



Gridstore All-Flash Hyper-Converged Infrastructure for the Cloud-Inspired Data Center

Architectural Overview





Table of Contents

The Cost of Traditional Infrastructure has Become Oppressive
The Cloud-Inspired Data Center
All-Flash HyperConverged Infrastructure Overview
Radically Simple Infrastructure
Automated Environment: Single-Pane-of-Glass Management
Gridstore Hardware Packaging
HyperConverged Appliances: 2U with Four Independent Nodes
Storage Appliances: 1U or 2U with a Single Storage Node
Existing Infrastructure
Gridstore Distributed Software Architecture
Grid Nodes
Architecture Components Overview
Architectural Features
Non-Disruptive Scaling / Upgrades
Fault Tolerant
Distributed Everything
Distributed Parallel Flash Architecture
HyperConverged (Compute and Storage) Architecture
Independent Node Scaling (Scaling Compute and Storage Separately) \ldots . 7
Programmable Infrastructure
HyperConverged Infrastructure Design Goals
Overview
Gridstore HyperConverged Infrastructure Design Goals
Conclusion15



The Cloud-Inspired Data Center reduces capital costs, management effort, operating costs and operating risk.

The Cost of Traditional Infrastructure has Become Oppressive

According to Gartner,¹ HyperConverged Infrastructure is growing at 50% cumulative annual growth rate. By 2017, 50% of enterprises will be using deploying it for Webscale IT. The reason: "The cost of traditional infrastructure has become oppressive." Traditional infrastructure in today's data center has become a series of complex technology silos that require deep technology skills for different layers of the infrastructure stack and large up-front investments. When outgrown, they require rip-and-replace upgrades that are risky, disruptive, and expensive. The larger these environments become, the greater the complexity grows, and the higher the risk and operational cost become.

The Cloud-Inspired Data Center

For the past decade or so, companies that were built on the concept of scale—the hyperscale Web properties such as Amazon, Azure, Facebook, and Google—had to find a better way to build data center infrastructure. These companies built the largest data centers in existence and demonstrated how to build highly scalable, resilient infrastructures using low-cost commodity hardware and intelligent software efficiently managed by a handful of people.

The key to these hyperscale data centers, characterized by unprecedented scale, economics, and efficiency, is simplicity and a software layer that pools compute and storage resources from many standard X86 building blocks into a massive resource. Intelligent software automates the provisioning and control of resources to each workload.

This approach to infrastructure has become widely known as HyperConverged. Three layers of traditional infrastructure are consolidated into a single tier built from X86 building blocks and intelligent software that automates everything. Eliminating layers of infrastructure results in:

- **Reduced capital costs:** Two of the three infrastructure layers are completely eliminated.
- Reduced management effort: Less infrastructure means less to manage and a geometric reduction in discrete management points between discrete infrastructure components, which grows geometrically as more are added.
- Reduced operating costs: Reducing management effort frees staff for value-added activities. Decreasing infrastructure with the potential to break reduces management requirements. Reduced space, power and cooling all combine to drive down the cost of operations.
- Reduced operating risk: Instead of risk being concentrated within an infrastructure silo, HyperConverged architectures are designed to expect many component failures. By using standard components as shared resources, the system simply works around faults and self-heals using added X86 building blocks.

¹ Gartner Magic Quadrant for Integrated Systems, June 16, 2014

WHITE PAPER | GRIDSTORE ALL-FLASH HYPER-CONVERGED INFRASTRUCTURE





Legacy Infrastructure Silos (Compute, Network, Storage)

All-Flash HyperConverged Infrastructure Overview

Radically Simple Infrastructure

HyperConverged Infrastructure collapses three layers of infrastructure into a single layer of commodity hardware that reduces cost and complexity. A Gridstore HyperConverged



Appliance combines everything on the left in the diagram above into a single 2U appliance that contains four independent servers, SAN and all-flash storage.

HyperConverged Infrastructures can be config-

ured with any combination of either HyperConverged Nodes (which contain compute and storage) or Storage Nodes (which allow independent scaling of compute and storage resources).

Compute and storage resources are pooled together into shared resource pools that are available to the virtualized data center environment. Up to 256 HyperConverged Nodes or Storage Nodes can be incrementally added to a pool to linearly scale the environment, as more resources are required.



Any combination of HCNs and/or SNs

Micr Cor	r osoft htrol, A	Syster utoma	n Cen te, Mor	ter + A hitor, S	zure F elf Ser\	Pack vice
VM.1	VM.2	VM.3	VM.4	VM.5	VM.6	VM.n
5,000 Min	9,500 < Min	—vм- ↓	I/O Con	itrol —	2,000 Max	2,800 Max
	El	astic (Compu	ite Po	ol	
HyperConverged Infrastructure						
	El	astic All-Fla	Storag sh I	e Poo Hybrid	s	

Automated Environment: Single-Pane-of-Glass Management

Management of the environment resides on a single Microsoft System Center console. All resources in the environment are provisioned automatically and monitored through Microsoft System Center.

Gridstore Hardware Packaging

The Gridstore hardware packaging uses two form factors that can be combined seamlessly into the same Grid[™]. Each form factor contains one or more nodes that add resources to the Grid.

Regardless of form factor, each node functionality is the same and connects to the Grid over either 1 or 10GbE. Form factors only differ in density of compute and storage resources in order to better meet different scaling requirements.

HyperConverged Appliances: 2U with Four Independent Nodes

HyperConverged Infrastructures contain up to four independent nodes that can be a combination of HyperConverged Nodes, and Storage Nodes.

Each HyperConverged Node runs Windows Server 2012 R2 Hyper-V and Gridstore vController[™] software to access storage resources in the pool (see Architecture Components Overview for more). HyperConverged Nodes can vary in CPU RAM specifications to meet different workload requirements.

Each Storage Node contains either all-flash storage (eMLC SSD) or a combination of flash and SATA disk in Hybrid Storage Nodes. Storage Nodes are bare-metal appliances and do not require additional licensing of OS or hypervisor. Storage Nodes run Gridstore software to manage the storage resource as part of the storage pool.

Typically, HyperConverged Nodes are used to meet the needs of scaling Tier 1 compute and storage resources (all-flash storage). All-flash Storage Nodes can be incrementally added to HyperConverged Infrastructures to scale Tier 1 storage resources as required.



Storage Appliances: 1U or 2U with a Single Storage Node

Independent storage nodes packaged in a single appliance can be added in various-sized increments.

Storage Nodes add direct-attached storage into the storage pool. Storage Nodes are bare-metal X86 servers which do not run a hypervisor or provide any compute resource to the pool. As single-purpose appliances, they provide only storage resources to the Grid (see Gridstore Hardware Packaging for more).

Existing Infrastructure

Other Compute Nodes: Compute-only nodes can be any X86 server running either Hyper-V or Windows Server 2012 R2. Existing customer-owned compute nodes can access storage resources in the Grid through a Gridstore vController running on this host (see Gridstore Hardware Packaging for more). vControllers present a local SCSI block device to the host with the same feature set as a HyperConverged Node and access to the shared storage pools.

Network Fabric: Network I/O between all Grid nodes is over 1 or 10 GbE connections. A separate VLAN is used for storage I/O. Storage pools must be contained within the same subnet.

Gridstore Distributed Software Architecture

Grid Nodes

A Grid is made up of physical nodes that run Gridstore services on bare metal. A Grid node can be one of three types: Compute Nodes (CNs run Gridstore Virtual Controller (vController)), Storage Nodes (Storage Nodes run Gridstore Resource Controller (rController™)) or HyperConverged Nodes (HyperConverged Nodes run both vController and rController). This bare metal peer-to-peer architecture allows for maximum flexibility in deployment and scaling options. Compute-Only Nodes can be any X86 host running Windows Server 2008 or later. Storage Nodes and HyperConverged Nodes are available from Gridstore.



Architecture Components Overview

vControllers: A virtual controller presents a local SCSI block device to a host. vControllers run as kernel mode drivers in the Windows Server OS host. A host can be either a HyperConverged Node or a Compute-Only Node (any customer server that runs Windows 2008 R2 or higher). vControllers communicate directly and in parallel with a number of rControllers over a proprietary distributed peer-to-peer protocol.

rControllers: Resource controllers provide storage resources in parallel to a number of vControllers over a proprietary peer-to-peer protocol. They are delivered in both HyperConverged Nodes and Storage Nodes. rControllers are delivered as a complete appliance.

Tie in existing infrastructure with any X86 server running Hyper-V or Windows Server 2012 R2.



Parallel Block Protocol (PBP): This is a peer-to-peer direct parallel protocol that controls I/O between (*n*) vControllers and (*n*) rControllers. PBP controls the distribution of data blocks within a storage pool of (*n*) rControllers. PBP is the data path between any vController and the rControllers that provide storage resources to the vController. PBP controls block distribution, fault detection, self-healing, and maintains fault tolerance across a number of rControllers in the Grid.

Parallel Control Protocol (PCP): This is a peer-to-peer direct parallel management protocol between (*n*) nodes in a Grid (also known as GridControl[™]). PCP operates out of band to coordinate and manage events within the Grid. PCP manages all processes that deal with hardware provisioning (nodes joining or leaving the Grid), storage pool management, storage provisioning (to vControllers), fault monitoring, detection and self-healing, operational telemetry gathering and monitoring, and non-disruptive upgrade processes.

Storage Pool (SP): Storage pools aggregate storage resources from both HyperConverged Nodes and Storage Nodes. A storage pool can span up to 256 physical nodes (either HyperConverged Nodes or Storage Nodes) and reside within the same subnet. Subject to security, the storage pool can only be accessed by vControllers running on any node within the Grid. Storage pools automatically expand when a new Gridstore HyperConverged Node or Storage Node is added to the network.

Bare-Metal Server Architecture: Gridstore is a peer-to-peer architecture where components/drivers/services run on bare-metal servers and do not require any services from the hypervisor. This architecture allows the greatest number of deployment options, consumes the fewest possible resources from the host, and provides the highest performance. Most other HyperConverged architectures operate at the guest-VM level and run as full iSCSI or NFS servers. This consumes significantly higher resources, impacts performance, and adds configuration complexity.

Architectural Features

Non-Disruptive Scaling / Upgrades

After the initial setup and configuration, any type of node can be added to the Grid non-disruptively to add resources to the compute or storage pool (or both). New nodes are auto-discovered and auto-provisioned. They can be added to the pool immediately or can wait in a standby mode for future provisioning. Each Grid can contain multiple resource pools. Each resource pool can contain up to 256 nodes (either HyperConverged Nodes or Storage Nodes).

Software upgrades (of Gridstore software or the host OS) can be performed non-disruptively to Grid operations. For HyperConverged Nodes where the OS is patched or upgraded and that require a reboot, the Grid will control the process of taking nodes offline, performing the patch or upgrade, restarting the nodes and bringing the nodes back online and to a healthy state.



With currently available HyperConverged Nodes, compute pools can contain up to 6,144 CPU cores, 256TB RAM and 10.2TB of parallel bandwidth (shared with storage).

256 Independent Nodes per Pool

Storage can be scaled up to 1.5PB of flash storage for Tier 1 and up to 12PB of Hybrid Storage for Tier 2/3 workloads.

Add resources nondisruptively to compute or storage pools, or both.





End-to-end parallel allflash architecture reduces latency to the extreme. Performance of both reads and writes are enhanced.

Fault Tolerant

Instead of building reliability at the hardware level, which adds complexity and expense, Gridstore followed the design principles of Web-scale pioneers like Google to build fault tolerance in software at the node level. This architecture assumes components will fail, and the system works around failures. Gridstore is fault tolerant to multiple failures of entire nodes without disruption or down time. If failures occur, the system is self-healing to retain the desired protection levels in the Grid. When a node fails or any fault occurs, the Grid will automatically determine what is required to return to a fully protected state. This can be as simple as replaying the last sequence of I/Os or as complex as the complete rebuild of a failed storage node.

Distributed Everything

Every component in the Grid is distributed and autonomous. This includes the control plane—which manages configuration, provisioning, monitoring, and alerting—and the data plane, which manages both metadata and block data. This architecture eliminates any single points-of-failure and eliminates performance bottlenecks through a highly parallelized protocol between nodes in the Grid.

Distributed Parallel Flash Architecture

Gridstore leverages an end-to-end parallel flash architecture to reduce latency to beyond what is possible even with all-flash storage arrays. Because Gridstore operates on both the compute and storage sides, flash media can be used to accelerate I/O at both ends of the network. HyperConverged Nodes can optionally utilize a compute-side flash read cache to deliver performance with the latency of a local direct-attached flash device (typically measured in microseconds). This minimizes latency by removing the network hop, packet processing, and erasure encoding and offloads reads from the underlying storage nodes.

This improves overall performance in two significant ways. First, reads are distributed to the local node and performed with minimal latency from local direct-attached storage. Second, reads are off-loaded from the storage resources, freeing these resources to process inbound writes as quickly as possible. In a typical environment, read/write ratios can be as high as 80/20 read/write. This architecture allows reads to perform with microsecond latency while freeing storage resources to primarily process only writes to yield the fastest possible combination.

HyperConverged (Compute and Storage) Architecture

The architecture is a peer-peer grid of vControllers and rControllers communicating in parallel with each other. The Grid architecture allows these components to operate concurrently on the same host. In this model, each vController and rController operates autonomously to perform its service.

Independent Node Scaling (Scaling Compute and Storage Separately)

The Grid architecture allows independent scaling of compute and storage to address one of the major limitations of current hyper-converged solutions that force customers to grow servers at the rate of storage growth (dependent scaling).

Dependent scaling happens in architectures that run their distributed services either in a guest VM on a full-blown NFS, iSCSI, or SMB server or that run in the hypervisor itself. Both of these models require a full hypervisor running on a full compute stack in order to add storage capacity to their pools.



Gridstore addresses a major limitation of prior solutions with independent scaling of compute and storage. The Grid architecture eliminates this through independent bare-metal scaling. Independent storage nodes that do not require a hypervisor or full compute stack can be added to the Grid. Storage nodes (appliances) use a stripped down embedded OS and sufficient compute resources for the Grid rController to operate. This architecture eliminates the licensing and management costs of additional hypervisors and reduces the CPU, RAM, and network costs of the appliance.

The independent scaling Grid architecture also allows existing compute to be used by non-disruptively adding a vController to physical or virtual hosts. These compute nodes can then access resources in the Grid like any other node. This capability allows customers to maximize their current investments and avoid creating another infrastructure silo.

Programmable Infrastructure

The concept of the programmable infrastructure (PI) was pioneered by Web-scale companies, and is one of the keys to simplifying the data center.

The first step in simplifying the data center is the use of standard commodity hardware components (building blocks) that are aggregated together into shared resource pools through intelligent software. The infrastructure becomes standardized and easy to scale by adding more resource blocks to the shared pool and growing the aggregate resources linearly. In effect, this architecture combines simple resource building blocks into a massively parallel machine that can be orders of magnitude more powerful than the largest monolithic system.

Programmable infrastructure dynamically carves up this powerful resource pool and provisions a virtual infrastructure of virtual machines, each with specific resources to meet the specific virtual machine's workload requirement. PI provisions, monitors, and controls the real-time allocation of resources from the pool to each VM. Effectively PI provides the real-time creation of virtual infrastructure to meet the demands of each workload running on that infrastructure.

HyperConverged Infrastructure Design Goals

Overview

The Gridstore HyperConverged Infrastructure is designed to meet five goals that address critical issues raised by customers using first generation HyperConverged Infrastructure. These critical issues include:

The Performance Impact of Guest-VM-Level Integration of Compute and Storage Running distributed storage services in a guest VM adds to cost, complexity, and performance overhead. Having to run full-blown iSCSI or NFS servers in a guest VM consumes valuable CPU cycles and anywhere from 32GB RAM up to 96GB as the minimum recommended memory allocation for the controller VM from different vendors. This either adds considerable cost for the added resources or impacts the VM density running on the

The Cost of Three-Way Infrastructure Replication

hosts, which adds cost for more hosts to carry all VMs.

First-generation HyperConverged Infrastructures all use three-way replication to protect against data loss in the distributed storage pool. While replicating low-cost storage capacity is a common practice, replicating the entire infrastructure stack 3X is cost prohibitive and wasteful.



Independent Scaling of Storage from Compute

One of the major drawbacks of first-generation HyperConverged Infrastructure is that it forces users to scale compute with storage. Storage is consumed daily and typically grows at four to six times the rate of compute. When they consolidate compute and storage into a HyperConverged Appliance, customers are forced to grow the entire infrastructure at the rate of storage. This leads to a massive compute and hypervisor cost and sprawl that adds to management cost and complexity.

Gridstore HyperConverged Infrastructure Design Goals

Gridstore eliminates these three critical issues through its architecture and five design goals.

1. NATIVE BARE-METAL INTEGRATION

One of the key drivers of HyperConverged Infrastructure is simplicity. Due to tight integration, it should also deliver the best-performing system.

Running as a guest VM is complex, resource intensive, and significantly slower. Aside from VMware VSAN, which runs in the hypervisor kernel (which has some common issues), all



other hyper-converged products run as guest VMs using an NFS server or iSCSI target in a full Linux server. Issues encountered include:

- **Complexity:** Setup, configuration, and maintainance of walled-off resources for a controller VM (cVM—runs an NFS, iSCSI or SMB server) on each hypervisor is time-consuming and adds management complexity.
- Resource Intensity: Running a full VM consumes unnecessary CPU and memory resources. Vendors' guidance for minimum memory allocations to cVMs range from 32GB to 96GB. The maximum size is not uncommon for an entire host running 10-20 or more guest-server VMs. This must be allocated to the controller alone.
- Latency: A full virtual stack must be traversed twice—up and down—for each I/O, adding significant latency to every I/O. Moving between kernel and guest VMs requires many context switches that add significant CPU processing requirements and latency into the I/O path.
- Lack of I/O Control: Presenting an NFS mount point or iSCSI target does not eliminate the I/O blender. This is one of the original downsides of virtualization—visibility and control of I/O on a per-VM basis is lost because all I/O is blended together. The ability to control storage I/O for each VM is critical to enable a programmable Infrastructure. A guest VM running an NFS server in the host is exactly the same as having the NFS server across the network. Nothing has changed, it is effectively the same traditional storage, and the problem persists.

HyperConverged

Infrastructure design goals:

- Native integration
- Predictable performance
- Efficient scaling
- All-flash infrastructure
- Independent scaling



Bare-Metal Native Windows Architecture

As described in the architecture section, Gridstore software has two core components—one on the compute side and one on the storage side. Each compute node runs a vController. This is a native windows OS kernel driver that operates at the host level (below the hypervisor) and requires no services from a hypervisor. The vController presents a local SCSI block

HyperConverged Compute / Storage Node



Fastest Data Path – Minimized Resource Utilization

device to the Windows Server (2008 or later). This block device is SCSI-3 compliant, allowing multiple hosts to share the same device through a Clustered Shared Volume (CSV). As a result, it delivers a number of benefits, including:

- Eliminating Complexity: With no setup or configuration required, users can power on and provision directly out of the box. All storage is added to the storage pool, giving each host immediate access to this pool upon initial startup.
- Efficiency: By nature, kernel-level drivers are small and efficient. With Gridstore's native kernel-level SCSI device, the fastest and most efficient storage path is achieved as it reduces guest VM overhead, context switches, and memory overhead.
- **Fastest Data Path:** Operating in the kernel is the fastest data path. A SCSI block device presents the fastest device with the least amount of overhead.
- **Flexibility:** Allows independent resource scaling.

2. PREDICTABLE PERFORMANCE AND GRANULAR CONTROL PER VM

Traditional storage was designed to present a LUN to a server. This worked well until that server became a hypervisor running 10, 20 or 30 virtual servers, all pointing to the same LUN.

Adding to the complexity, all hypervisors need to point to the same LUN to enable VM mobility or high availability. Architecturally, this creates a many-to-one relationship that makes optimization impossible. Each VM has a different I/O pattern. When they all mix together, the aggregate I/O becomes highly random, creating



performance issues with no way to isolate problems or control I/O per VM.

HyperConverged Node



Isolate and Optimize I/O per VM

The core component of the Gridstore architecture is the vController that operates in the host kernel and appears as a local SCSI device. From this privileged position, it isolates I/O from each of the VMs into independent swim lanes or channels from the hypervisor across the network and onto storage nodes in the Grid. This recreates the 1:1 relationship between VM and storage. Once I/O is isolated, each channel can be independently optimized according to what the VM is trying to achieve. Optimizations occur dynamically optimizing the I/O pattern to the demands of the VM.



Predictable performance per VM

With the I/O isolated into channels per VM, the vController can now overlay QoS policies or set specific IOPS minimum/maximum per VM to give predictable performance per VM.

Upon provisioning VMs through Virtual Machine Manager (VMM) and classifying them as a platinum service, Gridstore will deliver a platinum storage service. Furthermore, administrators can dial in a



specific amount of IOPS to a given VM to ensure that it allocates a minimum amount of I/O to enable it to run optimally.

Equally, lower-grade classes can be created or limits on IOPS can be dialed in to ensure that no VM consumes more than it requires. This is beneficial for service providers who want to offer different levels of service to customers, since it ensures they can deliver what they promise, resulting in predictable environments that run as desired for all workloads.

End-to-End QoS Architecture

Due to the unpredictable performance in virtual environments, QoS has become a hot feature. However not all QoS is the same. Three models for QoS are:

- A. Hypervisor based
- B. Array based
- C. End-to-end

A. Hypervisor- (Host-) Based QoS

Hypervisors create QoS in the hypervisor by throttling I/O. This is effective to ensure that one or more VMs do not consume all the IOPS, the noisy neighbor problem. This model, however, only operates in a single hypervisor host. It does not provide a global view (i.e. storage resources being consumed by other hosts, like a database on a physical server). Furthermore, a hypervisor-based model is not in control of the actual storage resource, making it impossible to allocate more resources to a particular VM or physical host to meet a minimum IOPS reserve.

B. Array-Based QoS

This is the opposite of the hypervisor model.

In this configuration a storage array presents multiple LUNs to the hypervisor. Each LUN has a QoS priority attached to it, and I/O is more or less "allocated" to these LUNs based on the prioritization. Problems with this model include:

- Being LUN based, it is not granular to the VM level
- The I/O blender is still blending, which means I/O that passes through a LUN is blended together from multiple VMs, resulting in no per-VM granularity. With this, hundreds of VMs on that LUN compete for resources. When a VM on the highest priority LUN runs wild and consumes all resources, every other VM is impacted. According to the QoS policy, it did its job. It served the highest-priority LUN and starved the other LUNs. However, it does not know that all this I/O is coming from a single VM at the expense of others. It does not know that they are on the same highest-priority LUN.

Traditional Storage Architecture

Server VM.1 VM.2 VM.3 VM.1 VM.2 VM.3 VM.n + + + + + + + I/O Blender
Hypervisor LUN
Storage Array Priority I/O
Control Plane
Data Services
Data Plane (I/O)
Capacity



An array-based QoS model does not have any ability to control the flow from the hypervisor end. Rather, its domain of control is within the array.

C. End-to-End Per-VM I/O Control

In a highly virtualized environment, delivering predictable performance to VMs from a shared elastic resource is critical.

Gridstore's vController technology isolates I/O from each VM at the source and enables endto-end per-VM I/O control through the entire storage stack.

By controlling both the resource consumption (VM hosts) and the resource allocation (storage resources from Grid nodes), Gridstore can precisely monitor demand as well as control the allocation of resources across the hypervisor hosts and storage resources. End-to-End VM-I/O Control



End-to-end VM I/O control enables the ability to dial in specific I/O per VM or class of VMs, thus ensuring SLAs are met. The result: complete control of resource allocation and utilization.

3. EFFICIENT INFRASTRUCTURE SCALING

Reduce Infrastructure Cost and TCO by 50%

The three-way replication architecture of first-generation HyperConverged Infrastructure is considered standard in Webscale architectures using commodity hardware. While this model delivers the required protection, it comes at a high cost.

When a write occurs on one node, the full write is replicated to two other nodes in the cluster.

This generates the following costs:

- 1. It creates high latency due to the synchronous replications that must complete before the I/O if committed
- It doubles network traffic to achieve this—for every write, two replicas are sent across the network to two other hosts
- 3. It consumes 3X the storage resources—for every TB written, three TBs are consumed
- **4.** It wastes 50% of IOPS available in the system—assuming a 75/25 read/write split, the replica writes consume 50% of the available IOPS

All of this replica activity costs host resources, with the CPU and memory handling all this additional TCP/IP traffic. In addition to wasting resources, it consumes VMs, resulting in lower VM density, thus driving even greater hardware requirements.





Eliminate the Three-way Replica

Because Gridstore operates in the kernel at the source, it uses erasure encoding to protect the data before it leaves the host. Erasure encoding is an advanced forward error correction method used for communications.

Erasure encoding breaks a block of I/O into N fragments and encodes these fragments. This means that a number (K) of these fragments can be lost but the data still can be reassembled as long as (N-K) fragments are available. The encoded fragments are distributed in parallel to N resource controllers anywhere in the Grid. This method protects against multiple component failures, communications failures, and entire node failures.

Scale with 50% of the required resources

The illustration shows an apples-apples comparison for HyperConverged Infrastructure appliances using the same form factor.

By eliminating the replicas, Gridstore delivers the same usable capacity and the same protection (any two complete node failures) with 50% less equipment required.

While eliminating 50% of the cost of required equipment up front, it also eliminates 50% of the ongoing operational costs, including data center footprint, power, and cooling, as well as the licensing stack for hypervisors, management software, and maintenance.

Gridstore also operates with significantly higher efficiency in terms of memory and CPU utilization. This enables us to achieve our next design goal, the All-Flash HyperConverged Infrastructure.

4. ALL-FLASH HYPERCONVERGED INFRASTRUCTURE

Efficiently and cost effectively deliver the highest possible performance

Because 66% of storage resources are wasted on three-way replicas, all-flash hyperconverged appliances have been cost prohibitive. Gridstore's next-generation architecture is the first and so far only design that eliminates this problem and makes all-flash HyperConverged Infrastructure practical.

Gridstore uses flash in an end-to-end architecture to both accelerate performance and to eliminate as much as 80% of storage traffic, returning CPU cycles to hosts. Gridstore All-Flash HyperConverged Nodes use flash as a local cache and as a high-performance storage resource. By putting a large flash cache (1TB) close to the workload, the entire working set of approximately 30 server VMs or 200 desktop VMs can be cached locally. Our vController sits in front of this cache, becoming part of a controlled data path with VM granularity. This critical

feature ensures the cache resources are utilized according to per-VM policies and not consumed just by the most active VMs.

With Gridstore, approximately 80% of I/O (reads) is served from the local low-latency flash cache. This eliminates 80% of





3-Way Replicas 12 Servers, 72TB

Replica



storage traffic that would otherwise go across the network and allows storage nodes to focus on serving the remaining 20% (writes).

By comparison, an all-flash array must handle 100% of the I/O (reads/writes) in the same controller through the same network connections.

Gridstore's end-to-end distributed all-flash architecture will serve about 80% of its I/O with microsecond latency by not going across the network for reads. When compared to millisecond latency for an all-flash array where all I/O reads/writes incur the network latency, this results in 10X lower latency, which is critical for high performance workloads like SQL, server virtualization, VDI, and big data.

5. INDEPENDENT SCALING

Scale to fit your exact requirements

Since the arrival of HyperConverged Infrastructure, the biggest customer complaint has been inefficient scaling. First-generation products force customers to scale servers at the rate of storage growth. To make matters worse, three-way replication force customers to grow servers at 3X the storage growth rate. For some workloads that are uniform (such as VDI), this is not a problem and many installations have started with these workloads. For many other general purpose workloads, however, storage grows considerably faster than the compute workloads do.



Unlike other HyperConverged Infrastructures, Gridstore is not iSCSI, NFS, or SMB serving I/O. Rather, Gridstore is a peer-to-peer Grid architecture designed specifically for this purpose, addressing the following critical scenarios:

- Independent Scaling: Independently scale storage only using purpose-built storage nodes that can be deployed incrementally as storage is required. This supports mixing different classes of available storage (flash, hybrid, and capacity) to allow different types of capacity pools to grow to meet different workload requirements or budgets.
- 2. Compute Scaling: Scale compute only using any X86 server that runs Windows. A Gridstore vController can be deployed in less than a minute to any Windows server. That server, in turn, can become a node in the Grid, with full access to all shared storage. Adding SSDs to any of these servers enables the vController to use them to accelerate workloads on servers. Hosts can be physical servers or hypervisors. If they are physical



servers, features like end-to-end storage QoS can be applied to deliver precise amounts of IOPS to each of these servers.

Hence, the Gridstore architecture allows scaling of both compute and storage separately. Compute can be scaled using any or existing X86 servers while storage can be scaled to fit precise requirements using storage-only nodes.

Conclusion

Gridstore's HyperConverged Infrastructure is purpose-built for the Microsoft Cloud. Either with flash or hybrid storage, elastic pools enable massive, independent scaling of compute or storage by eliminating the use of three-way replicas. Furthermore, Gridstore's patented vController provides both affordable fault tolerance and end-to-end flash performance.

For more information, contact a Gridstore representative or email info@gridstore.com.



www.PaperlessProductivity.com





FOLLOW US ()/gridstore (in /company/gridstore ()/user/GridstoreInc

t US 855.786.7065 US 650.316.5515 UK +44(0)20 3553 3662 e info@gridstore.com www.gridstore.com

© 2015 Gridstore. All rights reserved. Gridstore, the Gridstore logo, AutoPilot, Direct I/O, FlashGrid, Grid, GridControl, GridProtect, GridScale, GridSensor, HyperConverged Appliance, rController, Server-side Virtual Controller Technology (SVCT). Thin-Provisioned VLUNS, TrueQoS, vController, vmOptimized, vPool, and vStore are registered trademarks or pending trademarks of Gridstore in the U.S. and other countries. All other tradems are the property of their respective owners. Information regarding products, services and offerings may be superseded by subsequent documents and are subject to change without notice. For the latest information and specifications regarding Gridstore and any of its offerings or services, please visit www.gridstore.com. 040115