https://www.weka.io/wp-content/uploads/2017/12/WekaIO-Architectural_WhitePaper-W02WP201712.pdf

https://www.weka.io/wp-content/uploads/2017/06/WekaIO-WP-Distributed-Data-Protection-W01WP201706.pdf

SFD15 - WekaIO

Barbara Murphy (VP of Marketing @scaleoutlady) - https://www.linkedin.com/in/bamurphy/

“Fastest, most scalable filesystem for AI and technical workloads that ensures your applications never wait for data”

Ex XIV, NetApp, EMC, IBM folks
Partner with HPE, Intel, Mellanox, AWS and SuperMicro

Only parallel file system built to take advantage of NVMe/F

What are they?
SW-only solution
Massive scale
- trillions of files
- billions of files inside a directory
- hundreds of PB
- Millions of IOPS
- hundreds of GB of bandwidth
Lowest latency
Cloud economics

WekaIO Matrix
Fully coherent POSIX file system that delivers local file system performance
Distributed Coding, more resilient at scale, fast rebuilds, end to end DP
Instantaneous snapshots, clones, tiering to S3, partial file rehydration
InfiniBand or Ethernet, Hyper-converged or Dedicated Storage Server
Bare-metal, containerised, running in a VM

Dedicated storage servers generally running 100GbE

Internally tiers to an object store
Everything managed as a single, global namespace

Hyper-converged mode

Servers can be added to the cluster with different profiles

Storage performance is defined by the amount of cores available - add more compute, you get more performance

Scale out model

Native in AWS
Storage on S3
I3.2xlarge Storage Servers
C3 CPU instances, P3 GPU instances

Customer example running 300 instances inside AWS

Focused on:
Machine learning / AI
Digital Radiology / Pathology
Algorithmic Trading
Genomic Sequencing and Analytics

Millions of small files
Metadata intensive
Latency sensitive
Huge capacity
Huge data growth

“Speed, Simplicity and Scale”

Scale out - NetApp, Qumulo, Pure Storage, EMC
Parallel - Lustre
Speed - E8 Storage, Excelero, vexata

10 CPU-only servers -> 1 GPU accelerated server

“NFS - Not For Speed”
A protocol developed in 1984 trying to solve a 2018 problem
pNFS tried to fix NFS but failed when metadata workloads exploded

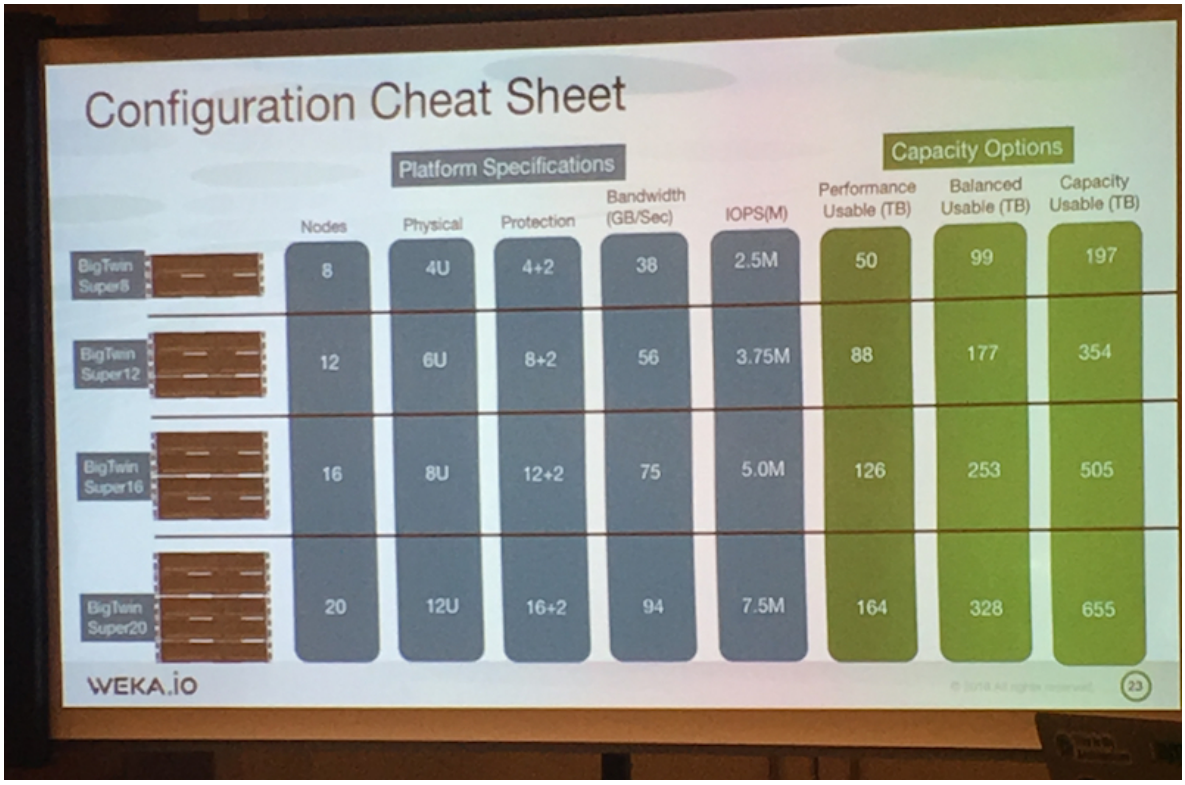
WekaIO
- shared, parallel file system written for NVMe
- POSIX client runs on GPU servers

Access the same data through all interfaces

SDSC - key partner

No more than 30% of sales will be direct
OEM (HPE and Penguin Computing)

Goal is never to be an appliance vendor



*Liran Zvibel (Co-founder and CEO @liranzvibel) - https://www.linkedin.com/in/liranzvibel/
Core IP
Software based for dynamic scalability
- software scales to thousands of nodes and trillions of records
- significantly more scalable than any appliance offering
- metadata scales to thousands of servers
Patented erasure coding technology
- Allows us to use 66% less NVMe compared to triple replication
- Fully distributed data and metadata for best parallelism / performance
- Snapshots for “free” with no performance impact
Integrated tiering in a single namespace
- allows for unlimited namespace critical for deep learning
- enables backup and cloud bursting to public cloud

Real scale = 64k servers

Architecture
File Services (NFS, SMB, HDFS, REST)
File System Clustering
SSD Access (run over InfiniBand, Ethernet)
Object Connector (Object Store - S3 or SWIFT, On-premises, Cloud Object)

Whiteboard

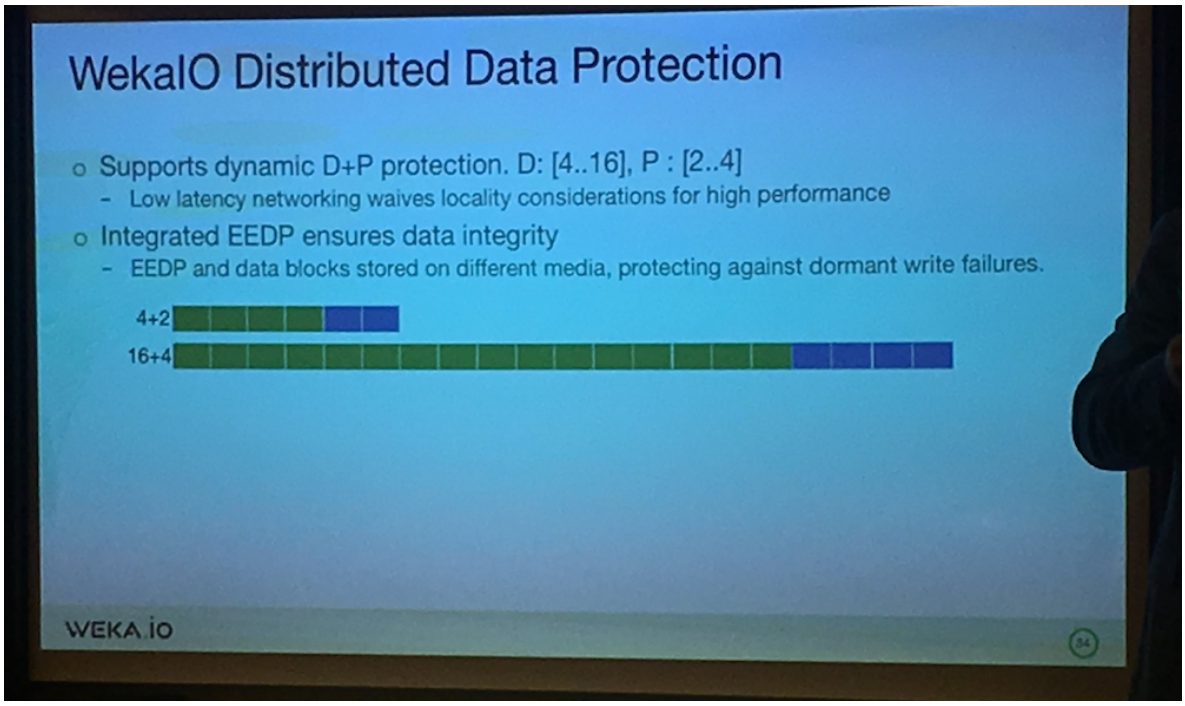
User Space
Backend - Protection, Metadata, Tiering
Front End - S3, NFS, SMB, HDFS

Clustering

Why data locality is irrelevant
Local copy architecture (e.g. Hadoop, or caching solutions) were developed when 1GbE and HDDs were standard
Modern networks on 10GbE are 10x faster than SSDs
It is much easier to create distributed algorithms when locality is not important
With right networking stack, shared storage is faster than local storage

https://www.weka.io/blog/data-locality-irrelevant/

WekaIO Distributed Data Protection



Data Distribution for Parallelism
- data is distributed across all nodes
- cluster resiliency increases on bigger clusters
- in case of failures (even transient) the system diverts writes / reads away dynamically

Failure domain level selected at cluster creation, or when it's extended

Fully Distributed Snapshots
- 4K granularity
- Instantaneous and no impact on performance
- Supports clones (writable snapshots)
- Supports file system wide or file-based snaps
- Redirect-on-write snaps

Security
- FS adheres to the UNIX permission model

- CLI access is role based authentication - LDAP
- V3.2 - enhanced security features
- - authenticated hosts (based on PKI)
- - file system mount rules per host
- - encryption of data at rest
- - encryption of data in flight

Tiering
- HDD integration provides optimal economic model
- Performance of NAND Flash and the economics of S3
- Parallel FS algorithms over the HDDs similar to traditional solutions
- Large files are chopped down into small objects to achieve parallelism - can rehydrate and modify partial files (great for image and large files)
- Stage next workload to NVMe while current workload is running
- data demoted to object storage remains on SSD as a cache until space is needed

You always want your snapshots stored on third-party storage

Deep Learning Requirements
- Actually very close to HPC problems
- Store a vast amount of data - effectively “stage” working set back on fast storage for efficient access
- high bandwidth, low latency
- Very good metadata performance, traverse files quickly
- Very high single host performance
- Support multi-protocol (S3, HDFS, SMB, NFS)

Unique Defensible Features
- Distributed Data Protection with no performance loss and lowest wear on flash
- distributed snapshots with no performance loss
- integrated tiering from flash to disk for best economics
- highest performance file system with native NVMe/F support
- only file system that integrates NVMe and S3 storage on-premises, and burst to the cloud
- Only parallel file system with full protocol access (POSIX, NFS, S3, HDFS, SMB)

Demo Time with Shimon Ben-David (https://www.linkedin.com/in/shimonbd/)

Hint versus Policy