WHITE PAPER

pernixdata

All-Flash Arrays Weren't Built for Dynamic Environments. Here's Why...

This whitepaper is based on content originally posted at www.frankdenneman.nl

Monolithic shared storage architectures are ill-suited to support the modern virtualized data center due to limitations in the storage fabric, constrained port densities, and finite amounts of available CPU in storage controllers. The result is poor application performance and limited scalability.

In this paper, we'll explore the best way to scale storage performance in a virtual data center. We will do a technical deep dive into the challenges of using traditional shared storage (including newer hybrid and All-Flash Arrays), and discuss how these obstacles can be overcome with newer decoupled storage architectures.

Look Back to Move Forward

Storage arrays were originally designed to provide centralized data storage to a low number of machines. I/O performance was not the primary pain point as most arrays could easily handle the requests from a small set of servers. That all changed with the advent of modern day virtualization.

The concept of virtualization has existed for a long time. Mainframes, for example, were the first to use this concept in the 1970s. However, this model of virtualization was monolithic – i.e. based on a single compute structure (Mainframe) that communicates with a single shared data repository (storage array) through a data path on an isolated network.

The modern data center uses virtualization much differently. It is implemented in a distributed fashion, whereby many workloads across multiple servers are consuming I/O. However, data is still predominantly stored in a shared monolithic device. A Storage Area Network (SAN) has basically replaced the role of a mainframe in this respect.

Does it still make sense for this data to be stored in a single shared array? Especially given the fact that the same array is also serving dedicated servers running workloads on bare metal.

Applying a system built for an old model introduces significant challenges, leading to inadequate performance. Let's explore this further...

Storage Area Network (SAN) topology

The most common virtual data center architecture consists of a group of ESXi hosts connected via a network to a centralized storage array. SAN design typically follows the Core-Edge fabric topology. In this model, a high capacity core switch (or a few switches) are placed at the center of the fabric. The servers are sometimes connected directly to the core switch but is more often linked to a switch at the edge. An inter-switch link connects the edge switch to the core switch. The same design applies to the storage array; it's either connected directly to the core switch or to an edge switch. (See Figure 1)

Each layer has its inherent role in the design. Although core switches can offer a lot of capacity, there is a limitation on the number of ports. Hence, edge switches are used to extend the port count and connect a greater number of devices.

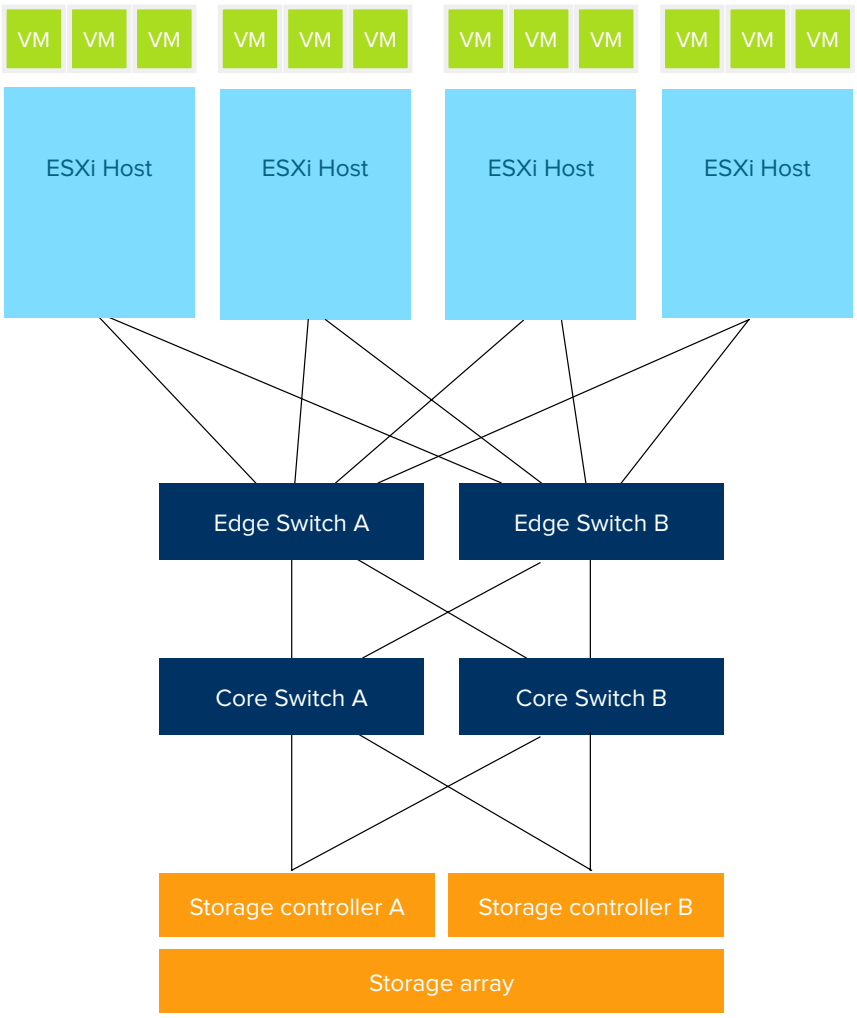

Figure 1: The typical SAN design follows a core-edge topology which provides resiliency but introduces potential performance and scale issues

While the Core-Edge fabric topology delivers the necessary resilience and performance along with scalable port density, there are a number of (circular) design tradeoffs to consider.

- The placement of systems matter as each additional network hop increases latency.
- Connecting more systems per switch decreases the number of hops to reduce latency but this impacts the port count and limits future growth. It also restricts bandwidth availability, as network oversubscription is amplified with more edge ports sharing a single connection to a core port, and more edge switches connected to each core switch.
- Adding switches to increase port count and available bandwidth brings you back to device placement problems again.

As latency plays a pivotal role in application performance, most SANs aim to be as flat as possible. Some use a single switch layer to connect the host layer to the storage layer. Let's take a closer look on how scaling out compute impacts storage performance in this type of network topology.

Oversubscription Ratios

Consider a representative architecture with two hosts connected with two 10 Gigabit Ethernet (GbE) links to a storage array that can deliver 33,000 IOPS (Figure 2). The SAN is also 10 GbE-based and each storage controller has two 10 GbE ports. To reduce latency as much as possible, a single redundant switch layer is used to connect the ESXi hosts to the storage controller ports. In this case, the oversubscription ratio of the links between the switch and the storage controller is 1:1 – the ratio of consumer network connectivity to resource network connectivity is equal.

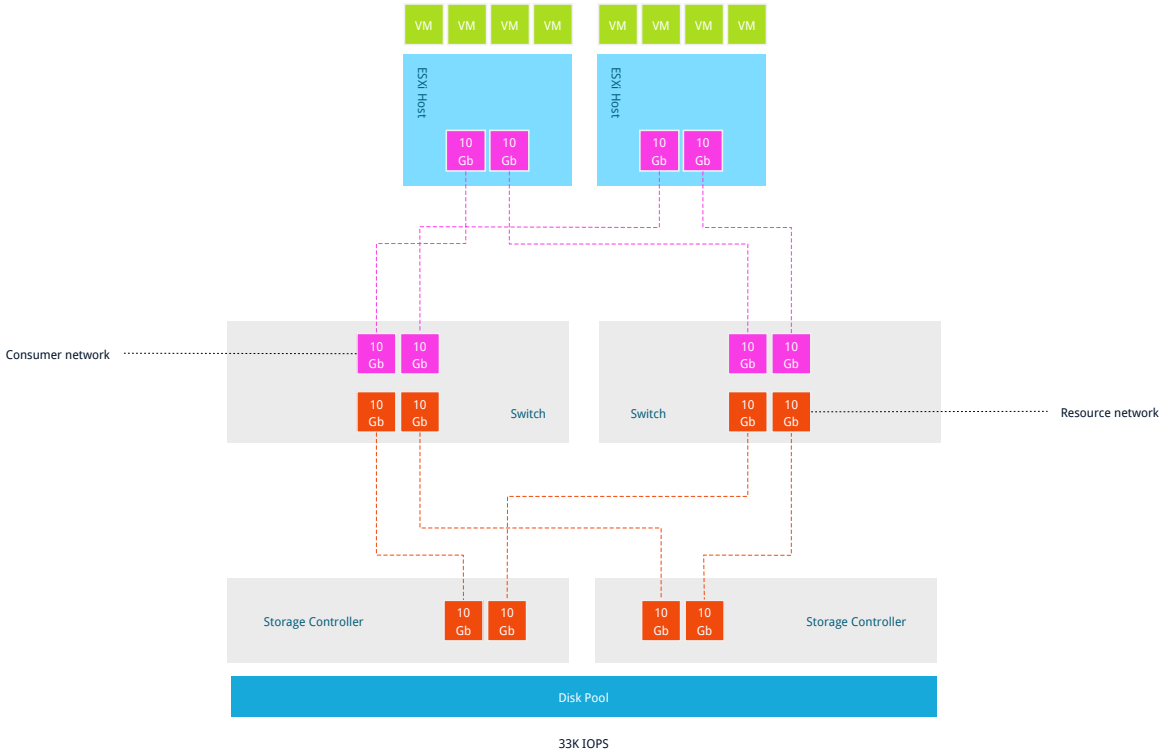

Figure 2: In a typical deployment a single switch provides redundancy with the consumer network connectivity equal to resource network connectivity in a 1:1 ratio.

Now, let's imagine that new workloads are introduced and more compute resources are required. To accommodate, we expand the ESXi cluster from two to six hosts. In turn, the spindle count of the array is increased by adding two disk shelves to improve performance (while adding capacity) at the storage level; based on this upgrade, the storage subsystem can now provide 100,000 IOPS. Although both the compute and storage resources have expanded, no additional links between the storage controllers were added (Figure 3). Thus, each 10 GbE link from an ESXi host has to share the connection with 5 other hosts. Hence, the oversubscription ratio has ballooned to 3:1.

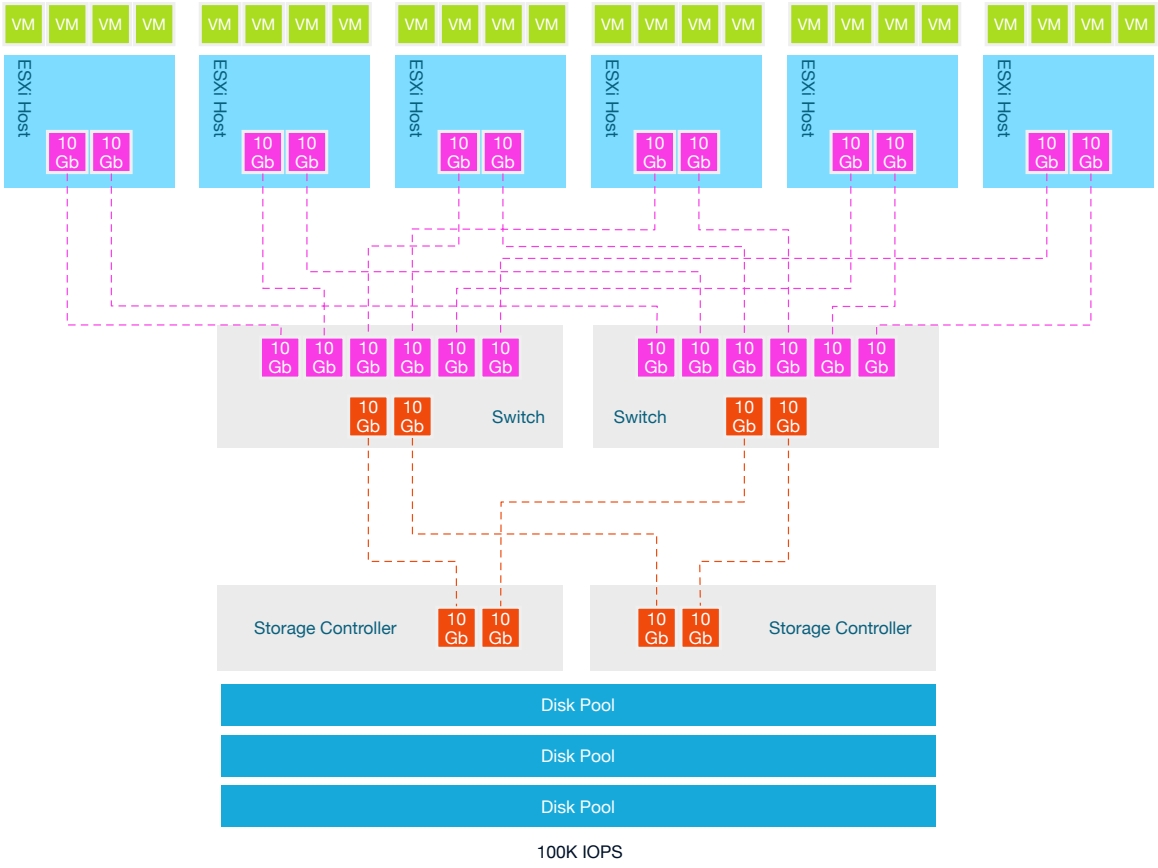

Figure 3: Adding more hosts creates a network oversubscription ratio of 3:1.

The obvious remedy is to increase the number of links between the switch and the storage controllers. However, most storage controllers don't allow for scaling of the network ports. This is due to the fact that most SAN designs are based on best practices rooted prior to the virtualization era where storage arrays where connected to a small number of hosts running a single application. By making the assumption that not every application will be active at the same time or with the same intensity, the bandwidth requirements of the virtualized environment with ever-expanding consolidation ratios are often underserved. And, although many vendors stress the desire for a low oversubscription ratio, the limitation of storage controller ports generally prevents this constraint from being removed easily.

Keep in mind that the above scenario incorporated only six ESXi hosts. Typically, you will see many more servers connected to the same shared storage array, stressing the oversubscription ratio even further. With more I/O squeezing through a fixed funnel, latency and bandwidth performance can be severely impacted.

IOPS per Host

Diagnosing scaling problems with traditional storage architectures frequently starts by dividing the total number of I/Os Per Second (IOPS) provided by the array by the number of hosts to calculate the average number of IOPS per host. In the previous example, this metric remained constant at 16,500 IOPS due to the expansion of the storage resources staying proportionate to the increase in compute resources (i.e. from 33,000 IOPS with two hosts to 100,000 IOPS with six hosts).

However, this is an imprecise tool as the behavior of the combined set of virtualized workloads as well as their rate of growth typically do not align with initial expectations. Or, new applications can turn up unexpectedly when business needs change. It's not unusual for organizations to experience performance problems due to lack of proper visibility in workload behavior.

Arguably, this should not pose a problem if sizing was done correctly. Due to the way storage is generally procured (i.e. sized for the expected peak performance at the end of its life cycle), when the first hosts are connected, the available bandwidth and performance should be more than sufficient. As new workloads are added to a fixed set of servers, though, the consolidation ratio grows to increase the number of I/O requests proliferating from each host. This leads to a general reduction in bandwidth and IOPS available to each VM.

To counter, more capacity is added to the storage array in an attempt to satisfy the performance requirement. But, this ignores a fundamental problem – the bottleneck created by the oversubscription ratio of the links connected to the storage controller ports. As previously discussed, the storage controller port count is a major inhibitor in scaling storage performance. Another problem is the way bandwidth is consumed. The activity of the applications and the distribution of the virtual machines across the compute layer has a distinct effect on the storage performance. On top of that, the workload might not be evenly distributed across the links to the storage controllers.

Storage Controller Architecture

Storage controllers are based on a foundation of commodity server hardware running proprietary software supporting storage protocols and providing data services. They are equipped with I/O ports to establish communication with the array of disks on the back end and interface with the attached hosts on the front end.

The most popular storage arrays are configured with storage controller with two CPUs ranging from four to eight cores. This means that the typical enterprise storage array with a redundant pair of controllers is equipped with 16 to 32 cores in total. The storage controller CPU cycles are dedicated for activities such as,

- Setting up and maintaining data paths.
- Mirror writes for write-back cache between the storage controllers for redundancy and data availability.
- Data movement and data integrity.
- Maintaining RAID levels and calculating and writing parity data.
- Data services such as snapshots and replication.
- Internal data saving services such as deduplication and compression.
- Executing multi-tiering algorithms and promoting and demoting data to the appropriate tier level.
- Running integrated management software providing management and monitoring functionality of the array.

While the de facto method for adding storage performance is to add more disks to the array, the storage controller cycles are a fixed resource. This means that it becomes a greater constraint as more disks, each requiring management and data services, are placed behind it.

Storage Controller Cache

Almost all storage arrays contain cache structures to speed up both reads and writes. Speeding up writes provide benefits to both the application and the array itself. Writing to NVRAM, where typically the write cache resides, is much faster than writing to (RAID-configured) disk structures allowing for faster write acknowledgments. As the acknowledgment is provided to the application, the array can “leisurely” structure the writes in the most optimum way to commit to data the backend disks.

However, in order to avoid making the storage controller a single point of failure, redundancy is necessary to avoid data loss. Some vendors provide consistency points for redundancy purposes; most vendors mirror writes

between the cache areas of both controllers. Mirrored write cache requires coordination between the controllers to ensure data coherency. Typically, messaging is used via the backplane between controllers to ensure correctness. Mirroring data and messaging consumes CPU cycles from both controllers. Storage controller CPUs can be overwhelmed as the incoming I/O has to be mirrored between cache structures and coherency has to be guaranteed. Wasting precious CPU cycles on messaging between controllers instead of using it for other data services and features limits performance further.

Increasing cache sizes at the controller layer just delays the point at which the write performance problems begin. No matter the size or speed of the NVRAM, it's still the write processing ability of the disks on the back-end that is being overwhelmed. Typically, this occurs when there is a spike of writes. As most ESX environments already generate a high, constant flow of I/Os, adding a spike of requests is usually adding insult to injury to the already strained storage controller. Some controllers exacerbate this further by reserving a static portion of the cache for mirrored writes, forcing the controller to flush data to disk when that portion begins to fill up. As the I/O keeps pouring in, the write cache has to wait to complete the incoming requests until the current write data is committed to disk resulting in high latency for the application.

Decoupled Storage – The Path Ahead

By placing active data closer to the VM at the server tier using high-speed media such as flash or RAM, PernixData FVP software enables a decoupled storage architecture that addresses performance and capacity independently. This mechanism greatly reduces the amount of data traffic traversing the storage network to dampen the inhibitors – oversubscription ratio, storage controller port count, limited storage controller cycles, etc. – preventing performance scalability.

A clustered solution capable of adding IOPS as needed by incorporating server-side resources (e.g. greater memory footprint, additional flash devices, new hosts) non-disruptively, the decoupled storage architecture can expand seamlessly with minimal effort. Because, by definition, the IOPS per host increases linearly in this type of structure, performance can be added precisely and cost-effectively – the need to overprovision storage from Day 1 is completely obviated.

Focused on host-to-host connectivity instead of an edge-to-core topology to connect the shared storage system to a group of ESXi hosts, decoupled storage allows for a fast and simple means to grow the infrastructure. The use of non-blocking, point-to-point connectivity permits copious amounts of accelerated data to be processed more readily. Leveraging point-to-point connections and being able to leverage the context-aware hypervisor allows you not only to scale easily, it allows you to create environments that provide consistent and deterministic performance levels.

The era of traditional storage is at an end. The fundamental architectural limitations of a monolithic array is an imperfect fit for the requirements of the virtualized world. The emergence of the decoupled architecture provides a better solution. The future data center is one step closer!