

# SOFTWARE DEFINED STORAGE VS. TRADITIONAL SAN AND NAS

This white paper describes, from a storage vendor perspective, the major differences between Software Defined Storage and traditional SAN or NAS arrays



## **EXECUTIVE SUMMARY**

There is a new trend of Software Defined “Everything” in IT, and the storage industry is going through the same transformation. While there is a lot of marketing buzz around the term *Software Defined Storage*, this article is intended to introduce clarity by describing—from a storage vendor perspective—the major differences between Software Defined Storage and traditional SAN or NAS arrays.

## **SOFTWARE VS. HARDWARE**

This is the most obvious difference, and it has a big impact on both the cost and agility of deployment of storage systems.

## **COST**

On the cost side, traditional SAN and NAS storage array vendors spend a lot of money in developing custom ASICs, custom circuit boards and custom real time operating systems (OSs). A custom ASIC can cost between \$10M and \$20M in development that, in turn, needs to be paid for via inclusion in the price of the storage array. In fact, only a minimal part of the R&D expenses of a traditional storage vendor is spent on storage functionality. The majority is spent on development of custom hardware and software infrastructure. In addition, because all the components are proprietary, traditional storage vendors need to spend a lot of money on interoperability tests. As an example, a simple new drive qualification cycle can cost millions of dollars and take many months.

On the other hand, Software Defined Storage companies use standard x86 hardware and standard OSs. This means that the semiconductors used are designed by companies such as Intel, Broadcom and LSI. The hardware and interoperability is completed by companies such as Dell, HP and IBM. And the OS components are developed by Microsoft, VMware or Linux developers. This also means that a company like Zadara® Storage can leverage existing and proven technologies and concentrate all its R&D efforts purely on storage functionality, bringing forth a better product at a fraction of the cost of a traditional SAN or NAS storage array.

If so, why do SAN and NAS storage array vendors prefer custom hardware? If you ask any storage vendor why they need expensive custom hardware they will tell you that the reliability of their hardware is much better than that of standard x86 hardware. In general this is true, but the other side of the same coin is that their legacy software assumes a great degree of reliability from the underlying hardware. Their software protects against rare “single points of failure” as opposed to multiple and more frequent failures that could happen with commodity hardware.

Unlike storage based on legacy software, Software Defined Storage companies like Zadara actually assume that hardware is unreliable, and the software is the one that needs to continue operating in the face of concurrent hardware failures. By understanding that eventually all hardware will fail (even proprietary hardware), and architecting a software-defined solution with no single point of failure, SDS solutions create a more resilient and reliable solution than traditional proprietary arrays.

## **AGILITY**

The benefits do not end with cost savings. The other big advantage that Software Defined Storage provides is agility. Agility means doing things faster; preferably much faster. Agility is a significant value-add because it allows rapid, effective response to changing business requirements or conditions, In fact, agility can be even more important than cost savings.

For instance, while deploying a traditional storage array can take weeks from the moment of the decision to buy to the moment it is actually installed and connected to the application servers, the exact same thing takes exactly one minute with Zadara. “Yes”, we said, “weeks versus a minute”!

Moreover, another component of the agility is automation. While traditional SAN and NAS arrays use command line interfaces (CLIs) or the SMI-S API to automate certain operations, Software Defined Storage typically uses REST APIs that are much simpler to use and integrate.

## **SCALABILITY**

Beyond hardware vs. software, there are other major differences between Software Defined Storage and traditional SAN and NAS arrays. One of the most important ones is scalability. Traditional SAN arrays are based on a dual controller configuration, and even the ones that “scale out” can scale to a maximum of about 16 controllers. This means that there is a real limit to the number of drives one can attach behind those controllers. When using solid state drives (SSDs), this scalability limit is dramatically lower due to the high IOPS (input/output operations per second) rate of the drives compared to the IOPS rate supported by the controllers. On the other hand, Software Defined Storage is designed like the cloud. It assumes hundreds to thousands of nodes (or more), interconnected with standard networks. This means that the number of drives you can put behind it is practically unlimited, even when using SSDs.

From a software architecture perspective there is a world of difference between designing a distributed system that can scale up to a dozen or so nodes and one that can scale up to thousands of nodes. In the case of the latter system, one has to assume that there are no blocking activities. Different components can be up or down at any given time and the software cannot rely on specific modules being up or down. There is no such a thing as a “shutdown” or “reboot” of the entire storage system, even for a software upgrade. Software upgrades need to be performed on a rolling basis, and the software needs to assume that at any given time different nodes will have different versions of the software (and, as mentioned earlier, that some nodes will be down).

## **RELIABILITY**

As discussed previously, traditional SAN and NAS vendors base their availability on the reliability of their custom hardware, while for Software Defined Storage vendors the reliability comes from the software—which assumes that hardware will fail, especially in highly scalable environments. It is the job of the software to make sure everything continues to work with intelligent, self-healing processes. This difference in concept actually makes the Software Defined Storage systems much more reliable than traditional SAN and NAS storage devices. This kind of thinking is what made technologies like the Internet and its underlying protocol, TCP/IP, very reliable. Much in the same way, TCP/IP assumes that routers will

fail between two points but has enough intelligence to reroute packets via alternate routes—resulting in resiliency and reliability.

## **MULTI-TENANCY**

Because the traditional SAN and NAS storage devices are limited in scalability, they usually serve a single purpose, and typically only one administrator or a very limited number thereof. There is no good separation among workloads in a SAN or NAS storage array that is shared by multiple applications. This causes performance problems and lack of self-provisioning by the applications. Typically, customers buy multiple storage arrays and employ physical separation among workloads. Obviously, this is not cost effective, as unused space in one storage array cannot be used by another array. It is also not efficient for storage administrators, who need to manage multiple storage arrays, which they purchase, at times, from different vendors and have different management consoles.

On the other hand, a Software Defined Storage system is built with multi-tenancy in mind. The ability to separate workloads and even self-provision storage by every tenant is critical if all the storage for all the applications is to come from a single system. This allows isolation of the performance among tenants and at the same time allows the user to quickly reallocate unused storage to other applications or other tenants.

Another aspect of multi-tenancy is chargeback. Most of the traditional SAN and NAS arrays lack the ability to track the usage and cost of particular applications, while in a multi-tenant environment, it is important to understand the cost incurred by every tenant. Software Defined Storage designed for the cloud, such as the VPSA<sup>®</sup> Storage Array by Zadara, has metering and billing capabilities built in providing users the ability to charge-back to various departments based on usage, rather than a flat CapEx percentage.

## **SUMMARY**

Behind the marketing buzz of Software Defined Storage there is real substance. Real Software Defined Storage delivers significant cost savings, unprecedented

agility, cloud scale, and multi-tenancy. Storage that falls short of these criteria is not software defined. It may talk the talk, but it doesn't walk the walk.

## CONTACT US

### USA Sales

[sales@zadarastorage.com](mailto:sales@zadarastorage.com)

+1-949-251-0360

### International Sales

[sales@zadarastorage.com](mailto:sales@zadarastorage.com)

+44-1285-610045

### 24/7 Support

[support@zadarastorage.com](mailto:support@zadarastorage.com)

+1-949-284-0713

© 2017 Zadara® Storage. All rights reserved.  
Company or Product names may be the property of their respective owners.

FEB092017

