for perpetual support.

Given that it's a service, organizations can consume it on demand. All the design, deployment, and scaling aspects of the brokering layer are handled for you. Also, you're typically running the apps and desktops in the public cloud next to the DaaS offering, so you don't have to manage the infrastructure these are running on either, which in turn gets you out of monitoring and upgrading these layers. These DaaS offerings can potentially get your project off the ground faster and are less effort from the operational side.

While the DaaS service covers the brokering and infrastructure layers, there remain operational tasks to perform. These duties include image building and updates, application installs and updates, user data, VPNs, and so on.

Another benefit that cloud offers is the ability to potentially use data centers around the world to offer pools of applications or desktops near a group of users. This benefit doesn't always work out. For example, if the users have a tight reliance on data or a specific app in a datacenter that is far away, this ability is null.



Hybrid Deployments

Now that we've discussed the different architectures for EUC brokers let's look at how they're being deployed. Despite some predictions that all workloads are going to public cloud, it's clear that most deployments are still on-prem and that most organizations are moving toward using a hybrid architecture. For the purposes of this discussion, hybrid simply refers to a mix of on-prem and public cloud. The percentages vary for each depending on a number of factors. When it comes to hybrid in the EUC world, there are ways it works for both traditional and DaaS offerings. Let's first look at how each of the architectures can fit in a hybrid world and then dive into some sample use cases.

The DaaS solution is the most common approach to hybrid offerings, and not all DaaS vendors offer the ability to deploy in a hybrid architecture. The control plane typically remains in the cloud for the DaaS version of hybrid and those that support it have a method to control and manage the private resource pools, usually by deploying a local VM on your on-prem clusters to act as a local connector. Through these cloud connector VMs, the DaaS provider can now provision VMs, control their power states, and broker connections to them. The connector VMs are easy to set up and you can deploy them in a pair to provide HA. This is the architecture that Citrix CVADS, Horizon Cloud and Nutanix Frame support.

Overall, the DaaS version of hybrid is still far less complex to design, deploy, and operate than the traditional approach. You're still responsible for managing the infrastructure the on-prem VMs are running, but you don't have to manage the broker layer.

Use cases

Let's talk about the different scenarios and use cases that are a good fit for hybrid. If used correctly, a hybrid design can offer great flexibility in terms of cost and features. Most use cases fall into three different scenarios.

- **Disaster Recovery (DR).** This scenario probably fits most traditional on-prem deployments with a need for a business continuity (BC). Historically, DR involved building or renting a secondary site and using it to deploy EUC resources for failover. The public cloud offers a very attractive alternative to the legacy approach in the sense that you can reserve some cloud capacity and then burst it out to full capacity in the event of a disaster. In this approach, you pay to run your broker infrastructure VMs and a small pool of desktops constantly with storage capacity for the replicated user data and profiles. Then, should an event happen, you can quickly expand that pool of desktops to whatever size you need and point your users to it to begin working again. When all is done, you shrink back down to the steady-state size to reduce costs again.

- **Bursting.** There are some use cases or projects that only need resources for a short time, and it might not make sense to keep space on-prem for them. In this case, you spin the resources up, typically in a public cloud, and destroy them when the need goes away. There are a range of scenarios that could effectively use bursting. Student labs are a common use case, because the needs are seasonal and scheduled around the student calendar. Students may also have GPU needs that you don't have on-prem. Special projects and temporary work are also common bursting use cases.

- **True Hybrid.** This option likely captures most of the other use cases in the sense that they aren't temporary needs, which means you're regularly running some workloads on-prem and some in the cloud and decide where to deploy based on parameters that makes sense for your organization and design. For example, you may have a use case where you need GPUs and you don't have them on-prem; you may need to deploy in the APAC region rather than have local users traverse back to your North American sites; or you may simply be out of capacity in your on-prem environment.

Architectural Principles

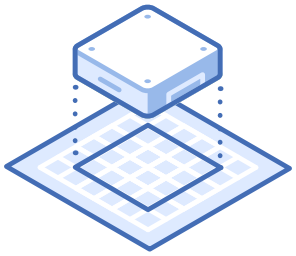
There are several important factors that should be considered in the EUC infrastructure design process. By using these factors along with the organization's requirements, one is better able to consider what the architecture alternatives are going to be. The following factors should be considered when evaluating architecture alternatives and vendor options on EUC projects:

- Entry point
- Scalability
- Performance
- Monitoring
- Capacity

Entry Point

The entry or starting point for infrastructure can often be a make-or-break decision on a project. This is how much infrastructure and cost it will take for an organization to start the application or desktop virtualization and delivery deployment based on different starting point sizes.

If the project is planned to reach 10,000 users when fully deployed with the starting deployment phase of 5,000 users, the organization is probably less inclined to be shocked by starting costs. The reasoning is that depending on the type of infrastructure that is selected, the per-user cost may not begin to make sense until you have deployed a few thousand users.



Architectural Principles 10 The flip side of this is that if an organization is going to deploy 10,000 users, but only intends to start with 500 users and scale up at a steady pace over the project timeline. They are going to look more closely at the cost of the initial infrastructure deployment at this size over taking a larger first step. The per-user cost at this size can hold steady as the environment scales, or it can look really skewed in the beginning, due to a larger starting infrastructure spend.

While per-user cost can be seen as vague and almost irrelevant as a factor to determine your infrastructure costs, you are going to be asked about this when trying to sell the project to the business or justifying your infrastructure selection to leadership. If you choose an alternative that has a higher up-front user cost, you need to be prepared to explain the details. Evaluate solutions that you believe would be better suited for your environment. Otherwise, be prepared to define the decision on how costs will play out. A sample of these two scenarios is shown in Figure 1.

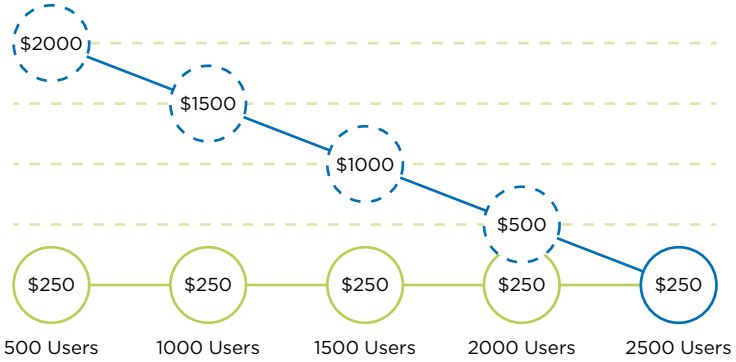


Figure 1:
Entry Points Per Desktop

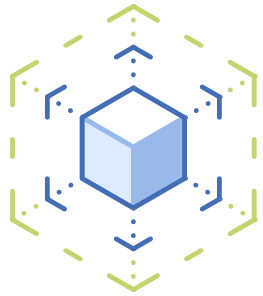
Scalability

The scalability of architecture is an important factor when evaluating project viability. An architect will need to understand the starting size options for the different alternatives; this loops back to the entry point topic that was just covered. Will the alternative easily allow the design to start at a smaller size as required, or will an organization need to purchase more infrastructure than would be needed to satisfy a project's starting size—and not be able to utilize all the resources until the project grows into it?

Aside from how small of a scale the alternative can start with, it is equally important to consider how large the alternative can scale to. If the desire is to start at 500 and still be able to scale to 10,000 users, what will the alternative look like at both ends of that spectrum? Will the organization be happy with the low or high points—or both?

The scalability topic is not just a storage discussion. It also holds true for compute, networking, and possibly other layers within the design. If adjustments are made to the configuration of the compute layer to achieve a smaller density of VMs per host server, how might this affect different design choices when scaling? An example would be if the initial host design starts with 128GB of memory per host and the final choice is 256GB or larger, one will need to ensure that the right size DIMMs are used to allow for the configuration to be scaled in the future. If the wrong choices are made to save costs, it will affect the density due to constraints, or cost more in the long run with DIMMs that were unable to be reused.

The architect should focus on how the solution will be able to start small, as well being able to scale to the largest point. But one cannot ignore all the points in between either, because depending on how one scales the deployment, there could be many scaling points in between the start and finish. It is ideal to look for something that is going to allow the design to easily scale in buckets of user counts that the project identifies, while not outpacing the deployment timeline and capabilities. The ideal scaling bucket size for a project may be in increments of 500-1000 users. But if the architecture alternative chosen scales greater than this, understand how this affects the costs and deployment.



Performance

EUC performance as measured by end user experience is always a top consideration. Your selected architecture must be able to meet requirements at any phase of the project. This can be a tricky path to walk with some alternatives. If one scales a solution down to meet the minimum starting user requirements, it may end up sacrificing performance if they are unable to scale linearly. Architects do not want to make compromises in the architecture to reach this small starting point that may affect the overall maximum performance options of a solution. If you spend time upfront making the right decision, you can avoid issues later on.

An EUC solution design will typically have many different performance requirements. Select an architecture alternative that is flexible enough to meet all performance requirements within a single option. Whether the design will provide several types of EUC services or just focus on app and desktop virtualization, multiple performance needs must be accounted for. Understanding how each alternative will or will not be able to meet individual performance requirements will heavily affect your evaluation and design process.

Capacity

The capacity discussion is similar to the performance one. There are a number of different capacity requirements within EUC designs that will need to be provided. The solution will call for running server VMs, desktop VMs, applications, user profiles, and user data for this type of architecture. Each layer within the design may have very different capacity requirements. Some use large amounts of data that typically deduplicate well. Other portions, such as user profiles and data, consist of smaller amounts of data per user, but multiplied by thousands of users, turn out to be a large portion in the end.

A larger problem in years past was purchasing too much or too little capacity, while trying to achieve the required performance levels. Closely look at architecture alternatives during the design phase to see how they will be able to provide required capacity, while ensuring that minimum performance requirements are also met. The alternative should not provide 2-3x or more capacity to meet storage performance requirements, or add significant additional performance to meet capacity requirements.

The ideal solution is one that allows enough flexibility to scale performance and capacity at similar rates, so neither gets too far out of pace from the other.

In the past, this topic has caused a lot of issues. Many organizations have gotten themselves into performance and capacity planning trouble by scaling capacity faster than performance. Just because the solution has 5TB of free space does not mean it is able to scale by another 500 users. This scenario may cause the performance to suffer greatly. Administrators and IT leadership that do not have a solid understanding of how the solution scales can fall into this trap.

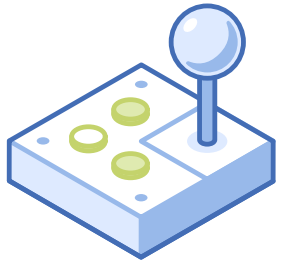
Monitoring

Monitoring is very important and often overlooked. When it comes to monitoring infrastructure in an EUC environment, administrators typically focus on the performance aspect. They need the ability to understand what is normal and when there is an active issue.

Using the monitoring should be simple, while still providing a wealth of detailed information. This is not the case for many manufacturers, so one should look closely at what the monitoring experience is with each alternative.

Another requirement is the ability to provide performance monitoring at the virtual machine level. Unfortunately, the majority of infrastructure vendors still cannot offer this level of visibility into the virtualization environment. Best-in-class infrastructure performance monitoring should give admins the ability to quickly look at the storage layer and determine if the storage performance issue is global or if it's isolated to a host, group of VMs, or just a single VM.

By managing storage performance at the VM level, one can use a similar approach to managing the CPU and memory performance of a VM at the host level. Administrators need to know if a VM is temporarily using additional performance, or if it is a regular consumer of more storage performance than typical users. This will allow one to understand when there is a spike and when to be looking into something further to identify the issue.



Building Blocks

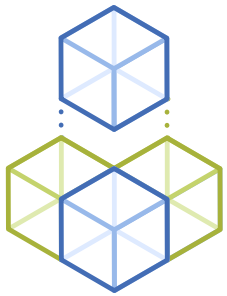
A building block is a predefined set of infrastructure that maps to a specific amount of resources or number of users. This approach is one of the best ways to approach infrastructure design with end user computing.

By using this approach, one can develop an architecture that offers a predictable cost, performance, and capacity scaling model. When determining building block size, choose what increments you need to scale users and how the infrastructure selection can accommodate the choices. For instance, one may want to scale users in increments of 50 to 100 users, but the infrastructure choice does not scale in that small of increments well. This may force the design to scale in larger increments of 500 or 1,000 users. If the infrastructure choice scales in large blocks, one can choose to scale to mesh with that or just accept the fact that the infrastructure costs will not scale in the same way the user deployment blocks will. This simply means that the organization would be purchasing infrastructure in blocks of 1,000 users and only be deploying in groups of 50 to 100 users.

It does make the costs of the virtual desktops or user sessions look expensive when purchasing the large block to deploy a smaller amount of users. This evens out if the organization does deploy all of the planned users.

Building block style architectures are helpful in any design project, but EUC deployments always have common chunks of users and use cases that have similar characteristics and are deployed in groups. To continue with the example of a 100 user block size, by understanding the resource requirements of 100 users, one can ensure that the block of infrastructure is able to provide everything those users require.

If each user requires 15 IOPS at steady state and 30GB of storage capacity, along with 2GB of memory, and 200MHZ of CPU, the architect then knows that the building blocks must provide 1500 IOPS, 3TB of capacity, 200GB of memory, and 20GHZ of CPU. The architect can design the building blocks to contain additional resources, but none of them can be below those values. We also want to avoid the waste of including too much extra in each block that we can't utilize.



Building Blocks 15 With this approach and granularity in the design, one can now scale the environment in groups of 100 users. This allows for a slow and steady approach and provides predictable values that organizations can plan around for deployment, performance, capacity, and costs. If organizations want to scale faster and in larger quantities, they just drop in multiple building blocks at once.

Lastly, the building block approach has proven attractive as a majority of customer deployments like to start with smaller deployments and scale up from there. The "start small and pay as you grow" model enables them to invest smaller amounts of capital up front, and to gain experience as the deployment grows. The next section covers the different types of infrastructure architectures available today and how each of them supports or doesn't support the building block approach.

Hypervisors

The hypervisor is an important layer in your infrastructure design. It's directly responsible for a healthy share of the performance, availability, resilience, and manageability of your solution.

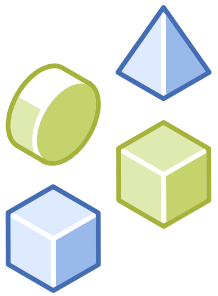
Hypervisor Landscape

In today's virtualization environment there is a solid list of enterprise-ready hypervisor options. The list narrows when you look at hypervisors that routinely have EUC and VDI use cases deployed on them, and consists of the following:

- Citrix Hypervisor (XenServer)
- Nutanix AHV
- VMware vSphere (ESXi)
- Microsoft Hyper-V

Reasons to consider change

There are any number of reasons for organizations to consider changing their hypervisor, ranging from simplifying the architecture and operations of the hypervisor layer, to increasing security, to reducing vendor lock-in and eliminating costs.



Evaluation Criteria

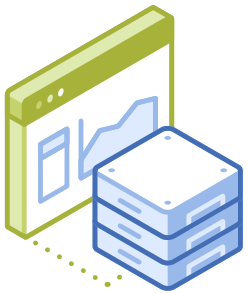
Hypervisors are complex pieces of software that are comprised of hundreds of possible features. Basic features include resource scheduling, high availability, virtual networks, and multiple OS compatibility. These are now commodity features that are readily available in all of the hypervisor alternatives discussed above.

Beyond this, organizations should focus on the value each alternative offers the business and how it might change or improve operations.

A good place to begin is to look at the design phase for the different hypervisor alternatives. Use the design requirements for a recent project or an upcoming one as the solution that you would be architecting for. Develop an understanding of the amount of effort to design a solution for the selected project. Note if this phase consumes days or weeks of effort to design the hypervisor layer of the solution due to the complexity of design choices. Ideally the hypervisor should deliver all the feature and functionality required, while simplifying the design phase down to a few clear, simple choices.

Next be sure to understand what the deployment effort looks like. Based on the proposed design selected in your evaluation, identify the effort required to deploy the solution and how this differs, if at all, from a basic deployment. Ideally just like the design phase, the deployment phase should require minimal input from the engineer and be highly automated. Deploying a new cluster is not something that should require days to weeks of effort.

And finally taking a close look at the operations side of a hypervisor will help understand any difference from your current efforts. Patching and upgrading a hypervisor has historically been one of the larger operational efforts. Do any of the hypervisor alternatives offer any advantages around simplifying this process in both terms of effort and reliability of upgrades? Apart from upgrades, how does the daily management of virtual machines look? Can this be accomplished from a single simple interface?



Infrastructure Alternatives

There are currently three primary architecture alternatives for application and desktop virtualization, or EUC solutions broadly. The alternatives are build your own (BYO), converged infrastructure (CI), and hyperconverged infrastructure (HCI).

Build Your Own

The BYO infrastructure alternative is really just what the title implies, the architect or team independently chooses products that they like or feel are best of breed. This alternative results in a significant increase in the upfront planning and research period, as the team must evaluate each product separately and how they may or may not work together.

This alternative also provides the ability to select and follow a reference architecture that a vendor has published for the type of solution that is being built. These reference architectures are typically published by a single vendor and focus on their product. These DIY reference architectures can save time and reduce some risk, but they do not always apply to your design requirements, use cases, and environment.

At minimum, a BYO alternative for an EUC-based design will contain compute and storage resources. You may be able to use existing network connectivity, so it may not be a component of this alternative. Figure 2 illustrates a simple example of the parts of a BYO alternative. With flexibility in scaling, costs are fairly predictable; the only exception would be on the storage side. Depending on the maximum size of your design and the storage choice made, you may require multiple storage arrays or appliances. As you scale the storage and need to add a new array or appliance, the cost will spike at those points.

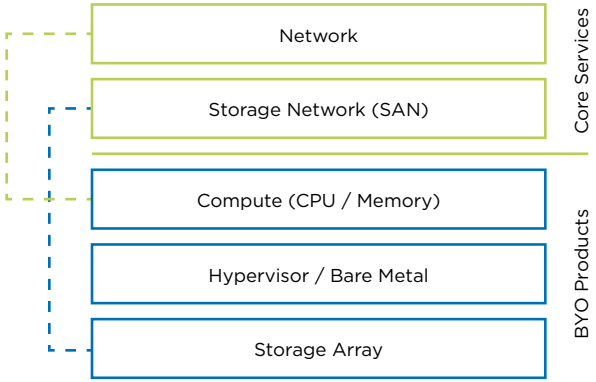
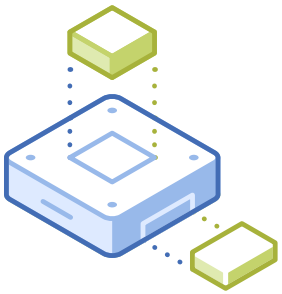


Figure 2:
Bring Your Own (BYO) Infrastructure

Anytime you are assembling a number of products from the same vendor or multiple vendors without prior experience, there is additional risk. There will be a level of uncertainty about the performance and reliability of the solution until the actual infrastructure is purchased and deployed in the architected manor.

If one can accept the unknowns and additional risk, the BYO alternative does maximize flexibility. Since one is able to make nearly any vendor and product decision that is capable of working together, this allows you to stay with existing vendors you have had good luck with, while moving to new vendors in other areas.

The BYO alternative is able to scale the compute and storage resources independently. The only limits to the scaling method or the maximum size would be a constraint of the individual product choice. Since products are purchased separately, there are no minimums or set amounts the products need to be scaled in. This allows flexibility in trying to account for the building block approach mentioned earlier.



Converged Infrastructure

A converged infrastructure (CI) alternative is an architecture that was brought to market around 2010. Converged infrastructure offerings typically offer the same products that might be selected as part of the BYO alternative, and package them together into a productized solution. This means that a CI vendor will include computer, storage, and networking in their offering. Typically, most CI offerings contain products from multiple vendors and be included as part of a single offering, or a vendor can offer all the layers of a CI offering from their own product line. Figure 3 illustrates a simple example of a converged infrastructure alternative.

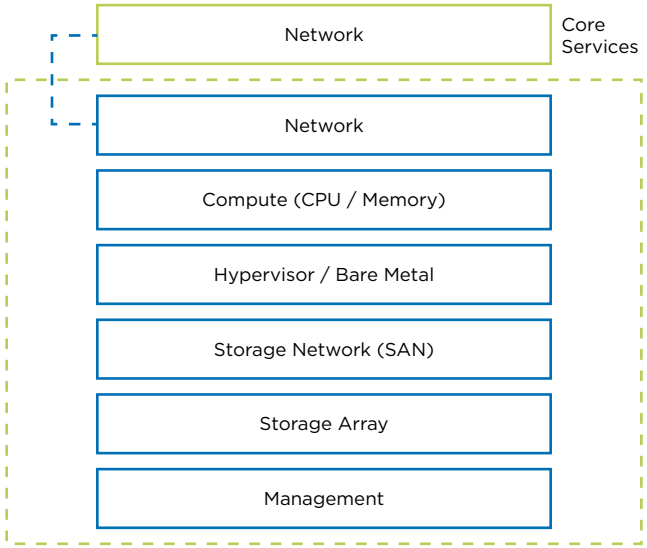
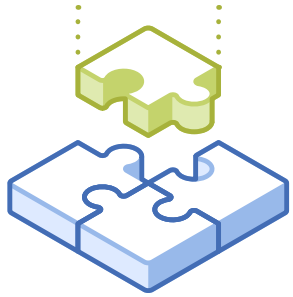


Figure 3:
Converged Infrastructure

A converged infrastructure offering will enable you to purchase familiar products that have been packaged into a single solution. This can be thought of as a reference architecture that can be purchased as a product. Depending on the CI product that is evaluated, the product may or may not offer any additional convergence than if you purchased the products separately in a BYO alternative.



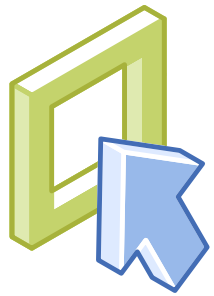
Typically, most CI vendors and products will offer the ability to purchase all of the infrastructure parts in a single product SKU. The CI vendor should be able to offer single call support for the entire CI solution, which means that the CI vendor can support all of the products within the solution. This is an added benefit as it allows customers to eliminate the need to work with multiple vendors in the troubleshooting process.

With most CI offerings, there is a limited number of products offered within the solution. This allows the CI vendor to pre-test and validate all parts and pieces to ensure they work properly together, removing much of the risk in the BYO alternative.

Even after several years of CI products being sold in the market, little has been done by the CI vendors to simplify the management of these products. With CI offerings that include the same products as the BYO alternatives, one will typically manage both alternatives in a similar and dispersed manner. This alternative may converge the purchase and or some of the products, but it typically does not converge the daily operational management of the solution.

A converged infrastructure product should be able to scale the resources within it independently of each other. This would mean that you can just add compute, although there may be minimum increments that one can scale in. The other resource that would be scaled in a CI offering is storage, and this will be heavily dependent on the type of storage solution selected as part of the CI offering. A converged infrastructure product will have a maximum size, meaning it will have a limit on the number of servers it can support, and a storage limit based on the included storage array.

Scaling limits of a CI offering are typically fairly large, but at some point as one scales the resources within the CI product, it will hit the maximums. To continue to scale the design at this point, one will need to purchase an additional CI product. This will cause large peaks in infrastructure costs at different points of the scaling process depending on the maximum size of your design.



Hyperconverged Infrastructure

The hyperconverged architecture was introduced to the market approximately one year after CI. True hyperconverged architectures are achieved by converging the compute resources, storage resources, and management layer into a single product. It is possible to deploy a hyperconverged solution in a software-only (SWO) model or in an appliance model that includes purpose-built hardware.

By including a hardware appliance as part of the product, the vendor can now include the management of the infrastructure along with the other resources that are being converged in the product. Figure 4 illustrates a simple example of a hyperconverged infrastructure alternative.

The software-only architecture can provide great flexibility at the hardware layer, allowing choice of platform to deploy on. When considering SWO options they should still provide an appliance-like experience. This is different from offerings with a loosely coupled hardware compatibility list that only incorporates minimal testing.

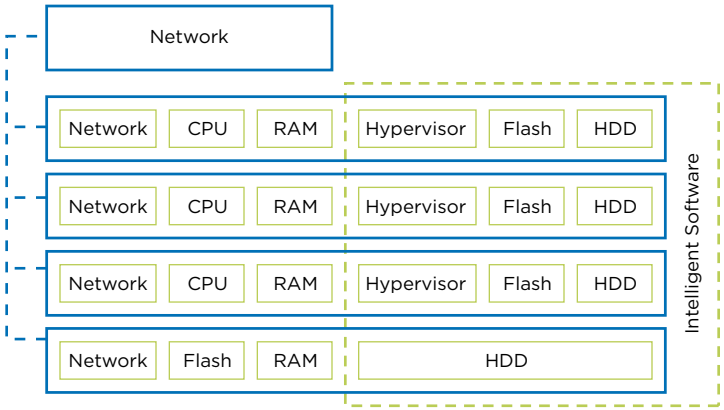
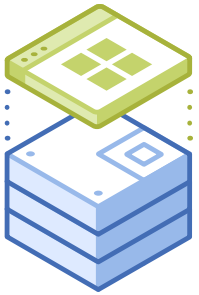


Figure 4:
Hyperconverged Infrastructure



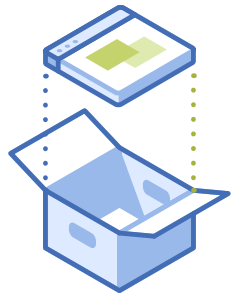
Simple Installation – The leading HCI products should install nodes within minutes and hours, not days or weeks, using a highly automated process.

Easy Scalability – The product should be easy to scale up or down. The addition of new nodes to the environment should happen easily and quickly through the management interface.

Modern management – A modern management interface must focus on the virtual machine (VM) as the point of management. An administrator must be able to see how VMs are performing, the amount of resources each VM is consuming, if there are any events or errors, and be able to easily pull reports based on VMs.

Extensibility – You must be able to integrate the infrastructure with other parts of the solution easily and control it programmatically. This requires the HCI product to offer an API, and possibly another method, such as PowerShell cmdlets. With an API, you will be able to automate the communication and control among products to further reduce the effort and increase the accuracy of the environment.

Performance was intentionally left out of the HCI benefit list because everyone expects a modern hybrid or flash-based solution to perform well. HCI is about creating an infrastructure layer that is simple and efficient. It enables teams to stop spending time turning knobs and enables them to provide additional value to the business at the automation or application level.



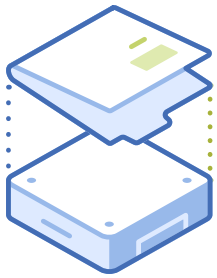
Storage Requirements

There are a number of different storage resource requirements that exist with any EUC design. It will need to account for server-based VMs, user data, and virtual desktop infrastructure (VDI). The VDI storage requirements will be the most demanding within the environment and are also the ones that cause most VDI projects to fail or suffer from a bad experience.

For this reason, the storage portion of this eBook focuses on the needs of the VDI service of the solution. The needs of each virtual desktop can often seem small and insignificant, but when you combine them into large groups as the storage scales, performance demands can easily overwhelm storage that was not properly designed to meet these needs.

If each virtual desktop averages 15 IOPS and one expects 2,000 concurrent users, that amounts to 30,000 IOPS. That number is pretty large and could overwhelm the average storage array. But one cannot simply design the storage solution to meet the average I/O of the environment, the design must account for peaks, including desktop boots and user login events.

Virtual desktop workloads have very spiky I/O, which makes them very different from other types of workloads within the average enterprise datacenter. For example, opening an application like Outlook for the first time in a session can generate upwards of 1,000 IOPS for that one user session. That is far beyond the average 15 IOPS discussed earlier. An example of different application IOP impact is shown in Figure 5.



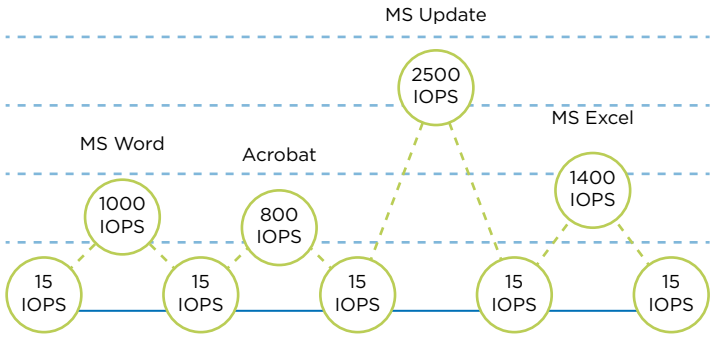
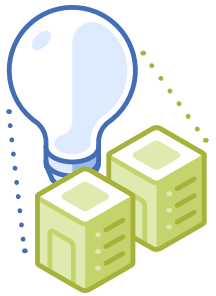


Figure 5:
VDI IOPS

Other deployment and operational items, such as patching and environment refreshes, can also create tremendous spikes in IOPS and will affect performance if not accounted for and planned accordingly. If one deploys another 50 virtual desktops, that action can create a significant I/O spike. For these reasons, the storage architecture must be designed to accommodate peak IOPS from maintenance operations.

There are a number of ways to architect VDI solutions with full clones or shared image presentation and each can have different effects on storage requirements in terms of both capacity and performance. Since full clones consume additional capacity and storage, deduplication will be important. Full clones must also be patched independently, which will increase the I/O during those operations.

The shared image approach that Citrix offers with MCS or PVS, and VMware with linked clones, presents different I/O challenges. By nature, these shared image approaches require less storage capacity since the parent image is shared and each virtual desktop is only consuming a smaller amount of space for its unique data. The shared image has different performance requirements than the typical VM. This image is now used by hundreds or thousands of virtual desktops and must be able to generate large amounts of IOPS. If the shared image is a bottleneck, all virtual desktops using it will be negatively affected and user experience will be bad.



Taking into these considerations for peaks and different types of app/desktop virtualization architectures one must select and design a storage solution that is capable of meeting the peak boot, login, and steady state demands of the environment. To understand the storage requirements of the design, one should perform a desktop assessment on the existing physical PC environment. This desktop assessment will gather the real performance and capacity details from the user base so that one can apply these to the design calculations.

A final thought on app/desktop virtualization-related storage requirements is that aside from being very unpredictable in the I/O side of things, desktop workloads are also very write heavy. Unlike many server workloads that are mostly reading data and serving it to users, desktops are typically spending more time writing to disk. Writes are more intensive to the storage array than reads are. A typical server workload might be 80% reads and 20% writes, while the steady state virtual desktop workload might be the opposite. When evaluating your storage choices, be sure to pay close attention to how the storage solution buffers and commits writes, versus some large promise that it does an “excellent job” at caching commonly read blocks.

Storage Types

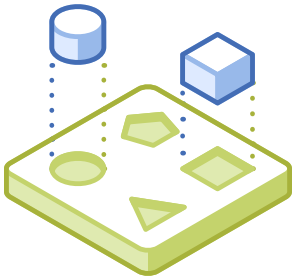
There are a number of different types of storage. The primary storage alternatives available today are legacy tiered-storage arrays, hybrid flash arrays, and all-flash arrays. Each alternative takes a different approach to providing performance and capacity to workloads. Within each alternative, vendors take different approaches in building their offerings, so a brief explanation of each is listed below.

Legacy Tiered Architectures

These are the legacy enterprise arrays that have been used for server-based workloads for the last 10-20 years. They are typically dual controller-based architectures, and within the last decade, have been modified to allow for multiple tiers of performance and capacity disks to be included in the architecture. Different tiers of disks are provided to try and service the capacity and performance demands of dispersed workloads. There are two options in this approach: you can design for performance by creating dedicated pools of high performing disks for a workload, but this can be very expensive and limiting. The other option is to try and take advantage of tiering that was added to this architecture to ask the array to promote or demote blocks of data based upon demand. The trouble with this auto-tiering is that it often takes too much time to make those decisions for VDI workloads.

All-Flash

All-flash storage arrays are entirely made up of flash-based storage. There are many different types of flash that can be used within these storage arrays. Modern all-flash arrays were designed to take advantage of the characteristics of flash storage, meaning that the operating system and file system were designed with flash in mind. Some products have taken a legacy array design and simply replaced the spinning disks with all flash. While this is still faster than the older option, the final product was not designed for this purpose.

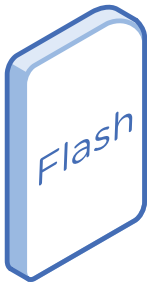


All-flash storage arrays are very fast, with only one level of performance in the product. To ensure the array can also provide the capacity required for the design at an affordable price, you should look for arrays that offer deduplication and compression. While nearly every modern all-flash array is easier to manage than their legacy counterparts, they don't always offer the same ease of management and per-VM management that many of the hybrid flash offerings do.

Hybrid Flash

Hybrid storage arrays are modern architectures that were designed to efficiently use a combination of flash drives and spinning disks. Vendors have taken different architecture approaches on how they use capacity and performance in their arrays, but the end results are similar. They are all able to offer impressive performance from a smaller amount of flash, while still providing a large amount of capacity by storing data on large spinning disks in the array. Ideal hybrid storage architecture alternatives use built-in intelligence to automatically tier data across flash and disk drives based on demand, eliminating the need for manual tuning and potential performance pitfalls.

The architectures that are the best fit for a modern VDI design are hybrid and all flash storage architectures. These architectures are capable of providing the performance required for VDI environments and typically also offer the modern management experiences discussed earlier. VDI workloads are very unpredictable by nature, and if your storage solution must wait to make storage decisions or promote blocks to a caching tier, the performance demand will be long gone before that happens and the experience will have been negatively affected.



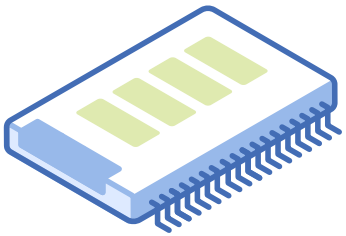
GPUs

A Graphics Processing Unit (GPU) is an important part of a modern desktop experience. Physical desktop and laptop computers have always had a GPU included, but were not important unless you wanted to play video games or do high-end graphics-intensive work. While these still hold true, modern operating systems and new workloads such as machine learning (MI) and artificial intelligence (AI) are also taking advantage of GPUs to enhance their performance.

In the VDI space, GPUs are not generally required and are typically evaluated on a use case basis to see if the added performance is required versus balancing additional costs versus perceived value for use cases with lower requirements. Windows 10 or watching streaming video are examples of use cases with lower requirements that can benefit from GPU but may not deliver enough value as compared to offering an acceptable user experience without GPUs at a lower total cost. However, as with CPU manufacturing, GPUs are gaining performance at a lower cost and the value derived from GPUs needs to be part of an overall evaluation when designing an EUC environment.

NVIDIA is the best performing and most known manufacturer of graphics cards designed for desktop virtualization. AMD's graphics cards work only in certain use cases and do not deliver the same optimizations that NVIDIA cards offer.

NVIDIA Grid boards allow GPU virtualization, enabling multiple users to share a single graphics card. GPU virtualization not only supports higher user densities, but also delivers native performance while accessing a virtual desktop. NVIDIA GPUs also have an engine for H.264 encoding that offloads processes from the CPU, which further increases user density on your hardware. NVIDIA Grid cards typically have multiple GPUs, which improves scaling.



Dedicated GPU

With GPU passthrough, you can create a VM with a dedicated GPU. This configuration provides user experience comparable to using a fat client with a high-end graphics card. However, assigning a GPU core to a single VM, either a hosted shared desktop (SBC) or a hosted private desktop (VDI), limits scalability.

Shared GPU

Grid technology lets multiple virtual desktops share a GPU, while offering the same user experience as native GPUs. This sharing is commonly known as a vGPU and is a function of the Grid software and hypervisor. An NVIDIA Grid M10 card, for example, has four physical GPU cores that can host up to sixteen users per core, resulting in 64 users, each with a vGPU-enabled desktop, per M10 card.

The GPU processes VM graphics commands directly, which means that users get high-end graphics without a performance penalty due to hypervisor interference. vGPU is more scalable than passthrough, as we assign vGPU profiles to our users and thus get more users on the same card.

vGPU profiles deliver dedicated graphics memory through the vGPU Manager, which assigns the configured memory for each desktop. A vSphere installation bundle (VIB) installs the vGPU Manager on the hypervisor. With AHV, an RPM package manager performs this task. Each VDI instance has preset resources based on the applications' needs.

Grid Licensing

Something unique to NVIDIA is that to utilize the vGPU functionality, licensing is required. There are different levels of licensing for virtual apps that are used for RDSH based solutions. Power user and designer for most VDI use cases covers high-end applications such as Adobe applications, engineering, and CAD applications. The Grid licensing is available in named or concurrent user options.

File Services

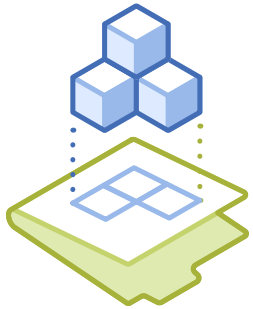
File services are widely utilized and an important layer in VDI solution architectures. File services have traditionally been provided by a NAS device or a Windows server based device. This is presented as an SMB share that different services and guest OS devices consume for shared or private access.

Within a VDI design there are typically several different requirements for file services. Managing and capturing user profiles are important to provide a persistent user experience and most profile management solutions store the data on SMB shares. It is also very common to redirect folders as part of the user's profile to an SMB share. And lastly the user created data, such as documents, media and images. are stored on private or shared folders on SMB shares.

Nutanix File Services

Nutanix offers Nutanix Files as a native feature in the Nutanix cloud platform. Files is an integrated scale-out file service platform that is managed from Prism along with all other Nutanix functions. This allows for the simple 1-click deployments and non-disruptive upgrades. This simplification allows for the possibility of allowing VDI teams to manage more of the complete solution when desired.

Files provides a highly scalable architecture that allows for additional capacity to be added to file services VMs with 1-click to allow for additional capacity or user connections. Files instances can run in the same cluster and your server or VDI VMs or in a dedicated cluster, if desired. To allow for flexibility in connectivity, Files supports SMB 2.1, SMB 3.0 as well as NFS V3 and V4 connections.



Compute Sizing

There are different schools of thought on sizing the compute layer of the design. The first is the scale up approach, which uses fewer large hosts to provide resources, while the scale out approach uses more small hosts to provide resources. The preferred method is somewhere in between the two approaches, one that utilizes 2 socket hosts and makes them as dense as possible without violating the consolidation ratios set as part of the design. This eBook is focused on helping size the compute resources for the VDI workload.

There are three primary calculations that one will be focused on when sizing the compute resources in the design. They are the amount of physical memory in each host, the amount of CPU clock speed and the number of CPU cores and the CPU ratio for them.

Physical memory

First and foremost, one should never overcommit memory in a VDI design. Violating this rule has very little value and will only lead to performance issues in the environment.

CPU clock speed calculation

The CPU clock speed calculation depends heavily on the details gathered in the previous desktop assessment. Reports from the assessment will provide the amount of CPU that user sessions used on average and peak. One will use those details along with the memory details from the assessment to make the calculations.

Host utilization limits

A couple of other host and virtualization cluster recommendations are to never exceed 80% host utilization and always size your cluster for N+1. The 80% host utilization is not just for app/desktop virtualization deployments, it's a recommendation that applies to any workload running on a hypervisor. If you are running your hosts past the 80% mark you have very little room for peaks and may also not have enough resource overhead to account for a host failure, depending on the size of your cluster.

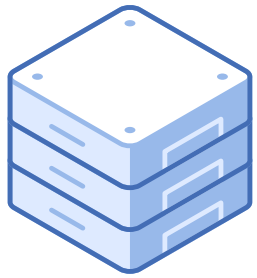
Always size your cluster for N+1

The second item of figuring for N+1 in your cluster sizing is to ensure that there are enough resources in your cluster to account for a single host failure, to ensure that all VMs can keep running and failed ones will restart without issues. A single host failure is the most common level of resiliency; there is a small set of customers that require N+2 to account for higher SLA requirements.

CPU ratios

The final item on the compute sizing topic is the CPU ratio, which focuses on the number of virtual CPUs to physical CPUs (vCPU:pCPU). This ratio is very important because if one goes too high with this ratio, it will reach a point where a CPU scheduling issue will arise, dramatically affecting performance and user experience. When a CPU scheduling issue happens on vSphere hosts, the amount of CPU-ready time increases and this lets one know that the scheduler is having trouble getting all of the vCPUs scheduled onto pCPUs. This means that the vCPU will have to wait, even though it's ready. The CPU ratio is very different for the various types of workloads that are virtualized on VMware clusters. Typically, server and database workloads have a much smaller ratio, while VDI workloads are able to have a higher ratio.

The use of vCPUs is not a linear calculation, meaning that one can build a host that has a higher consolidation ratio if all VMs have only a single vCPU. When many VMs have two or more vCPUs, this will affect the calculations. It's not as easy as dividing by two to account for twice as many vCPUs. Figure 6 represents a range that has proven to work with real customer deployments. Manufacturers that do synthetic testing may show higher ratios. One should be careful with these, as they do not always apply to real-world designs.



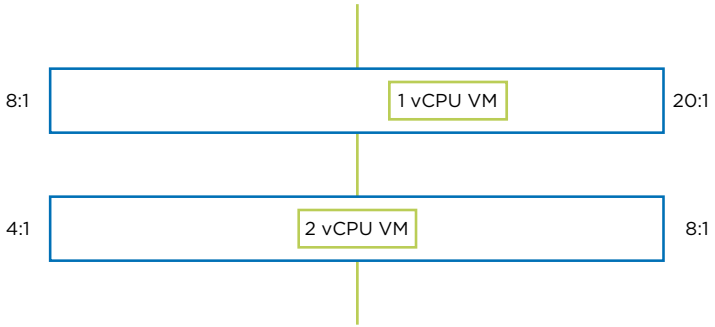
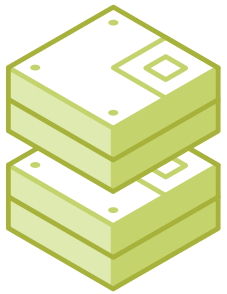


Figure 6:
VDI consolidation is heavily driven by the ratio of vCPU your virtual desktops will be configured with. The chart represents a range that experience has proven to be safe.

The working range to operate normally for single vCPU virtual desktops is between 8:1 and 20:1. This is a large range, and where one would land in that range is driven by different choices. One would be how large the hosts are, the number of VMs per host, and the customer's comfort level with that number. An example would be a dual socket host with dual 18 core CPUs. This could accommodate upwards of 700+ VMs on the high side, providing you have the right amount of memory and enough clock speed available. Typically, having that many VMs on a single host would scare most customers. Consequently, there are two choices to make in this scenario, first is to choose a lower density that one is artificially limiting. If one chooses the lower end of the ratio, it would net 288 VMs on the same host. The second option would be to choose CPUs with fewer cores, but choose a ratio somewhere in the middle. If one chooses 12 core CPUs and uses a 12:1 ratio, that would net 288 VMs. This decision is typically a combination of customer feedback, architect's recommendations, and infrastructure pricing. There may be significant cost savings from choosing different physical CPU configurations.

The calculations for a dual vCPU virtual desktop are similar, except that one is now dealing with double the amount of vCPUs. The range to operate here is between 4:1 and 8:1. Some vendors promise higher, but these recommendations are driven by real customer deployments. One should use the same decision points as the previous example, just with a different CPU ratio range.

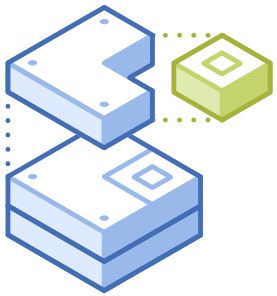


Another thing to keep in mind is that if you select a CPU ratio in the middle of those ranges, it will provide the freedom to scale the consolidation density upwards should the environment continue to perform within tolerances. One thing to note is that there is no place to configure these CPU ratios as a setting in any other tools today. These are attributes that must be declared in the design and become data points that one will need to account for in the management and scaling of the environment. Just as much as memory and clock speed, the CPU ratio needs to be calculated into the decision to add more VMs to a cluster, and when to add another host to a cluster to provide more resources.

One can manage the CPU ratio through manual calculations by gathering data. Some administrators use a PowerShell script that will gather data and present the ratio as the output from the script. With a script, it could run as a scheduled job daily to ensure one is not violating the ratio and be in danger on any of the clusters.

The RAM or memory bus frequency is also associated with the compute sizing. The rule of thumb when sizing memory is to aim for the highest density with the fastest bus speed budgets will allow. The challenge often faced with memory is that slower memory can result in idle CPU cycles waiting for read/write transactions to RAM to complete.

Incorporating GPUs into your VDI clusters and design will typically affect the density of users per host also. This is directly related to the number of GPU cards and number of GPUs per card that can be placed in your desired host and then the vGPU profile selected for your users. Simple example of a host that can accept two GPU cards, each with one GPU. Then a vGPU profile that allows 16 users per card is chosen, this means that only 32 users that need GPU would fit on that type of host. If there is available CPU and memory on the host it can still run non-GPU VMs. Identifying the correct vGPU profile is important for efficient sizing.

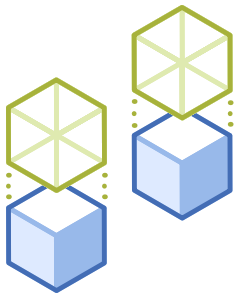


Virtualization
Cluster Design

There are a number of reasons to create different virtualization clusters in an EUC design. The decision to have different clusters is typically going to be driven by different workloads and cluster size. A lot of time won't be spent on this subject in this eBook, but here are a few recommendations that build on the topics covered elsewhere in the broader book and online.

First and foremost, when building a VDI design of more than a few hundred users, it is essential to separate the virtualization management infrastructure from the VDI workload. This means that all of the management servers, VDI brokers, file servers, application management servers, and any other functions that are not virtual desktops should be running in a different cluster.

Whether the management cluster needs to be one just dedicated to the EUC design is going to depend on how large the environment will be. If the design is smaller, one can run management VMs in an existing server virtualization cluster. It is possible to scale these virtual desktop clusters up to reach a size that is between 16-32 hosts. This range allows for a larger resource pool to be created for VMs to use, and also pushes most customers to adopt a cluster that is larger than their typical sizes. Recent hypervisor updates allow for clusters up to 64 hosts, but it will take time for many architects and customers to feel comfortable going that large. If the environment is large enough that the host counts would exceed these ranges, there would be a need for more than one VDI cluster.



Another reason one would design for multiple virtualization clusters besides environment size would be for different workloads. There are different workloads within the VDI clusters. If there is a significant amount of 1 vCPU and 2 vCPU virtual desktops, one should design a separate cluster for each. Figure 7 illustrates a multi-cluster design approach. This enables one to manage the CPU ratio differently in each cluster, allowing for an easier to manage design. If one was to blend the different CPU configurations, there would be a new blended ratio that would need to be calculated and that just confuses things.

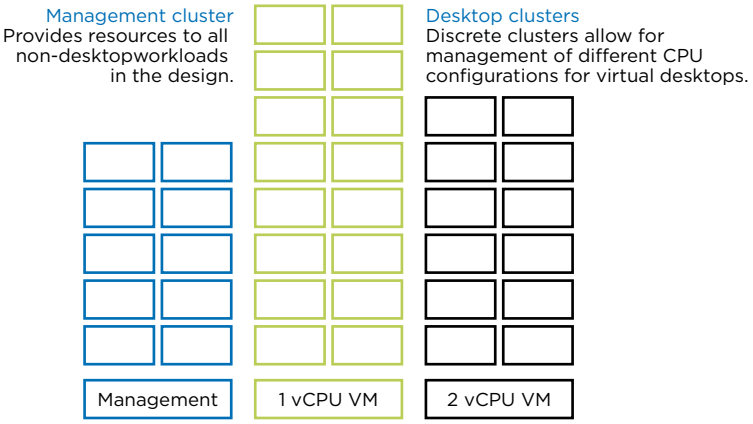


Figure 7:
Management and Desktop clusters

The use of GPUs can be another reason to consider a dedicated cluster for GPU users. You can mix GPU and non-GPU users in the same cluster, but this can make the operations and scheduling of these users a little more complex. If you have enough GPU users to meet the minimum requirements for a small cluster then it's usually worth doing so.

Provide Excellent End User Experience

Wherever You Deploy

Explore Nutanix solutions for End User Computing at **nutanix.com/euc**

NUTANIX