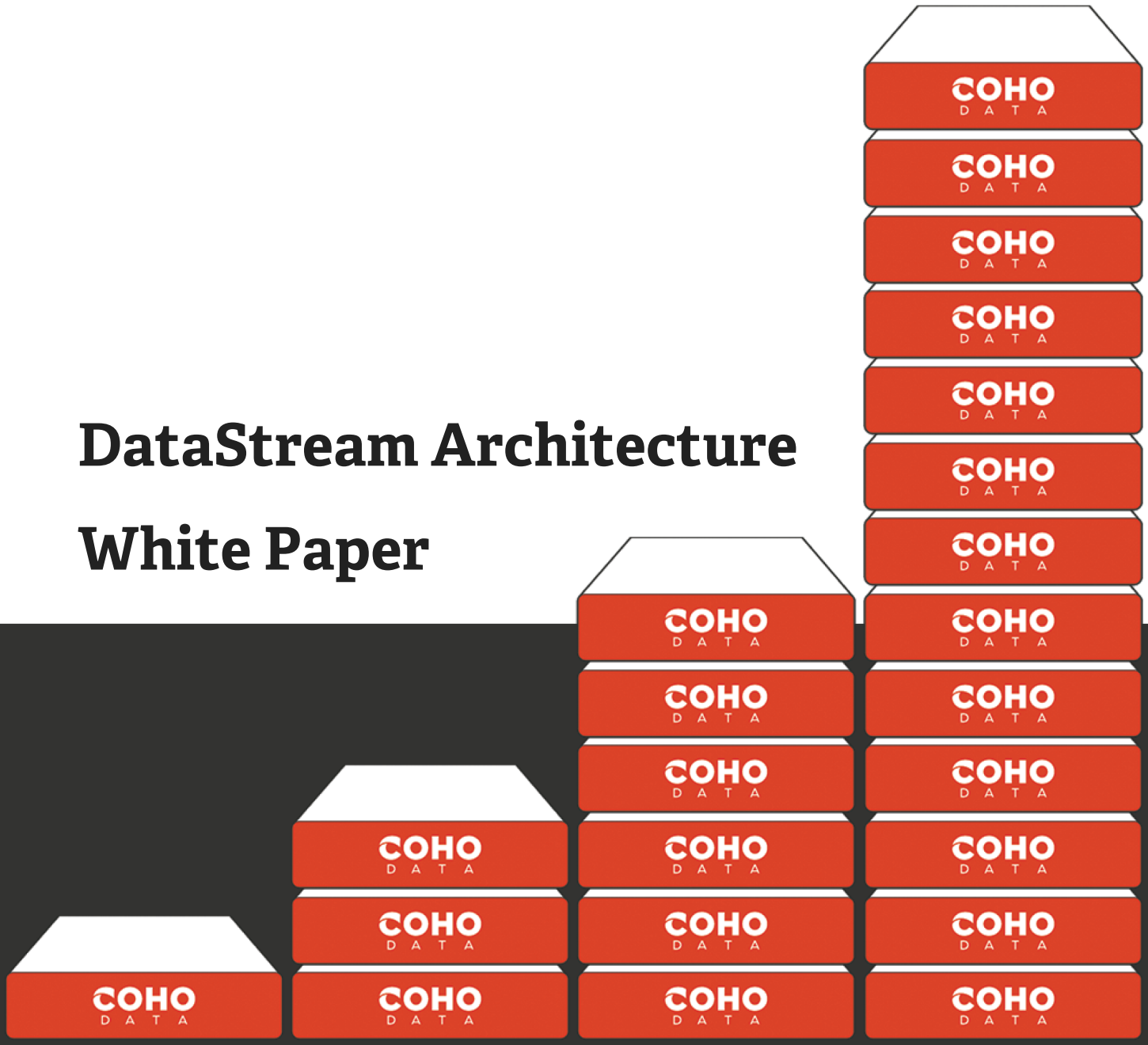
DataStream Architecture White Paper



COHO DATA

Storage for the Cloud Generation

Coho DataStream White Paper

The Era of the Monolithic Array is Over.

Virtualization has created a new paradigm for how IT can respond to business needs to provision workloads on demand, independent of underlying hardware. Software-defined networking is now enabling a similar model at the network layer, but storage remains the dinosaur that hasn't changed for decades. At Coho Data, we believe that era is coming to an end.

The number of ways by which applications talk to data has multiplied over the past decade. What used to be a handful of storage protocols - NFS, iSCSI, and Fiber Channel - has diversified to include object storage interfaces such as Amazon's S3, analytics-specific file systems such as Hadoop's HDFS, and NoSQL stores like Mongo, Cassandra, and Riak. Over this same decade, the hardware that is available to datacenter environments has also changed dramatically. First, flash emerged, went through several generations of development, and is now being offered as an enterprise-grade PCIe-integrated component. Second, 10Gb interfaces became commonplace on servers, and Ethernet switches inherited "software defined" capabilities including support for OpenFlow.

In fact, about the only thing related to data that hasn't changed over the past 10 years is enterprise storage itself. Despite fundamental shifts in workloads, interfaces, and technology, storage vendors continue to push the same thing they've been selling since Reagan was President: monolithic arrays. These boxes represent a tried and true business model, not unlike automobile or washing machine sales. As a business, you are forced to anticipate your needs over the next five years, buy a single box with a fixed set of parameters to support those needs, and then amortize cost until it's time to do another refresh.

It's All About the Data

At Coho Data, we believe that enterprise storage remains a critical part of the datacenter environment but the focus should be on the data, not the container. Your data is currency and it needs to be carefully protected, reliably available, and instantly accessible to any application that needs it.

Coho Data's scalable DataStream platform is built to free your business from the need to manage storage as a fixed asset with fixed interfaces. Inspired by the pay-as-you-grow economics of the public cloud and leveraging our founding team's experience supporting Amazon EC2 during their XenSource days, we enable you to invest in only the storage you need now, adapt to your application storage demands as they evolve, and to grow your storage investment flexibly over time. We take advantage of emerging technologies such as software-defined networking and commodity hardware like high-performance PCIe flash to provide breakthrough innovations in rapid scaling of capacity and performance with zero bottlenecks. This white paper explains the goals behind our approach to storage, and some of the core aspects of our product architecture.

Here are the key ideas that drive our approach to data management:

1. **Monolithic storage stacks need to go on a diet.** Modern flash devices are literally one thousand times faster than spinning disks. Array designers have always assumed that disks are slow, and so today's controller-based systems have layered functionality on top of them because of the lazy assumption that you can't really make disk performance worse than it already is. One of the array's main jobs has always been to aggregate spindles in order to present the illusion of a single, *faster* disk. On enterprise flash, this isn't true. A single PCIe flash device is fast enough to saturate a 10Gb NIC, so this historic approach of centralized array functionality in a single controller acts as a stack of *performance-damaging layers* that punish all clients equally. The base of the Coho DataStream platform is a ***data hypervisor*** for high-performance flash; the system virtualizes the bare metal storage hardware, and then gets out of the way of application performance.

   The result of virtualizing bare metal flash resources is a system that is based on individual **MicroArrays.** These are balanced combinations of PCIe flash, CPU and 10Gb network resources that form the new building block for a scalable, high-performance data path. The DataStream storage hypervisor enables storage memories to be safely shared by multiple tenants, over multiple protocols, without prescribing the central and heavy-weight storage stack that monolithic arrays have worn long in the tooth over the past 25 years.

2. **Buying storage shouldn't be like buying a car.** Just like an automobile purchase, enterprise storage has always been an up-front commitment, and an appeal to your lifestyle. Whether it's the sporty all-flash array, or the practical JBOD minivan, the decision is made up front and the resulting product parks in your server room, filling that niche for the next five years. This focus on packaging results in businesses having to buy and manage a myriad of containers for their various data needs. At Coho Data, we *started* by addressing scale and built it into every aspect of the system: you can add additional performance and capacity whenever you need, and the system will adapt dynamically to incorporate it. This means you buy only the storage you need now, and can let your business drive incremental purchases instead of the old model of buying enough for 3-5 years.

   An important aspect of this approach is that the Coho DataStream product line is entirely software-based. We package and sell on qualified OEM hardware because this allows us to deliver a storage offering that has been carefully tested and can be trusted to perform and be reliable in enterprise environments, but the product architecture does not depend on specific custom hardware. This approach to building datacenter storage allows our stack to incorporate the performance and density improvements that we know are emerging in flash, server, and networking hardware over the next decade of data center evolution. The design of our system anticipates hardware heterogeneity: while the hardware you buy from us next year may be twice as fast, it will still integrate seamlessly with the hardware you buy from us today. This approach allows you to benefit from the fast evolution of commodity hardware while responsively scaling your storage infrastructure investment on demand.

3. **Access matters. Protocols shouldn't.** Your storage system should support the most efficient possible interaction with your data. Legacy protocols like NFS are important because applications don't have to change to support them. Virtualized applications can easily access a DataStream NFS datastore via a single IP address while the business seamless grows capacity and performance behind that datastore. The DataStream platform allows data to be accessed over a linearly scalable NFS implementation that moves dynamically from 180K IOPs at 2 MicroArrays to 1.8M IOPs at 20 MicroArrays. But NFS is really just table stakes, and it's only a single example of a ***Data Profile*** running on DataStream storage.

   Our platform is extensible and supports a range of legacy and modern ways to integrate applications and data. One example of this extensibility is the ability for users to leverage the ***DataStream DirectConnect APIs*** to bypass NFS altogether and build applications that interact directly with MicroArrays at the lowest possible latency. Current joint customer development includes medical imaging application integration to create workflow-intelligent data fetching for a new level of data access speed for medical diagnosis, and digital content creation integration to create new high speed workflows with large media datasets. Another dimension of this extensibility is the ability to support ***Zero-copy Analytics*** by allowing users to launch Hadoop jobs on their data *directly within* the storage platform and run in the background behind production workloads. That's data analytics directly inside your enterprise storage platform, and not as an independent silo in your data center. Virtualizing at the storage layer means we support multiple storage tenant workloads, multiple forms of data presentation, all on isolated, private virtual networks.

4. **Manage data, not disks.** Enterprise arrays have always put the complexity of storage management squarely on the shoulders of the storage administrator. The organization wanted performance and durability, so it was the storage administrator's job to understand RAID, volume management, masking and zoning. Disks filled up and arrays reach the end of their support, so admins needed to carefully manage data migration across storage devices. We've taken enormous effort to make these issues the responsibility of the storage system itself: dynamic scalability means that data migration is obsolete, while storage profiles allow you to easily provision new data and choose performance and durability characteristics on demand.

   Our user experience aims to change the things that a storage administrator can spend their limited hours focused on: ***Performance costing*** allows administrators to identify workloads that are most expensive to support in flash from a performance perspective and to make clear decisions about whether to invest in a critical workload or to deprioritize an unimportant one. ***Data accounting/showback and chargeback*** enables administrators to associate data with the specific business units in their organization that are consuming it, and even supports the generation of monthly internal invoices on storage consumption to make the organizational accounting of IT clear and justifiable.

The Coho DataStream architecture is a complete redesign of the enterprise storage stack. It is incrementally deployable, high-performance, and lets you manage your data fluidly and flexibly. Achieving the four goals above involved solving some very challenging problems. In the following sections, we will dive into the details of the components of the system.

Section 2: The Birth of the MicroArray: A balanced storage building block

FLASH KILLS THE MONOLITHIC ARRAY

While flash-based storage hardware has been available for almost a decade, the past few years have seen a fundamental change to the technology. First, flash products have matured enough that vendors are now offering "enterprise-grade" devices that have strong guarantees in terms of durability and device lifetime. Second, and more importantly, flash has stopped trying to pretend that it's a disk. Devices have moved off of the slow (600MB/s) SAS and SATA buses where disks lived and onto the much faster (32GB/s) PCIe bus, which supports performance-critical hardware like network interfaces and GPUs. This change in connectivity is a big deal: PCIe flash devices are fundamentally different in terms of price, cost, and performance tradeoffs.

	Capacity	Performance	Latency	Power	Cost
15K RPM Disk	3TB	200 IOPS	10ms	10W	$200
PCIe Flash	800GB	50,000 IOPS	10µs	25W	$3000

The table above shows some characterizations of commodity disk and PCIe flash hardware that is available today; note that these numbers are optimistic about disks and conservative about enterprise flash. There are a few important observations to be made:

1. PCIe flash exhibits about one thousand times lower latency than disk and is about 250 times faster on a per-device basis. This performance density means that data stored in flash can serve workloads less expensively (16x cheaper by IOPS) and with less power (100x fewer Watts by IOPS). As a result, environments that have any performance sensitivity at all should be seriously exploring ways to incorporate PCIe flash into their storage hierarchies.
2. At full rate, a single modern PCIe flash card is capable of saturating a 10Gb/s network interface. As a result, the established technique of using RAID and on-array file system layers to combine multiple storage devices simply doesn't add up. There is no additional value on offer, other than capacity, to adding additional expensive flash hardware behind a single network interface. Moreover, unlike disks, the performance of flash is incredibly demanding on CPU. Using the numbers in the table above, the CPU driving the single PCIe flash device has to handle the same request rate of a RAID system using 250 spinning disks.
3. PCIe flash is about sixty times more expensive by capacity. So unless compression and deduplication is buying you more than a 60x reduction in capacity, you would be ill-advised to store all of your data in flash. Hybrid storage environments are going to be the best

approach to achieving value in enterprise storage, but the 1000x performance rift between disk and flash makes keeping hot data in flash absolutely critical.

4. Finally, it's worth noting that PCIe-attached flash isn't really the last point in a progression of flash memory form factors. It's the first example of a class of emerging solid-state memory technologies that will appear on the PCIe bus. Technologies including phase-change memory (PCM), spin-torque transfer (STT) and others will follow over the next few years. While these emerging technologies will take time to mature, they are all even faster than flash. Regardless of media technology, it's the movement to the PCIe bus and the resulting raw single-device performance that begs reconsidering storage system architecture.

In response to these emerging hardware trends, we reconsidered storage system design based around the primitive of the MicroArray as the building block for scale. A MicroArray is an evenly balanced combination of flash, CPU, and network connectivity, and it effectively results in a hardware component that directly attaches flash storage devices to the network and includes enough computing horsepower to drive them at full speed.

EACH 2U DATASTREAM CHASSIS HOUSES TWO COMMODITY-BASED SERVERS THAT SERVE AS INDIVIDUAL MICROARRAYS.

We build MicroArrays using the densest commodity hardware that we can find. Today, this means basing the system on a modular 2u, 2-server ***DataStream Chassis*** that houses two independent servers that each include two 10Gb NICs, two Intel Xeon CPUs, and two Intel 910 Series PCIe SSDs. Each of these servers is a MicroArray that independently connects to the network.

But what about reliability and scale?

The move to a MicroArray-based architecture leaves a challenging problem in terms of where storage logic should reside. Clearly, data needs to be stored across multiple MicroArrays in order to survive failure and to scale both load and capacity. Furthermore, IP storage systems like NFS and iSCSI have always assumed that they are talking to a single network-attached target and *not* a collection of independent nodes.

In solving this problem, we realized that Ethernet -- the datacenter network itself -- would need to replace the SAS or SATA buses used by monolithic arrays as the interconnect for storage devices. Rather than treat the network as a dumb medium that is outside our control, we elected to explicitly include it in the architecture. As a result, ***MicroArrays are aggregated using a 10Gb software-***

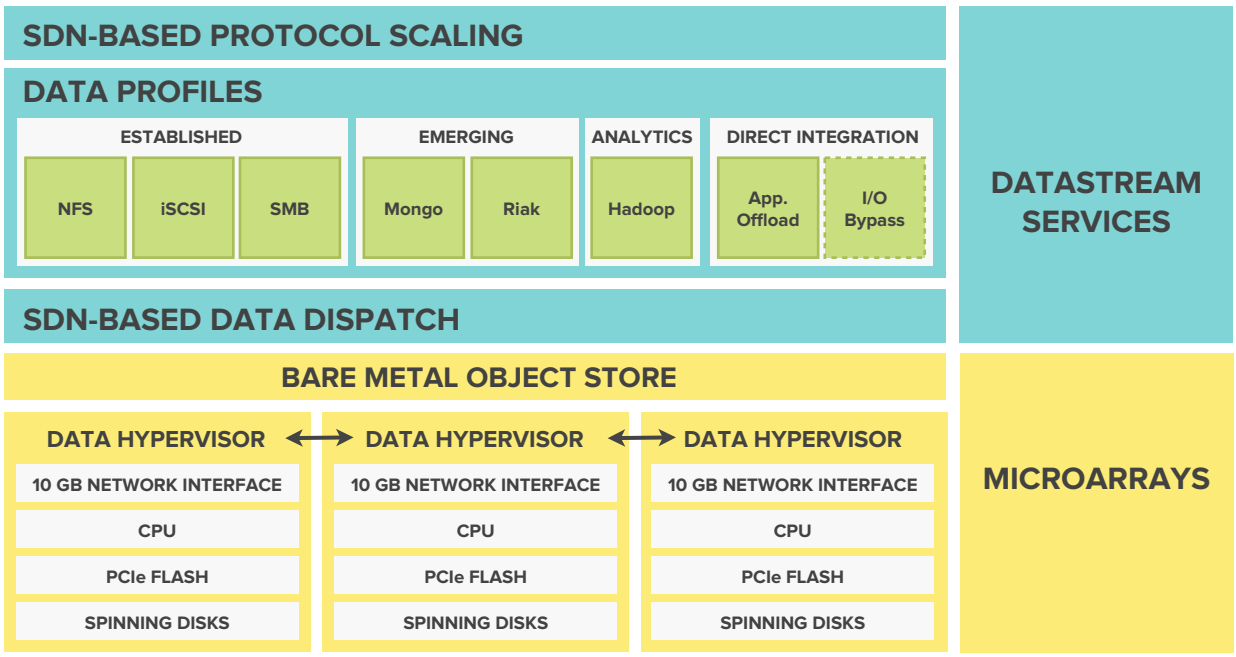
defined network switch that provides a rich set of functionality with which to present storage in a datacenter environment. The DataStream 1000 solution ships with a 52-port 10Gb Openflow-Enabled SDN switch, with support for dual switches in an active/active redundant configuration, to which all MicroArrays and clients are directly attached. SDN capabilities on the switch allow significant aspects of storage system logic to be pushed directly into the network, resulting in a completely unique approach to achieving scale and performance that wasn't possible five years ago.

What about spinning disks?

As mentioned above, spinning disks are absolutely critical in order to balance the capacity cost equation for most storage workloads. Unless you are using all of your data all of the time, there is very little benefit in paying to house it all in flash. In the DataStream 1000 packaging, each MicroArray includes 2 x 800GB Intel PCIe SSDs and 6 x 3TB spinning disk drives, and dynamically tiers data between those drives and flash. The DataStream software is designed to allow flash and disk capacities to scale completely independently of one another, so expect to see the introduction of additional form factors in our product line that will allow flash and disk to be deployed and scaled in arbitrary proportions.

Section 3: The DataStream Architecture from 10K feet

Using MicroArrays as a scalable building block and a commodity 10Gb Ethernet switching fabric as the interconnect, the DataStream architecture is a low-overhead and high performance software architecture that ties things together. The high level view of the architecture appears in the diagram below.



DATASTREAM ARCHITECTURE OVERVIEW

The design of the system divides storage functionalities into two broad and independent areas. At the bottom, MicroArrays and the data hypervisor that they host are responsible for bare-metal virtualization of storage media and for allowing hardware to be securely isolated between multiple simultaneous clients. Like a VMM, coordinated services at this level work alongside the virtualized resources to dynamically migrate data in response to the addition or failure of MicroArrays. They also provide base-layer services such as lightweight remapping facilities that can be used to implement deduplication and snapshots.

Above this base layer, our architecture allows the inclusion of an extensible set of ***hosted, scalable Data Profiles*** that are able to layer additional functionalities above the direct storage interfaces that lie below. These personalities integrate directly with the SDN switch and may be hosted in isolated containers directly on the individual MicroArrays. This approach allows a development environment in which things like NFS controller logic, which has traditionally been a bottleneck in terms of storage system processing, to transparently scale as a storage system grows. The hosted NFS implementation in our initial product runs on every single MicroArray, but interacts with the switch to present a *single external IP address*.

The interface between these two layers again involves the SDN switch. In this situation, the switch provides a private, internal interconnect between personalities and the individual MicroArrays. A reusable library of dispatch logic allows new clients to integrate onto this data-path protocol with direct and configurable support for striping, replication, snapshots, and object range remapping.

Dividing the architecture in this manner allows us to focus on obtaining performance, scalability, and reliability right at the base, while providing extensibility to easily incorporate new interfaces for presenting and interacting with your data over time. The DataStream 1000 offers a high-performance NFS target for VMware and non-virtualized workloads. With future releases, new data profiles will emerge, including the ability to deploy Hadoop-based analytics directly on your stored data and to take advantage of HTTP-based key/value APIs. With the extensibility of the DataStream DirectConnect APIs, users may elect to integrate their in-house applications to interact directly with the bottom-level MicroArrays and reduce protocol, library, and OS overheads.

Section 4: Object storage on a data hypervisor

BARE-METAL ISOLATION AND SCALABILITY

High-performance solid state storage inverts the parameters in designing enterprise storage: when the disk was the bottleneck, we used techniques like RAID-based striping or erasure coding to add up the individual performance contributions of many disks connected to a single controller. Since a single PCIe flash device can saturate a 10Gb network port, adding extra processing in front of it is only going to get in the way of performance. The right way to deal with the presentation of shared, high-performance storage hardware is to *get out of the way* -- to present the lightest possible interface that allows applications to use storage as efficiently as possible. Coho Data's MicroArray model achieves this by implementing a ***data hypervisor*** that does for enterprise storage what a VMM does for a CPU: it is a minimal, high-performance layer that allows tenants to safely share storage hardware with isolated performance. Just like a VMM, the data hypervisor is concerned

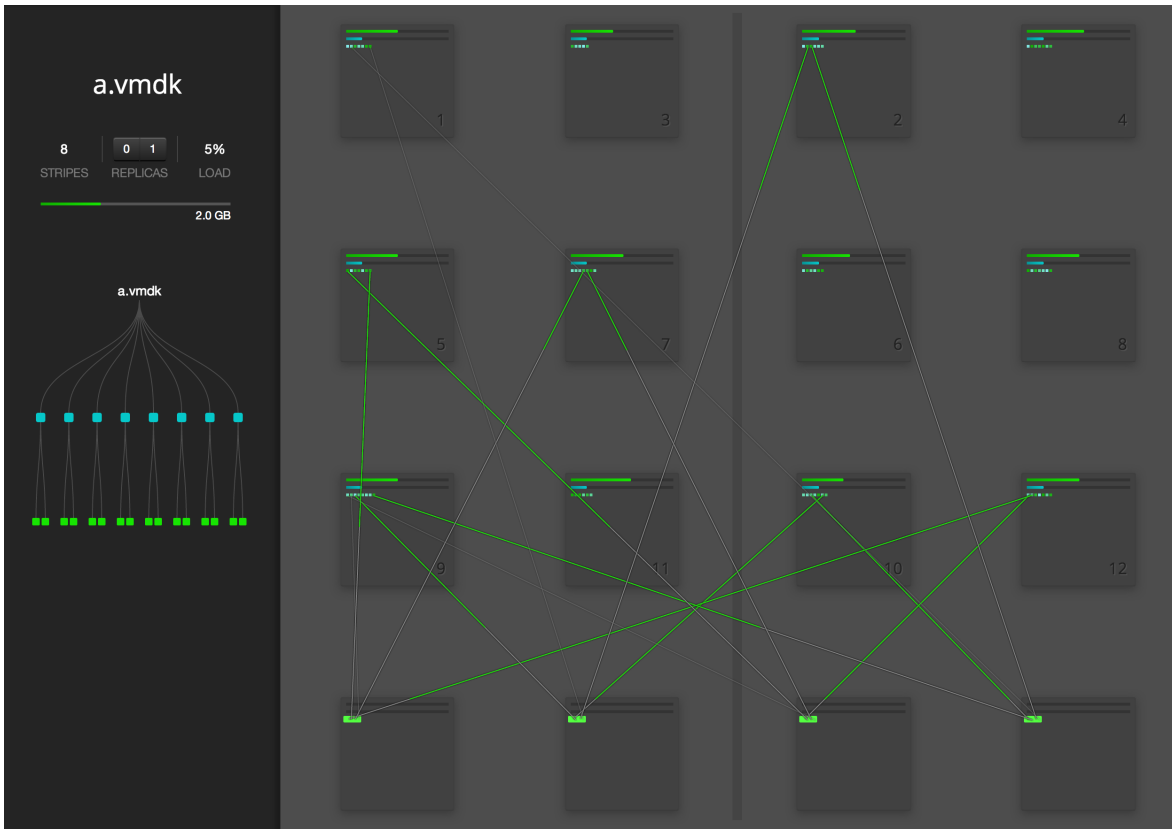
only with isolation and performance. It doesn't add the fat of additional layered file systems, volume managers, RAID parity calculation, or prescriptive protocol implementations. Its job is just to allow you to safely and efficiently share scalable storage hardware across multiple applications.

The data hypervisor is responsible for virtualizing the flash address space and allowing secure and isolated access to the flash device by multiple simultaneous clients. Just like a VMM divides a single machine into VMs, and as VLANs allow the partitioning and isolation of a shared physical network, the data hypervisor virtualizes flash and allows multiple clients to access their own isolated data at the same time. This virtualization creates a high-performance, bare-metal object store. Each MicroArray allows the creation of arbitrarily many sparse objects and ensures that these objects can only be accessed by their owners. Individual objects on each MicroArray are coarse-grained containers that can be used as primitives by the data personalities to build more complex storage abstractions.

Dynamically managing objects as data containers.

The data hypervisors on the MicroArrays work together to manage and maintain objects over time. Background coordination tasks at this layer monitor performance and capacity within the storage environment and dynamically migrate objects in response to environmental changes. If a new DataStream Chassis is added, two additional MicroArrays will come online. A balanced subset of objects from across the existing MicroArrays will be scheduled to migrate, while the system is still serving live requests, onto the new MicroArrays. Similarly, in the event of a failure, this same placement logic recognizes that replication constraints have been violated and triggers the reconstruction of lost objects. This reconstruction can involve all the MicroArrays that currently house replicas and can create new replicas on any other MicroArray in the system. As a result, recovery time after device failure actually *decreases* as the system scales out.

It is important to recognize that the placement of data in the system is *explicit*. Old approaches to storage -- such as RAID and the erasure coding techniques that are common in object storage systems -- involve an opaque statistical assignment that tries to evenly balance data across multiple devices. This approach is fine if you have large numbers of devices and data that is accessed very uniformly. It is less useful if, as in the case of PCIe flash, you are capable of building a very high-performance system with even a relatively small number of devices or if you have data that has severe hot spots on a subset of very popular data.



A VISUALIZATION OF DATA REBALANCING AS FOUR ADDITIONAL MARRAYS ARE ADDED TO AN ACTIVE 12-MICROARRAY SYSTEM.

The figure above shows a visualization of a running system in which four new MicroArrays have just been added. The data hypervisor's placement logic has responded to the arrival of new nodes by forming a rebalancing plan to move some existing objects onto the new nodes. The system then transparently migrates these objects in the background and immediately presents improved performance and capacity to the system. Note also that the left margin of the figure shows how the top-level image file a.vmdk is mapped onto 16 MicroArray-based objects.

Section 5: The DataStream Switch: Any Data Over any Interface

SOFTWARE DEFINED NETWORKING (SDN) AND SCALABLE DATA PROFILES

The DataStream data hypervisor forms a base layer that enables storage performance and capacity to scale out dynamically and to respond to environment and workload changes. It achieves this by providing the thinnest possible interface to the underlying storage hardware. It also acts as a hosting layer for ***data profiles*** -- distributed implementations of storage protocols -- and provides an extensible environment for data-intensive compute to be placed next to the data that it computes on. Tight integration with the DataStream SDN switch allows these distributed profiles to be presented to clients as a single storage target, transparently scaling even legacy protocols, while a collection of composable SDN dispatch modules makes it quick and easy to implement new Data Profiles.

SDN Data Dispatch

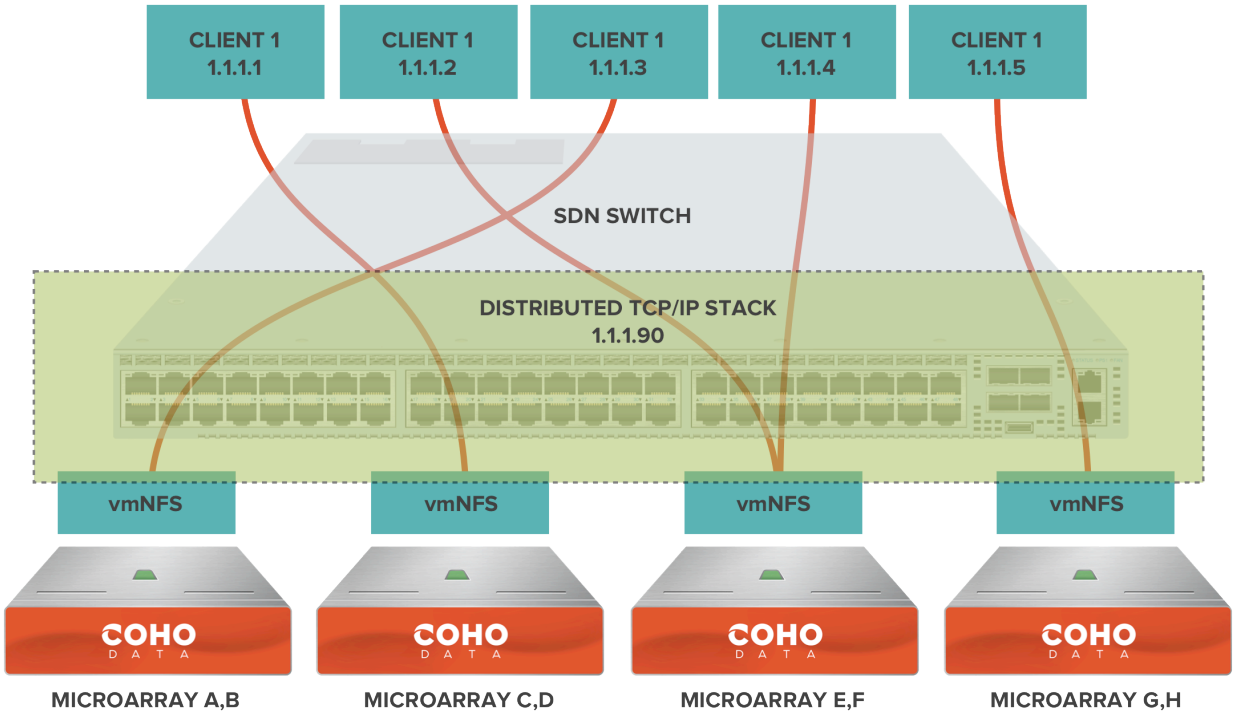
The DataStream dispatch library builds upon the blazingly fast low-level data hypervisor interface to provide a composable set of storage request transformations that can be combined to create high-level storage abstractions like striping and replication. These features are important both in terms of balancing request load to reduce latency and in surviving the failure of individual devices, but they should be optional and configurable so that they can applied appropriately depending on what is being stored and how it is being used.

The data dispatch interface allows complex and important portions of storage logic to be decoupled from the underlying low-overhead MicroArrays, but still reused across multiple data personalities. Dispatch tables are created by combining these pluggable modules, and requests are forwarded through them much like a packet is forwarded over a network according to rules on a switch or router. In fact, once an object is composed, this is *exactly* what happens: the dispatch library interacts with the SDN switch to forward read or write requests for a given data address to the appropriate MicroArrays that house that data. The library also provides an opportunity for applications to bypass conventional storage protocols and interact *directly* with the MicroArrays themselves, providing a low-latency network path that scales out both performance and capacity as additional MicroArrays are added to the system.

Scaling Storage Protocols

A key limitation of conventional IP storage protocols is that they present storage targets as a single IP address. While this may not be a problem for traditional monolithic arrays that sit behind a relatively narrow network interface, it quickly becomes a bottleneck for the DataStream architecture. The DataStream Switch addresses this limitation by using SDN capabilities to intelligently route client connections to multiple backend servers over a single logical IP address, making it easy for data profiles to scale legacy protocols.

One of the key Data Profiles in the DataStream 1000 is vmNFS, an NFSv3 server specifically designed to meet the high throughput, low-latency demands of virtual machine workloads. On existing NFS architectures, the NAS head (also commonly called a “controller”) is a choke point. It is limited by CPUs in terms of the number of requests that it can serve and limited by network links in terms of the volume of data that it can provide. In the case of IP storage, it also suffers from the fact that it generally lives on a specific network device, addressed by a single IP address.



VMNFS INSTANCES ON EACH MARRAY IMPLEMENT A DISTRIBUTED TCP/IP STACK THAT ALLOWS ACTIVE NFS SESSIONS TO SCALE ACROSS THE CLUSTER AND MIGRATE IN RESPONSE TO LOAD.

The vmNFS Data Profile in the DataStream architecture addresses the scalability of NFS request processing by distributing the NFS server implementation across multiple MicroArrays. As shown in the diagram above, vmNFS still presents a single NFS server IP address, but the TCP/IP stack and NFS processing load is effectively distributed across all of the backend hardware. The SDN switch assists in transparently steering NFS sessions to the appropriate backend targets and mitigates hot spots through active load balancing, while the vmNFS instances parse client requests and interact with the scalable DataStream dispatch libraries to issue requests through the data hypervisor.

The two graphs below show both static and dynamic scale of vmNFS running on between 2 and 20 MicroArrays. The graph on the left plots achieved IOPS for a fully random, 4K 80/20 read/write workload being issued by 80 concurrent clients across 10 physical VMware ESX hosts. All writes are fully replicated. Performance scales linearly up to almost 1M IOPS using the VMware NFS client, which is limited in the number of outstanding requests that it will issue at once. The second (blue) plot in this graph shows scale for NFS clients running within each of the 80 VMs, and achieves nearly 1.4M IOPS in 11u of rack space (including the 1u DataStream Switch). The second graph shows the same workload scaling dynamically: 4 initial nodes offer a load of about 180K IOPS, and as additional cables are attached, connecting new MicroArrays to the system, objects are migrated and performance scales up dynamically. Other than adding the cables, no reconfiguration is required of the system.

COHO DATA

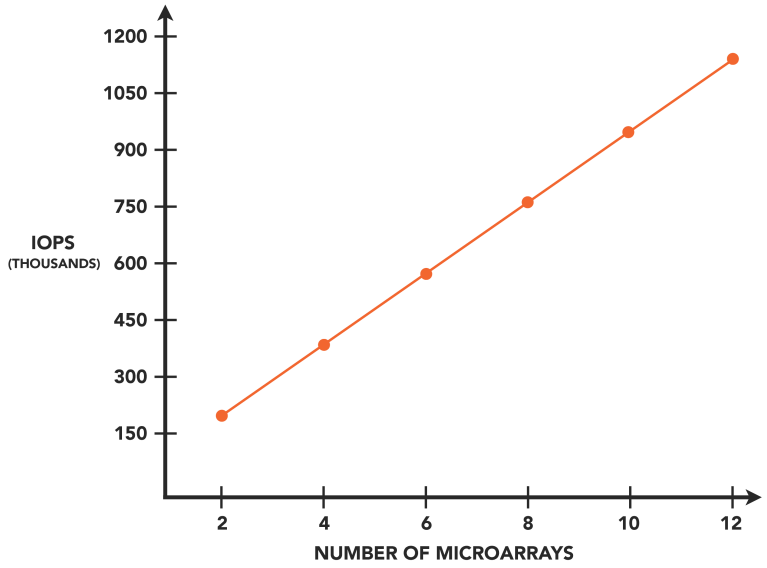
Joint Benchmark with Intel 910 Series SSDs

Workload: Random 80/20 read/write, 4k Block Size

Linear Performance Scaling with ~180K IOPS/2U COHO Chassis

1M+ IOPS in 1/3 Rack

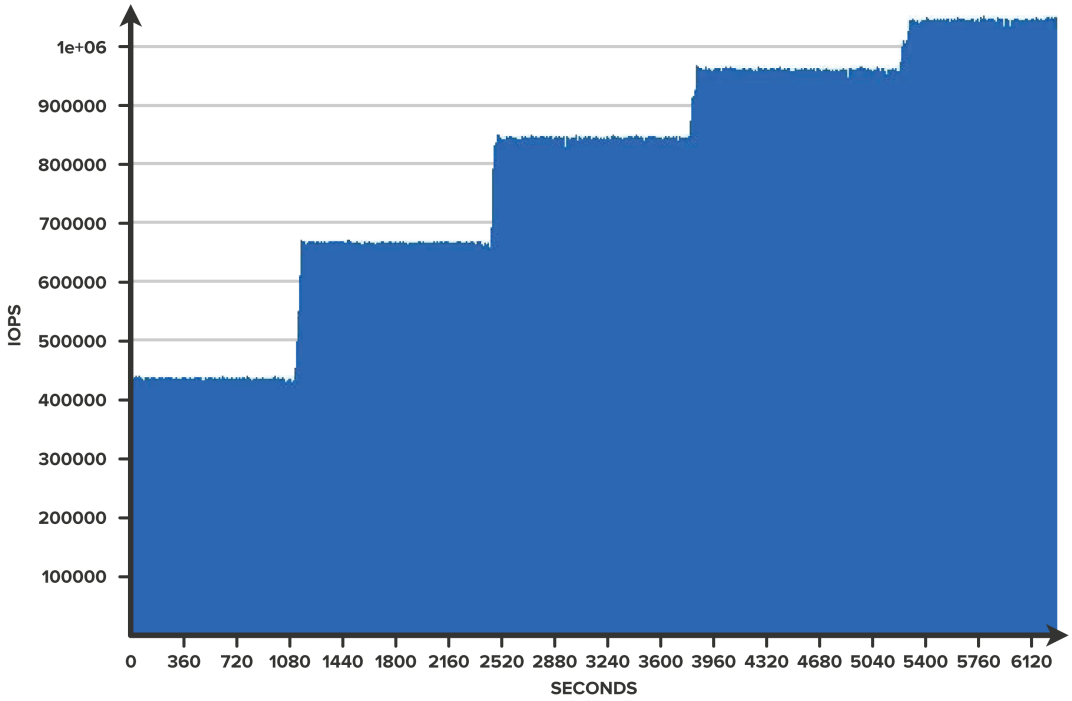
Validated by Enterprise Strategy Group Labs



COHO DATA

COHO DataStream MicroArray Rebalancing

GROWING TO 10 MICROARRAYS



vmNFS is a full-featured NFS v3 target that is integrated with VMware's storage APIs in order to support the offload of object cloning, as well as supporting a specification of storage profiles for workload-specific performance prioritization. It allows administrators to specify performance requirements on a per-object basis, allowing the system to optimize its use of flash across the cluster automatically without the creation of fixed tiers. This workload metadata supplements the access frequency based weighting so that only the right workloads are delivered higher performance, enabling greater overall efficiency of the system in aligning with business priorities.

Section 6: Turning Tiering Upside Down

ENGINEERING AGAINST A 1000X SLOW DOWN

Storage systems have always involved a hierarchy of progressively faster media, and there are a set of very well established techniques for attempting to keep hot data in smaller, faster memories. In general, storage system design has approached faster media from the perspective that slow disks represent primary storage, and that any form of faster memory (frequently DRAM on the controller, but more recently also flash-based caching accelerator cards) should be treated as cache. As a result, the problem that these systems set out to solve is how to *promote* the hottest set of data into cache, and how to keep it there in the face of other, lower-frequency accesses. Because caches have historically been much smaller than the total volume of primary storage, this has been a reasonable tactic: it is impractical to keep everything in cache all the time, and so a good caching algorithm gets the most value out of caching the small, but hottest subset of data.

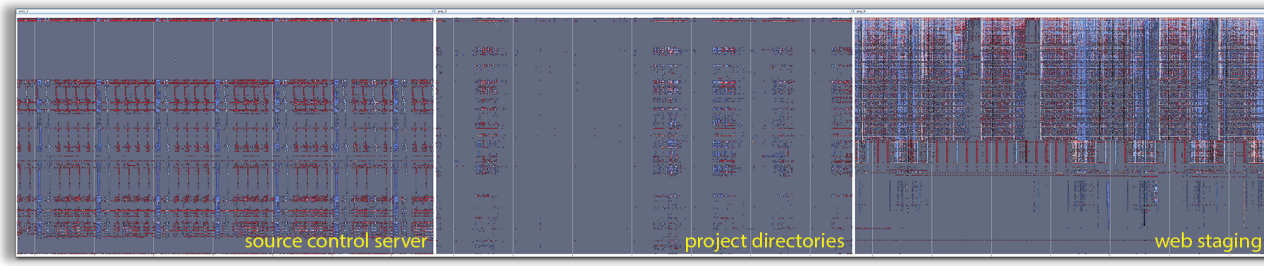
The economics of high-performance flash suggest a different approach to designing storage systems. Given that PCIe flash is about a thousand times faster in terms of both throughput and latency than random access to a spinning disk, our belief is that *all* hot data should reside in flash. As a result, our design approaches dynamic tiering from the opposite direction from that of most systems: rather than identifying the hot data that should be moved up to a higher-performance tier, we focus our efforts in identifying the cold data that is unlikely to be read again in the near future, and move that to disk.

This difference is subtle but profound: given a large amount of data and a high rate of access, it is much easier to identify data that *won't* be accessed in the near future than it is to identify the data that will. Moreover, as scalability is a core property of the system, the size of the higher-performance flash tier can be grown as necessary to accommodate a growing set of hot data. Each MicroArray in the DataStream 1000 line has a ratio of three 3TB disks per 800GB flash device, with new configurations to be made available in the near future. The data hypervisor uses an on-disk strategy that we call the ***continuous data layout*** to allow objects to be placed across a mix of flash and spinning disk in response to expected accesses.

One significant challenge faced by any tiering approach lies in automatically identifying what data is important. Disk-based systems have historically been very frugal in terms of investing resources to get caching right. Caches have historically been relatively small, and cycles have been limited. As a result, most caching strategies use some variant of "Least Recently Used" (LRU) caching, to evict

the stalest piece of data in the cache whenever a new item is accessed. Unfortunately, as you attempt to grow the high-performance tier to encompass *all* accessed data, you quickly find a situation in which large amounts of data are accessed repeatedly over time, but with inter-access gaps that are long enough that data will be retired from the cache before it is reused.

Rather than using demand-based strategies like LRU, the DataStream architecture's continuous data layout takes a more analytics-based approach to planning data placement. The high-speed nature of flash allows us to collect a continuous and detailed log of all storage accesses made by each client. As shown in the three diagrams below, these accesses are analyzed and clustered over time to identify highly correlated accesses, and accesses that happen predictably at certain times. The system uses the results of this internal, time-series analytics on data access to identify data that can be safely removed from flash altogether, and also to proactively pre-populate flash with data that is accessed on a predictable periodic pattern. One example of the latter are the bootstorms associated with virtual desktop infrastructure environments as users log-on in a predictable time range each weekday.



ACCESS ANALYTICS ON THREE DIFFERENT STORAGE WORKLOADS. TIME IS ON THE X-AXIS FOR A ONE-WEEK TRACE, AND THE Y-AXIS INDICATES THE DISKS'S ADDRESS SPACE. ACCESSES ARE REPRESENTED AS A HEATMAP OVER TIME, WHERE BLUE INDICATES READ-HEAVY ACCESS AND RED INDICATES LARGE NUMBERS OF WRITES.

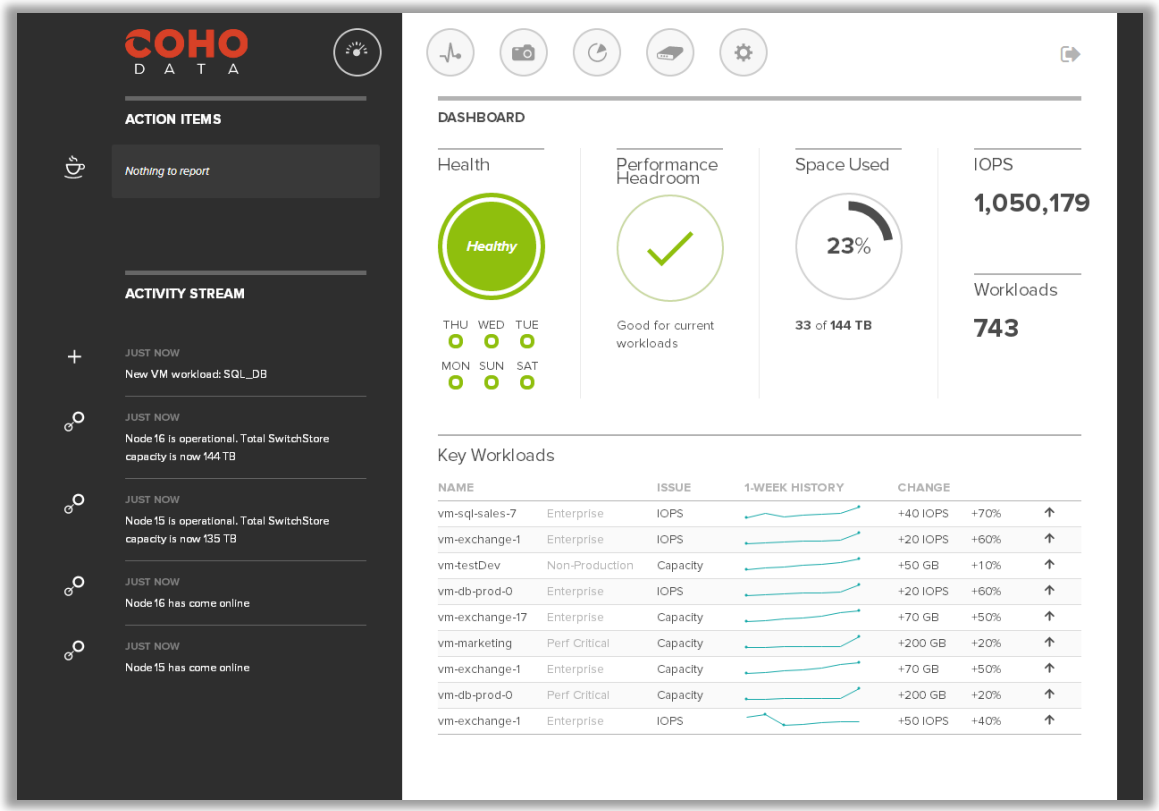
Our internal workload analytics provide significant opportunities to improve the efficiency of flash utilization and overall system behavior. They also allow us to characterize how well flash is serving each individual workload and to provide feedback for admins as to whether they would benefit from adding additional flash to the environment. This is seen in the management UI in the form of a Performance Headroom status indicator that reports on the overall performance utilization of the system and triggers alerts when the system believes additional flash is required. The goal is to significantly reduce the time-consuming storage administration task of monitoring and tuning traditional storage tiers to align with workloads as performance needs grow and shrink dynamically. We believe this set of storage tasks can be accomplished most efficiently by the system itself with the access analytics approach mentioned previously, enabling IT to use the DataStream solution as a storage service with lower configuration and ongoing management costs.

Section 7: Storage as a Service

BUILDING A SELF-MANAGING STORAGE SERVICE

The final aspect of our system design described here is a fundamental layer, as it is the interface to the IT team of the business and seeks to achieve the goal of delivering a highly self-managing storage service. In building pay-as-you-grow and grow-as-you-go scalable storage system, we set out to build a service-oriented management paradigm that was largely decoupled from any single piece of storage hardware. After extensive design and redesign based on direct customer trials, the resulting management UI includes fewof the features that you would normally expect in a storage UI, and allows administrators to focus their attention on a set of newer facilities that aim to help admins do their job in a very different way. The goal is minimize the time required by admins for both configuration and ongoing management of the DataStream storage service. Thus only issues that require user intervention are highlighted in the top-left (and also emailed to the admin if this option is configured). Ongoing tasks around managing performance like load balancing around the system are largely hidden behind the scenes, but summary indicators on the Dashboard provide IT with an overall health status for the system, the availability of performance headroom (calculated based on the capacity and access patterns of flash in the system), capacity utilization and total average IOPS being used. Key workloads are also reported on the main Dashboard so that notable changes in performance of specific workloads can be quickly and easily identified.

As the system grows, perhaps with different versions of DataStream hardware, this single management interface is used to view the health of the overall system, thus delivering continuous service as your business grows. For an in-depth look at the management UI, please refer to the website at cohodata.com for product demo videos.



Conclusion

This white paper has described many of the aspects of the Coho DataStream storage system design. Our approach to building enterprise storage is a dramatic redesign of conventional systems that is motivated by the changing demands of datacenter applications and the evolving capabilities of both network and storage hardware. By taking a clean-slate approach to storage system design, Coho Data's architecture achieves performance and scale through common and narrow low-level interfaces to MicroArrays, and then allows a flexible interface for building extensible data profiles in order to present that storage to applications over whatever interface suits them best.

We're very excited about the plans that we have to evolve this platform over the coming years, and we'd love to hear your thoughts on how it should grow.