Designing Next Generation FS for NVMe and NVMe-oF

Liran Zvibel

CTO, Co-founder Weka.IO

@liranzvibel

Flash Memory Summit

Clustered

Designing Next Generation FS for NVMe and NVMe-oF

Liran Zvibel

CTO, Co-founder Weka.IO

@liranzvibel

Does it work?

- ~~You're invited to the Intel Partner booth to see~~
- ~~Show GUI of the WRG~~

- We'll talk about creating filesystems that scale to 100s of PB providing 100 of M of IOPS at very low latencies

Why are we here?

- Most local file systems are not NAND FLASH optimized
- No high performance clustered FS created for a very long time

FS for NVMe and NVMe-oF ?!?

- Right, there are already Block based solutions leveraging NVMe-oF
- The shared FS world has not moved up to the challenge yet

Mission Statement

- Architect the best scale-out distributed file system based on NAND FLASH, optimized for NVMe and high speed networking
- Run natively in the cloud and scale much beyond the rack level

What can be improved

- Random 4k IOPS and with low latency
- Metadata performance
- Affordable throughput
- Write amplification

Changes even on a local scale

- Trees are horrible for NAND FLASH
  - So even though FS is a hierarchal in nature, other means need to support it
- atime updates means that reads create writes
  - Horrible from write amplification standpoint

Data protection must be flash friendly

- Triple replication too expensive for flash (or throughput!)
- Protection should be coded, large clusters can go 16+2 with very little protection overhead both on FLASH and Network

FLASH means more metadata power needed

- Current clustered filesystems are good at providing high throughput over many HDDs
- FLASH has the ability to provide much more IOPS and metadata operations
- Older filesystems did not try to optimize for 4k workloads, as HDDs could not carry them

Solving metadata

- The metadata processing associated with distributed FS is massive
- Up scaling is not enough
- The solution must be able to shard metadata into thousands of smalls pieces (compared to 10s currently supported)
- We can break it down to 64k pieces effectively

Solving metadata – cont

- Latency increases from coordination
- Each metadata shard takes care of the of the logic of the operation in a lockless manner while requiring minimal help from another networked component
  - Otherwise, cannot scale

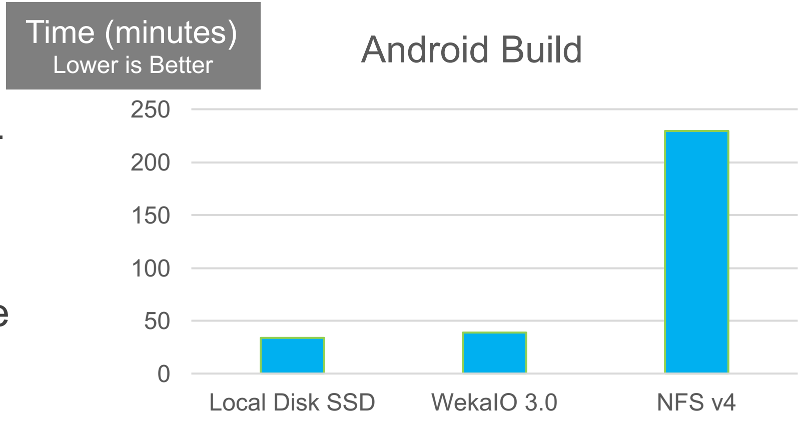
Protocol must change

- NFS over TCP/IP incompatible with low latency
- NFS is stateless and inefficient. “Chatty”
- Local state-full connector with POSIX semantics leveraging NVMe-oF for data transfer

EDA Software Compilation Results

- Problem: Only way to get fast build was to run software builds on local SSD.
- Pain Point: Wasted engineering time. Lots of data copying back and forth from NAS. Multiple copies of the same data taking up space.
- WekaIO Benefit:
  - Matrix completed in 38 min vs. 3.8 hours for NFS NAS.
  - Data never has to be copied. Data is fully sharable.
  - Massively scalable, sharable and simplified compared to local disk
- Test Platform – 14 Node HPE Proliant vs. Oracle ZFS
- State-full protocol key for such improvements

WekaIO is 6X Faster than Oracle ZFS



User-space is king

- Kernel bypass actually increases performance
- No need to depend on Linux kernel for Networking or the IO stack
- Our own memory management allows zero-copy stack
- Own scheduling reduced context-switch latency

Our networking stack

- We have an Ethernet native network stack that is 100% kernel bypass, supports zero-copy operations, and very low latency
- Routable, with retries flow control and congestion control
- We also support NVMe-oF :-)

Locality is irrelevant

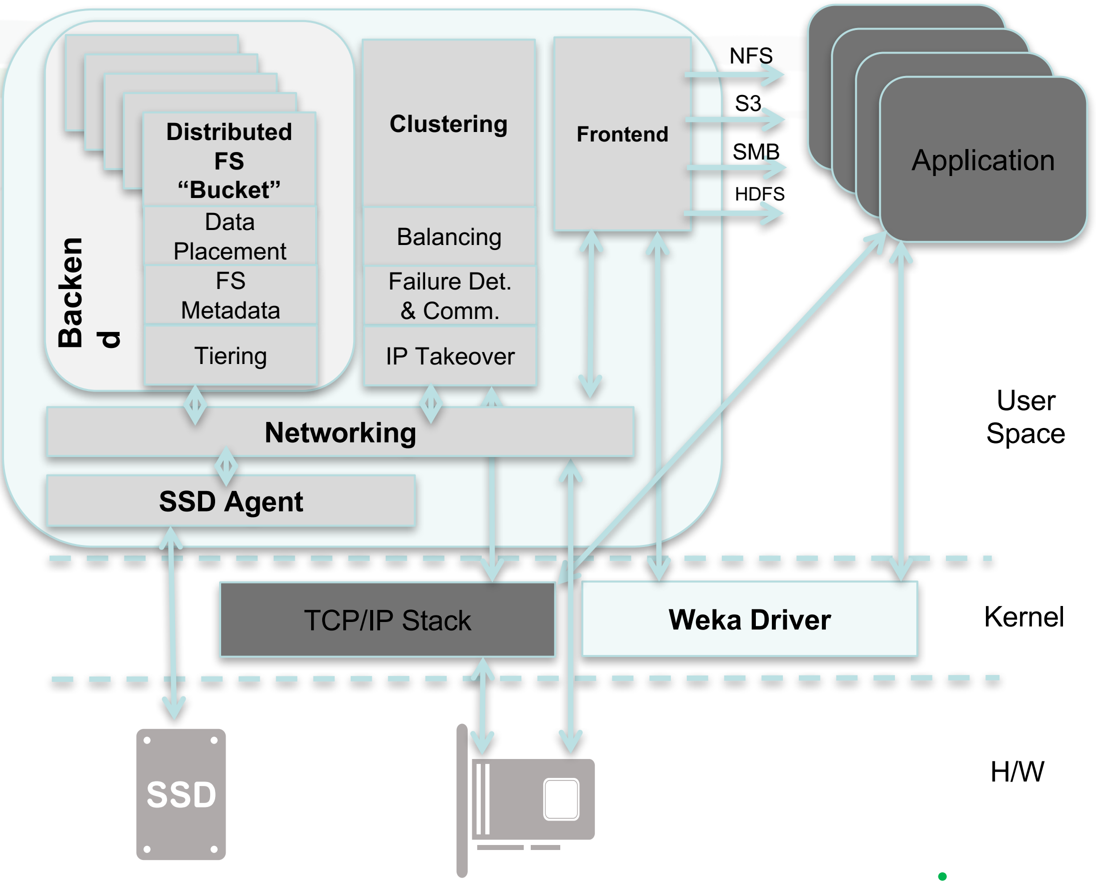
- Over 10Gb/e moving 4KB page takes 3.6 μsec
- Over 100Gb/e moving 4KB page takes .36 μsec
- NAND FLASH media is WAY slower!

- Once locality stopped being importnat, it‘s much easier to create distributed algorithms

Software Architecture

Flash Memory Summit

- 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
- Metadata split into many Buckets – Buckets quickly migrate ➔ no hot spots
- Support, bare iron, container & hypervisor



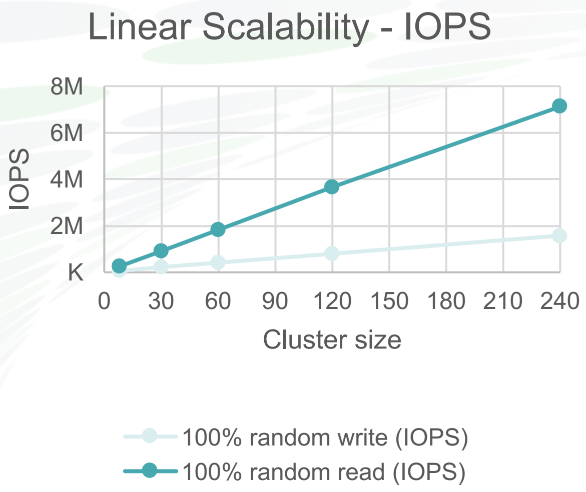
WEKA.IO
Radically Simple Storage

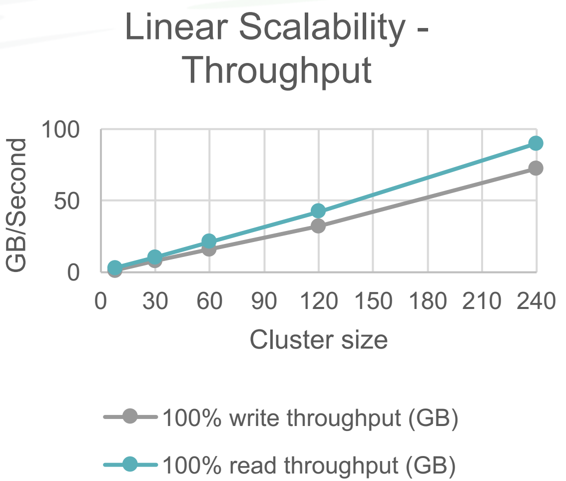
Our per-core NVMe performance

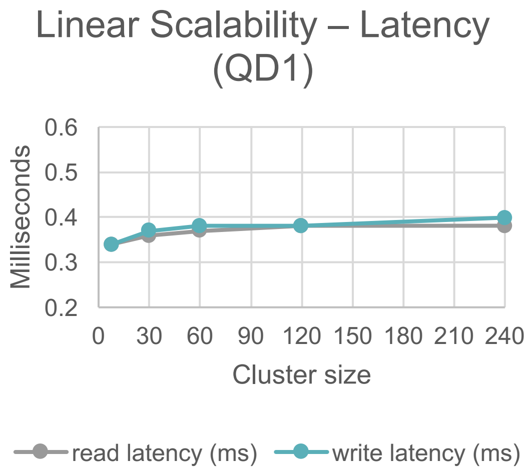
- Random 4k filesystem reads : 50,175 (less than 500 usec avg latency)
  - QD=1 read latency 188 usec to application (5237 IOPS)
- Random 4k filesystem writes: 11,320 (less than 700 usec avg latency)
  - QD=1 write latency 150 usec to application (6658 IOPS)

- Sequential reads: 980 MB/sec
- Sequential writes: 370 MB/sec

Perf Scales Linearly with Cluster Size





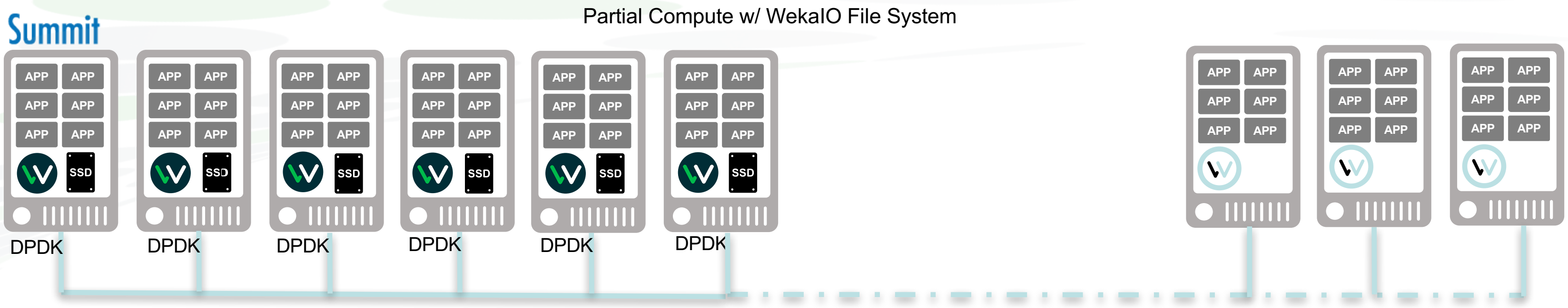


~30K OPS/AWS Instance
~375MB/sec/AWS Instance
<400 microsecond latency

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

WEKA.IO
Radically Simple Storage

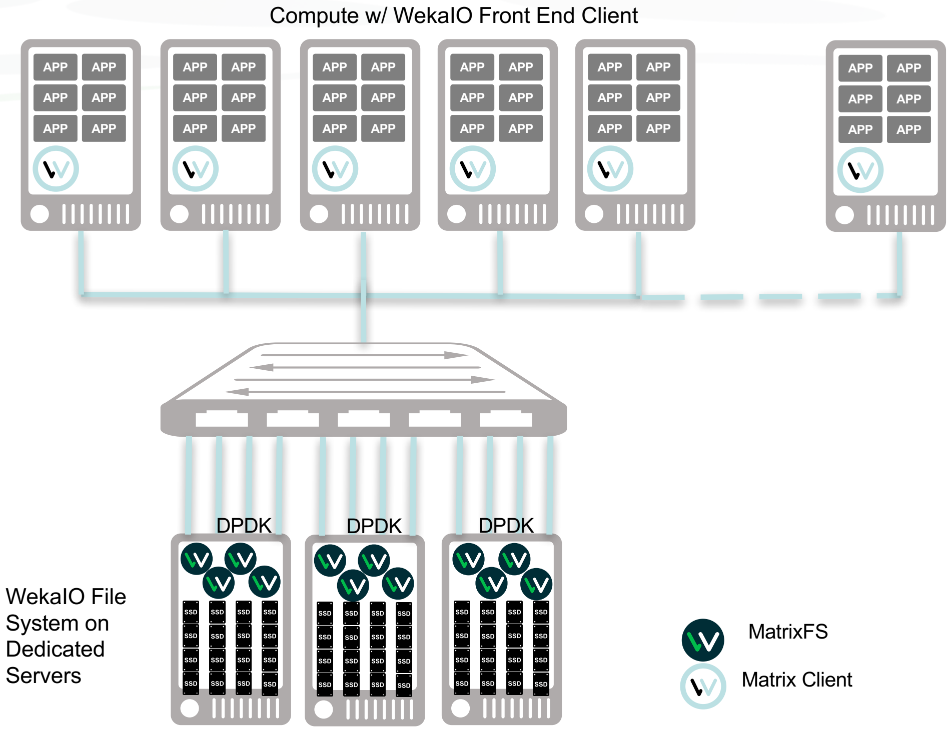
Hyperconverged Mixed environment



- Ideal for customers who have a mixed environment or who have limited capacity and performance needs
- WekaIO and SSD in every storage enabled node

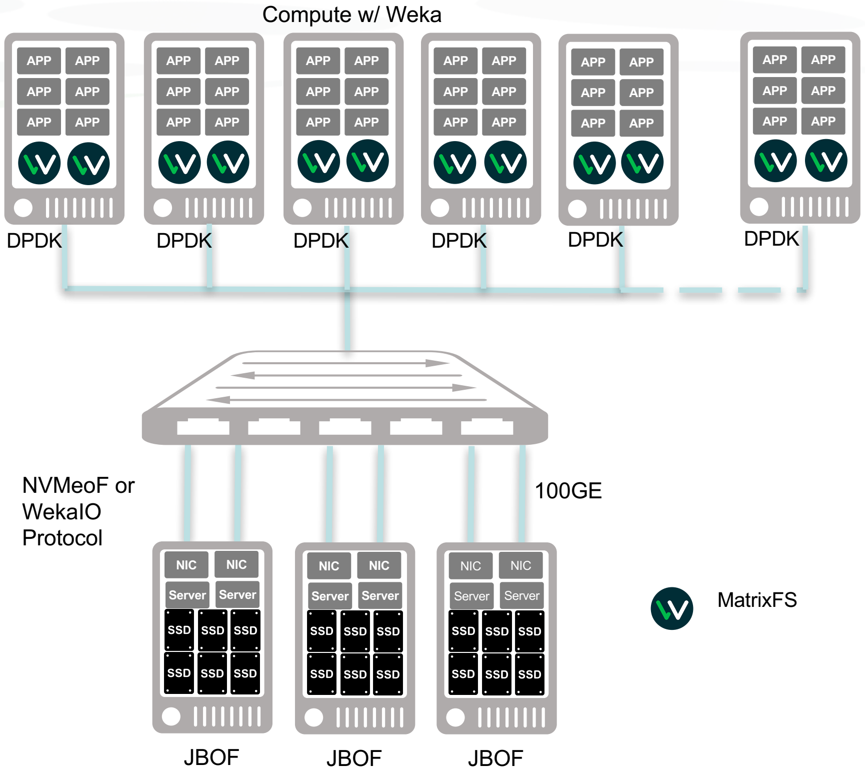
Dedicated Server Model

- Ideal for customers who do not want to run storage services with compute services
- Requires at least 3 storage servers. The more servers, the quicker the rebuild time



Disaggregated JBOF

- Ideal for customers who do not want to run storage services with compute services
- No need for a large number of JBOFs, can start with one
- Each SSD is its own failure domain



What is NVMe-oF?

- Couples NVMe devices with a networked Fabric
- Can be supported in SW or HW accelerated
- Several fabrics supported: RDMA based (IB, RoCE, iWarp), FC , etc
- We care about the RDMA based

Zoom into NVMe-oF w/ RDMA

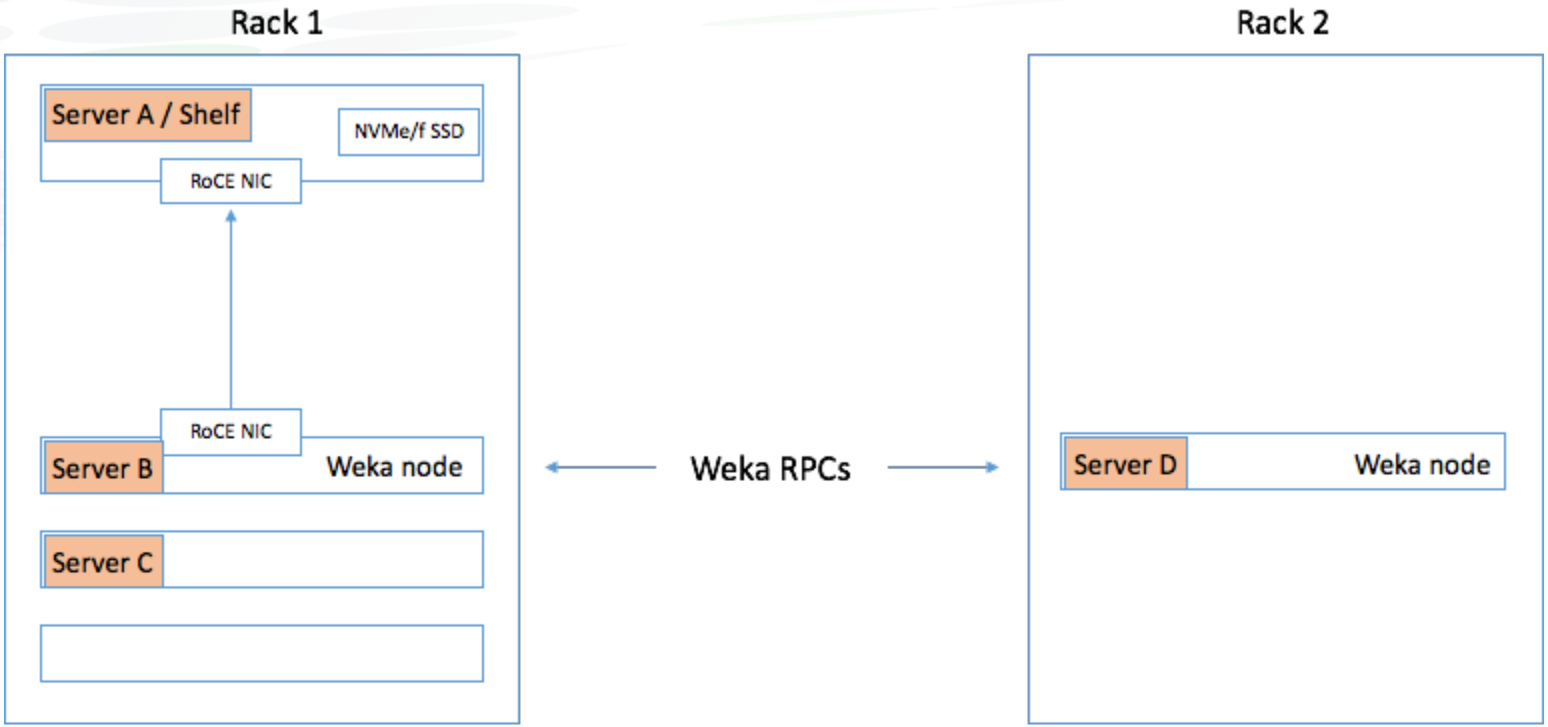
- IB is the best kind of RDMA, just works!
- Ethernet has RoCE support, requires PFC configured
  - Means inside a TOR works great, very difficult to get configured across the data center, or even several racks

But you promised datacenter scale!

- The trick is to use NVMe-oF inside an RDMA domain (or “rack scale”), and then Weka network protocol between racks

Flash Memory Summit

Multi-rack Architecture



WEKA.IO
Radically Simple Storage

NVMe-oF increasing reliability

- In HC mode, if NVMe device talks directly to the NIC then survives kernel panics, server failures as long as there is power
- Requires HW accelerated NICs that convert DAS NVMe SSD transparently to NVMe-oF connected device

NVMe-oF reducing overhead

- HA appliances are coming soon with a SoC instead of Intel Processor reducing overall solution price considerably

WEKA.IO

Radically Simple Storage™