SDC 18
September 24-27, 2018
Santa Clara, CA
www.storagedeveloper.org

NVMe-oF Parallel File System

Achieving 2X Performance Improvement Over Legacy Filesystems and Accelerating HPC Workloads

Liran Zvibel

WEKA.IO
World's Fastest File System

Introduction

WHO WE ARE

WekaIO Matrix is the fastest, most scalable parallel file system for AI and technical compute workloads that ensures your applications never wait for data.

OUR CUSTOMERS

OUR PARTNERS

OUR ACCOLADES

WEKA.IO

Customer Quotes

DREAMWORKS

*With WekaIO as part of our HPE High Performance Compute cluster, **file service scalability and reliability issues are a thing of the past**. We're using the Matrix file system as burst-buffer style transient storage for the **most demanding render and simulation workloads** in our pipeline.*

Scott Miller, Technology Fellow, Engineering and Infrastructure

SDSC SAN DIEGO SUPERCOMPUTER CENTER

*We are using WekaIO technologies over **InfiniBand** to address the challenges of **data analytics at extreme scale** in life sciences, particle physics, geosciences, and other fields. That process is still ongoing but to-date we've already achieved some promising results.*

Michael Norman, Director of San Diego Supercomputer Center at UCSD

tusimple

*WekaIO was the clear choice for our DNN training...**standard NAS would not scale** and Matrix [was] the **most performant of all the parallel file systems** we evaluated...we really liked that it was hardware-independent allowing us better **control over our infrastructure costs**.*

Dr. Xiaodi Hou, Co-founder and CTO

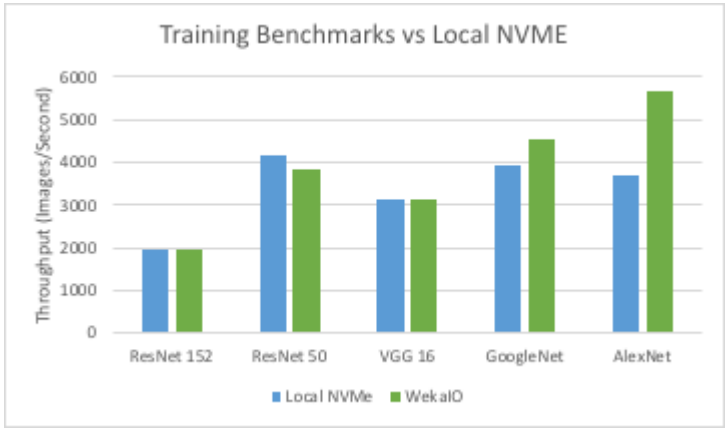
TRE ALTAMIRA
A CLS Group Company

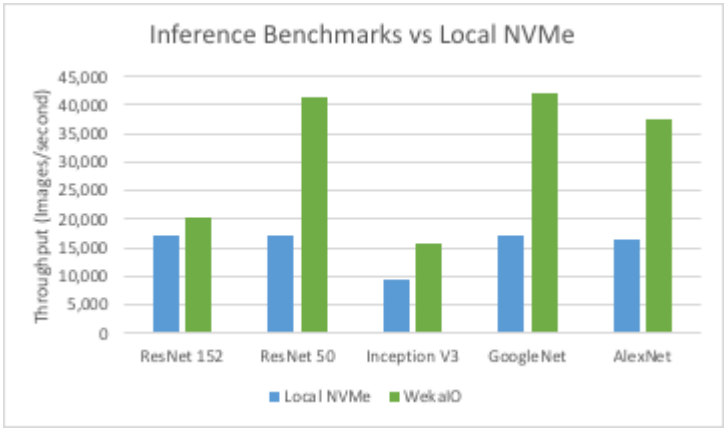
*We are using WekaIO shared file system instead of Lustre®in AWS for its **stability** and their **stellar support** for our geospatial workflows*

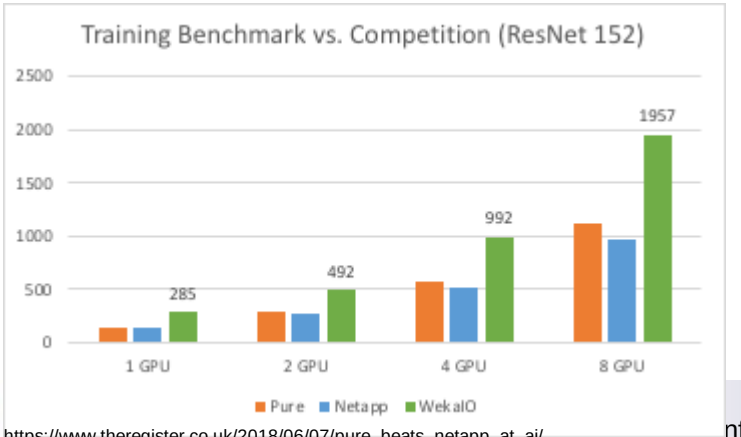
Alessandro Menegaz, Cloud IT Manager

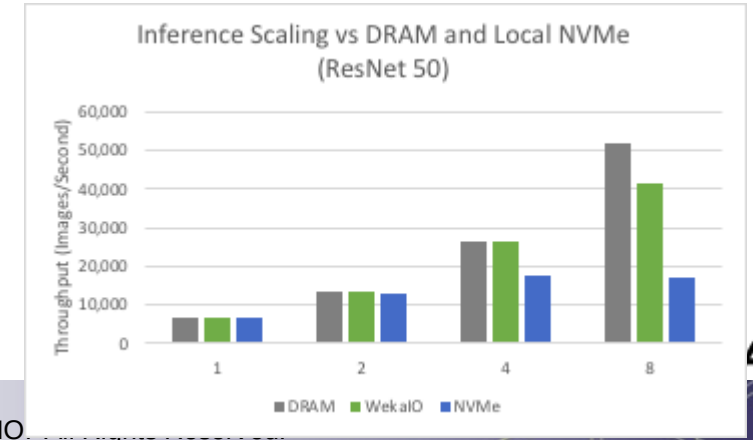
WEKA.IO

GPU Performance vs. Alternatives





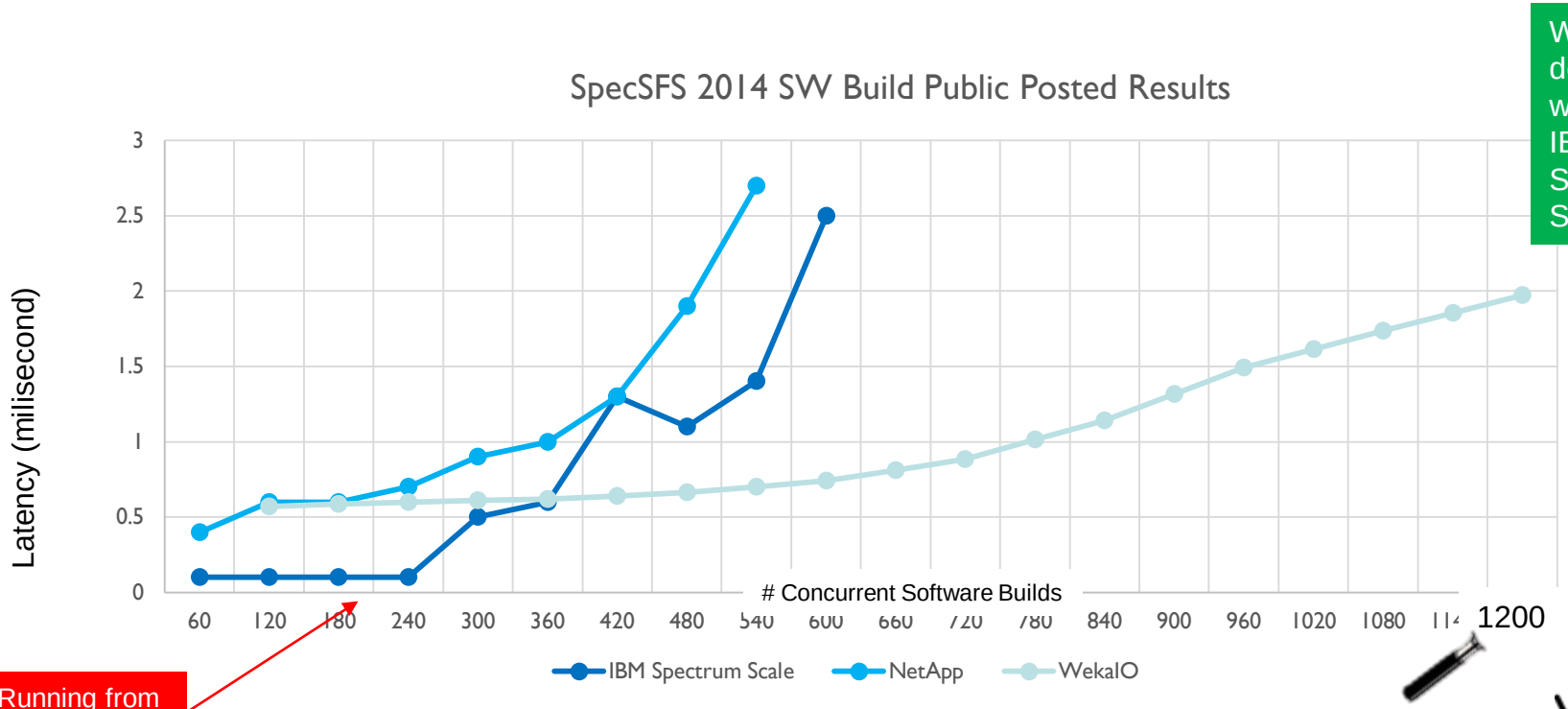




https://www.theregister.co.uk/2018/06/07/pure_beats_netapp_at_ai/

Measured Results Compared to All Flash Filer

Fastest NAND FLASH File System

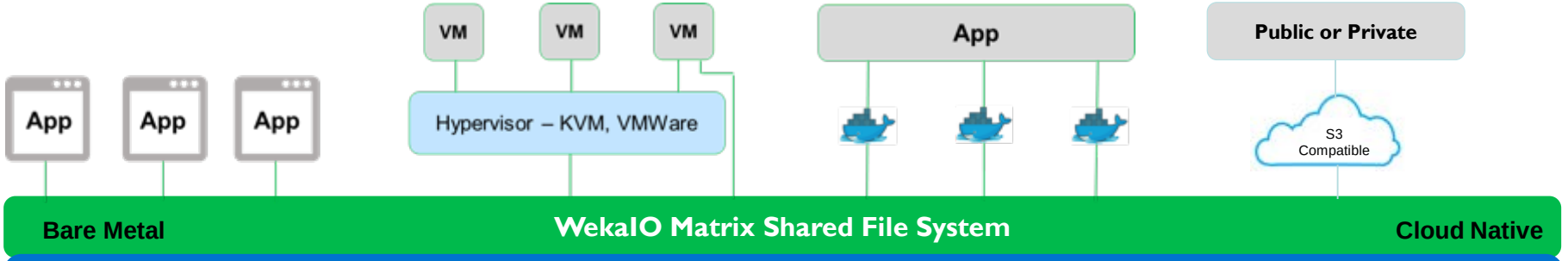


WekaIO does 2x the workload of IBM Spectrum Scale

Running from RAM cache

WEKA.IO

WekaIO Matrix: Full-featured and Flexible



Fully Coherent POSIX File System that is Faster than a Local FS

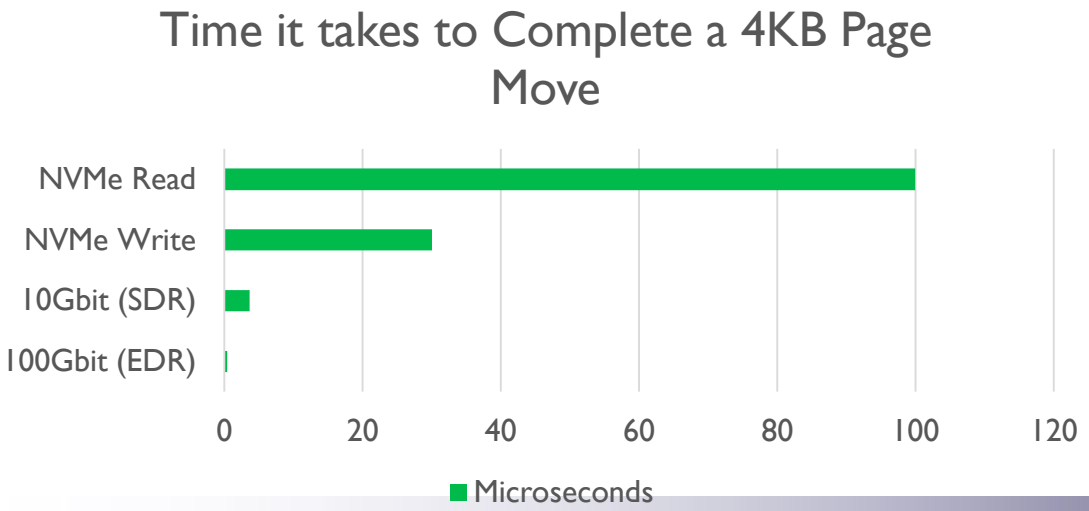
Distributed Coding, Scalable Metadata, Fast Tiering to S3, End-to-End DP

Instantaneous Snaps, Clones, Cloud Bursting, DR, Backup

InfiniBand or Ethernet, Dedicated or (Hyper)Converged Storage Server

Caching is Futile

- Local FS caching happening forever
- Modern networks on 100Gbit are 100x faster than SSD
- It is much easier to create distributed algorithms when locality is not important
- With right networking stack, shared storage is faster than local storage
- NFS has not adopted to fast networking, so cannot keep up

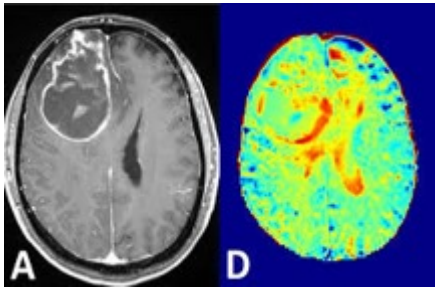


Focused On the Most Demanding Workloads

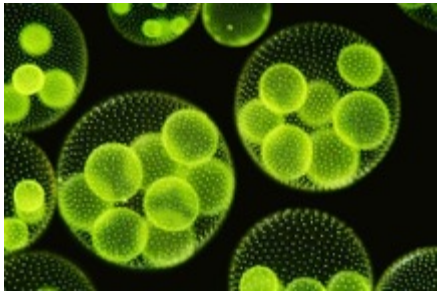
- Machine Learning\ AI
- AV Training systems
- Government image recognition

- Algorithmic trading
- Time series analytics (KDB+) ; ELK
- Risk analysis (Monte Carlo simulation)
- FusionIO\Local FS replacement

- ✓ Millions of small files
- ✓ Metadata intensive
- ✓ Latency sensitive
- ✓ Huge capacity
- ✓ Huge data growth
- ✓ Local FS caching today



- Digital Radiology/Pathology
- Medical Imaging ML



- Genomics sequencing and analytics
- Drug discovery

WEKA.IO

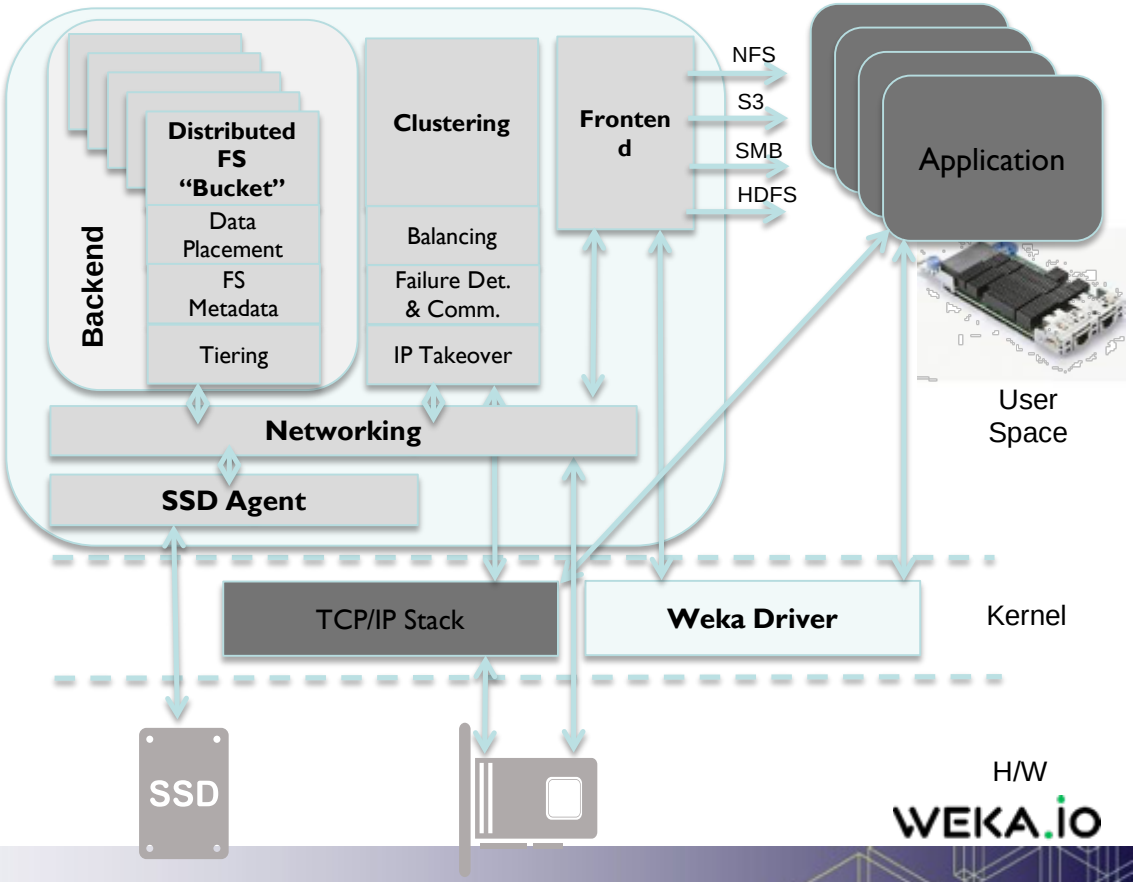
Shared File system Myths that we break

- Local file system is faster than shared file system
  - WekaIO is already faster for I\O
  - Exclusive mount option bridges the gap for local file system use cases (ELK, compilation, untar, etc)
  - Integration with leases will make it fully coherent by EOY
- File systems don’t scale in capacity
  - We can have 100s of PB of NVMe tier, EBs in obj. storage capacity
- File systems don’t scale in metadata, obj store must be used
  - Billions of files per directory
  - Trillions of files per namespace
- You need to have a PhD to operate a Parallel FS

WEKA.IO

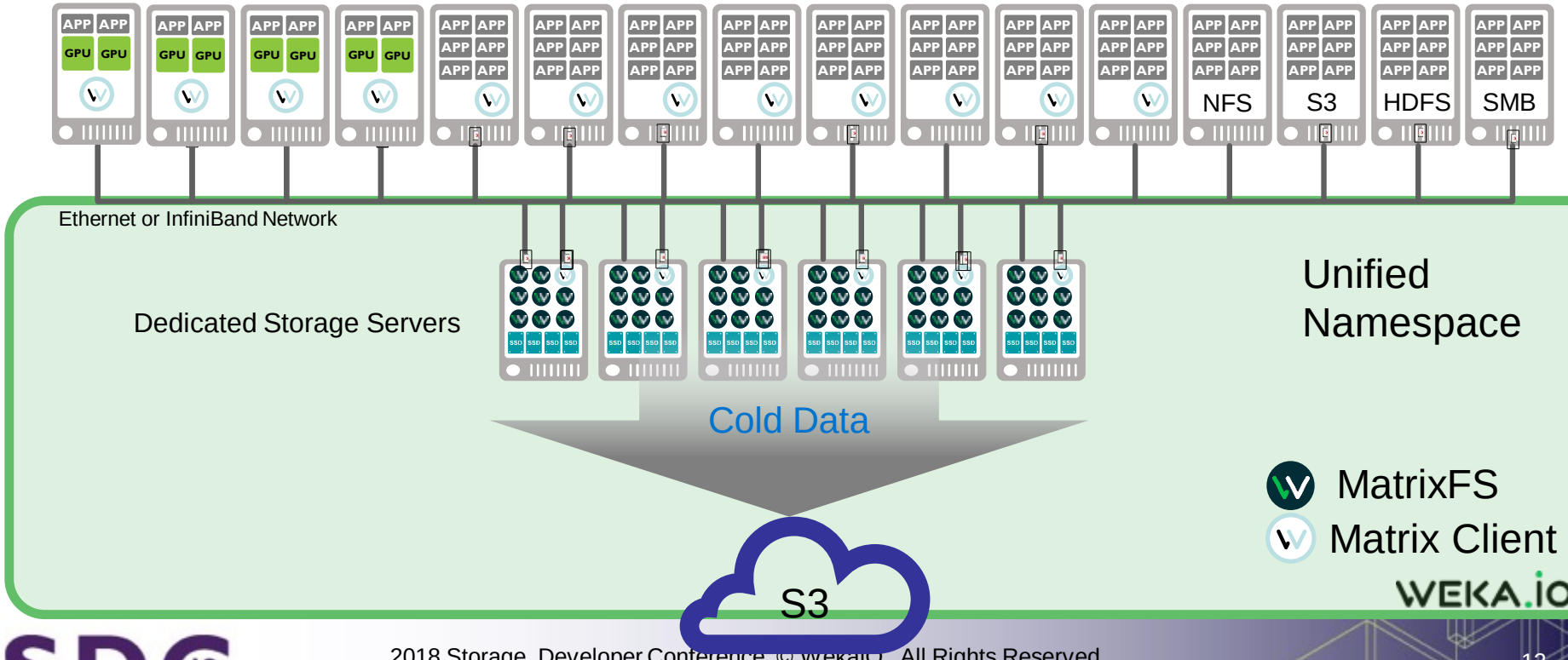
Software Architecture

- Runs inside LXC container for isolation
- SR-IOV to run network stack and NVMe in user space
- Provides POSIX VFS through lockless queues to WekaIO driver
- I/O stack bypasses kernel
- Scheduling and memory management also bypass kernel
- Metadata split into many Buckets – Buckets quickly migrate ➔ no hot spots
- Support, bare metal, container & hypervisor



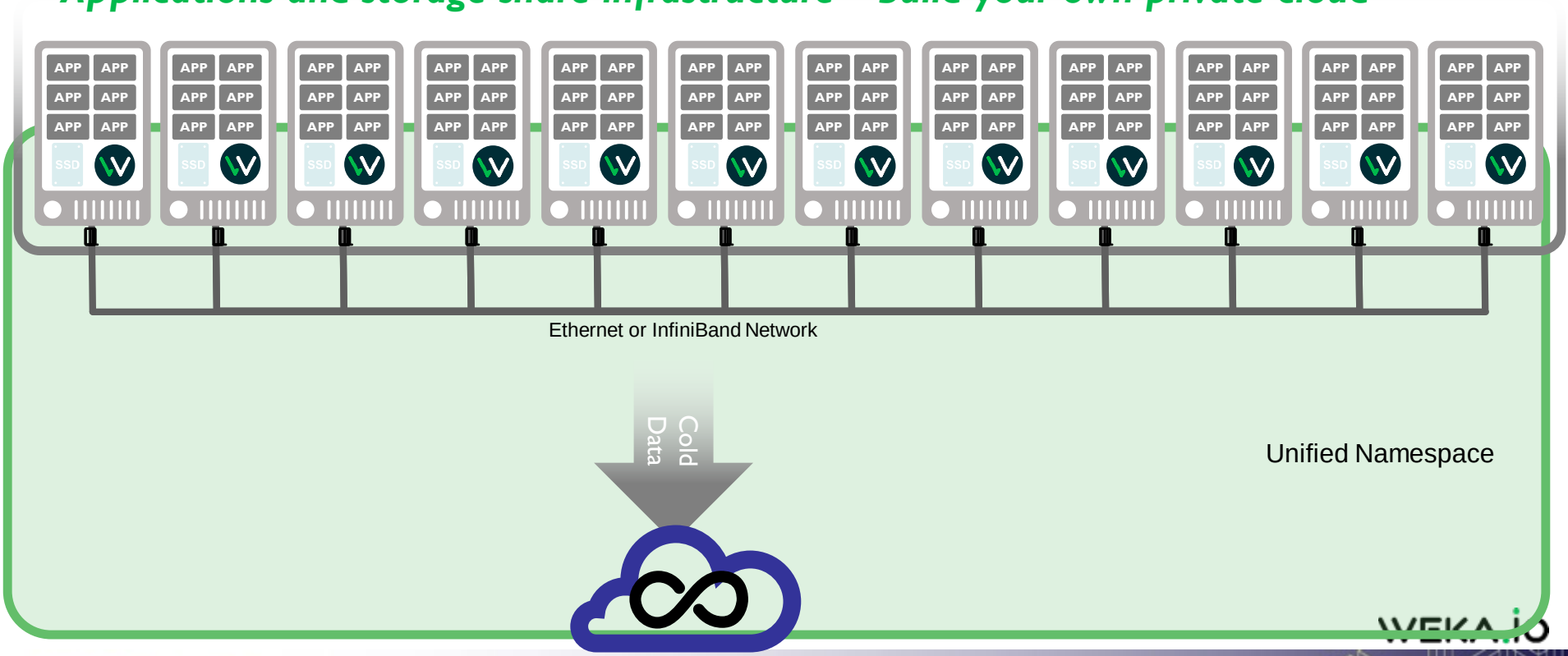
WekaIO Deployed as a Dedicated Server

Storage on separate infrastructure with parallel access to clients



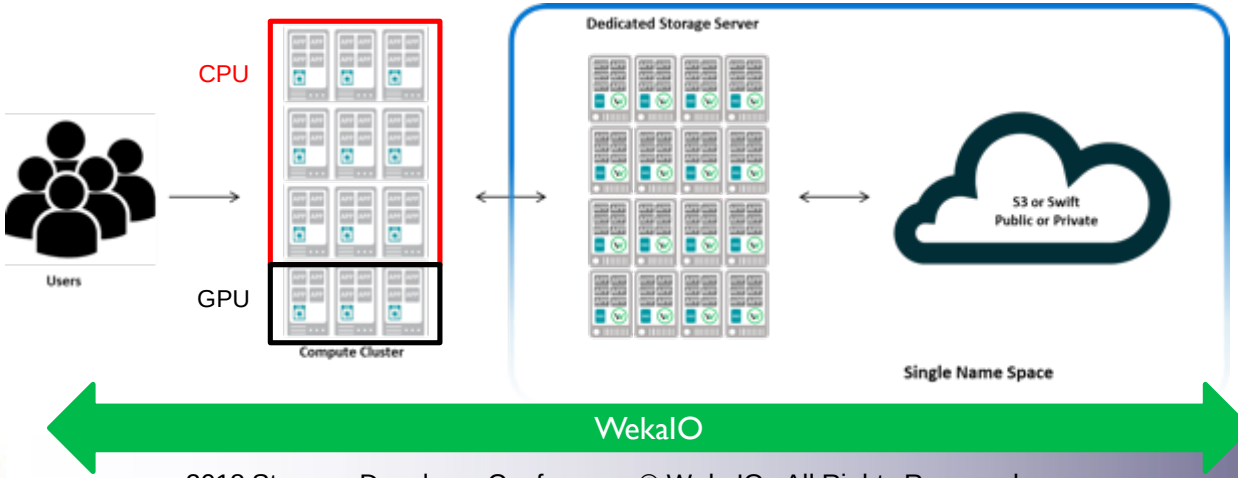
WekaIO Deployed in a (Hyper)Converged Mode

Applications and storage share infrastructure – build your own private cloud



Analytics Requires an Appropriate IT Architecture

INSTRUMENT INGEST	CLEAN TAG	EXPLORE TRAIN VALIDATE	ARCHIVE & CATALOG
S3 SMB NFS	CPU Based HDFS Interface	GPU Based Server Cluster	Growing Data Archive



Fully Distributed Snapshots

- 4K granularity
- Instantaneous and no impact on performance
- Supports clones (writable snapshots)
- Redirect-on-write based snaps
- Each file system is individually snapshotted

Snap-to-S3 DR/Backup/burst Functionality

The ability to coherently save a complete snapshot to the object storage → The original cluster can now shut off and the data is safe → Rehydrated cluster may be of different size

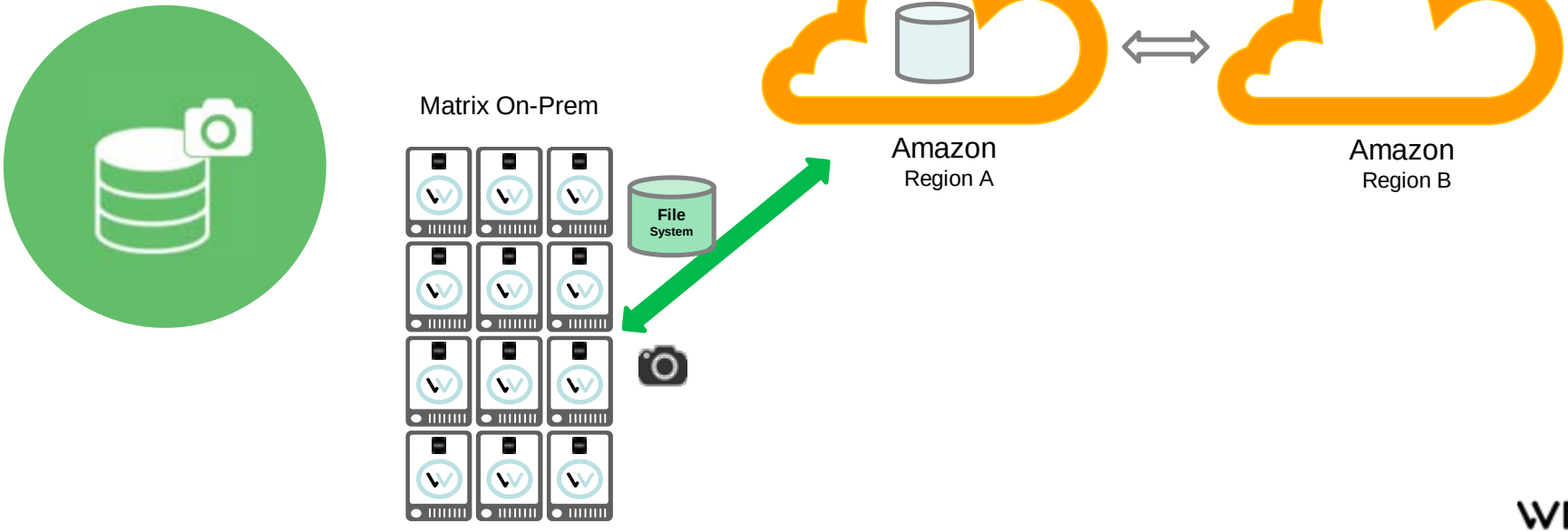
Follow-on snapshot will be saved in differential manner

Enabled use cases:

- Pause/resume : Allows a cluster to be shutdown when not needed, supporting cloud elasticity
- Backup : the original cluster is not needed in order to access the data
- DR : enabled via geo-replicated object storage, or tiering to the cloud
- Cloud bursting : launch a cluster in the cloud based on a snap saved on the object storage

WEKA.IO

Snapshot File System as DR Strategy

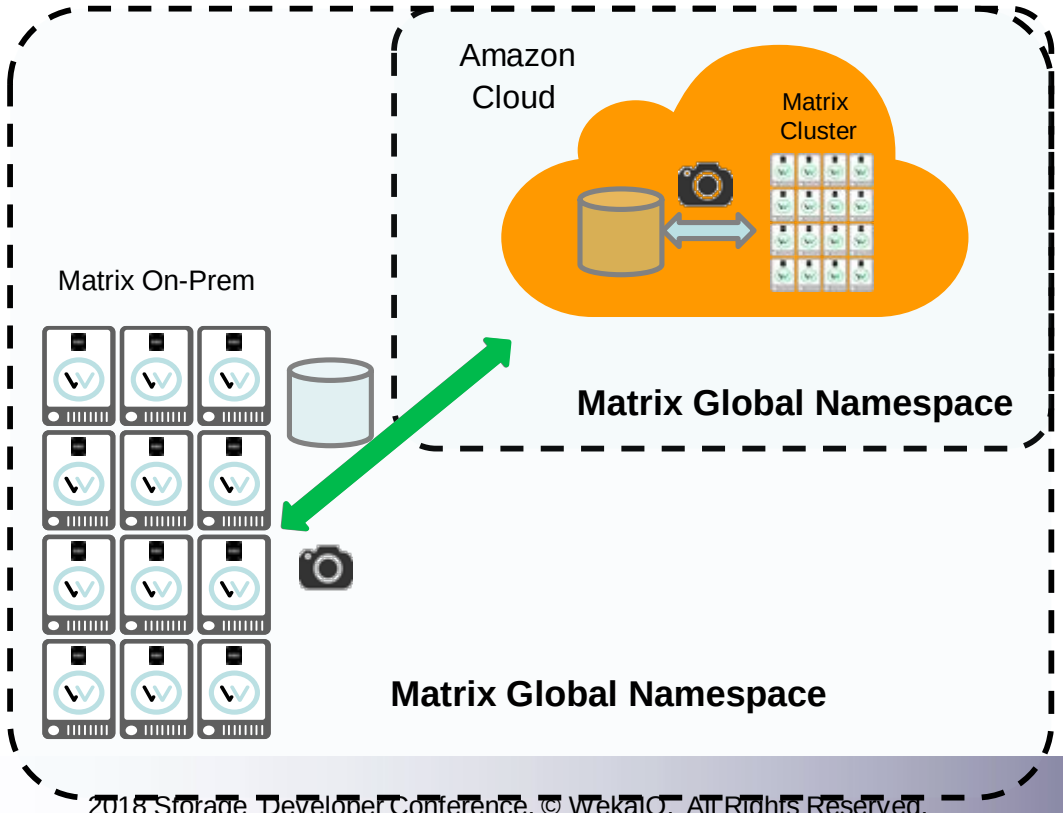


Cloud Bursting

- EC2 cluster can be formed based on S3 snap data
- The original on-premises cluster can continue running and take snaps
- The EC2 cluster can run concurrently and take snaps
- Each snap that is pushed to S3 can be linked back to the other system
- The data is viewed via the namespace using the .snapshots directory and data can be merged
- A hybrid-cloud solution that is “sticky” to the public cloud. Other solutions require network transfers that are transient
- The “data gravity” is in AWS rather than on-prem. A follow up burst is “cheaper”

WEKA.IO

Snapshot File System for Infrastructure Elasticity



Security

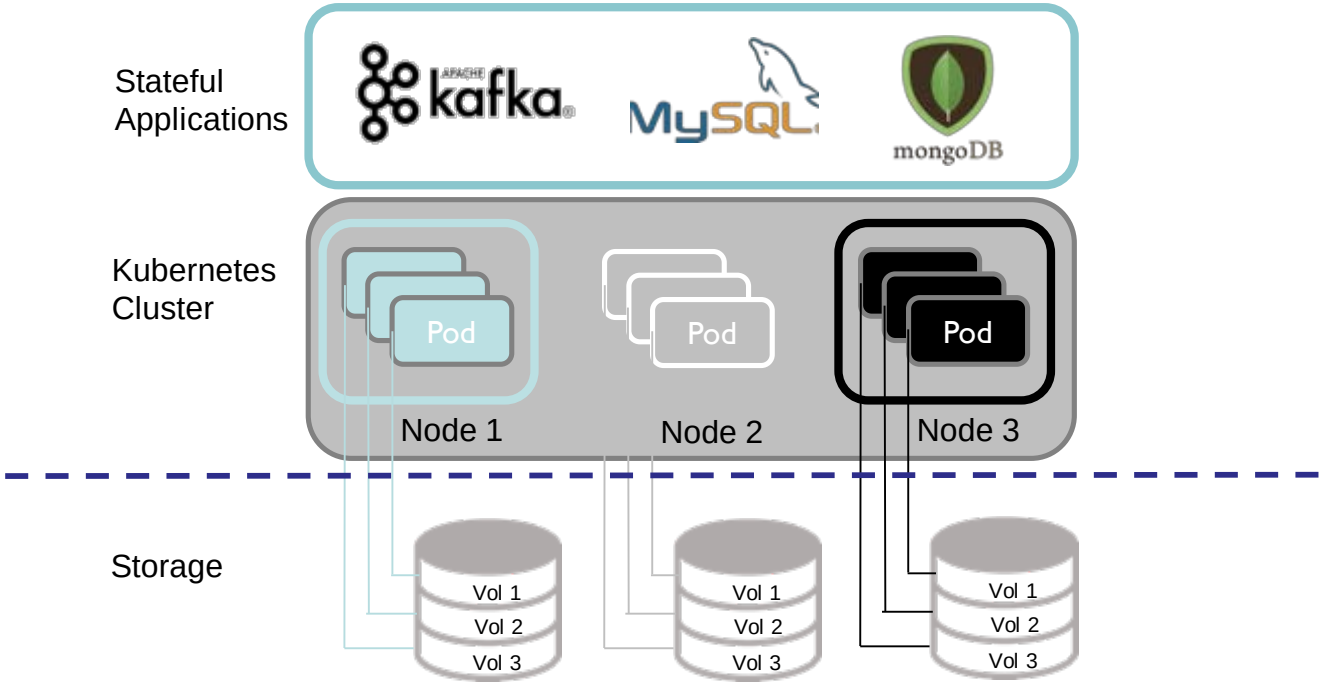
- FS adheres to the UNIX permission model
- CLI access is role base authentication - LDAP
- Enhanced security features
  - Authenticated hosts (based on PKI)
  - File System mount rules per host
  - Encryption of data at rest
  - Encryption of data in flight

Kubernetes

- Open Source software to orchestrate and manage containerized services
- Kubernetes is stateless and ephemeral
- However many applications require persistent storage
- Persistent Volumes
  - Un-sharable
  - Creates ”multi-copy” challenge
  - Cannot easily coordinate and share change data

WEKA.IO

Common Kubernetes Implementation

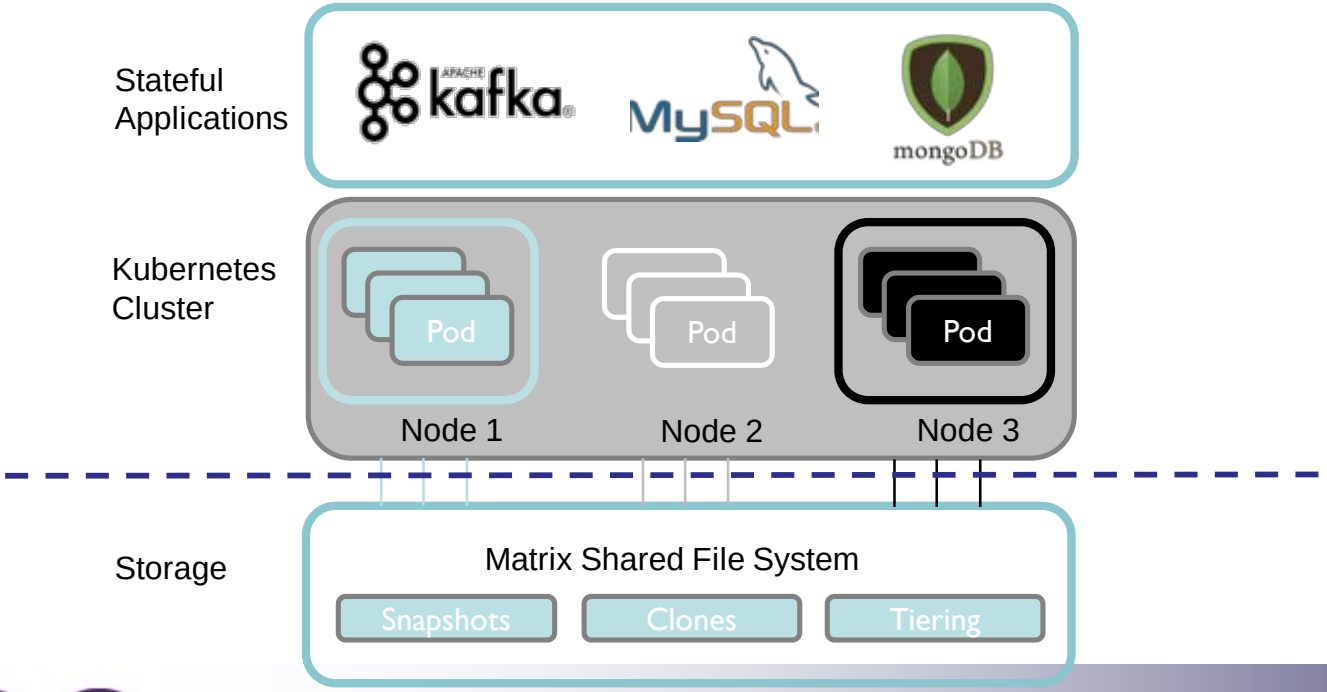


WEKA.IO

WekaIO for Kubernetes

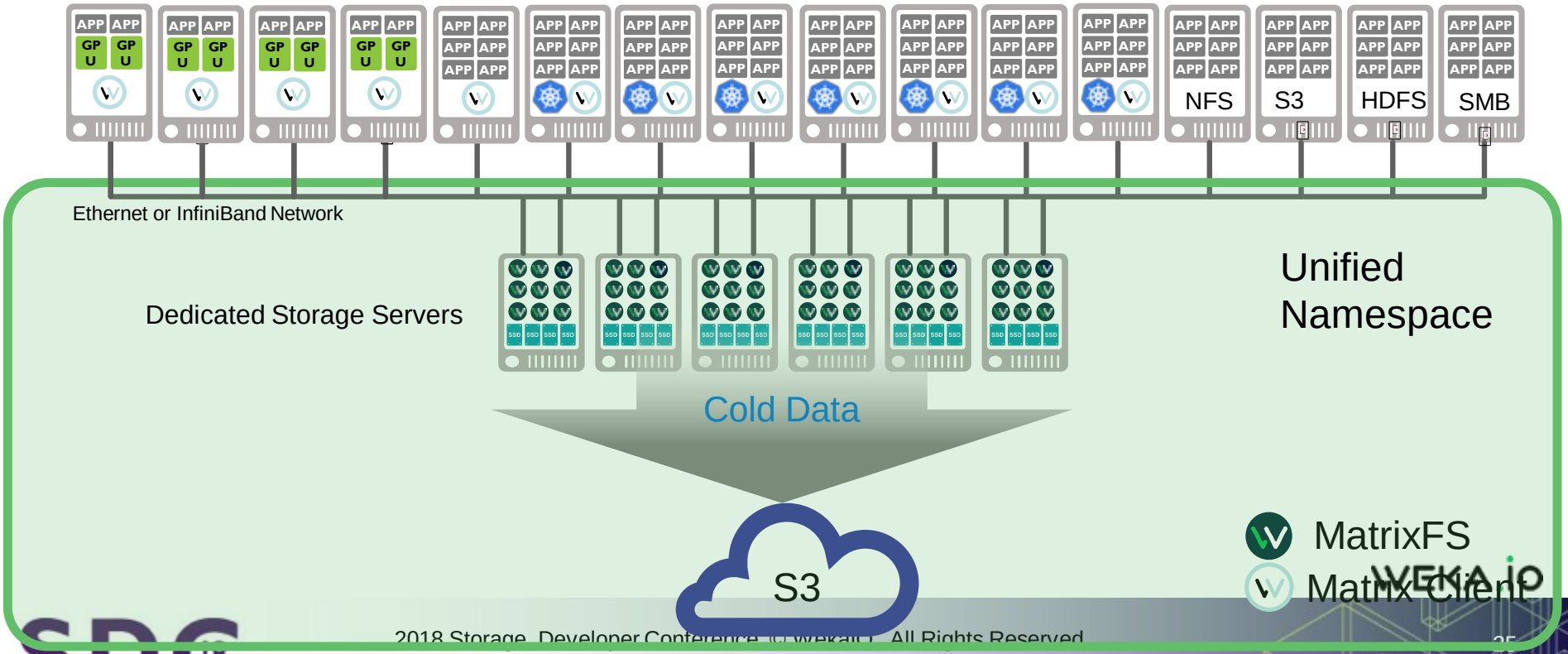
- Shared file services eliminates need for separate volumes
  - Sprawl is unmanageable at scale
- Provides better performance and lower latency than local disk or SAN
- Solves multi-copy issues
- Solves data-change issues
- Scales seamlessly as environment grows

Kubernetes With WekaIO Matrix



WekaIO For Containers

Storage on separate infrastructure with shared access to clients



WekaIO for Sharing Data Between Containers

- Projects usually start small scale and use local file system
- This works great as long as all containers that share data run on the same server, and no need for protection, backup, replication, etc.
- Traditional filers and shared FS have much reduced perf than a Linux local FS (try `untar` or `git clone` on such a mount)
- WekaIO has local FS performance for such workloads, making scheduling much easier, providing protection and shareability
- Kubernetes, Docker support, snapshots, tiering, replication
- Can burst workload from on-prem to public cloud or another datacenter

WEKA.IO

WEKA.IO

SYSTEM OVERVIEW

UP

ERASURE CODING

16 + 2

IOPS

R 10.08M W 0

10.08M

THROUGHPUT

R 38.44GB W 0

38.44GB

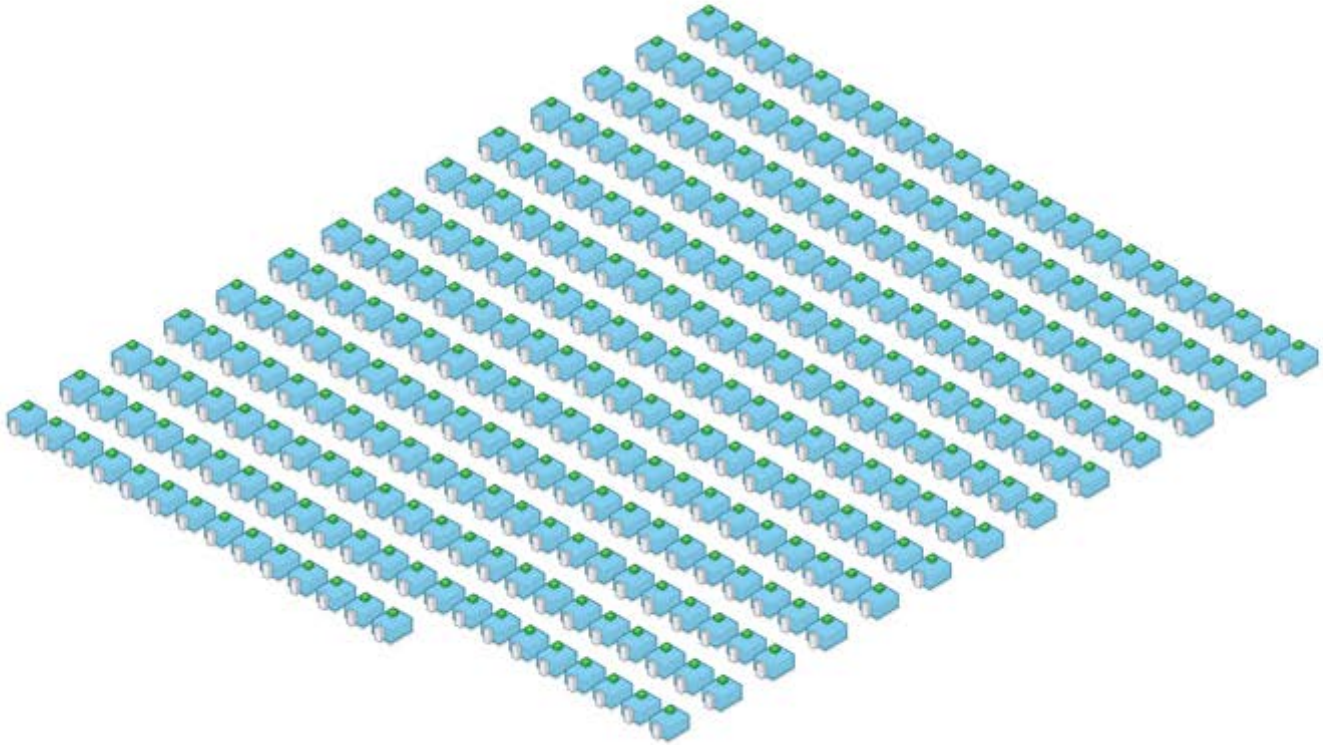
LATENCY

R 692.52µs W 0

CORE UTILIZATION

85%

CAPACITY	118.95 TB
HOSTS	300
SSDS	600
CORES	600



Unique Features

Distributed data protection and scale-out metadata services unlike any other FS

NVMe-oF enables shared FS faster than local. No caching strategy improves on WekaIO performance.

Distributed snapshots with no performance loss

Only file system that integrates NVMe and s3 storage on-premises, and burst to the cloud

Integrated tiering from flash to disk for best economics

Only parallel file system with full protocol access (POSIX, NFS, S3, HDFS, SMB)

WEKA.IO

Feature Comparison

Feature	Isilon Nitro	Pure Flashblade	Lustre	Spectrum Scale	WekaIO
Snapshots	Yes	Yes	ZFS	Yes - perf. impact	**Yes - instantaneous**
Snapshot-to-S3	No	No	No	No	**Yes - 8 targets**
Tiering to Cloud	Yes	No	No	No, data tiering = perf. impact	**Yes**
Independent cap/perf scaling	No	No	No	No	**Yes**
Thousands of nodes	No	No	Yes	Yes	**Yes**
Dynamic Perf. Scaling	No	No	No	No	**Yes**
Quality of Service	Yes	No	Load Sharing	Yes	**Yes**
Replication	Yes	Yes	No	Yes	**via Snapshot**
Data Protection EC – erasure coding	EC	EC, N+2	RAID, replication	N+3, EC only on ESS appliance	**N+4 Distributed data protection**
Compression/Dedup	No/Yes	Yes/No	No	Limited/No	**Q2'2019**
Encryption	Yes	Yes	Yes	Yes	**Yes**
S/W only, H/W independent	No	No	Yes	No	**Yes**
IB & GbE Support	No	No	No	Yes	**Yes**
End-to-end Checksum	?	No	Yes	Limited	**Yes**

Q&A

Follow us on Twitter:

@WekaIO @liranzvibel