THE DEFINITIVE GUIDE TO

End-User Computing and Virtual Desktop Infrastructure Solutions

NUTANIX™

Brian Suhr [author] has acquired 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, datacenter, and cloud-based engagements while working with high performance technical teams on global-reaching environments. As an independent author of the Datacenter Zombie and Virtualize Tips blogs, Brian focuses on creating content that revolves around virtualization, automation, infrastructure, and evangelizing products and services that benefit the technology community. Follow Brian on Twitter: @bsuhr

Sachin Chheda [editor] is the director of solutions and verticals marketing at Nutanix. He has long been involved in information technology with engineering, management and marketing positions at the industry's most innovative companies—developing and taking to market products and solutions that power some of the largest and forward-looking enterprises. Follow Sachin on Twitter: @StorSC

Version 1.1 / January 2016

TABLE OF CONTENTS

ABOUT THIS eBOOK

This eBook is focused on infrastructure design for VDI and end-user computing (EUC) environments. The content within this eBook has been leveraged from the infrastructure focused chapter from an upcoming book “Architecting EUC and VDI Solutions.”

INTRODUCTION

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

The compute and storage infrastructure is the foundation on which one will build services. Similar to electricity and water, we count on them, and they should just work when we turn the faucet or switch on.

Without a stable, highly available and high performance infrastructure underlying the design, IT will be facing any number of challenges during the deployment and operational phases of your EUC project. This reinforces the reality that infrastructure is very important, but spending a large portion of IT's time on the infrastructure is a black hole. Architects and engineers need to be focused on providing the EUC services and applications, rather than managing the plumbing.

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/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 upfront 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.

The flip side of this reasoning is if an organization plans to deploy the solution for 10,000 users, but only intends to start with 500 users, they can scale up at a steady pace over the project timeline. As a result, they are going to take a closer look at the cost of the initial infrastructure deployment at this size rather than 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 in determining 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.

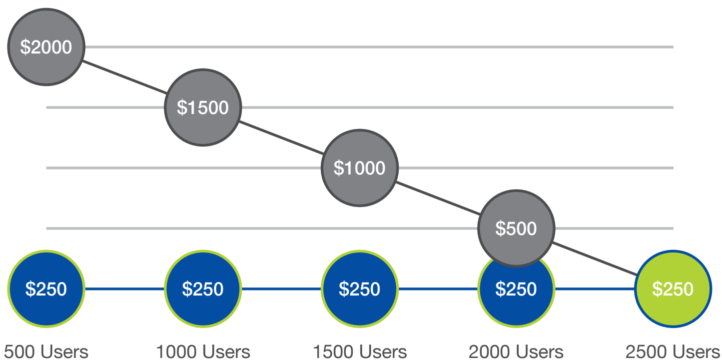

Figure 1:
Entry Points Per Desktop

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 users and still be able to scale to 10,000 users, what will the alternative look like at both ends of the 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 in order 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 in order to allow for the configuration to be scaled in the future. If the wrong choices are made up-front 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 as 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 100-200 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 looked at closely. 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 the minimum starting user requirements, one may be sacrificing performance if one is 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 up-front 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 only 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 compressible 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, many debates and issues have been caused by this topic. 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 does 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.

Monitoring needs to be simple to use, but still provide 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. The ability to quickly look at the storage layer and determine if the storage performance issue is at the global scale or if it's isolated to a host, group of VMs, or just a single VM is no longer an option.

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 well in such small increments. 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 don't want to wastefully include too much extra in each block that we can't utilize.

With this approach and granularity in the design, one can now scale the environment in smaller groups of 50-100 users. This allows for a slow and steady approach and provides predicable values that organizations can plan around for deployment, performance, capacity, and costs. If organizations want to scale faster and in larger quantities, then 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.

There are currently three primary architecture alternatives for application/desktop virtualization, or broadly, EUC solutions. 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 do-it-yourself (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.

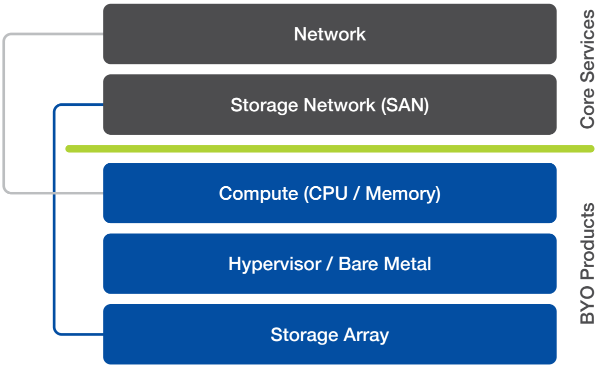

Figure 2:
Build Your Own (BYO) Infrastructure

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 manner.

If one can accept the unknowns and additional risk, the BYO alternative does maximize flexibility. Since you are 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 success 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 in which the products need to be scaled. 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 compute, storage, and networking in their offering. Typically most CI offerings will contain products from multiple vendors and be included as part of a single offering, or a vendor can offer all the layers of CI from their own product line. Figure 3 illustrates a simple example of a converged infrastructure alternative.

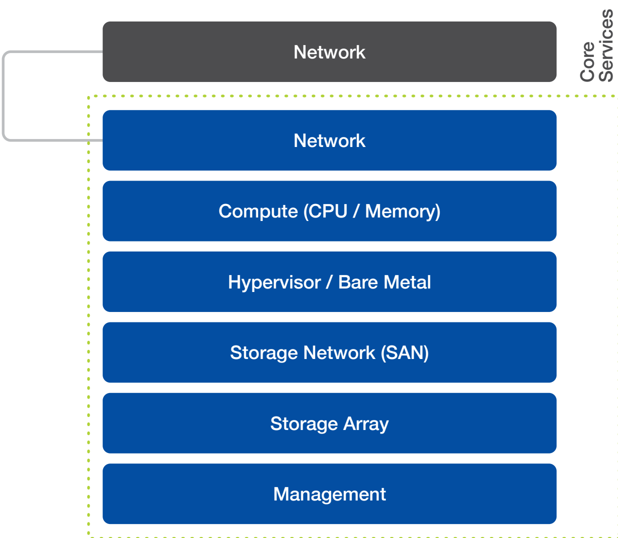

Figure 3:
Converged Infrastructure

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 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 disperse 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 add just compute, although there may be minimum increments in which one can scale. 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—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 BYO or reference architecture method, but to be truly hyperconverged, the product must include the hardware appliance.

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.

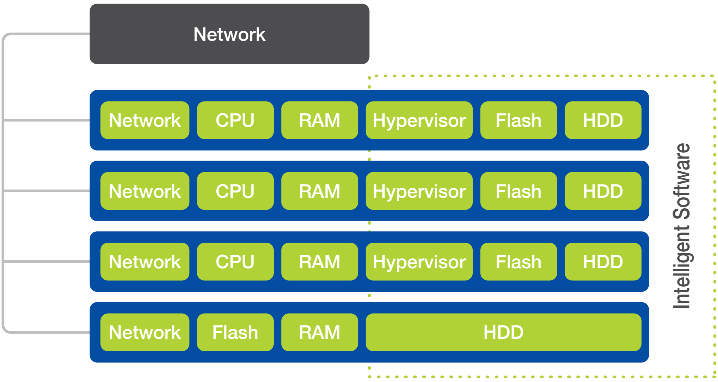

Figure 4:
Hyperconverged Infrastructure

A truly hyperconverged product offers a number of benefits that other reference architectures are unable to offer:

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 understand how VMs are performing, the amount of resources each VM is consuming, if there are any events or errors, and provide the ability 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 commandlets. With an API, you will be able to automate the communication and control between 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 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, also referred to as user VMs). The associated storage requirements will be one of the most demanding within the environment and are also the ones that cause most projects to fail or suffer from a bad experience.

For this reason, the storage portion of this eBook will be focused 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 at reasonable latency 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.

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

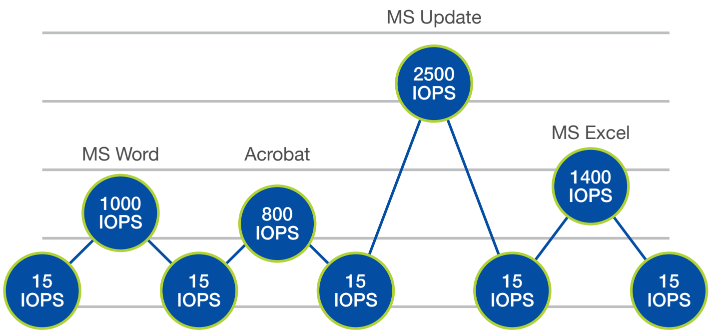

Figure 5:
VDI IOPS

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 spike in I/O. For these reasons one will need to account for maintenance operations into the storage architecture for peak IOPS.

There are a number of ways to architect VDI solutions with full clones or shared image 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 to handle situations like boot storms. If the shared image is a bottleneck, all virtual desktops using it will be negatively affected and the user experience will be bad.

Taking into consideration peak 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 just focusing on promise of storage doing an 'excellent job' at caching commonly read blocks to handle boot storms.

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 disperse 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 its legacy counterpart, 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.

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 two 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 sizing the compute resources for the VDI workload.

There are three primary calculations that you should focus 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. 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.

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.

Another important recommendation is around never exceeding 80% host (processor) utilization and sizing environments in a N+1 configuration for high availability. 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, then 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.

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.

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 2 or more vCPUs, this will affect the calculations. It's not as easy as dividing by 2 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.

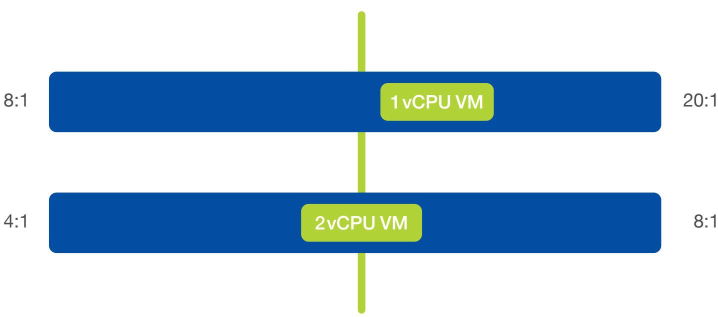

Figure 6:
VDI consolidation is heavily driven by the ratio of vCPU with which your virtual desktops will be configured. The chart represents a range that experience has proven to be safe.

The normal working range to operate 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. Multiple examples 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. So there are two choices to make in this scenario, first is to choose a lower density that 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 some place 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, architects' 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 in 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.

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 run on 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.

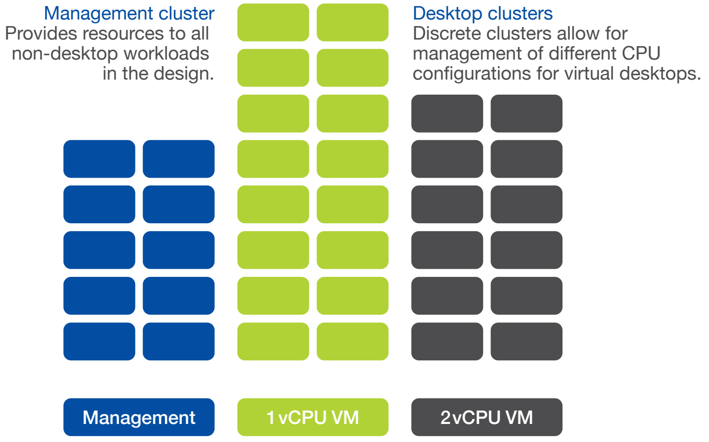

Figure 7:
Management and Desktop Clusters

If you have read this far, we hope you are at least intrigued by the possibilities of desktop and application virtualization. If you and your organization are ready to learn how to successfully deploy desktop and application virtualization, Nutanix would like to help. Nutanix Invisible Infrastructure can greatly simplify the journey through its award-winning web-scale architecture – the best VDI platform.

WHEN IT IS TIME TO START

It probably won't surprise you that Nutanix has put a lot of thought into figuring out the best ways to ensure successful desktop and application virtualization deployments.

Understanding Your Current Environment:

The process starts with a full understanding of your current end-user environment including:

- End-user metrics: Gather end-user profiles and related factors such as applications used, end-user access devices, location and connectivity.

- Network services and infrastructure-specific metrics: Gather appropriate information on different end-user services such as file services, authentication and access control, and firewalls/load balancing. Also gather performance metrics, latency, throughput, etc.

- Map everything to owners: Accountability is a key success factor.

Sizing the New Environment:

With the information above in hand, you can accurately size your new environment. Nutanix Sizer makes this task straightforward, but guidelines to keep in mind include:

- Always factor in high availability for key servers and desktops.

- Additional infrastructure and/or additional clusters may be required based on these considerations:
  - **Business:** SLAs, licensing, security, budget, politics
  - **Planning the transition:** Follow Nutanix and industry best practices and guidelines for P2V migrations and pay close attention to the golden image creation that will be used to create other desktops. If you are migrating an existing deployment, we recommend using Nutanix partner or native tools when possible.

Naturally, Nutanix Global Services can help you with any or all of these steps to put you on a path to greater infrastructure success. Through our Global Services organization, Nutanix offers the industry's only solution to eliminate the risk of incorrect infrastructure sizing for desktop virtualization projects.

Under the VDI Assurance program, Nutanix ensures that your virtual desktops always get the compute (virtual CPU and memory) and storage (performance and capacity) resources they need to meet end user VDI expectations. Simply determine the type and number of VDI users in their environment and transfer the risk of sizing infrastructure requirements to Nutanix with VDI Assurance.

Ready to learn more about Invisible Infrastructure for desktop and application virtualization? Contact us at info@nutanix.com, follow up on twitter with a DM @nutanix, or send us a request at www.nutanix.com/demo to set up your own customized briefing and demonstration to see how validated and certified solutions from Nutanix can simplify your desktop and application virtualization deployments.

Stay engaged with Nutanix experts and customers on the Nutanix Next online community (next.nutanix.com).

Nutanix delivers Invisible Infrastructure for next-generation enterprise computing, elevating IT to focus on the applications and services that power their business. The company's software-driven Xtreme Computing Platform natively converges compute, virtualization and storage into a single solution to drive simplicity in the datacenter. Using Nutanix, customers benefit from predictable performance, linear scalability and cloud-like infrastructure consumption. Learn more at www.nutanix.com or follow up on Twitter @nutanix.

NUTANIX™

T.855.NUTANIX (855.688.2649)
info@nutanix.com | www.nutanix.com | @nutanix