Understanding the DNA of Software Defined Storage

As Marc Andreessen famously wrote “software is eating the world”, and nowhere is this more true in changing how we think of data center architectures. Compute has already become “software-defined”, thanks to widespread adoption of virtualization. A similar process has already begun with networking – and storage.

Like any industry transition, software defined storage will take several years to be fully realized. But even at this early stage, the “genetic code” behind this new approach to storage is becoming better understood. While we will certainly see variations on a core theme, the essence behind software defined storage serves to differentiate it from other forms.

As software defined storage grows and matures, this blueprint has the ability change how we think about storage: both as a standalone discipline and as part of an integrated software defined data center.

vmware®

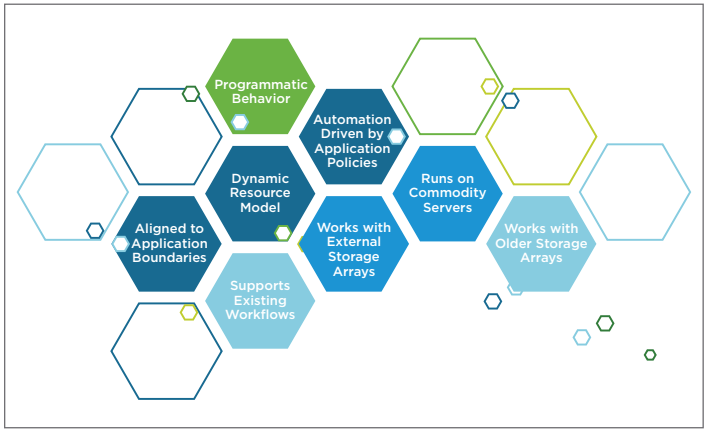

***Figure 1**. The DNA – Elements of Software defined storage*

Essential: Programmability of Behavior

If nothing else, software-defined storage should be capable of being entirely under external programmatic control, presumably using an API. This is the essence of **software-defined**: behavior is determined by external software control, most typically an automation framework.

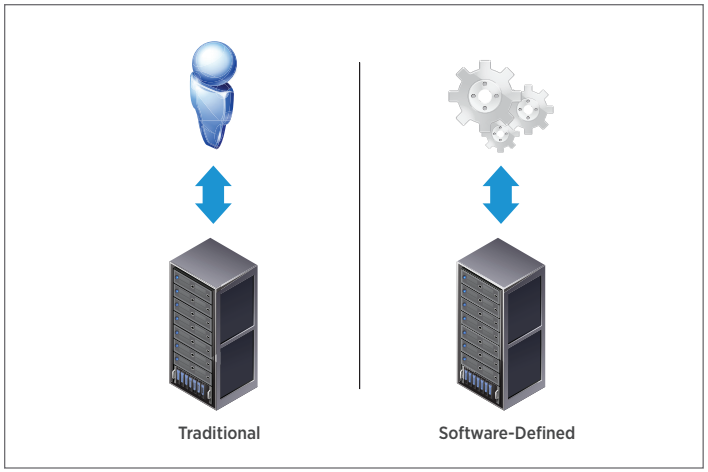

***Figure 2**. Programmability Enables Automation*

Without this **one key attribute**, there isn't a foundation for effective coordinated automation: a core goal of anything that is software defined. In an ideal world, a single standardized API set could be used to command and control many forms of storage, instead of a world where every vendor has their unique flavor.

A frequent area of current debate is the distinction between software-**based** storage, and software-**defined** storage. While there are many storage software products that can use industry-standard server hardware (software-**based**), there are precious few that can be entirely commanded via external software control (software-**defined**). The distinction is critical for IT architects that desire ever-increasing levels of automation and intelligence.

With this essential defining attribute in hand, we can look at other aspects that may be desirable, depending on specific requirements.

The essence of Software-defined storage lies in external programmatic control, presumably using an API. Programmability enables automation, a key attribute for IT architects. Without programmatic control, there's no foundation for effective, coordinated automation: a core goal of anything that is software defined.

In today's traditional static storage provisioning model, all classes of services are physically pre-provisioned and consumed as demand materializes. Contrast this with software defined storage model where applications are dynamically provisioned with the precise mix of capacity, performance, protection needed – no more, no less.

Very Desirable: Dynamic Composition vs. Static Allocation Model

The vast majority of storage products today use a static allocation model to deliver storage services. All classes of services are physically pre-provisioned, and then consumed as demand materializes. While this practice is currently widespread, software defined storage offers the potential of a more dynamic and responsive model, much as we do with virtualized compute today.

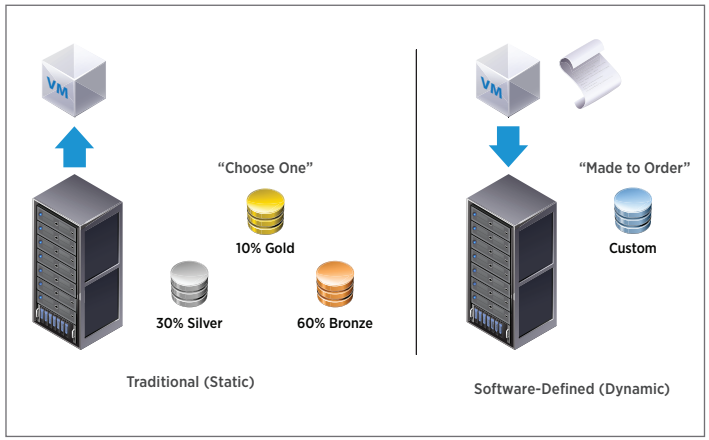

***Figure 3**. Dynamic Composition vs. Static Allocation*

With today's prevalent static storage model, the storage administrator must determine — well in advance — what different applications might eventually need, acquire physical storage hardware, allocate it into pre-provisioned resource pools with different service levels (capacity, performance, protection, etc.), and then – finally — advertise it for consumption. A quick examination will show why today's static storage model isn't ideal.

If an application's requirement doesn't fit nicely with one of the pre-established storage service levels, compromises must be made, or additional work performed. If an application's requirements move up or down outside the range provided by the pre-defined storage resource pool, more work is required to move the application to the appropriate pool. If actual aggregate storage demand doesn't line up with the pre-allocated buckets that have been forecasted, resources are wasted, or come up short in meeting demand.

Ideally, software defined storage would present all resources (hardware, software, etc.) as a large pool of potentially allocatable storage services. When application requests come in for a specific storage service, it would be dynamically composed with the precise mix of capacity, performance, protection, etc. desired by the application.

Much as we do with compute today, an ideal software defined storage environment enables IT administrators to precisely match demand and supply, without extensive foreknowledge of specific application requirements.

Very Desirable: Storage Services Aligned on Application Boundaries

At its essence, enterprise IT is about application delivery, and that's how users see the world: through the lens of their applications. Ideally, we would be able to tailor and adjust all IT services (including storage) along precise application boundaries – as users see things, and not how traditional IT infrastructure defines boundaries.

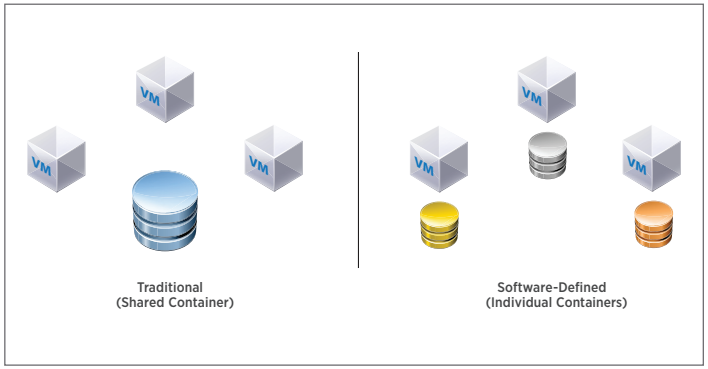

***Figure 4**. Aligned on Application Boundaries*

Today's storage usually has scant knowledge of applications or their boundaries; it sees the world as storage-centric LUNs and filesystems, each one offering up a container with a static combination of capacity, performance, protection, etc. A given storage container may have multiple applications consuming it.

If an application requires something different than offered by the physical storage container, it must be moved to a different, pre-allocated physical storage container that provides the desired level of services – assuming one is available! This results in more work by IT administrators, in addition to poor alignment with specific application requirements.

Software defined storage introduces the notion of application-specific storage containers: precise mixes driven by application requirements, and not physical storage constraints. This enables storage services to be precisely tailored and adjusted as needed on a per-application basis, without affecting neighboring applications: more for this application, less for that one.

Very Desirable: Automation Driven By Application Policy

Continual investment in automation pays continual dividends for enterprise IT. When done right, unit costs can drop, quality of services can improve, and new requirements can be satisfied more rapidly. Modest benefits are achieved by low-level automation of routine tasks. Significantly greater benefits can be achieved by automating from application requirements downwards into the IT stack.

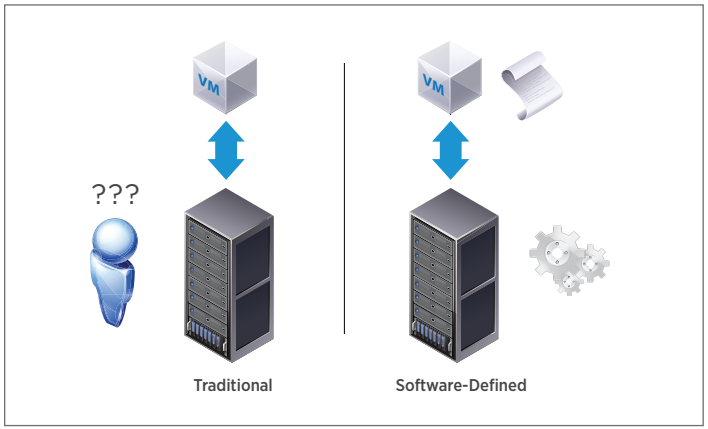

***Figure 5**. Policy Drives Automation*

This approach to application-centric automation is based on the notion of an application policy: a manifest of requirements desired: compute, network – and storage. In our software defined storage world, this application policy is used to request, monitor and adjust higher-level storage services.

By comparison, most storage today requires a very granular and low-level approach: specific requests for specific array types, disk types, RAID protection, amounts of caching required, specific replication schedules, and the like. What is missing is the ability for an application to express its requirements at a high level (e.g. performance, protection), and then enable the storage layer to figure out how best to satisfy those requirements.

Ideally, software defined storage could interpret a single, shared policy that is used for other resource and service requirements, without the need for specific and granular knowledge of the underlying storage resources.

Continual investment in automation pays dividends for enterprise IT. When done right, unit costs drop, quality of service improves, and new requirements satisfied more rapidly.

Desirable: Works with External Storage Arrays

The majority of today's external storage arrays are essentially intelligent software running on highly-configured standardized hardware. While based on commodity components, today's are designed and optimized for the task at hand. External storage arrays have represented the majority of enterprise IT implementations until today, and will likely continue for the foreseeable future.

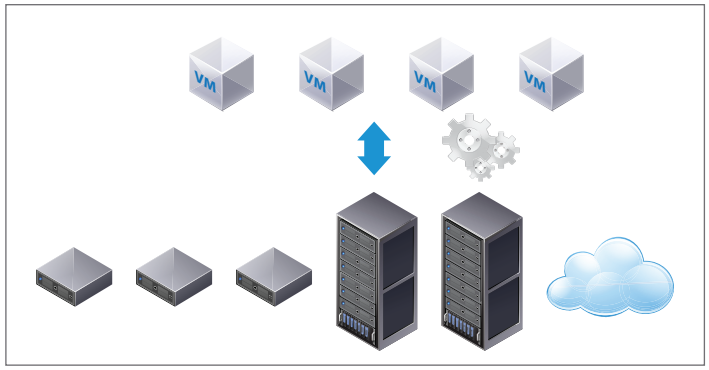

***Figure 6**. Works with Internal Storage, External Storage and Cloud Storage*

One area of debate is whether or not software-defined storage must imply a software-only storage stack that exclusively runs on the customer's choice of commodity hardware. This philosophical restriction is both illogical and unrealistic.

Enterprise IT organizations value highly the ability to work with existing investments, and that includes external storage arrays. Additionally, IT groups should have choices in how they consume storage functionality: as hardware, as pure software, or potentially an external cloud service.

To the extent that any external storage array can support the functions described here, it should be seen as a potential full participant in a software defined storage environment.

Desirable: Storage Runs on Commodity Servers

While traditional external arrays will continue to part of the landscape, there is also strong interest in software-only storage products that use familiar servers to offer storage servers. Software-only storage thus becomes an important option that our ideal software-defined storage environment should support.

These newer software stacks give customers the freedom to select hardware of their choosing, based on their unique requirements. The same physical server infrastructure can be used to provide compute, network as well as storage functions.

These newer storage stacks broadly fall into two architectural categories: storage software that runs in a virtual machine, and storage software that is tightly integrated with the hypervisor, or **hypervisor-converged**.

Desirable: Storage Can Be an External Cloud Service

Not all storage choices for software defined storage should be limited to the data center. Cloud-based storage services are becoming increasingly popular: whether to directly support an application, or as a target for backup, archiving or disaster recovery.

As long as the external cloud storage service supports important elements of our software defined storage definition (programmability, dynamic composition of services, aligned on application boundaries, driven by application policy, etc.) they too can be full participants in a software defined storage environment.

Somewhat Desirable: Works with Older Storage Arrays

While it's always desirable to extend the useful life of older assets (including storage), this presents a very difficult challenge for implementers. Simply put, older storage arrays weren't designed to participate as part of a software-defined storage environment – they just don't have the required intelligence.

Supporting older legacy storage arrays will require the use of an additional abstraction layer to communicate and participate, and – even then – the results will be less satisfying than using more modern storage arrays that were designed to be externally automated.

Somewhat Desirable: Supports Existing Storage Workflows

A significant benefit of software-defined storage is that it can support newer converged operational workflows: ones where all resources are managed by an application-centric policy. Indeed, one of the major motivations behind a software defined data center is the ability to support advanced forms of automation.

However, in today's world is that storage runs on familiar manual workflows, using low-level tools. While the industry progression to software defined storage appears inevitable, it is somewhat advantageous to support some of the existing processes while experience is being gained with the newer approaches.

At the same time, many IT organizations have built (or are building) greenfield environments with clear breaks from legacy approaches, and thus not requiring operational compatibility with existing tools and workflows.

The VMware Perspective

VMware believes that software defined storage is the next evolutionary step for storage: whether as part of a software defined data center, or as a standalone discipline. Many current VMware products implement key components of a software defined storage environment as described here, with more under development. As IT organizations drive virtualization concepts beyond server compute into network and storage, VMware plans to offer the required software technology, coupled with broad participation from our industry partners.

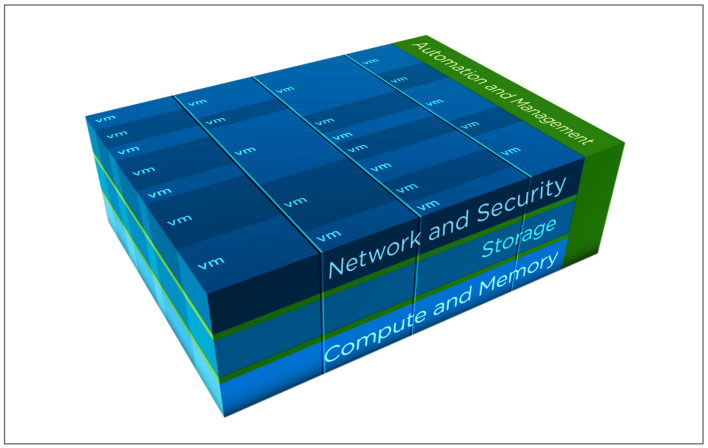

***Figure 7**. Software-Defined Datacenter*

Software defined storage is the next evolutionary step for storage: whether as part of a software defined data center, or as a standalone discipline.

Software Defined Storage — The Road Ahead

The IT industry is changing at a faster rate than ever before. These seismic shifts require that progressive IT leaders fully understand emerging industry trends, and begin to think through how they can work in their specific environments.

In the IT portfolio, storage is demanding more attention. Data volumes continue to explode, aggregate costs continue to escalate, and – in many ways – storage is more resistant to change than other IT disciplines .

When it comes to software defined storage, the time is now for leaders to create their own informed opinions as to what it is, and why it might matter to them. Exploiting the new power of the tools will require more than just new technology, it will demand a re-thinking of how storage services are created, presented and consumed by applications.

Workflows will change, operational procedures will change; roles and required skills will inevitably change as a result.

Historical precedents are illustrative here. As we know from previous experience, what it meant to be a "server expert" changed dramatically when virtualization became prevalent.

And it's safe to say the same thing is going to happen with storage defined storage.

Learn More

To learn more about Virtual SAN and software defined storage, visit **www.vmware.com/products/virtual-san**

Information on VMware's vision for software-defined data centers (SDDC) can be found at: Visit our blog **http://blogs.vmware.com/vsphere/storage**

1. "Why Software Is Eating The World", Marc Andreessen, Wall Street Journal http://online.wsj.com/news/articles/SB10001424053111903480904576512250915629460
2. "Server SAN" Wikibon, http://wikibon.org/wiki/v/Server_SAN_Market_Definition
3. "Digital Universe: Executive Summary", IDC, http://www.emc.com/leadership/digital-universe/2014iview/executive-summary.htm

AUTHOR

Chuck Hollis

Chuck Hollis is Chief Strategist for VMware's Storage and Application services. Chuck joined VMware in 2013 after a 17-year stint at EMC and is a well-regarded industry blogger. During his tenure at EMC, he ran EMC's Technology Alliances organization, was VP of Strategic Marketing and VP of Storage Platforms Marketing, among other roles.

Join Us Online

Blog: http://blogs.vmware.com/vsphere/storage

Twitter: @VMwareVSAN

Facebook: https://www.facebook.com/vmwarevsan

vmware

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 www.vmware.com
Copyright © 2014 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at http://www.vmware.com/go/patents. VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies. Item No: VMW-TECH-TRENDS-MGZN-VSAN-ESS-ELMNT-SDS-USLET-102