ActualTech Media

Tintri

Hybrid Cloud Storage Foundation

Author: Scott D. Lowe
Partner, ActualTech Media
November 2016

Contents

Introduction

In previous papers in this series, we talked about cloud and storage in relatively general terms. In this paper, we'll get into the details and focus on storage challenges today and the foundation of hybrid cloud storage.

Challenges with Legacy Block Storage and Hyperconverged Infrastructure

There is no shortage of storage options at your disposal in today's marketplace. These choices include legacy block storage systems and more modern hyperconverged infrastructure solutions. While these solutions have enjoyed a lot of positive feedback, they do come with some associated challenges that may not make them suitable for cloud requirements.

Physical-era Architecture

Although there are some people in the world who truly love managing storage and all the elements that come with it, the reality is that the constructs we've gotten used to are there due to inherent limitations that were present in legacy storage systems. We accept these constructs because *that's the way we've always done it*—as poor an excuse as any.

There are two common constructs in use in legacy storage systems:

- Logical Unit Numbers (LUNs). In the days of physical storage, LUNs represented physical devices that existed in the SCSI bus. Since then, however, LUNs have also become a bit virtualized themselves and can represent specific portions of carved up RAID groups as just one example. They have evolved to represent logical groupings of storage elements to hosts. LUNs are generally SAN-centric and are presented to hosts for the creation of volumes.
- Volumes. Volumes are areas of storage capacity carved out of one or more LUNs. Generally, volumes are created on host systems, such as vSphere hosts.

These terms are sometimes used a bit differently by different storage vendors, but these are the basic ideas. The end result is the same, though—they suffer from underlying fundamental architectural challenges.

In the modern data center, there are several needs that simply can't be easily met with physical-era architectures. These include:

- Simple scalability of both capacity and performance
- Predictable performance as the environment grows—Physical-era infrastructure in a scale up environment or stuffing too many virtual machines inside too few LUNs, making them difficult to isolate, thus subjecting them all to unpredictable performance
- Application of granular quality of service policies, for example, at the individual VM level to isolate VMs and eliminate conflict over resources
- Quick and easy identification of performance problems, including root cause determination of the exact point in the stack (compute, network and/or storage) that is causing the poor performance

As storage systems grow to span multiple data centers and even multiple clouds, these problems become even more profound. While technologies such as VMware's Virtual Volumes (VVols) have been developed to help address some of these challenges, VVols exhibits its own problems:

- Extreme dependence on storage array vendors to implement the complete VVols specification, which is carried out with varying levels of success and attention
- VVols doesn't fundamentally change the underlying architecture; it just masks architectural issues to help vSphere associate virtual machines with disks, which *does* help to solve some problems; VVols can make it possible for your storage to provide more granular per-VM policy management instead of per-LUN management. For example, per-VM controls on performance and redundancy.
- VVols only works in a complete way if you're using Sphere 6 or higher; if you're using an older version of vSphere or another hypervisor altogether, you're out of luck

That last point is particularly important to understand. While VMware's vSphere today enjoys a commanding market presence, ActualTech Media research shows a surge of interest in alternatives, most notably Microsoft Hyper-V. With the risk of VMware beginning to lose share, there are potential challenges for storage vendors that depend solely on VVols.

Storage is Not a Web Service

Storage is really important. In fact, of the resources in the data center, it's the *most* important. Why? It holds the keys to the business kingdom. As a business, if a server completely fails or if your network crashes, you can recover, even if you haven't done a good job building contingency plans. If your storage suffers catastrophic failure and you've not put into place contingency plans—e.g. backup systems—you're essentially out of business.

As we look at some of the emerging storage systems on the market, the components responsible for managing storage are not always as robust as we might like. For example, although there are hyperconverged infrastructure systems on the market that are quite good, there are others that have bolted storage on to the platform. In a data-centric world, companies can't afford to rely on systems that treat storage as "just another service." Moreover, as the market continues to extend support for container-based workloads, new challenges emerge, such as the need to maintain storage persistence inside these ephemeral constructs. More robust storage layers that can provide a persistent storage layer for non-persistent containers will be an increasing need as container adoption accelerates.

APIs are Not Clean

Traditional storage arrays lack any programmatic management. They are managed by a proprietary interface and, other than storage protocols, interface with no other external services than perhaps logging and statistical gathering.

Policy-based management and automation are a critical piece of any cloud construct. As companies move to leverage enterprise cloud, they require storage systems that support the automation necessary in a cloud model.

By using modern storage that provides a RESTful API you'll be able to integrate your storage with automation/orchestration tools, provisioning, chargeback, and more.

Infrastructure Components Must be Welded Together

When you look at a legacy data center environment, it quickly becomes clear just how much welding you must do to make things work in a reasonable way. Even when things are designed to interoperate, you still need to create all kinds of constructs—such as the aforementioned LUNs and volumes—to make it all work.

There are obvious seams around these welding points, too. These are the seams that create security concerns, but that is just a small part of the problem. Instead, each of these welding points results in additional friction in the data center. Each is a point that needs to be touched all the time.

Data center friction causes latency in achieving the goals of the business and drives business units into the waiting arms of outside providers that can move faster. The hybrid cloud storage environment requires a storage architecture that is a smoother and more flexible than traditional physical-era environments allow.

Hyperconverged Infrastructure Linear Scale and Lack of Choice

Hyperconverged infrastructure has emerged as a popular way for organizations to quickly implement data center environments. While hyperconvergence is a good solution for many, for others, it carries with it some inherent challenges.

First, the need to scale hyperconverged infrastructure node resources and software licenses in a linear way is a detriment. You're adding more compute power than you probably need as you add more storage capacity. This scale methodology may sound like scale-out storage, but there's a key difference: hyperconverged infrastructure nodes often have less overall capacity than storage arrays, so you're adding *way* more compute than you probably need. In addition, every time you add a new node, you also have to pay for another processor license for your favorite hypervisor. Those costs can add up really fast.

Moreover, you don't get choice in your storage. You have to use what you're provided and may not have the flexibility to choose an optimal combination of compute and storage.

Native Hybrid Cloud Storage Requirements

Although physical-era storage environments can coexist with and support hybrid cloud storage initiatives, in terms of efficiency and flexibility, a lot is left on the table. In the world of the hybrid cloud, virtualization is a core component and, as such, the use of a filesystem purpose-built for virtualization is important.

The sections below expand on this thinking.

Virtualization-specific Architecture

As the saying goes, “you can put lipstick on a pig” but, at the end of the day, you just have a pretty pig, not an awesome cheetah. You’re still hampered by the limitations inherent in the underlying animal. In the world of storage, you can add all the extensions you like to a legacy architecture, but you will never be able to get the same level of benefits that you do from a modern storage system.

Rather, it’s important to customers to look at storage systems that don’t depend on uncontrollable and inconsistently delivered API implementations—VVols is an API implementation—to deliver value. VM-Aware storage platforms are far better positioned for hybrid cloud storage environments since they eschew complexity in favor of simple implementations that increase performance and manageability. As organizations seek to implement hybrid cloud architectures, improved performance potential and easier administration are significant parts of their journey.

Automation, Orchestration, and Clean APIs

Legacy storage architecture is, well… not friendly. Much has been written about the complexity of legacy storage systems and the negative impacts that this complexity has on IT and the business as a whole. Remember, your storage is the lifeblood of your business. If you have difficulty with that resource, it will, thanks to the hooks it has into everything you do, create challenges for you.

Modern storage with a virtualization-specific architecture eschews this complexity, which also makes it easier to automate. Comprehensive automation is a key tenet of the Cloud and, without it, you can’t implement higher order features, such as self-service capabilities, inter-system orchestration, and DevOps-friendly capabilities. In Figure 1, note that automation drives orchestration and self-service.

Cloud is a Journey, Not Just a Destination

Consider the Cloud. Often, when thinking about the Cloud, Amazon and Azure immediately come to mind. However, for many organizations, there is hesitancy around simply throwing workloads over the fence to these public cloud providers. Rather, organizations considering the Cloud are really looking for the kinds of outcomes that are achievable with public cloud services, such as improved economics and frictionless operations. For them, the Cloud is not about a destination, but about a journey of improvement

Clean REST-based APIs are an additional key underpinning for enabling automation, orchestration, and user self-service in the world of hybrid cloud storage. A complete set of APIs enables a storage system to be fully managed without a person clicking a mouse.

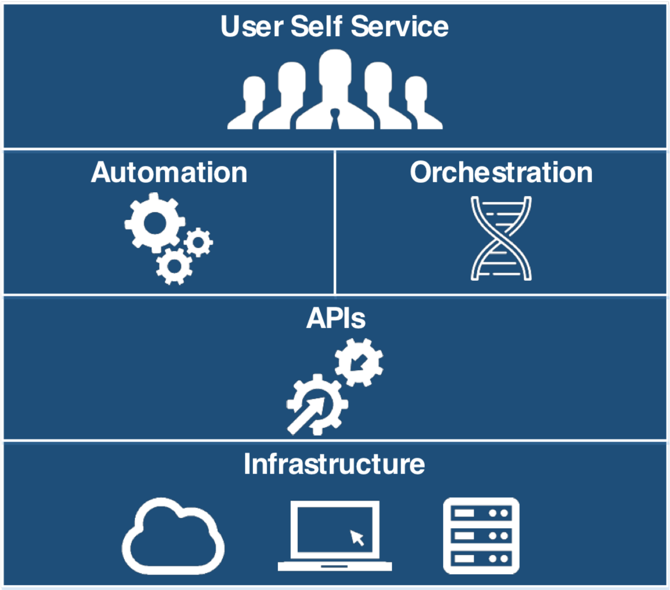

Figure 1. APIs, automation, and orchestration are key characteristics of hybrid cloud storage services

Let's for a moment consider a web-based, web-scale application that straddles both a local data center and the cloud. Imagine, if you will, this common scenario: The company that developed this application experiences occasional surges in its use. Perhaps the application is one that supports an ecommerce site and holidays drive additional traffic. During such surges, there is a need to continually adjust resources assigned to support the application. Without automation and orchestration driven by a clean API, this work will need to be handled manually.

In other words, someone will need to keep constant watch over the application's performance levels and storage capacity usage and then make reactive adjustments to resource levels. This reactive, manual method basically ensures that some customers will experience very poor results while they wait for IT to assign new resources.

Now, let's assume that the organization deployed their application atop an API-laden system and one that embraces a modern virtualization-specific architecture. As a part of the application development process, the developer can integrate deeply with the storage environment. They can write routines that keep an eye on storage latency and capacity and, as latency increases due to load or as storage capacity dwindles, enable the application to proactively deploy new virtual machines and storage services without an IT staff person having to be involved.

That's the potential power behind the software-centric nature of today's virtualization-aware storage products.

Unified Model Across Different Components

Virtual machines are just the tip of the iceberg in the modern data center. Today, containers are making a big splash in the data center pond, and with good reason: They are far more efficient than virtual machines.

If you consider the original reasons that people made the switch from physical servers to virtual ones, it was all about resource utilization. With physical servers, you had a ton of hardware to deploy, and you only ended up using between 5% and 15% of that hardware's potential, on average. There was just a ton of wasted capacity. With virtual machines, we used the same hardware, but we could cram more workloads onto it because we abstracted that hardware and turned each server into a software construct, eliminating a lot of hardware overhead and massively increasing the utilization of the hardware.

Virtualization brought relatively high levels of efficiency and flexibility to IT, particularly when compared to the world of the physical workload. With virtualization, companies can run operating systems from many vendors and different versions of each of those operating systems on the same hardware. Those workloads can be easily shifted between hosts, between data centers, and even to and from cloud providers.

But, for many, virtualization is inefficient. Every single virtual machine gets an operating system. Think about that for a second. If you install 100 identical virtual machines to support an application, that's 100 individual operating systems running in your environment. Every operating system imposes steep overhead by consuming CPU, memory, and storage resources.

To combat this inefficiency, *containers* have hit the market. Containers have an implicit assumption that the operating system for many environments is, in fact, the same, and can be safely abstracted, thus eliminating the overhead of individual operating systems. This has the immediate effect of increased density in the environment since you can cram even more workloads onto the hardware.

In terms of hybrid cloud storage, it's important for this resource to be able to equally support virtual machines as well as containers while continuing to enable deep understanding of performance bottlenecks. Containers will continue to increase in usage, particularly for web-scale applications and for tools that leverage web services. They provide a more efficient operational framework than is offered by virtual machine paradigms.

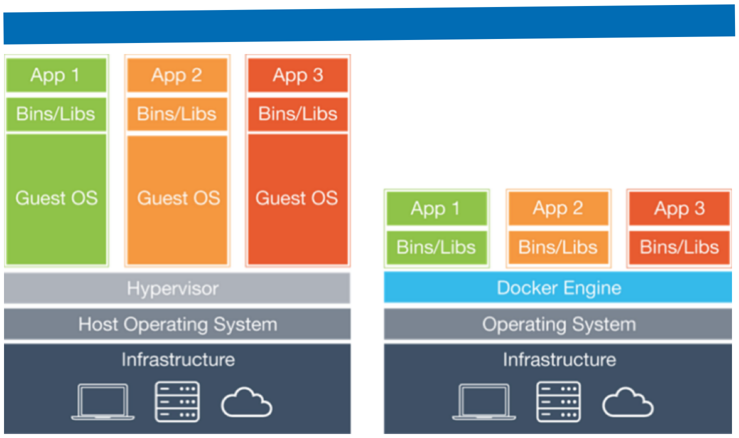

Figure 4-2: Containers vs. Virtualization

Containers in the Real World

Containers are still in their infancy and will undergo the same interest, scrutiny, and deployment challenges that initially faced virtualization, but with some twists. Whereas virtualization generally enabled companies to simply drag and drop applications from the physical environment to the virtual one, with containers, this is not the case. Applications must be purpose built for containers, at least for now. Over time, expect to see more tools put into place to enable more seamless support for migrating workloads to containers. For now, however, understand that the market is still digesting exactly how containers will ultimately fit into the IT landscape of the future.

Summary

Understanding where the storage market used to be and where it's heading is critically important to understand the eventual vision and potential of the environment. This is the topic of the next paper.