WekaIO Matrix™ Architecture

TECHNICAL WHITE PAPER

CONTENTS

Introduction

WekaIO was founded on the idea that current storage solutions have been largely incremental improvements to existing designs, causing the gap between advances in compute performance and those of data storage to continue to widen. Storage remains a bottleneck to application performance. In today's hyper-competitive market, organizations need flexible infrastructure; application workloads are becoming increasingly complex and data sets are continuing to grow unchecked, forcing enterprises to architect overly complicated, overly costly systems that reduce IT agility. As a result, important business insights remain locked away, out of reach of decision makers.

IT organizations are adopting cloud technology for its fluid, on-demand scalability that supports diverse workloads at scale. However, while network and compute can be virtualized to operate at scale very effectively, storage remains largely isolated in silos based on system performance profiles. Consequently, organizations are forced to architect a storage system that is highly customized for their environment and workloads from building blocks that do not scale. The result is a storage solution that is complex, temperamental, expensive, and slow.

What's needed is a radical change.

WekaIO Matrix™ is a software-only high-performance file based storage solution that is elastic, highly-scalable and easy to deploy, configure, manage, and expand. The design philosophy behind Matrix was to create a radically simple storage solution that has the performance of all-flash arrays with the scalability and economics of the cloud that runs on-premises or in the public cloud.

WekaIO Matrix (figure 1) highlights

- Supports bare metal, containerized, virtual, and cloud (on-prem, hybrid, public) environments
- Deployable as hyper-converged, a dedicated storage server, or native in the cloud
- Flexible application storage access, including POSIX, SMB, NFS, HDFS, and REST S3
- POSIX compliant adaptable performance for small and large files, mixed random and sequential I/O patterns
- Application level 4k IO latencies lower than any All-Flash-Array, unlimited random IOPS performance – scales linearly with the size of the cluster
- Supports more aggregated throughput than any parallel file system solution
- Automated built-in tiering across fast solid state devices (SSD) and cloud or on-prem object storage
- Distributed resilience eliminates bottlenecks of traditional data protection. End-to-end data protection provides additional protection against intermittent hardware failures
- Removes complexity of traditional storage and associated management

Figure 1 – WekaIO Matrix Feature Summary

Common Storage Challenges in the Cloud Era

IT organizations face a host of challenges as they try to match the flexibility and scalability of the cloud. Today's applications have a wide variety of performance (IOP, bandwidth, latency), scale, and other requirements that may make them unsuitable to run in existing cloud infrastructure. This coupled with the diversity of application file formats, access protocols, and data structures leads to data silos and IT complexity. Trying to eliminate these conditions through consolidation often results in user aggravation when application performance is negatively impacted.

Storage architects try to work around these limitations by adding multiple storage tiers and using costly all-flash arrays (AFAs). AFAs are optimized for performance but they do not provide the scale and resiliency of the cloud, and large applications may not fit within a single array. Furthermore, AFAs provide block storage protocols that are not sharable across servers, thus they are unsuitable for shared storage use cases. The result of these workarounds has been a costly and endless cycle of rearchitecting the infrastructure to keep pace with changing application requirements.

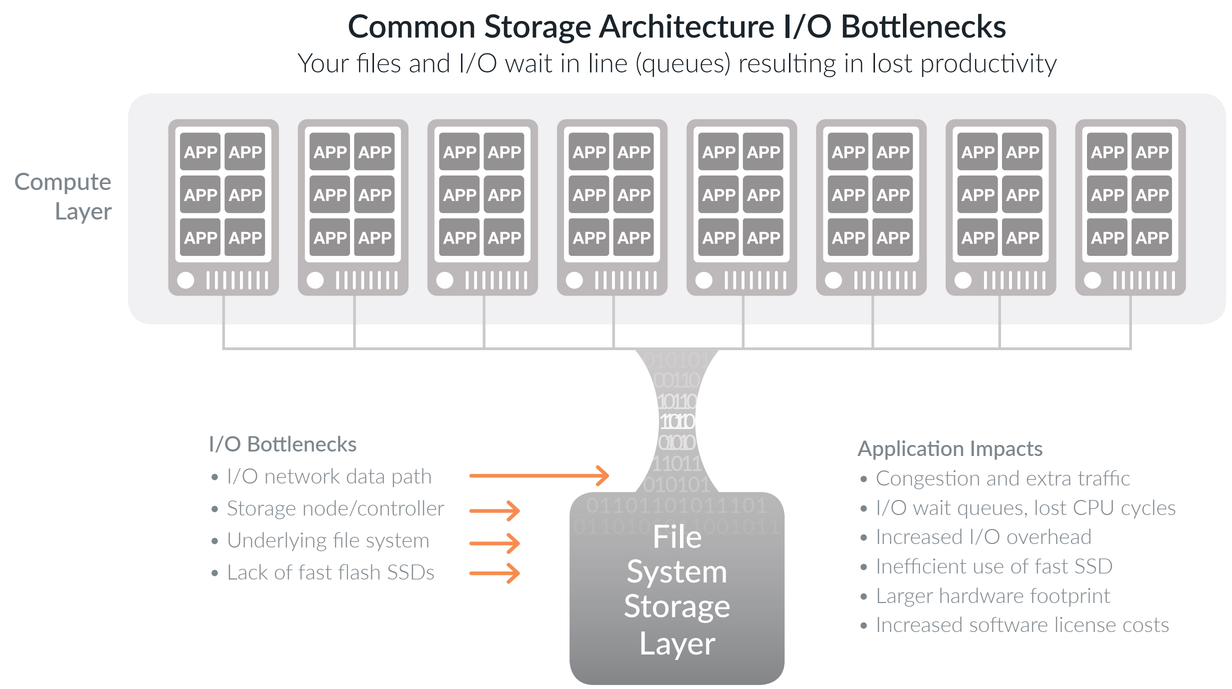


Figure 2 – Common storage system and implementation bottleneck impacts

Figure 2 above shows the locations of I/O bottlenecks found throughout the storage layer. Moving down through the I/O stack, the first chokepoint is encountered at the network layer along the data path between servers and storage. NFS is a very "chatty protocol" implemented over TCP/IP that results in extra I/O overhead between clients and file storage servers. Clients continually ask the storage server for metadata, which leads to longer application run times and lost productivity.

The second chokepoint involves the storage system and appliance controllers due to their hardware and software architecture limitations. Traditional storage architectures route traffic through a pair of controllers, so performance is bound by the controllers' abilities to process simultaneous I/O requests. In the case of file servers, the underlying file system can also induce latency despite having fast flash SSDs.

The greatest storage related challenge, however, is often overlooked; it is the time and effort required to deploy, provision, and expand storage as it is needed. This includes planning, the purchasing and delivery cycle, and installation, testing and provisioning, all of which results in application deployment delays.

Traditional Storage Architectures

Understanding which storage solution is best suited to a particular environment or application workload can be a daunting task given the variety of options available. Some solutions are optimized for performance (IOPs, bandwidth, low latency) while others are optimized for high-capacity and low cost. Especially problematic are workloads such as general HPC, EDA, geonomic research, and seismic analysis, all of which generate both large file sequential access and small file random access. No one storage design has been able to address all of these workload patterns. The workaround has always been to use multiple storage systems and complex tiering and management platforms.

Software Defined Storage was supposed to be the answer, yet many of these software based solutions have inherent design limitations. For example, performance and capacity scaling are often limited and rigidly tied together such that the need for more performance requires the purchase of more capacity. In addition, the software may be supported only on a limited set of specific hardware configurations or come with scaling or hypervisor based limitations. Finally, many of the current implementations use costly redundancy based protection schemes that, when coupled with current NAND FLASH pricing, end up being too expensive.

Designing a Unified, Cloud Era Storage Solution

When designing a modern storage solution, a key design consideration is that technology is inherently prone to failure and technology changes over time. A true software defined storage solution should accommodate such changes, which means that it must be able to run on any available hardware, adapt to customer environments, add cloud-like agility, scalability, and on-demand performance. It should also be simple to deploy and expand fluidly without incurring the typical procurement delays associated with traditional external storage appliances.

WekaIO Matrix is a software-only storage solution that was designed from the beginning to run on any standard Intel x86 based server hardware and commodity SSDs. This eliminates the cost of expensive specialized hardware and allows you to take advantage of improvements in technology without the pain of regular forklift upgrades to next generation architectures. The limitations created by legacy design constraints led the founders of WekaIO to develop and patent innovations in data distribution and protection, load balanced network file access, virtual file systems supporting multi-tiered storage, and many others.

WekaIO Matrix – Radically Simple Storage Without Compromise

Matrix solves the common storage challenges previously mentioned by eliminnating the chokepoints that impact application performance. It is well-suited for challenging environments that need sharable storage with low latency high performance, and cloud scalability.

Example use cases include:

- Electronic Design Automation (EDA)
- Life sciences
- Machine learning, artificial intelligence
- Web 2.0, on-line content, and cloud services
- Porting any application to a public or private cloud
- Media and Entertainment (rendering, after-effects, color correction)
- Financial trading and risk management
- General High-Performance Computing (HPC)

Matrix software delivers a solution that would have traditionally required several different types of storage systems by leveraging existing technologies in new ways and augmenting them with new innovations. The resulting software solution delivers high performance for all workloads (big and small files, reads and writes, random, sequential and metadata heavy), and because it is designed to run on the existing application compute cluster with direct cloud connectivity, it requires no additional datacenter hardware footprint.

At the core of Matrix software is a distributed, parallel file system (MatrixFS™) that was written entirely from scratch; it does not rely on open source components or legacy algorithms. The software also includes integrated tiering that seamlessly migrates data to and from the cloud (public, private, or hybrid) without special software or complex scripts; all data resides in a single namespace for easy management by our Trinity™ management console. The intuitive graphical user interface allows a single administrator without any specialized storage training to quickly and easily manage hundreds of petabytes.

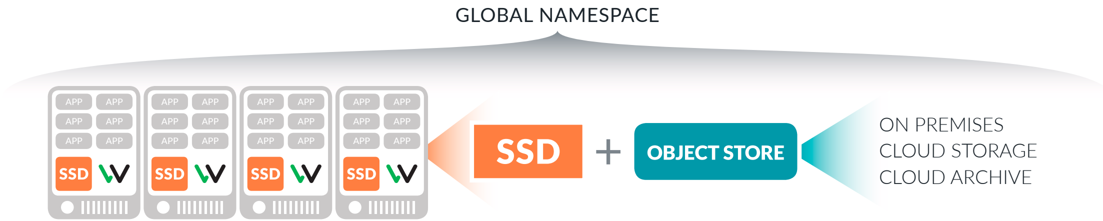


Figure 3 – Matrix Capacity combines fast SSD plus cloud object storage and global namespace

WekaIO's unique architecture (figure 3) is radically different than legacy storage systems, appliances, and hypervisor based software-defined storage solutions because it allows parallel file access via POSIX, SMB, NFS, HDFS, REST and S3 in addition to overcoming traditional storage scaling and file sharing limitations. Additional data services include both local snapshots and remote snapshots to the cloud, cloning, automated cloud tiering, and dynamic cluster rebalancing.

WekaIO Matrix Elastic Storage System

Matrix software provides unmatched multi-dimensional functionality (figure 5):

- Dynamically tunable performance that adapts to your workloads without compromise
- Scalable capacity to start small and fluidly grow as your needs change
- Strong resilience and data protection keeps your data safe and available
- Adaptable and flexible to adjust to your environment and changing application needs
- Easy to use cloud-like management for rapid deployment and simple provisioning
- Cloud economics by leveraging object storage to reduce costs at scale

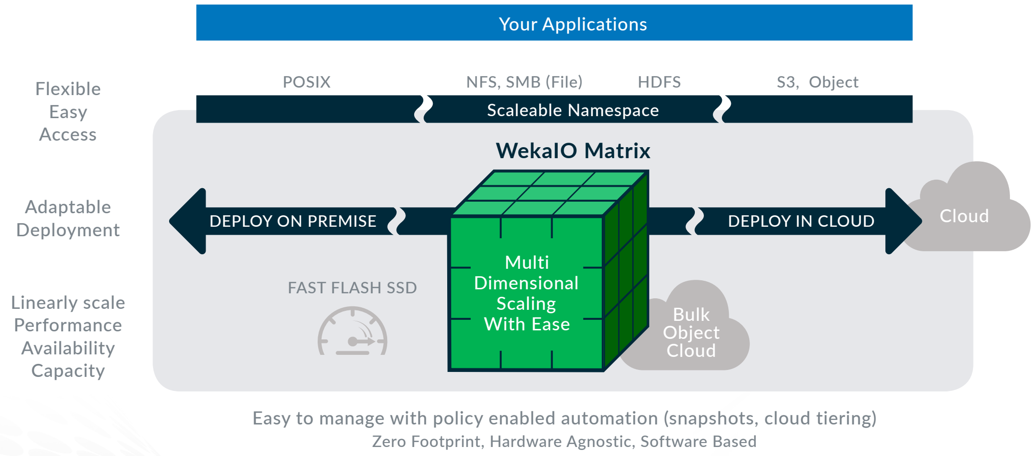


Figure 4 – WekaIO Matrix multi-dimensional functionality

WekaIO Matrix Architecture

WekaIO's Matrix cloud-scale file system is designed to provide a cloud-like experience whether you run your applications on-premises or plan to move them to the cloud. Matrix provides a seamless transition to the cloud and back.

Traditional storage systems layer file management software on top of block storage. MatrixFS is a distributed, parallel file system that eliminates the traditional block volume layer managing underlying storage resources. This vertically integrated architecture removes the limitations of other shareable storage solutions and delivers both capacity efficiency (e.g. space savings) and performance effectiveness (productivity). An important benefit is that MatrixFS can manage sparse data on a file basis vs. what traditional systems accomplish with volume based thin provisioning (e.g. storage efficiency).

Figure 5 shows the software architecture including flexible application access (top right). Matrix core components, including the MatrixFS unified namespace and other functions execute in user space, effectively eliminating time sharing and other kernel specific dependencies. The notable exception is the Weka VFS kernel driver, which provides the POSIX filesystem interface to applications. Using the kernel driver provides significantly higher performance than can be achieved using an NFS or SMB mount point, and it allows applications that require full POSIX compatibility to run on an NFS based solution.

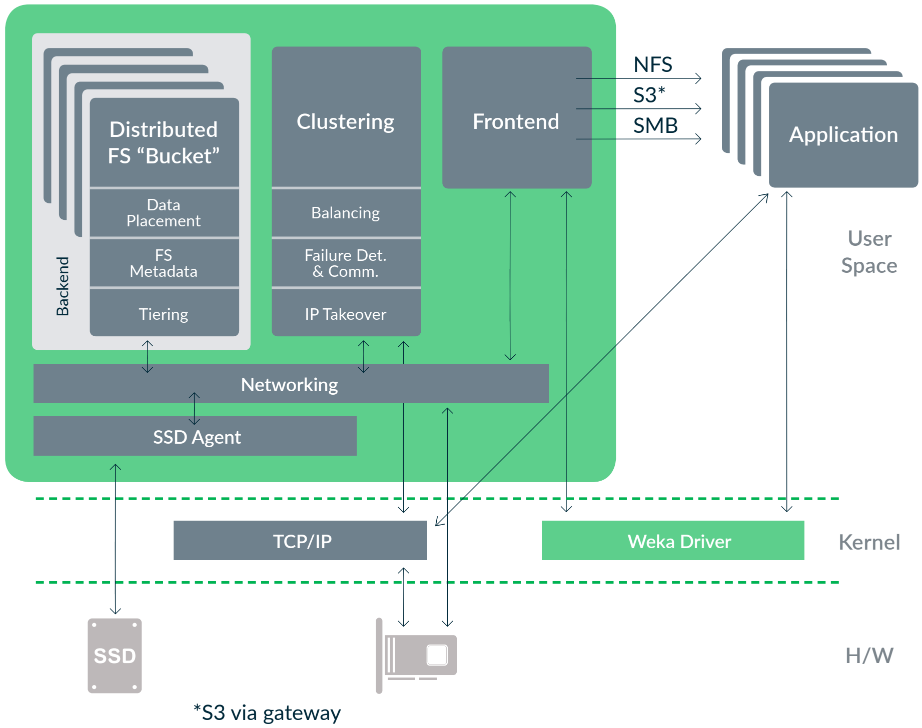


Figure 5 – Matrix software-based storage architecture

The Linux kernel was not optimized to run storage services. Many storage appliances deploy their solution over the Berkeley Software Distribution (BSD) operating system. This is not possible in an open architecture, shared environment because Linux is the most prevalent operating system being used.

Matrix supports all major Linux distributions and leverages virtualization and low level Linux container techniques to run its own RTOS (real-time operating system) in user space, alongside the original Linux kernel. Matrix manages its assigned resources (CPU cores, memory regions, network interface card, and SSDs) to provide process scheduling, memory management, and to control the IO and networking stacks. By not relying on the Linux kernel, Matrix effectively utilizes a zero-copy architecture with much more predictable latencies.

Matrix functionality running in its RTOS (figure 6) is comprised of the following software components:

- File Services (Front-end) – manages multi-protocol connectivity
- File System Clustering (Back-end) – manages data distribution, data protection and file system
- SSD Access Agent – transforms the SSD into an efficient networked device
- Management Node – manages events, CLI, statistics, and call-home
- Object Connector – read and write to the object store

Matrix core software runs inside an LXC (Linux) container which has the benefit of improved isolation from other server processes. A WekaIO VFS driver enables Matrix to support POSIX semantics leveraging lockless queues for I/O to achieve the best performance while enhancing interoperability. The Matrix POSIX file systems has the same runtime semantics of a local Linux file system (e.g. ext4, xfs and others), enabling applications that previously were not able to run on NFS shared storage due to POSIX locking requirements, mmap files, or other reasons.

Bypassing the kernel means that Matrix's I/O software stack is not only faster with lower latency, but also it is portable across different bare-metal, VM, containerized, and cloud instance environments.

Resource consumption is often a problem with traditional software based storage designs because these solutions either take over the entire server or share common resources with applications. This extra software overhead introduces latency and steals precious CPU cycles. By comparison, Matrix has a very small resource footprint, typically about five percent, leaving ninety five percent for application processing. Matrix only uses the resources that are allocated to it, from as little as one server core and a small amount of RAM to consuming all of the resources of the server. Administrators have complete control.

Trinity Enabling Easy Management of WekaIO Matrix

Continuing on the theme of radically simple storage, WekaIO provides two quick and easy ways to manage Matrix. Reporting, visualization, and overall system management functions are accessible using the command line interface (CLI) or the intuitive Trinity management console (figure 6). CLI functionality is also available via an easy-to-use API, allowing integration with current management stacks. Trinity's graphical user interface (GUI) has been designed so that administrators can quickly and easily manage a tiered, multi-petabyte storage system without specialized storage skills.

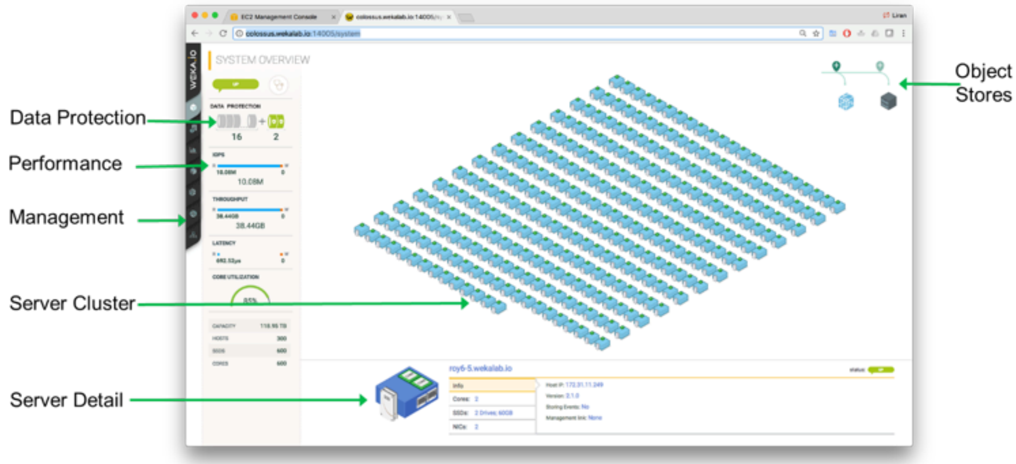


Figure 6 – Trinity management software user interface

Figure 7 – Time-series charts for event monitoring

A System Events tab lists the events that have occurred in a Matrix environment. The events displayed in this window are also transmitted to the WekaIO Support Cloud, so that they can be used by WekaIO support to actively assist you when necessary or to proactively notify you when action needs to be taken. Trinity is entirely cloud based, eliminating the need to physically install and maintain any software or dedicated hardware resources, and you always have access to the latest management console features.

Performance

Until now, IT organizations could only achieve their performance goals by running file based applications on a local file system provisioned over an AFA volume. Shared access had to be provided via NFS. Matrix delivers important performance benefits over this approach. Our POSIX compliant filesystem runs more efficiently than NFS and leverages an innovative, customized protocol to deliver file based semantics. As a result, Matrix allows you to meet your performance needs in a sharable filesystem solution through our revolutionary metadata handling. This, coupled with extremely low latency 4k IO to the application, make the need to use complex AFA volumes for a local file system obsolete.

In addition, our distributed data protection is highly efficient, so it greatly reduces write amplification and the resulting latency. Matrix provides three times more write throughtput than a triple replicated solution and can saturate any network link given enough SSD throughput. These performance innovations have a significant impact in use cases such as Dev/Test, EDA, Genomics, and machine learning where Matrix can serve as a persistent global cache to accelerate small I/O and analytics workloads. See the Proof Point section for an example use case.

Data Layout

MatrixFS manages all data within the storage system as part of a global namespace and supports two performance tiers – SSD for active data and object storage for inactive data. Tiering is an optional addition to the global namespace and can be configured at any time after installation in a few clicks from the Trinity console. The authoritative copy of a file resides on SSD while it is active or until it is tiered off to object storage based on preset or user defined policies. The original file is not deleted from SSD until the physical space is required, and hence acts as a cache until overwritten. As file system capacity runs low and reaches a predefined, user configurable high watermark, data is automatically tiered off to the object storage, which means you never have to worry about the file system running out of capacity.

The global namespace can be sub-divided into thousands of file systems, each with its own capacity, quality of service (QoS), and data protection level settings. Individual file system capacity can be expanded at any time simply by allocating more space to it. By segmenting the namespace, storage capacity can be allocated to individual users, projects, customers, or any other parameter, yet be easily and centrally managed. Data within a file system is fully isolated from every other file system to prevent noisy neighbor issues.

With Matrix, there is no sense of data locality, which improves performance and resiliency. Contrary to popular belief, data locality actually contributes to performance and reliability issues by creating data hot spots and system scalability issues. By directly managing data placement on the SSD layer, MatrixFS can shard the data and distribute it for optimal placement based on user configurable stripe sizes. Sharded data perfectly matches the block sizes used by the underlying flash memory to improve performance and extend SSD service life. Stripe sizes can be set to any value from 4 to 16 and can be changed at any time without impacting system performance. Figure 9 illustrates data placement across SSDs in a 4 + 2 configuration. It is important to note that while the illustration below shows stripes across each of the six failure domains in common, as Matrix scales these stripes are spread across more failure domains, resulting in fewer stripes with a common failure domain.

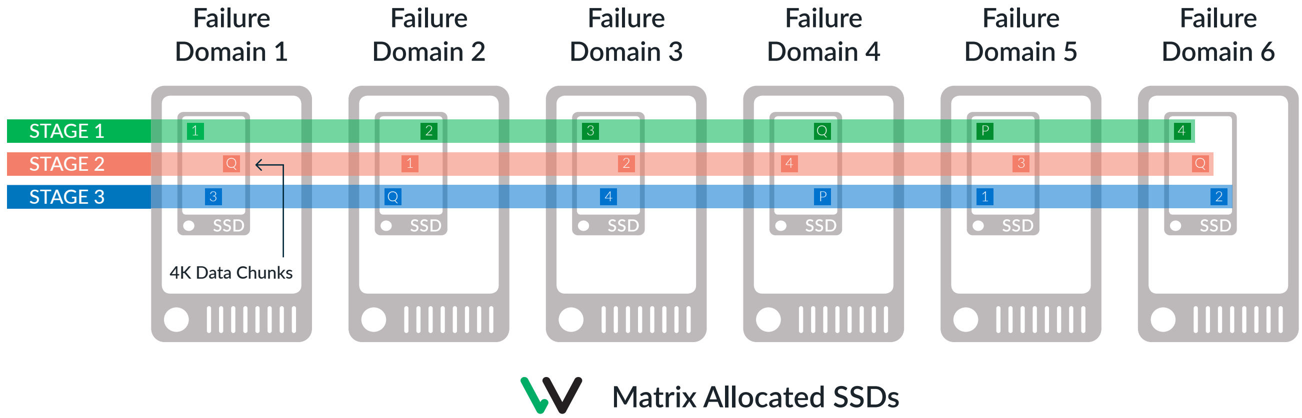


Figure 8 – WekaIO Matrix patented data distribution

MatrixDDP Fast and Efficient Data Protection

Data protection is a critical function of any storage system, especially at scale. Without an appropriate protection schema, file systems would need to be limited in size to accommodate the rebuild time window and minimize the data exposure risk. Popular data protection schemes such as RAID, Replication, and Erasure Coding are a compromise between scalability, protection, capacity, and performance.

Matrix uses failure domains to define data protection levels. Failure domains are fully configurable starting at the server node level, which provides single or multiple SSD level granularity. Data protection levels are flexible depending on the size and scale of the server cluster—the larger the cluster, the larger the recommended data stripe size for best utilization of SSD capacity and improved resiliency. For granular protection, the data protection level is set at the file system level and can be set to two or four, meaning that the system can survive two or four simultaneous drive or node failures without impacting data availability.

Our data protection scheme follows the convention of data (N) + parity (2 or 4), N + 4 data protection level is unique in cloud storage and delivers exponentially better resiliency than triple replication – which only protects to 2 failures - without the expensive storage and throughput impact. N + 2 level protection is ideal for hyperconverged clusters or clusters with a large number of nodes because it is the most space efficient while providing protection comparable to triple replication.

WekaIO Matrix manages protection so data is always safe and accessible:

- Configurable data protection levels
- Patented distributed data protection schema
- Configurable failure domains
- End-to-end data protection checksums
- Meta data journaling
- Network redundancy support for two top-of-the-rack switches over two network ports
- Local or cloud based snapshots and clones
- Automated tiering to the cloud

WekaIO Matrix Patented Data Distribution

WekaIO has invented a patented and patent pending distributed data protection (e.g. MatrixDDP) coding scheme that improves with scale. MatrixDDP delivers the scalability and durability of Erasure Coding but without the performance penalty. Unlike legacy hardware and software RAID and other data protection schemes, MatrixDDP actually gets faster and more resilient as you scale.

MatrixDDP, as its name implies, distributes data and metadata across logical nodes that span failure domains (FD). These failure domains (figure 9) include host server nodes and their SSD racks, as well as data centers, known in cloud parlance as availability zones (AZ).

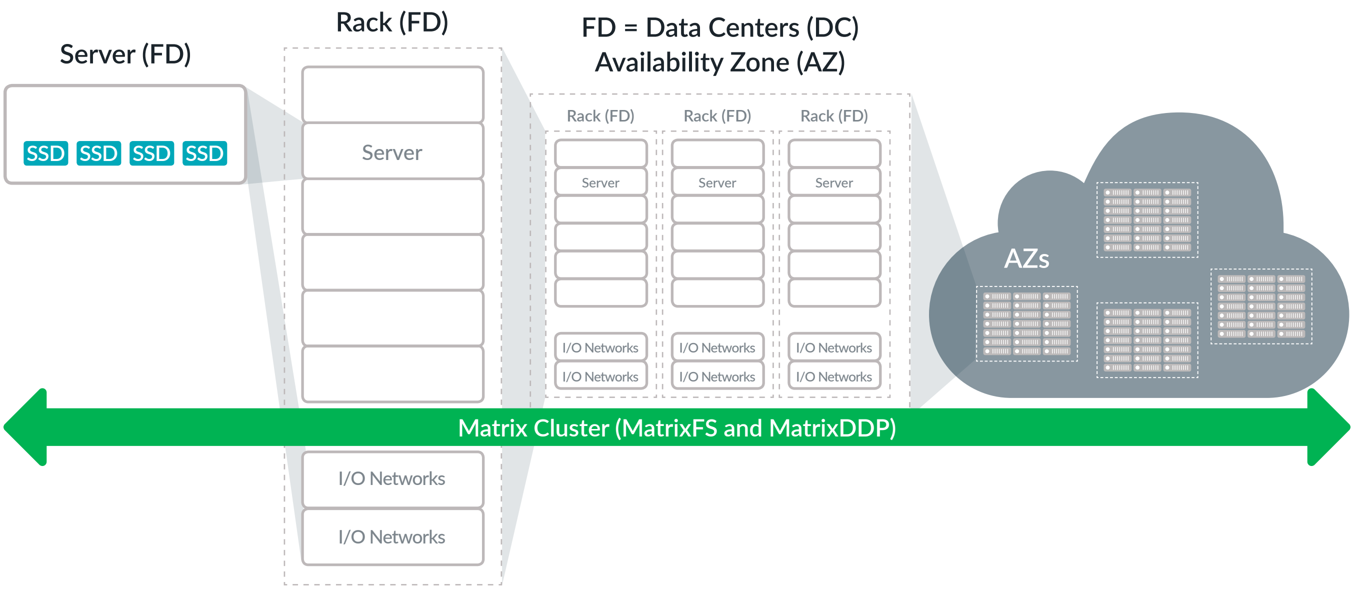


Figure 9 – MatrixDDP Failure Domains

Unlike traditional hardware and software data protection schemes, Matrix never places a single data stripe inside the same server (or FD), so in the event of a server failure there can never be the possibility of a double fault. Even if the node has multiple SSDs, data is spread across different server nodes, racks, or AZs depending on the resiliency chosen. Matrix resiliency is user-configurable to define the number of faults within a Matrix cluster to meet your application workload service level requirements. When a failure occurs, the system considers the FD a single failure, regardless of how large the domain is defined. By defining the FD as a data center, full data center resiliency is achieved for metro-cluster deployments. In addition to distributing stripes at the FD level, Matrix also ensures a highly randomized data placement for improved performance and resiliency.

MatrixDDP Fast Rebuilds (reduces exposure to subsequent faults)

MatrixDDP uses several innovative strategies to return the system to a fully protected state as quickly as possible and be ready to handle a subsequent failure. This ensures that applications are not impacted by long rebuild processes. Because Matrix protects data at the file level, it only needs to rebuild the data that is actively stored on the failed server or SSD. This means that the rebuild times are faster compared to a traditional RAID solution, including file servers that protect data at the block layer. RAID controller based systems typically rebuild all blocks on an affected device, including empty blocks, prolonging rebuilds and exposure. Furthermore, Matrix does not need to rebuild data on the SSD tier that has already been tiered off to an object store because the data is already protected by the object storage device.

[1] United States Patent 9448887 – http://www.freepatentsonline.com/9448887.html to learn more

All nodes in the Matrix cluster participate in the recovery process. This means that the larger the cluster size, the faster the rebuild and the more reliable the system becomes. By contrast, in a replicated system only the mirrored servers participate in the recovery process, impacting application performance significantly. Erasure Coding suffers from a similar problem where only a small subset of the servers participate in the recovery. With Matrix, the recovery rate is user configurable and can be changed at any time, so administrators have complete control to determine the best tradeoff between performance during recovery and recovery time.

Table 1 shows that a 100 node cluster with 30TB capacity per node (3PB total capacity) can be returned from 4 failures to 3 failure resiliency in under 1 minute, while a return to full redundancy would take 2 hours and 25 minutes, assuming the entire 120TB must be rebuilt.

Stripe Size(data+parity)	16+2	16+2	16+4
Number of Nodes	50	100	100
Capacity per Node	10TB	10TB	30TB
Rebuild Bandwidth (GB/Second)	0.5	0.5	1.0
Time to Full Protection (h:m:s)	2:54:34	1:26:22	2:25:50
Time to 1st failure resiliency (h:m:s)	1:00:37	0:14:52	0:28:00
Time to 2nd failure resiliency (h:m:s)			0:05:11
Time to 3rd failure resiliency (h:m:s)			0:00:55

Table 1 - Sample Rebuild Time to Return to Protected Level

Matrix Snapshots and Clones

Matrix supports user-definable snapshots for routine data protection including backup, as well as for cloud data migration and bursting. For example, Matrix snapshots can be used for backing up files locally, as well as making copies to cloud storage tiers for backup or disaster recovery. Also, Matrix snapshots can be utilized for saving (parking) files and data to lower cost cold storage such as cloud and object storage tiers. In addition to point in time snapshots, Matrix can create full clones, (snapshots that can be converted into writable snapshots), with pointers back to the originating data. Matrix snapshots and clones occur instantaneously and are differential after the first instance, which dramatically reduces the time and storage required for protection. Furthermore, overall system performance is unaffected by the snapshot process or when writing to a clone.

Matrix Powerfail and End-to-End Data Protection

Using a checksum process to ensure data consistency, Matrix provides end-to-end data protection of both reads and writes. Matrix always stores data and checksum information separate from each other for improved protection. Matrix provides additional data integrity capabilities by protecting against data loss due to power failures. When a write is acknowledged back to the application, it is safely protected from server failures or data center wide power failure through a journaling process. Matrix innovative data layout and algorithms enable it to recover from a data-center-wide power failure in minutes because there is no need to do a complete file system consistency check (FSCK). For most other file systems, the FSCK process recovery time is proportional to the size of the recovered file system. In large scale deployments, this recovery can take days.

Automated Data Rebalancing

MatrixDDP proactively monitors and manages the performance, availability, and capacity health status of a Matrix cluster. This allows the system to calculate the utilization levels (performance and capacity) of nodes to automatically and transparently redistribute data across the cluster to prevent hot spots.

The benefit is that Matrix can maintain well balanced cluster performance and data protection as capacity and usage change. Another advantage is that as additional SSDs are added to existing server nodes or the cluster is expanded with more nodes, Matrix automatically rebalances to enhance performance, resiliency, and capacity without scheduling costly downtime. Matched capacity SSDs are not required, which allows you to leverage new technology and save money as SSD prices decline.

Automated Policy Based Tiering

Built into Matrix is a policy based automated tiering feature that transparently migrates data across tiers. Matrix supports moving data from fast SSD storage tier to lower cost, higher capacity on-premise or cloud-based object storage (Figure 10). To improve performance, you may want to keep certain files local on SSD. In this case, Matrix allows you to pin the file to the SSD tier, preventing the designated file(s) from being migrated to the object store. Metadata is always kept on SSD, and a snapshot of the metadata can be stored on the object storage tier. The presented namespace shows all the files regardless of their placement, thus no change in application is needed in order to leverage cost optimized object storage based solutions.

This integrated capability eliminates the need for additional Hierarchy Storage Management (HSM) or tiering software that adds complexity and cost. The byproduct of the integrated tiering is an elastic unified namespace that can scale infinitely (or to the size of your cloud provider) to meet your needs.

Maximum tiering granularity is in 1MB ranges (chunks) of data, rather than complete files. Smaller files would be divided into respectively smaller chunks. The advantage of working with data chunks is that only the data that is required has to be retrieved from the object store, which significantly reduces data network traffic. Each chunk of data may reside on SSD only, object storage only, or in both locations. Storing chunks in both locations effectively serves as a cache for tiered data. This provides not only support for large files that are not uniformly accessed, but also concurrent and parallel access to large files.

Matrix can tier data to up to eight different object storage targets, allowing you complete flexibility and providing you with economical back-up and disaster recovery options.

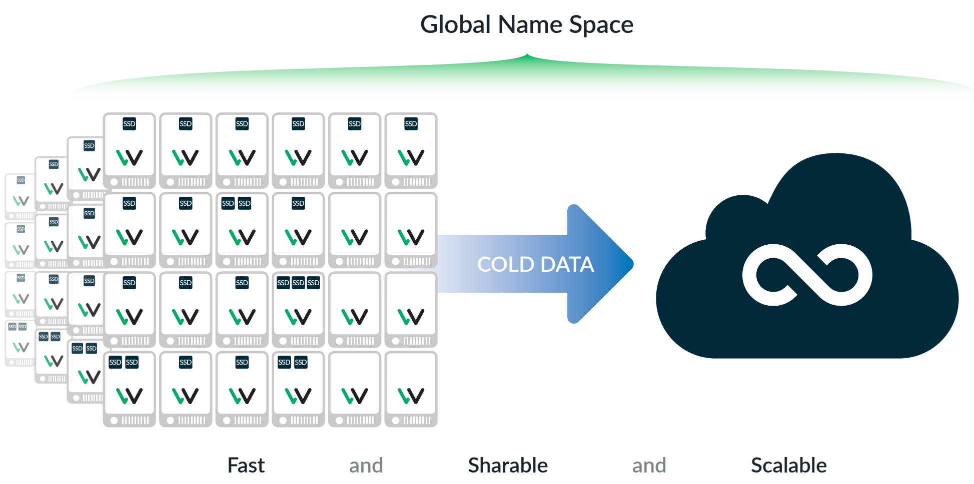


Figure 10 – Integrated tiering to any S3 or Swift compatible object store

Flexible deployment options and configuration (on premise and cloud)

Matrix provides industry leading flexibility by supporting an extremely broad array of operating environments and deployment models. It runs on standard x86 based servers using standard 10Gb or higher Ethernet network adapters and off-the-shelf SATA, SAS, or NVMe SSDs.

Starting at only six servers, Matrix scales to many thousands of servers. There is no need to match server types, brands, or configurations (CPU and memory), providing investment protection and extending your infrastructure's service life for a dramatically lower total cost of ownership. When infrastructure matures, Matrix enables expanding the cluster with new servers, then retiring older generation servers, thus enabling continuity over hardware refresh generations without the need to migrate the data to different storage.

Broad Operating Environment Support

Whether your applications run on bare-metal for performance, in a virtual environment for ease of deployment and resiliency, or entirely in the public cloud for on-demand scalability and elasticity, Matrix is a single, no compromise, storage solution providing the freedom to choose the environment best suited for your application based on performance, scale, and economics (figure 11).

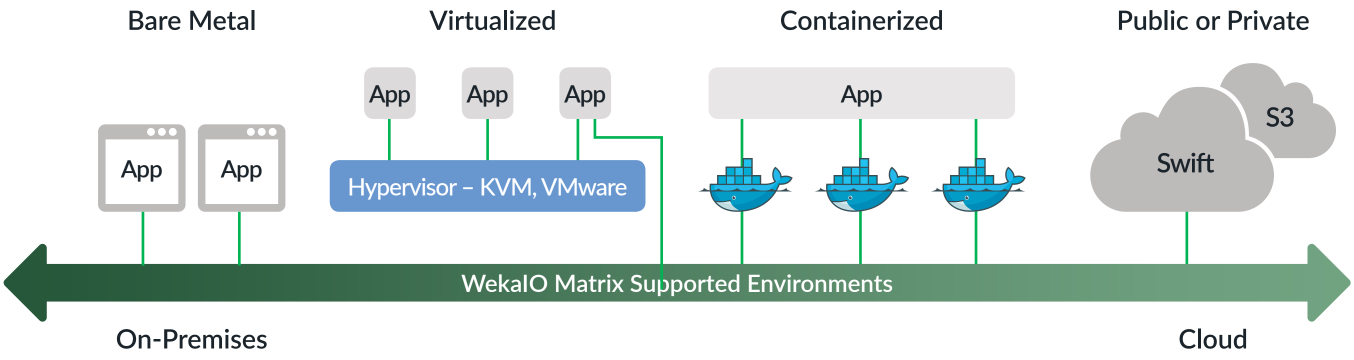


Figure 11 – Industry leading flexibility to support your operating environment

Benefit from extreme performance with the lowest latency access in bare metal environments where applications are co-located with storage on dedicated hardware. Integrated tiering to the cloud allows you to leverage the flexibility of the cloud from your data center.

Reduce total cost of ownership and maximize infrastructure utilization in a virtualized environment. Matrix provides VMs with unlimited shared storage capacity for scalability and ease of management. High throughput and random I/O performance can significantly reduce the impact of VM related boot-storms.

Take advantage of the elastic nature of the cloud and benefit from cloud economics. Matrix provides unmatched performance and scalability in public, private or hybrid cloud environments.

Hyperconverged

In a hyperconverged deployment (figure 12), Matrix is integrated into standard Intel-based application servers. Combining storage and compute resources with applications and data into a single building block delivers a highly optimized data center solution across a mix of workloads. This deployment model enables you to have a lower or zero external hardware footprint and reduce costs while boosting performance, availability, and capacity.

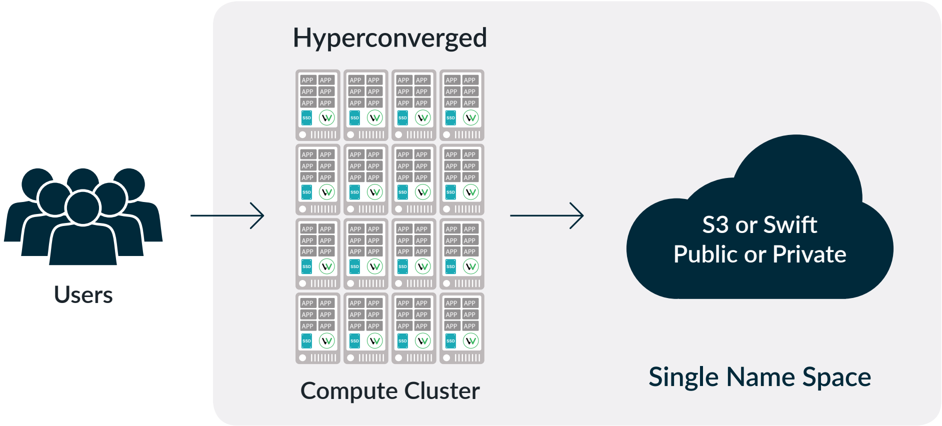


Figure 12 – Hyperconverged mode provides high performance and reduced footprint

Dedicated Storage Server

Matrix can also be configured as a dedicated storage server (figure 13) when all the resources of the system are dedicated to storage services while applications run on a separate compute infrastructure. For additional flexibility, Matrix also supports mixed environments where some servers are dedicated entirely to Matrix while others share resources between Matrix and applications. This allows you to move between deployment models as application workloads change or infrastructure needs dictate. This mode is popular among machine learning use cases where GPU servers only run the Matrix front-end while dedicated hardware runs the storage and back-end services.

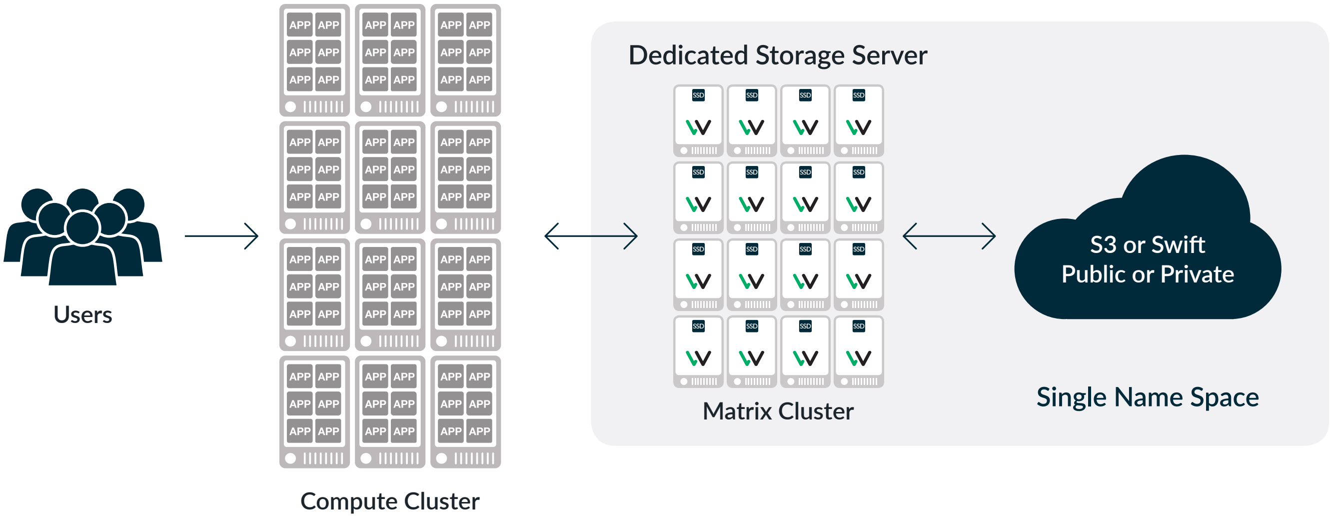


Figure 13 – Dedicated Storage Server with separate compute and storage

Native Public Cloud

Deploy Matrix on Amazon EC2 (virtual cloud servers) just as you would on-premises using EC2 instances with locally attached SSD(s). Matrix supports both hyperconverged and dedicated storage deployment models in the cloud on a selection of EC2 instance types. For compute intensive workloads or to take advantage of GPU based instances, it is recommended to use the dedicated storage server mode. If you need additional low-cost high-capacity storage capacity, leverage Matrix automated tiering to Amazon S3. If you want to burst or migrate data to the cloud, you can leverage Matrix snapshot-to-object functionality for data movement. For improved resiliency, Matrix also supports the spanning of AZs in the public cloud.

Refer to the proof points later in this document to see how Matrix scales performance and availability in a cloud environment.

WekaIO Matrix Zero Footprint

Running Matrix alongside other applications in a hyperconverged fashion using your existing compute infrastructure, internal SSDs, and tiering to the cloud allows you to effectively reduce your storage footprint to zero (figure 14). The result is a reduction in power, cooling, and floorspace costs by up to 80%.

Our software-only solution also eliminates silos of data and the complexity of multiple tiers of physical storage and specialty networks to radically simplify your storage and regain valuable data center space. Taken to the extreme, you can run existing applications without modification on Matrix entirely in the cloud for a zero data center footprint.

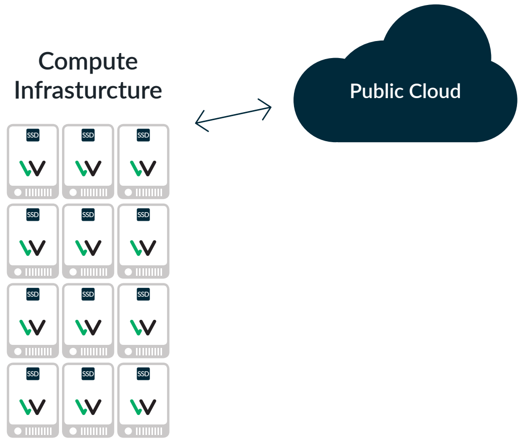


Figure 14 – Zero-footprint deployment

WekaIO Matrix Proof Points

Matrix easily adapts to scaling challenges and varying workload. Three proof point performance scenarios are shown below:

- Semiconductor software build of Android
- Genomic research file conversion
- Linear performance scaling from 6 to 240 nodes (IOPS, bandwidth, and latency)

Real-world Application: Semiconductor Software Build Time

Semiconductor design is extremely challenging and infrastructure intensive. Designers need to collaborate with other team members and access the same files simultaneously. Front-end design is typically I/O and metadata intensive with many small files and random access patterns. Back-end design is the exact opposite; it typically has very large files and sequential access patterns. As much as 60% of the design cycle is spent in verification and simulation, so any time saved during the design phase results in significantly faster chip tape-out.

This real-world application by a large semiconductor company involved running software builds of the Android operating system. To accomplish this quickly, files are usually copied to multiple external NAS systems to be run in parallel. Moving data wastes precious time, consumes valuable resources, and results in unnecessary infrastructure investment. The results of this test show that Matrix, as a distributed architecture with innovative metadata handling and very low IO latencies, is nearly as fast as running the build on a local file system with SSDs, completing the task in 38 minutes. By contrast, running the same build over battery backed DRAM based NAS NFS (common practice) took 3.8 hours (figure 15).

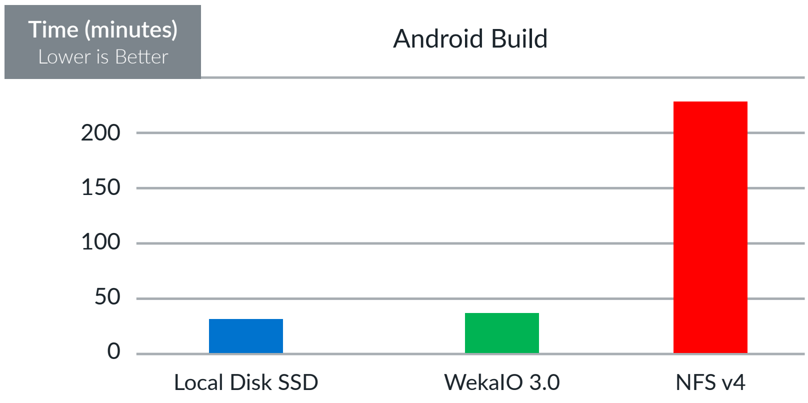


Figure 15 – Software Build on Local Disk, WekaIO and External ZFS based NAS system

Real-world Application: Illumina BCL2FASTQ File Conversion

Genomic research workloads are characterized by millions of small files and hundreds of very large files. Storage systems must be able to handle both of these workloads equally well. In clinical settings, lives depend on rapid sequencing analysis. This real-world application test was conducted by a leading genomics and cancer research center. The test involved converting BCL files from an Illumina sequencer to FASTQ files for use in the organization's bioinformatics pipeline. This particular step in the pipeline is a very I/O intensive operation.

Matrix was run on a six node cluster; the benchmark system was an all-flash Isilon. Matrix completed a single file conversion in approximately the same time (5 minutes longer). However, when six file conversion jobs were run, Matrix completed all six conversions in exactly the same amount of time as it took for a single run (figure 16). The benchmark system took over twice as long as Matrix to complete all six conversions.

In real world situations, research organizations generally run many of these jobs simultaneously. Matrix handles larger workloads than purpose built systems at a fraction of the cost, providing far greater value.

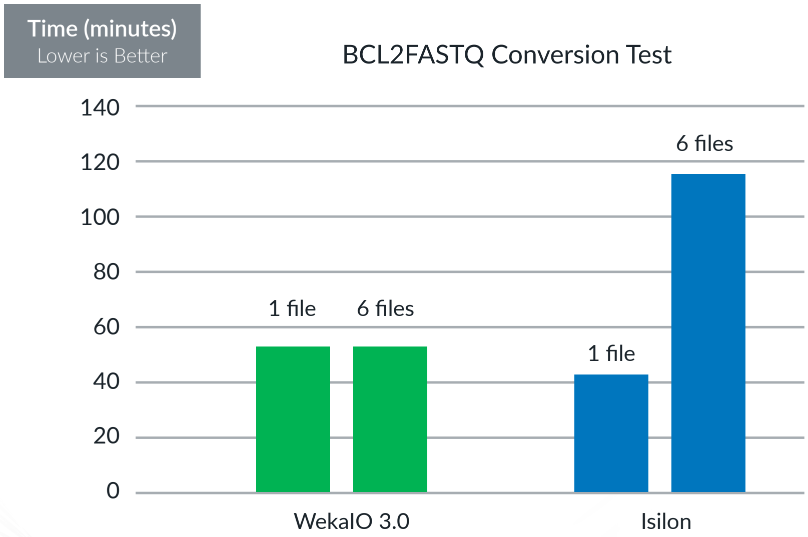
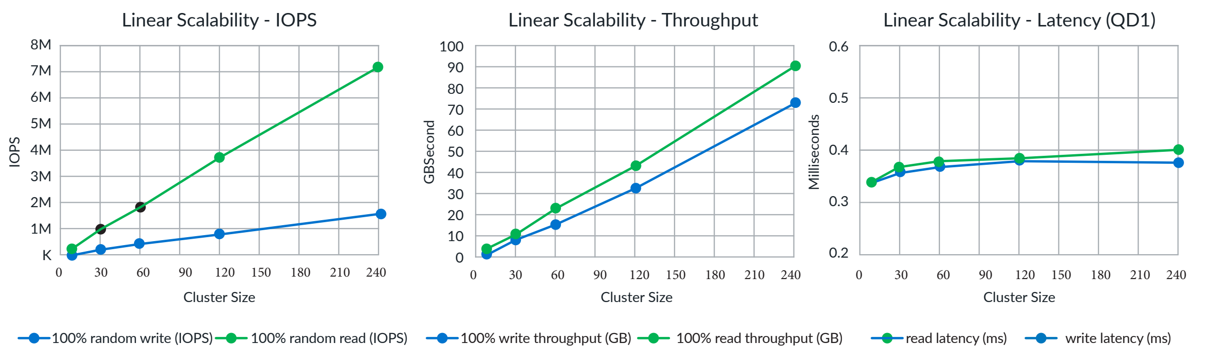


Figure 16 – File conversion comparison to an all-flash Isilon Nitro

Matrix Cluster Scaling

Media rendering benefits greatly from being run at massive scale on a very large compute infrastructure, a scale that is readily available in the public cloud. To get acceptable performance, studios typically use several Avere appliances for NAS acceleration. This approach is extremely expensive and still suffers from scaling limitations. Figure 17 illustrates how Matrix scales easily from 6 to 240 nodes with linear performance. As nodes are added to the cluster, not only is performance enhanced, but capacity and data protection are also enhanced. From left to right, figure 17 shows random IOPS performance, throughput (GBytes/second), and latency (response time).



Test Environment – 30-240 R3.8xlarge cluster, 1AZ, utilizing 2 cores, 2 local SSD drives & 10GB of RAM on each instance. About 5% of CPU/RAM

Figure 17 – Matrix linear and uniform scaling of performance

For this evaluation, Matrix was configured in hyperconverged mode, only 5% of the CPU and memory of each EC2 instance was allocated to Matrix processes while the remaining resources were available for other applications. The graphs show linear performance scaling as cluster size grows from 6 to 240 instances, delivering 7M IOPS and 90 GB/second read performance with sub-400 microsecond latency. Linear, predictable scaling means there is no need to purchase more infrastructure than you need.

Summary

Matrix is a software-based storage solution that adapts to your environment and application needs. It is ideal for the performance and scale needs of EDA, HPC, machine learning, life science, video rendering and special effects, content-streaming, or other large bandwidth and I/O intensive applications that rely on parallel file systems. Matrix reduces the cost and complexity of storage, requiring fewer hardware resources compared to traditional solutions.

WekaIO Matrix addresses common IT storage problems by providing a fast, efficient, and resilient distributed parallel file system that is cloud native and delivers the performance of all flash arrays, the simplicity of file storage and scalability of the cloud. Part of Matrix's ease of use and cloud-like experience includes rapid provisioning to reduce time to get new workloads deployed, along with elasticity scaling, resiliency, performance, and cost effectiveness.

Visit www.weka.io to learn more or ask about our free trial.

910 E Hamilton Avenue, Suite 430, Campbell, CA 95008 USA T 408.335.0085 E info@weka.io www.weka.io