WEKA

October 2022

Beyond Lift-And-Shift:
Optimizing Cloud Storage For IO Bound Workloads

Joel Kaufman

Senior Technical Marketing Manager

Agenda

Use cases

- Database and Analytics IO
- Cloud deployment

Architectural challenges

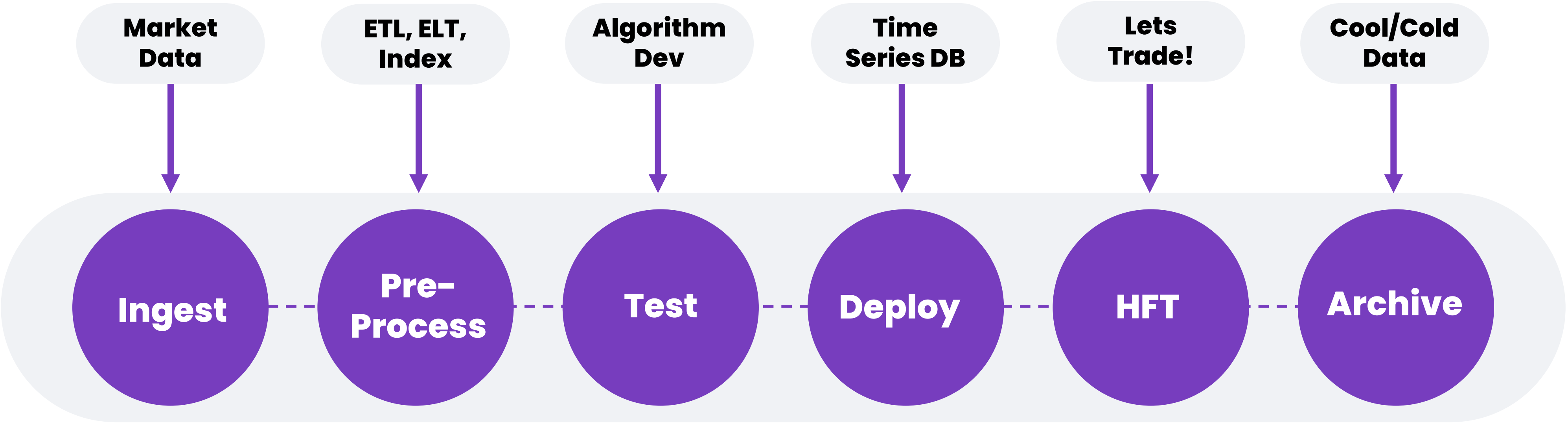
- Parallelism and low latency
- Scaling

MetaData Madness

- Appliance dedicated? Parallel but centralized MD? Distributed MD?

WEKA Conclusions

Use cases: WorkFlow and IO



Needs massive concurrency, Write throughput

Needs annotation, index, search, cloud bursting

Needs concurrency, Read throughput

Needs large no. of streams replay

Needs low latency access, high concurrency

Needs lifecycle mgmt., versioning, reproducibility

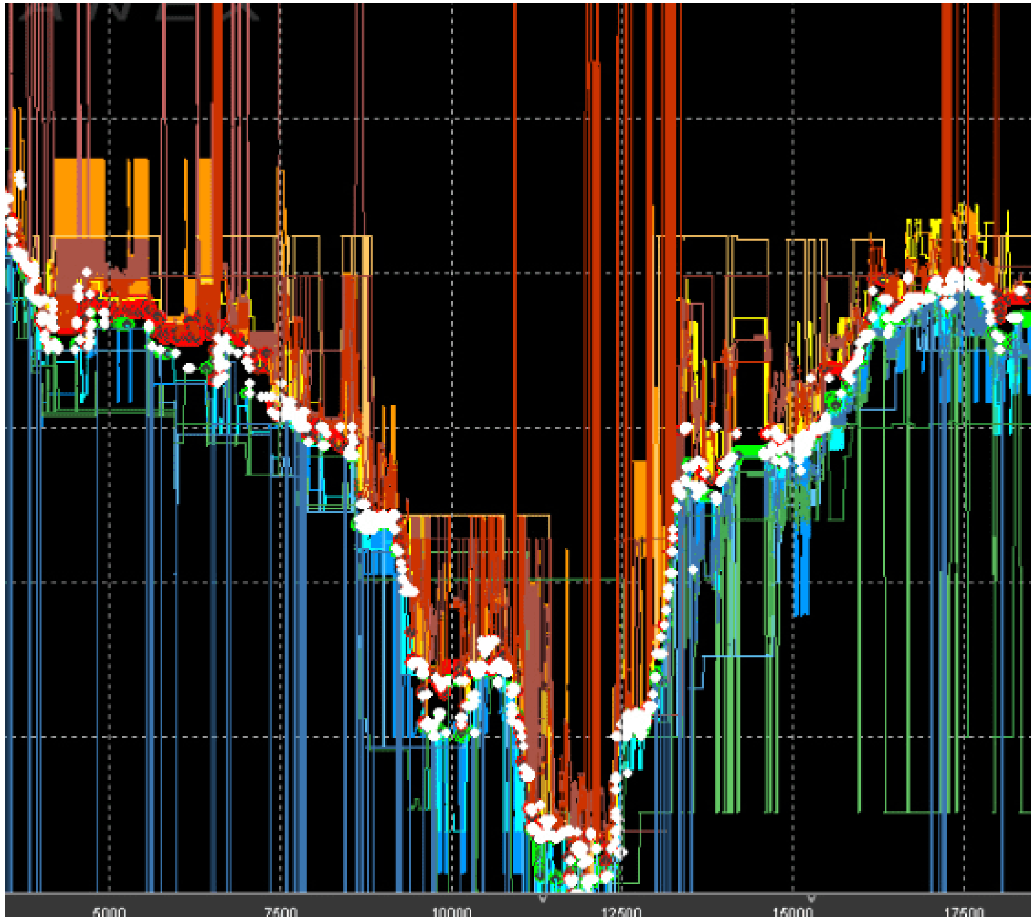
Use cases: Cloud usage

ON	WITHIN	TO	WITH	BETWEEN
Run Natively on the Cloud	Tier and Reduce Data Within a Cloud	Move or Backup to Clouds (Hybrid)	Use the Cloud for Data Tiering (Hybrid)	Migrate or DR Between Clouds

HFT Challenges: It’s All About Latency

Trader’s Tech Stack is a Major Competitive Advantage, Often its the Competitive Advantage

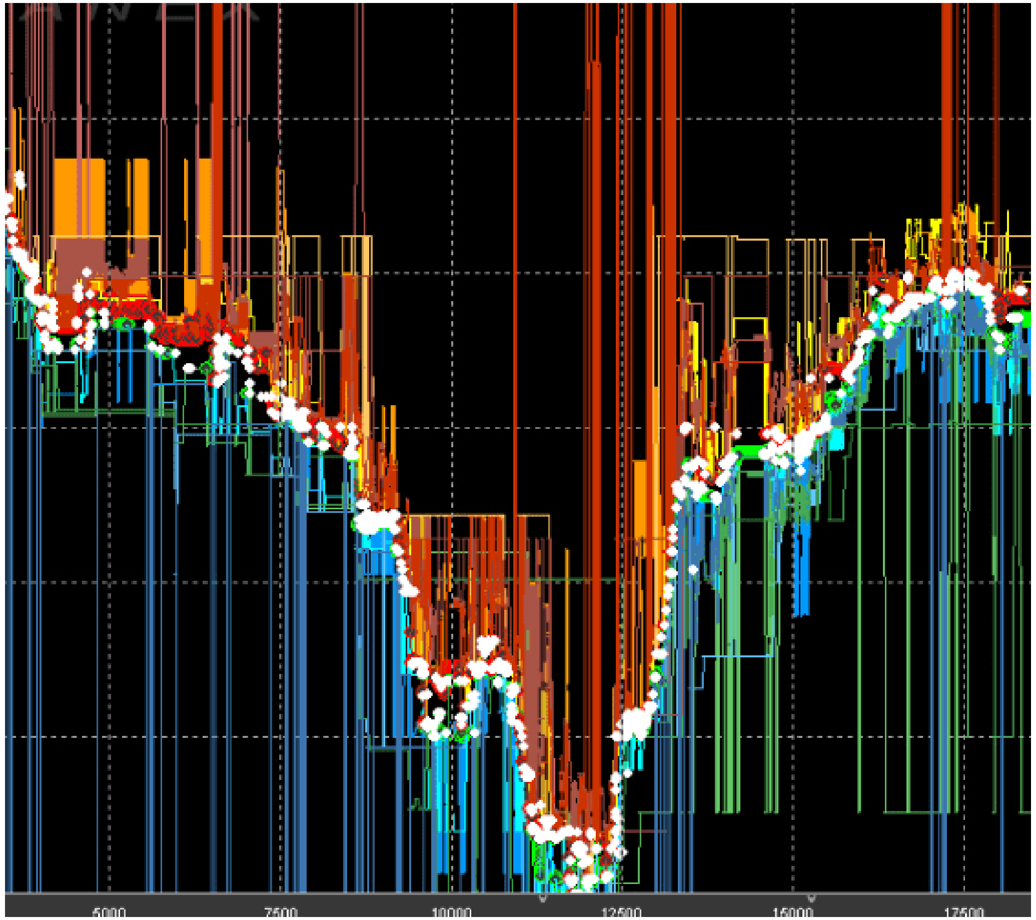
- High-Frequency Trading (HFT) slashes latency on market information to make profits before the competition
- In HFT, latency is often the only determinant for profit, or loss
- Not all algorithmic trading is high-frequency, but traders will still optimize their trading platform to execute any trade with low latency to get the best price execution



HFT Storage in the cloud

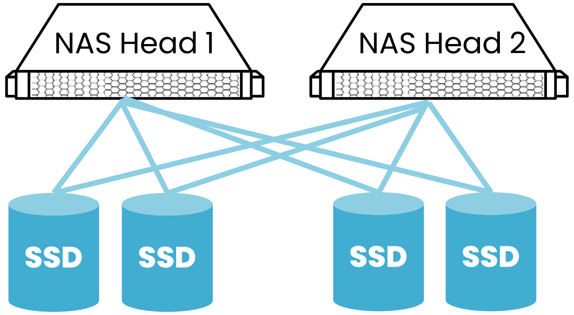
What to choose

- Most HFT needs lots of random IO for the analytics powering the recommendation system/ Time Series DB. Think STAC-M3 types of access.
- Low latency requires fast instances
  - NVMe drives, Fast networking
- Single Availability zones and instance grouping help keep Instances geographically and logically close to reduce network hops unless additional reliability is needed.
- RAM is usually more critical for the trading clients than in the storage itself**



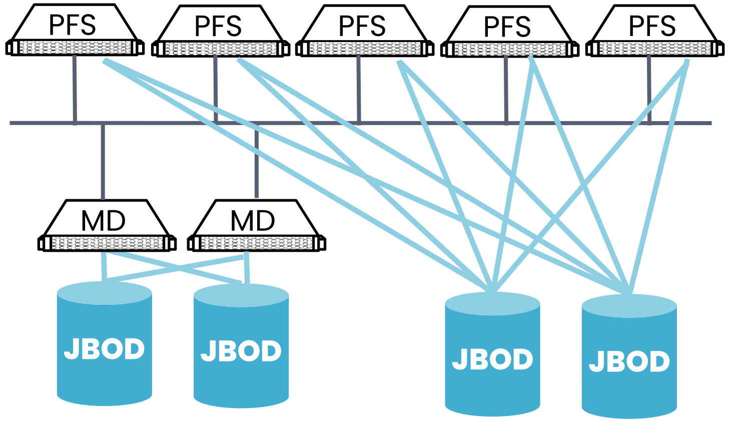
Different Architectures: Metadata Madness!

The Appliance



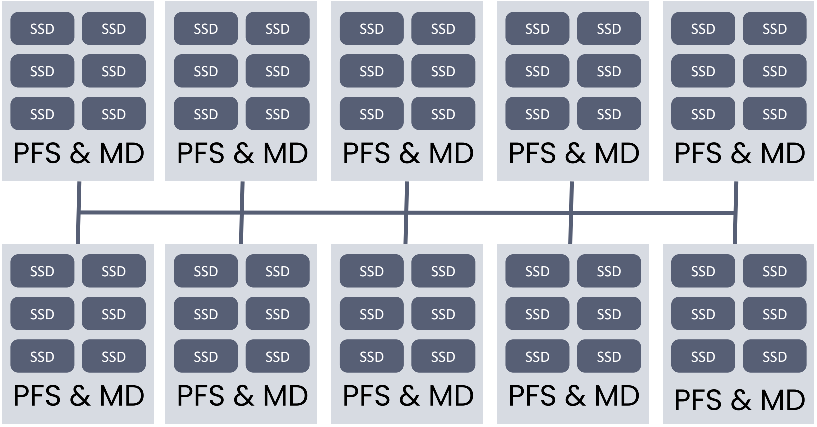
- NAS or SAN
- Limited to HA Pairs
- Metadata scale is limited to HA pair.
- Cloud Implementation recreates appliance in cloud even if it's software. May be co-located hardware.

Parallel FS



- Specialized client (POSIX)
- Scale out
- Each FS uses dedicated MD resources
- Cloud Implementation may be co-located hardware
- Native cloud implementations are non-tunable

Distributed Parallel

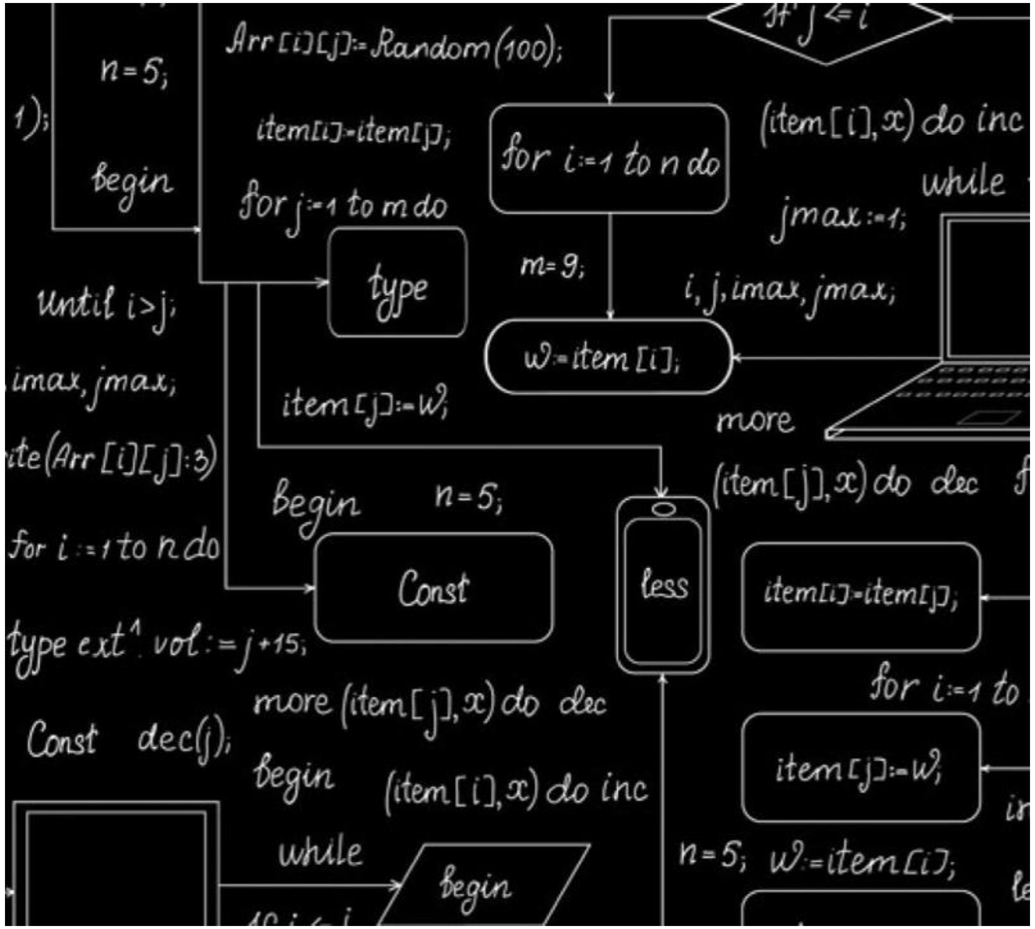


- Specialized client (POSIX)
- X86 based. Scale out or scale up
- Virtualized distributed MD in each server
- Cloud Implementation is Software running in instances
- Full SW implementation

Algorithm Development: Data Feeds More Data

Strains Infrastructure as Data Must be Continuously Fed to the Algorithm for Best Results

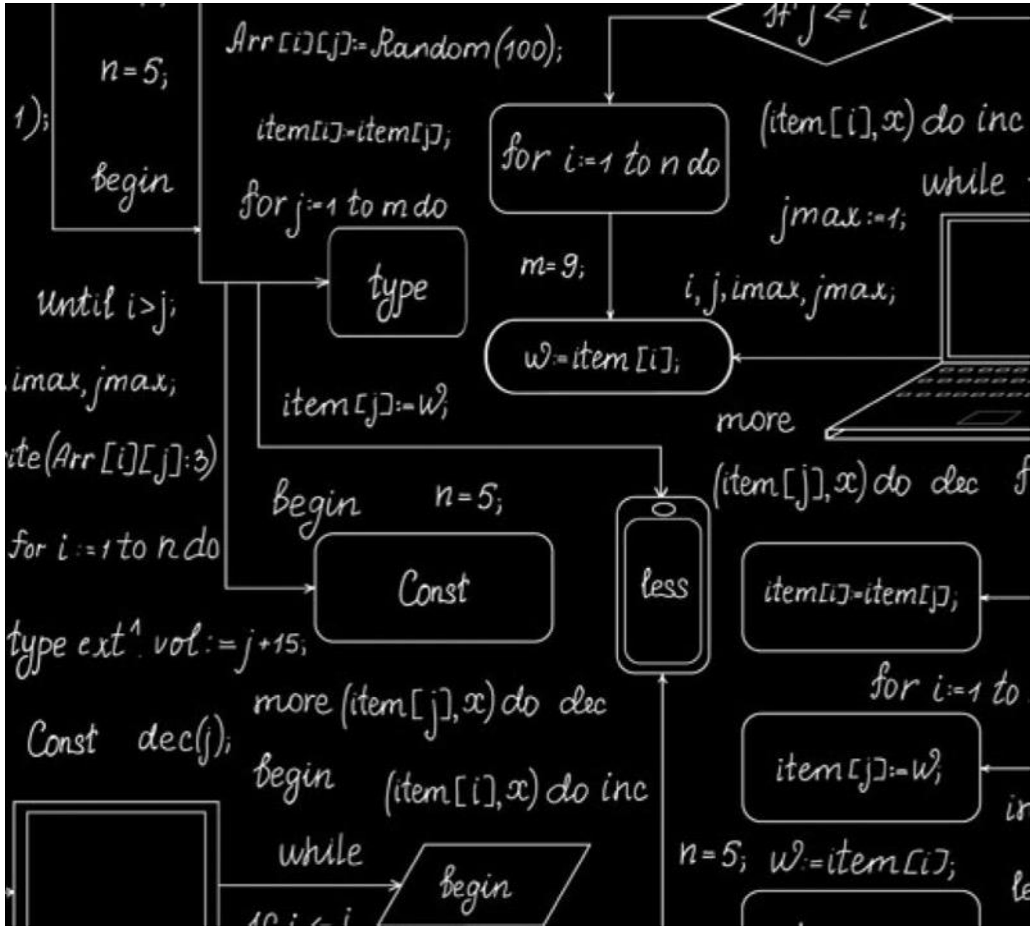
- A mix of data science, statistics, risk analysis and DevOps
- Algorithms are used for back-testing or experimenting against past data
    - Repeated to refine the algorithm
- Once results are verified, the algorithm is put in production
- Algorithm trading in the real-world markets will produce data that further feeds the algorithm backend



Algorithm Development

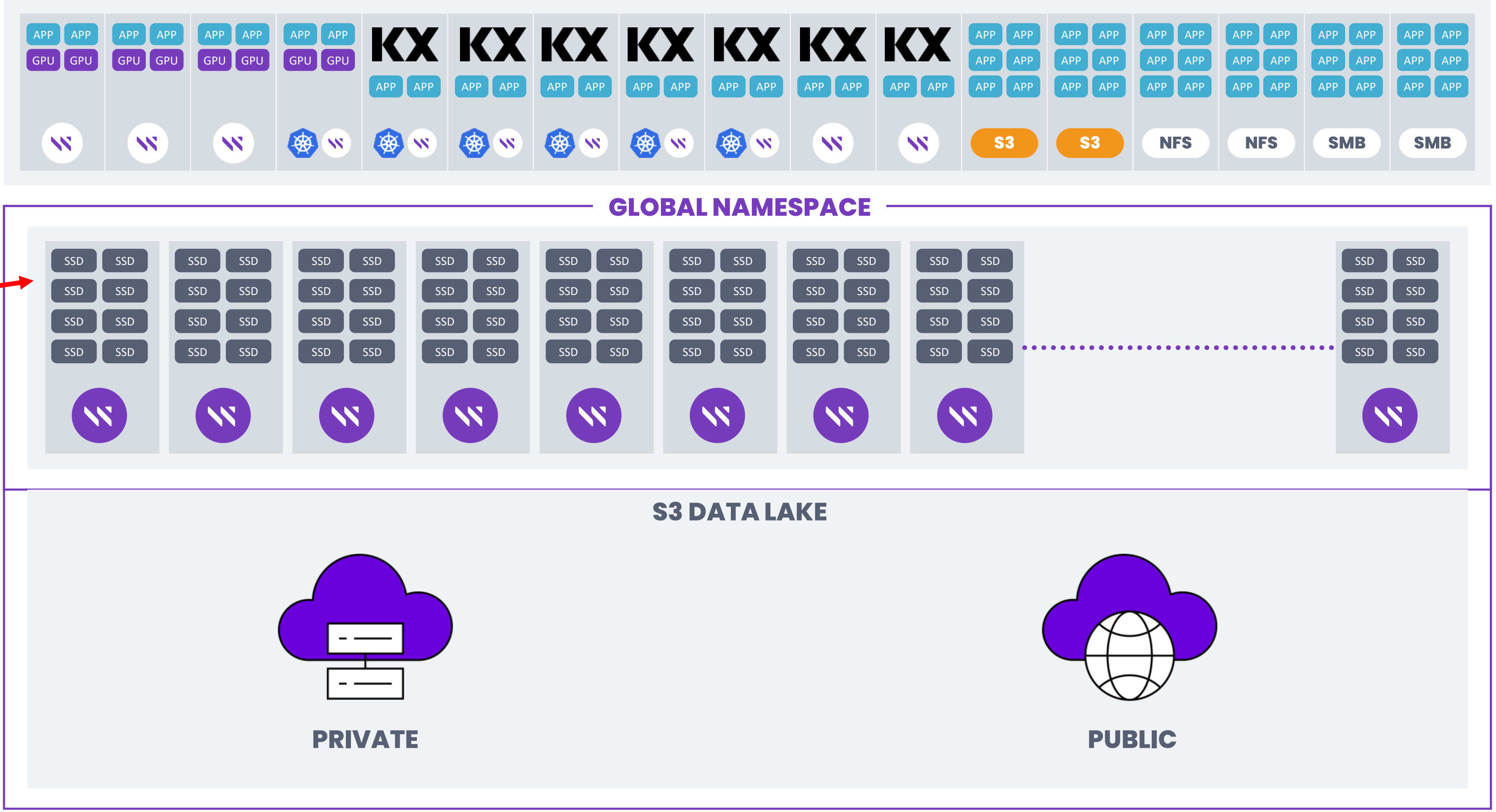
What to Choose

- High number of developers and test running in parallel = Data Blender
    - High performance in multiple dimensions is crucial. Storage needs to be flexible on types of IO
- Backtesting requires access to large amounts of historical data
    - Object store access for lower cost for cool/cold data. NVMe becomes expensive at scale.
- Scale can be thousands of clients and 10s of petabytes of data (or more)



Distributed Parallel: WekaFS

- Small 4K block size matches NVMe media 4K blocks to expose full performance of NVMe (others have 64K-1MB)
- Kernel bypass via DPDK, SPDK eliminate context switches, reduce kernel resources, pushes queuing towards zero
- All Metadata stays in the flash tier



● Weka Client

● WekaFS

Conclusion: Don’t be afraid of the cloud

- **Determine your workflow requirements**
  - High IOPS/small IO?
  - Big Throughtput?
  - Data set sharing?
- **High performance is available!**
  - Low latency
  - High Concurrency
  - Faster Tick Analytics
- **Choose which cloud meets your needs**
  - AWS, GCP, Oracle, Azure
- **Integration with cloud object stores for cool/cold data**

Q&A

Thank You!

in @wekaio

▶ /wekaio

@wekaio

Backup

Testing in the cloud: SUT# KDB210507 in AWS

3 outright records on STAC-M3 Kananga

- **WEKA was faster than other on-prem solutions as well**

	STAC System Under Test #	WekaFSv3.10 in Kanaga benchmarks	WekaFSv3.10 in Antuco benchmarks
Lustre + Appliance	*#KDB200915*	Faster in 20 of 24 benchmarks	Faster in 4 of 17 benchmarks
Direct attached 10 servers with Optane	*#KDB200603*	Faster by 16 of 24 benchmarks	Faster in 9 of 17 benchmarks
All-Flash NAS	*#KDB200914*	All-Flash NAS did not submit Kananga benchmark	Faster in 15 of 17 benchmarks

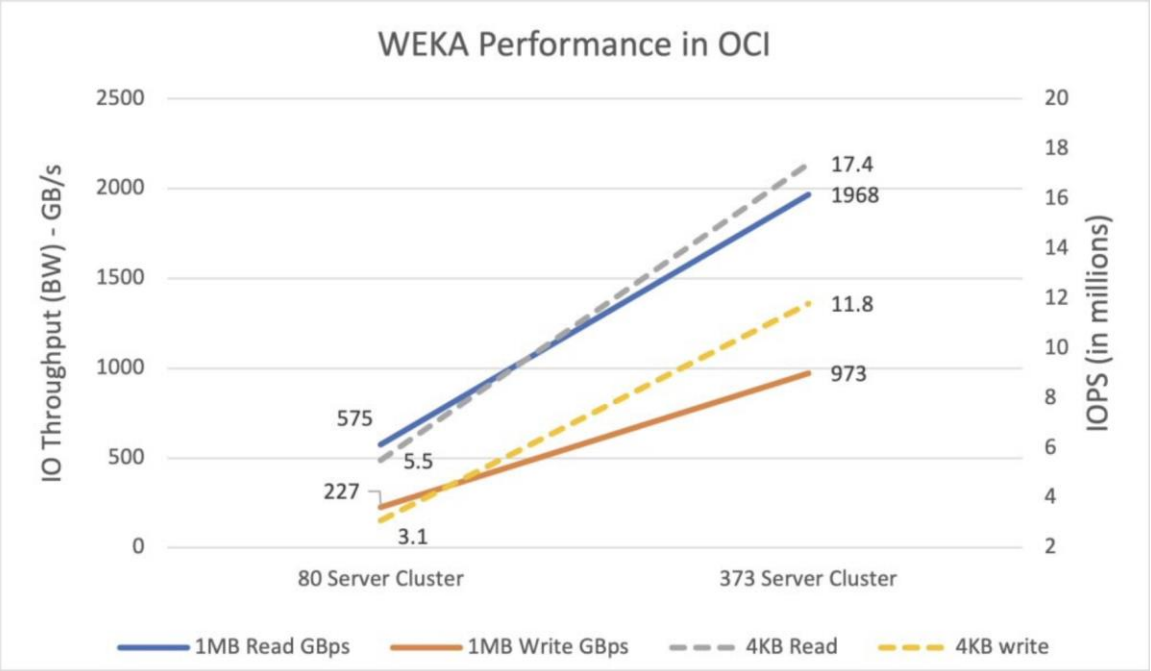
Multicloud Availability

- **Identical Code Base**
- **AWS**
  - CloudFormation scripts, full AWS API’s, Autoscaling
  - Better cost/performance profile than FSx native services
- **GCP**
  - Terraform deployment, Autoscaling
  - Brings high performance storage to GCP
- **OCI**
  - Terraform deployment, Autoscaling
  - Integration with Oracle workloads
  - Insane performance: 2TB/s!
- **Integration with ALL 3 cloud object stores!**

WEKA on OCI Delivers 2 TB per Second Performance

Maximum performance at cloud scale

- Run at petabyte scale in a high-performance file system
- NVMe SSDs for hot data and object storage for warm or cold data
- High-performance computing (HPC) bare metal Compute shape (BM.Optimized3.36)
- 100-Gbps RDMA over converged ethernet (RoCEv2) and 3.8 TB of local NVMe SSD



https://blogs.oracle.com/cloud-infrastructure/post/weka-on-oracle-cloud-infrastructure-delivers-2-terabytes-per-second-performance