AEROSPIKE

SUMMIT '19

Hewlett Packard Enterprise

AI at Hyperscale - How to go Faster with a Smaller Footprint

Theresa Melvin
Chief Architect of AI-Driven Big Data Solutions
HPE

The need to go 60% faster in an 80% smaller footprint

Fully Distributed Design

…Planning Matters

…Tools Matter

…Code Matters

…Skills Matter Most

Research, Data Science, ML Algorithms, AI Model Dev, & Deep Learning

jupyter, Apache Zeppelin, python, NVIDIA CUDA, Java, Scala, R, Spark ML, beam, FlinkML, PyTorch, Chainer, mxnet, ONNX, H_2O.ai, DL4J, theano, Caffe2, Microsoft CNTK, fast.ai, Keras, TensorRT, NMT, intel AI, TensorFlow

Applications & Frameworks

open source

Commercial Products: ververica, ALLUXIO, AEROSPIKE, WEKA.IO

Hadoop ISVs: cloudera, HORTONWORKS, MAPR

Container & Orchestration

docker, rkt, NVIDIA CUDA, Apache MESOS, SWARM, kubernetes by Google, Hewlett Packard Enterprise + bluedata

Infrastructure

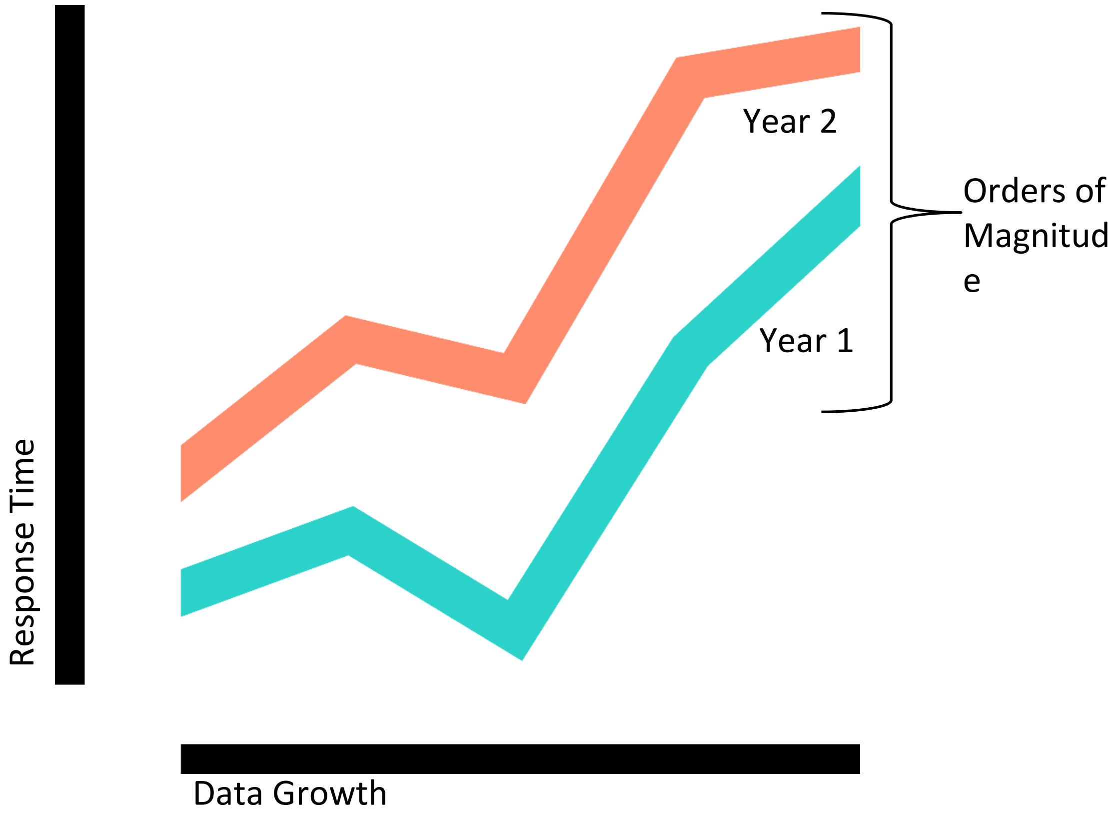
Hewlett Packard Enterprise, Google Cloud, Microsoft Azure, NVIDIA GPU CLOUD, amazon web services, 3rd Party Servers & Storage

 | Proprietary & Confidential |

Hewlett Packard Enterprise | AEROSPIKE SUMMIT '19

Welcome to the world of Big Data, where Big means Slow …

- **No Data Strategy**
  - Tools being used cannot be applied to all layers and dimensions of the data
- **Tech-Silos are the Status Quo**
  - No symmetry between IoT (if it exists), HPC, and Big Data resulting in an inability to perform AI-Driven workloads
- **Diminishing net return as more physical resources are thrown at technology problems**
  - Faults and Failures increase
  - Complexity increases
  - Visibility decreases



Hewlett Packard Enterprise | AEROSPIKE SUMMIT '19

Hello, Hyperscale! You lean, mean, efficient Machine!

- **There is no beginning, and no end to this "Solution-based" operational model**
  - Each 'Use Case' is purpose-built for maximum speed and agility
  - Designs are built from the ground-up for continuous financial and technical improvement
- **Designs are "end-to-end", starting with Ingest (IoT), moving to Research & Analytics (AI|HPC), and finishing with Storage (Big Data)**
  - Modular infrastructures
  - Portable code
  - Instantly Adaptable
  - Unified Management
  - Global Footprint
  - Aggressive Cost Controls

But wait! There's a new Extreme-Scale market emerging!

Extreme-Scale has become synonymous with Exascale

- **Exascale can mean several different things**
  - **Terabytes per second** (TB/s) **of sustained bandwidth**
    1 TB/s = 86.4 PB/day of raw ingest
  - **Exabytes of data processed (CPUs @horizontal scale)**
    Of the 2.6 Exabytes of raw data processed each month, only 30% or ~780 GB is "interesting" and needs to be stored
  - **Exabytes of stored data (Software-Defined Storage)**
    Everything has value, so all 2.6EB of monthly raw, plus another 500PB of aggregates, all of which needs to be stored for 5 or 10 years, depending on type
  - **ExaFLOP Processing (FPGA, ASIC, GPU, etc…)**
    These are currently non-existent HPC Systems capable of processing 1 quintillion (there are 18 zeros after that '1' using short form, or 30 zeros for long) calculations per second

Yeah, that's really cool, but who is seriously attempting Exascale?

- **Cloud Companies and the US Government**
  - Have been running Exascale footprints for years now
- **Academia manages Exabytes of data**
  - With no way to efficiently process it all
- **Manufacturing, Insurance, Finance, Healthcare, Pharmaceuticals & Biosciences**
  - Are well on the way
- **The transportation industry expects to have a serious Exascale problem**
  - In about 2-3 years

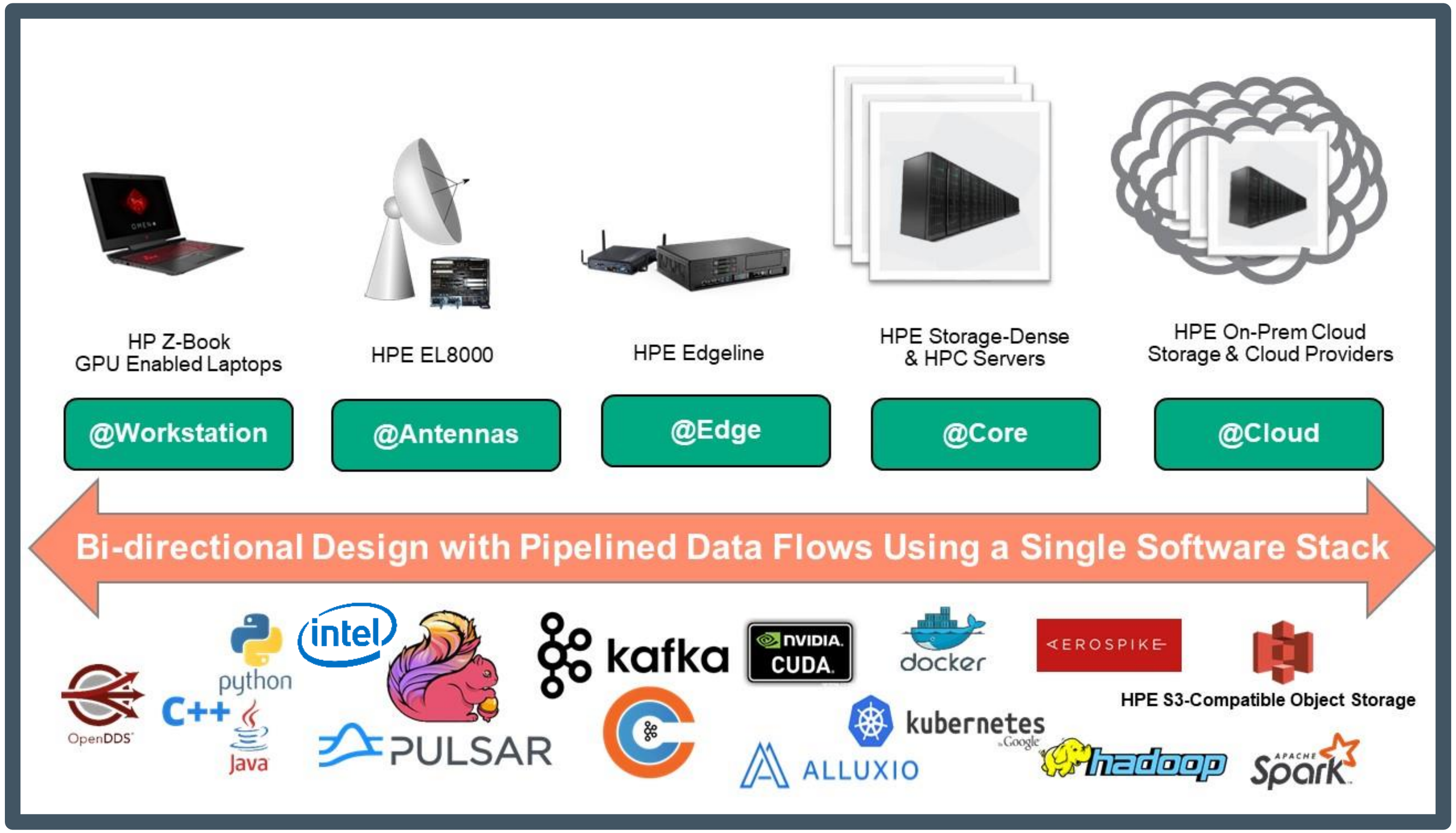
- **…And Astronomy will overtake them all, by several orders of magnitude, in less than a decade, if the smartest people in the world – from Government, Academia, and Industry can figure out how to do it**

How can this possibly be affordable?

- **Use Case (Achievable ROI in 2 years)**
  - Clear business problem → Solvable goal with current tech & skills
- **Layered Solution Design**
  - Capable of Phased Implementations
- **Skillsets**
  - What can be done in-house, versus going external
- **Software & Support**
  - Open Source and Community Supported Projects (In-House Support)
  - Commercial Offerings (Enterprise Support)
- **Hardware (+60% of the budget will go to hardware!)**
  - Minimum of 3 vendors to offset risk
- **Decentralized**
  - Data is processed in-place and as close to the compute, as possible
- **Infrastructure**
  - All Compute, Storage, and Networks operate in a mesh-grid design

Interesting, so what does a typical HPE Extreme-Scale design look like?

- **Development**
  - Rapid development, from anywhere in the world
- **Device**
  - In-place data processing
- **Edge**
  - Payload pre-processing
- **Core**
  - HPC & Batch Ops, as well as Bulk Storage
- **Cloud**
  - Global Portals for user access to distributed data



 | Proprietary & Confidential |

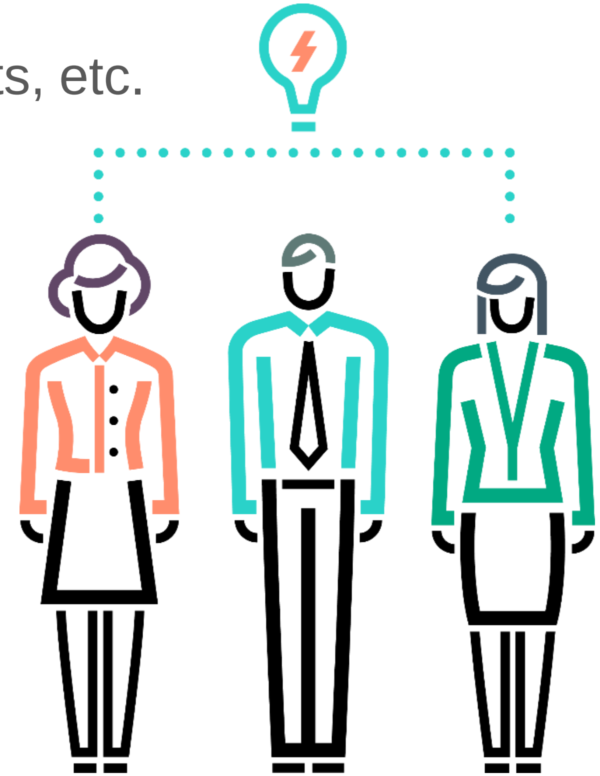
Hewlett Packard Enterprise

AEROSPIKE SUMMIT '19

Development Layer

Collaboration between (1) HPE (2) the Customer & (3) Partners

- **Business Stakeholders**
  - Management, Product Teams, Project Teams, Architects, Strategists, etc.
- **Developers**
  - Web, Middle-ware, Big Data, Data Science, HPC
- **Systems Teams**
  - Windows, UNIX, Linux, other?
- **Data Engineers**
  - Hadoop, Streams, Mesos, Docker, K8s, etc.
- **Domain-Specific Researchers**
  - Statisticians, Mathematicians, Data Scientists, etc.
- **Analysts**
  - Business, Data, Market, etc.

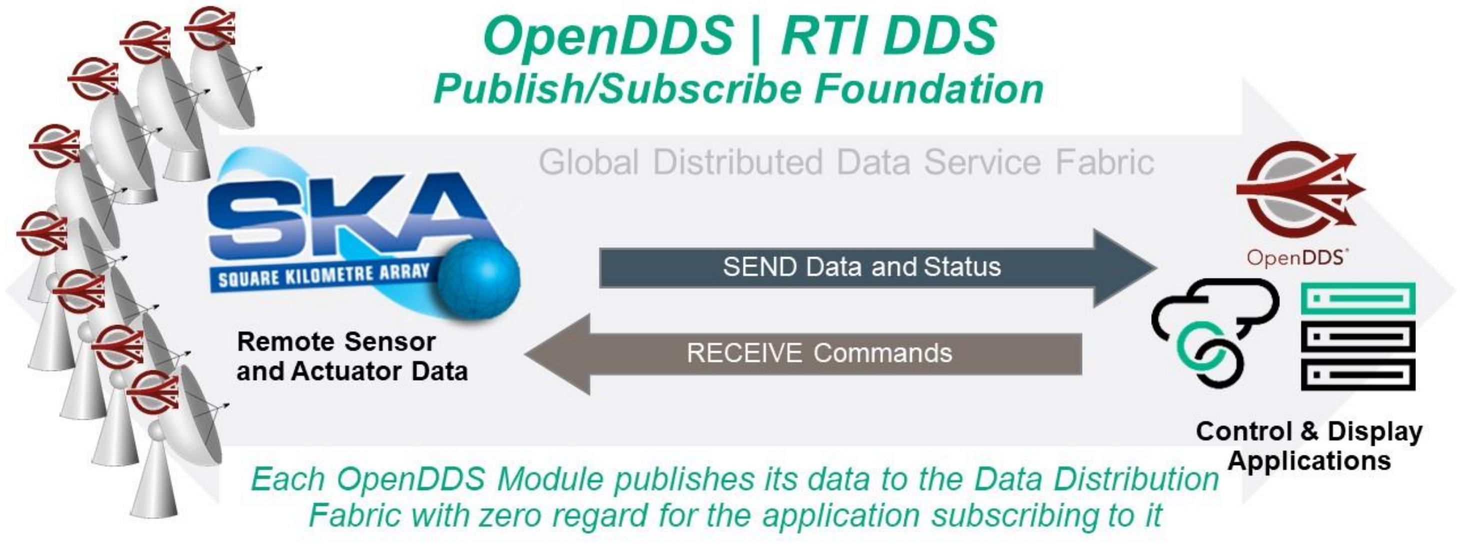


Hewlett Packard Enterprise

AEROSPIKE SUMMIT '19

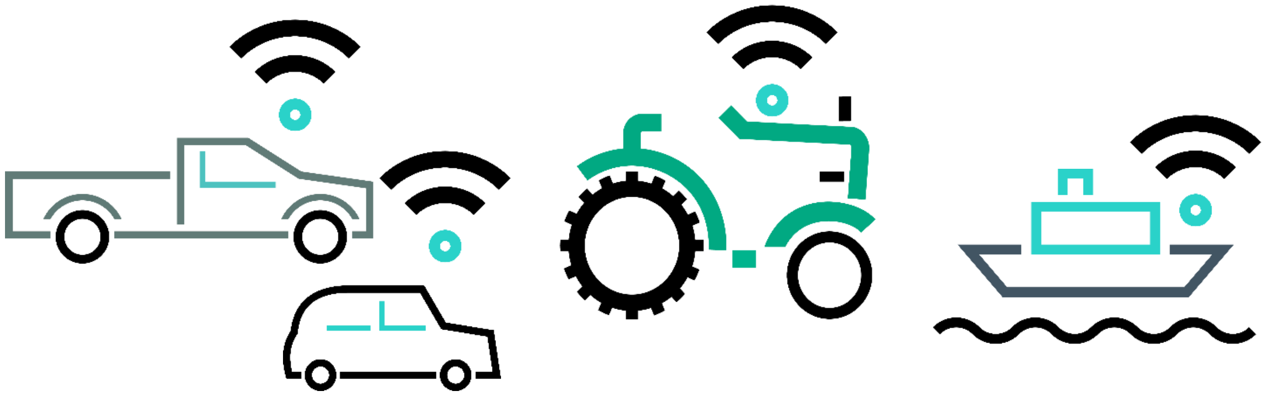
Device Layers (De-centralized & Autonomous Use Cases)

- In-place Processing
- Industry and Use Case Specific
- Can look similar to Edge or completely difference



HPE EL8000

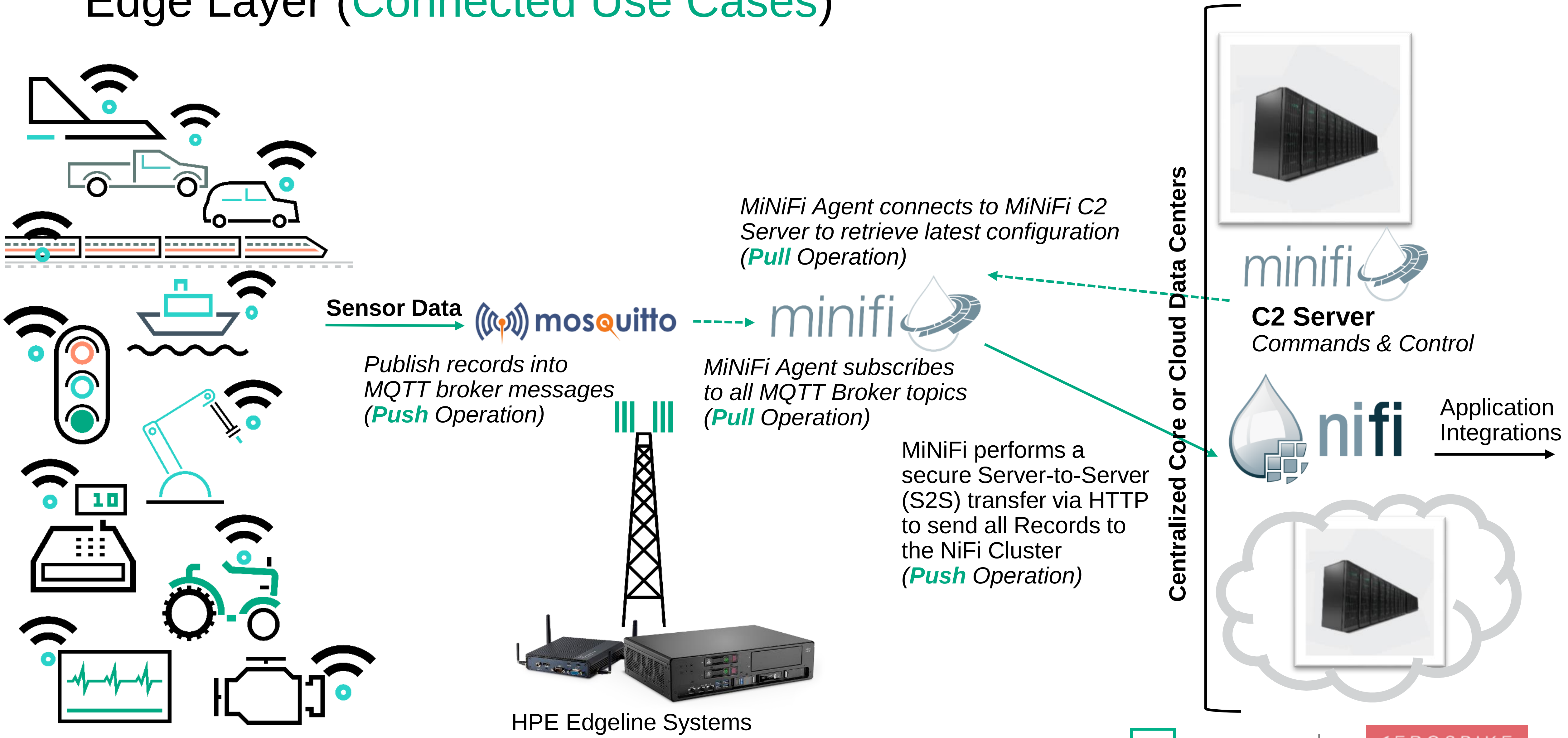
HPE EL300



Hewlett Packard Enterprise

AEROSPIKE SUMMIT '19

Edge Layer (Connected Use Cases)



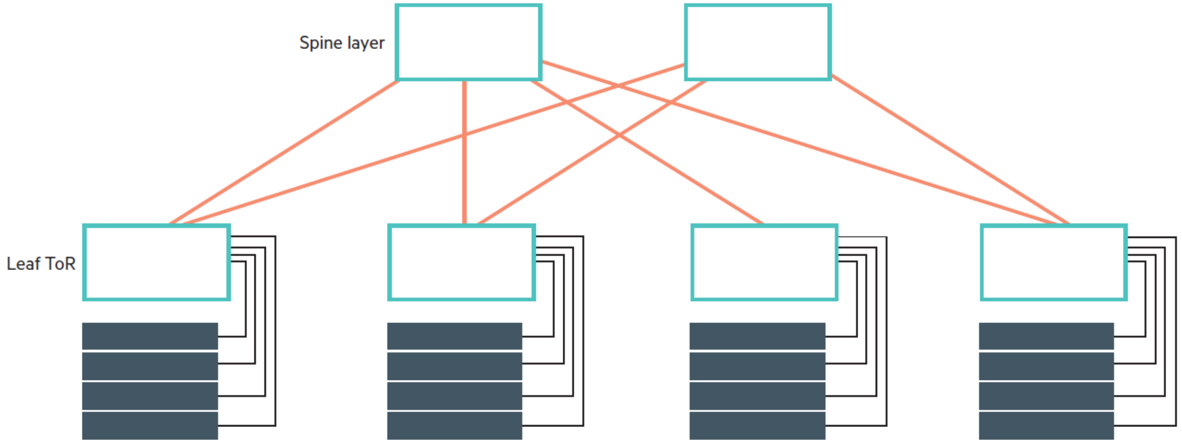
 | Proprietary & Confidential |

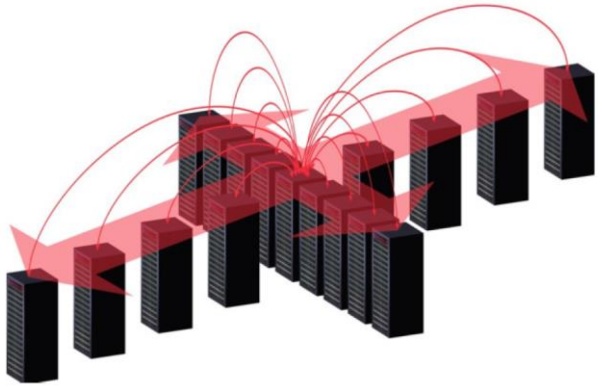
Hewlett Packard Enterprise

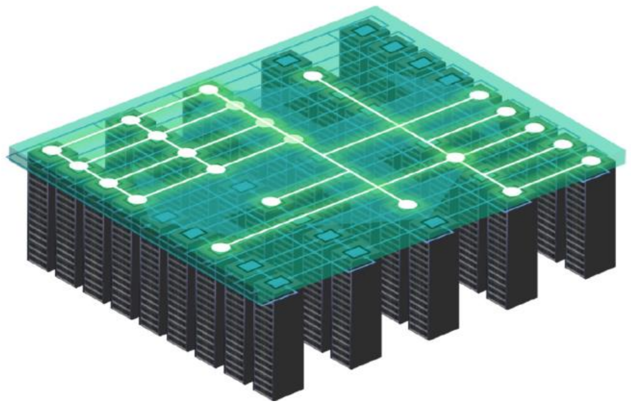
AEROSPIKE SUMMIT '19

Core and Cloud Network Fabric Layers

- **Network infrastructures @scale need to be**
  - Able to dynamically scale and evolve as load and requirements change
  - Simple enough for small teams to manage them
- **Cluster-network implementations have limitations**
  - Fabrics are disaggregated with balanced performance



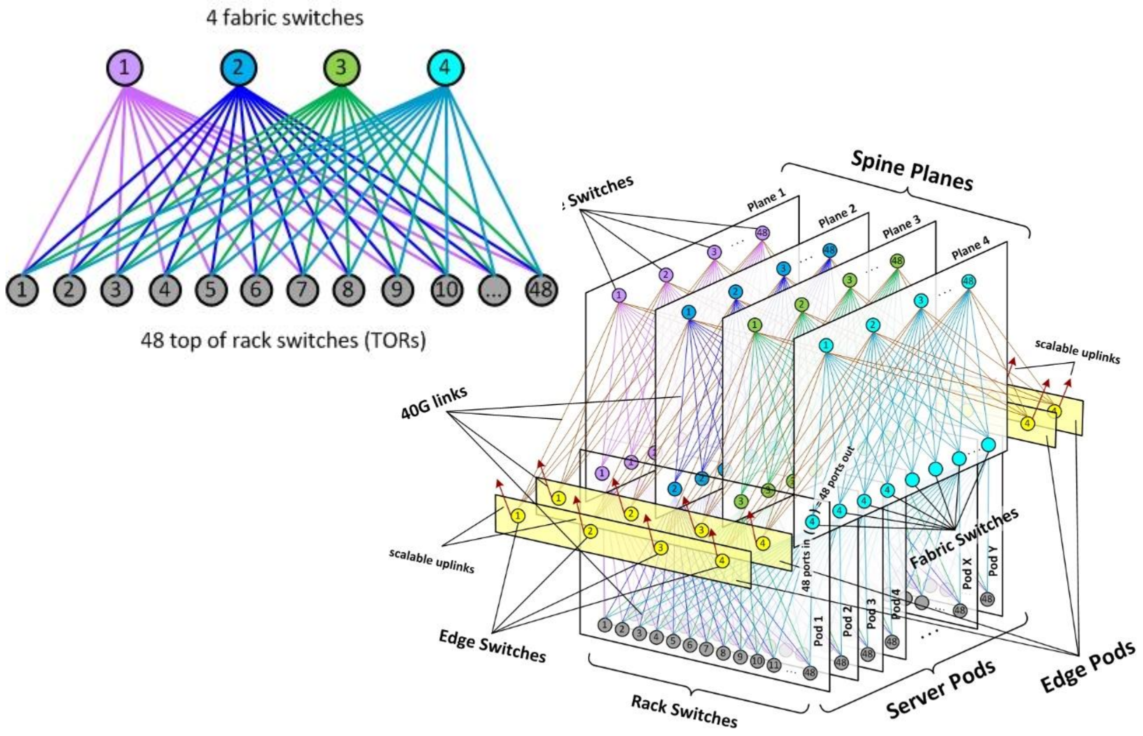




Images pulled from: https://h20195.www2.hpe.com/V2/getpdf.aspx/A00060583ENW.pdf?

Network Fabric Options

- **Open Source Network Fabric – Developed by Facebook**



Images pulled from: https://code.fb.com/production-engineering/introducing-data-center-fabric-the-next-generation-facebook-data-center-network/

- **HPE Composable Fabric – Formerly Plexxi**

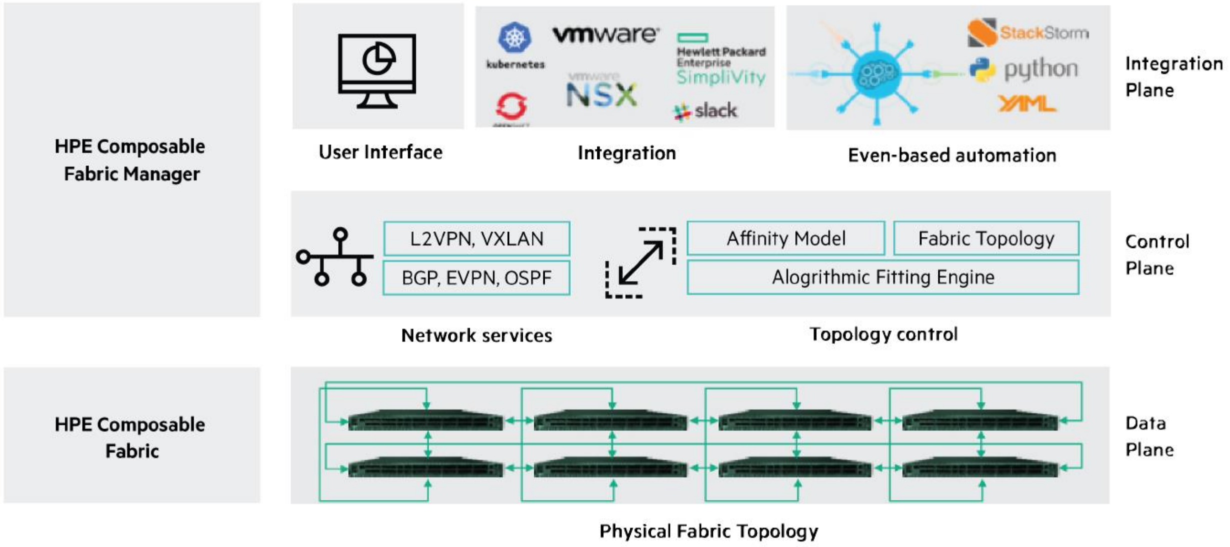


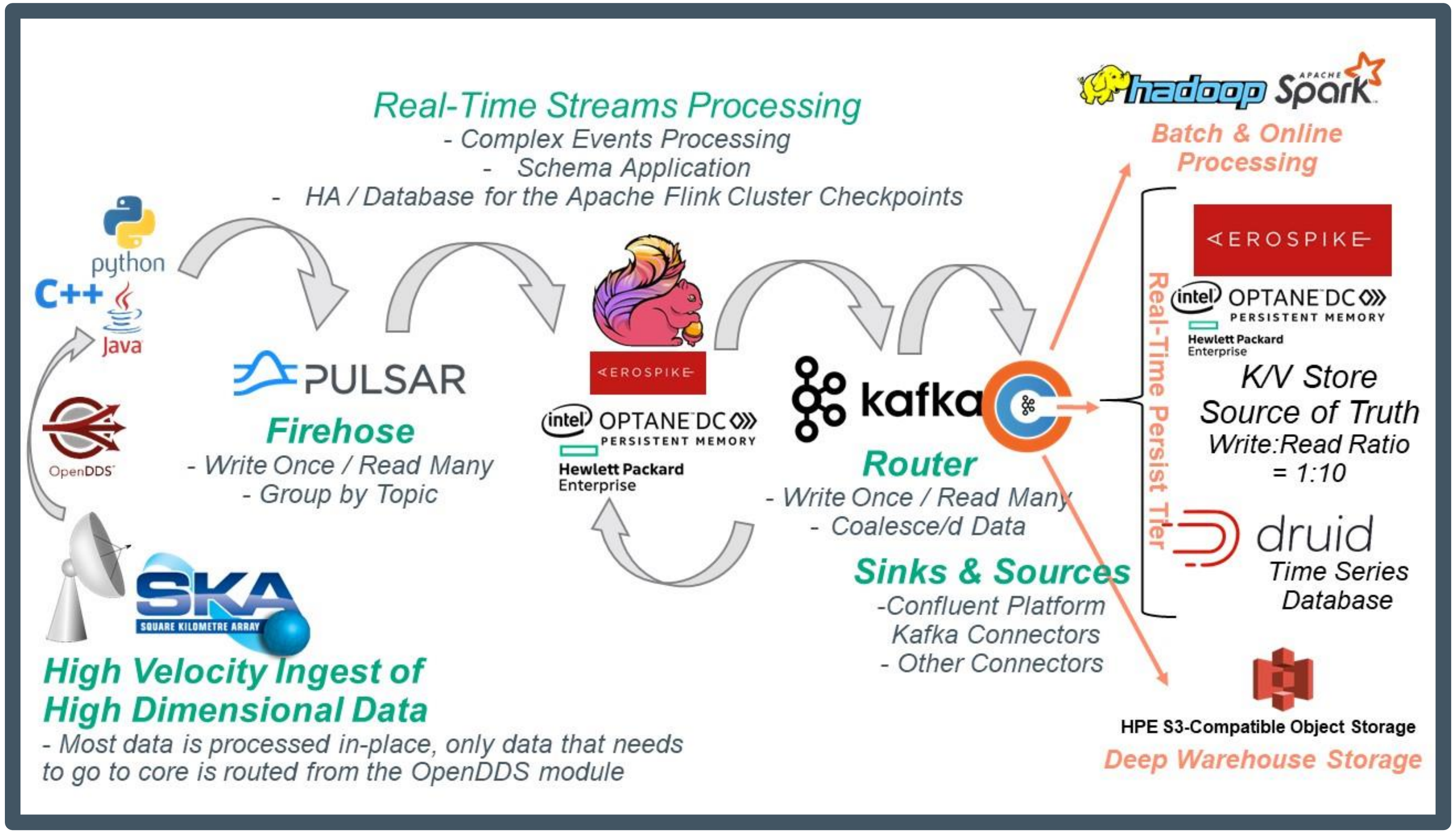
Image pulled from: https://h20195.www2.hpe.com/V2/getpdf.aspx/A00060583ENW.pdf?

Hewlett Packard Enterprise

AEROSPIKE SUMMIT '19

Core and Cloud Compute Fabric Layers

- **Real-Time Analysis**
  - Performed on live data streams
  - Model inference is also performed at this layer
- **Fast-persist is critical**
  - When using a single “source of truth”



 | Proprietary & Confidential |

Hewlett Packard Enterprise | AEROSPIKE SUMMIT ‘19

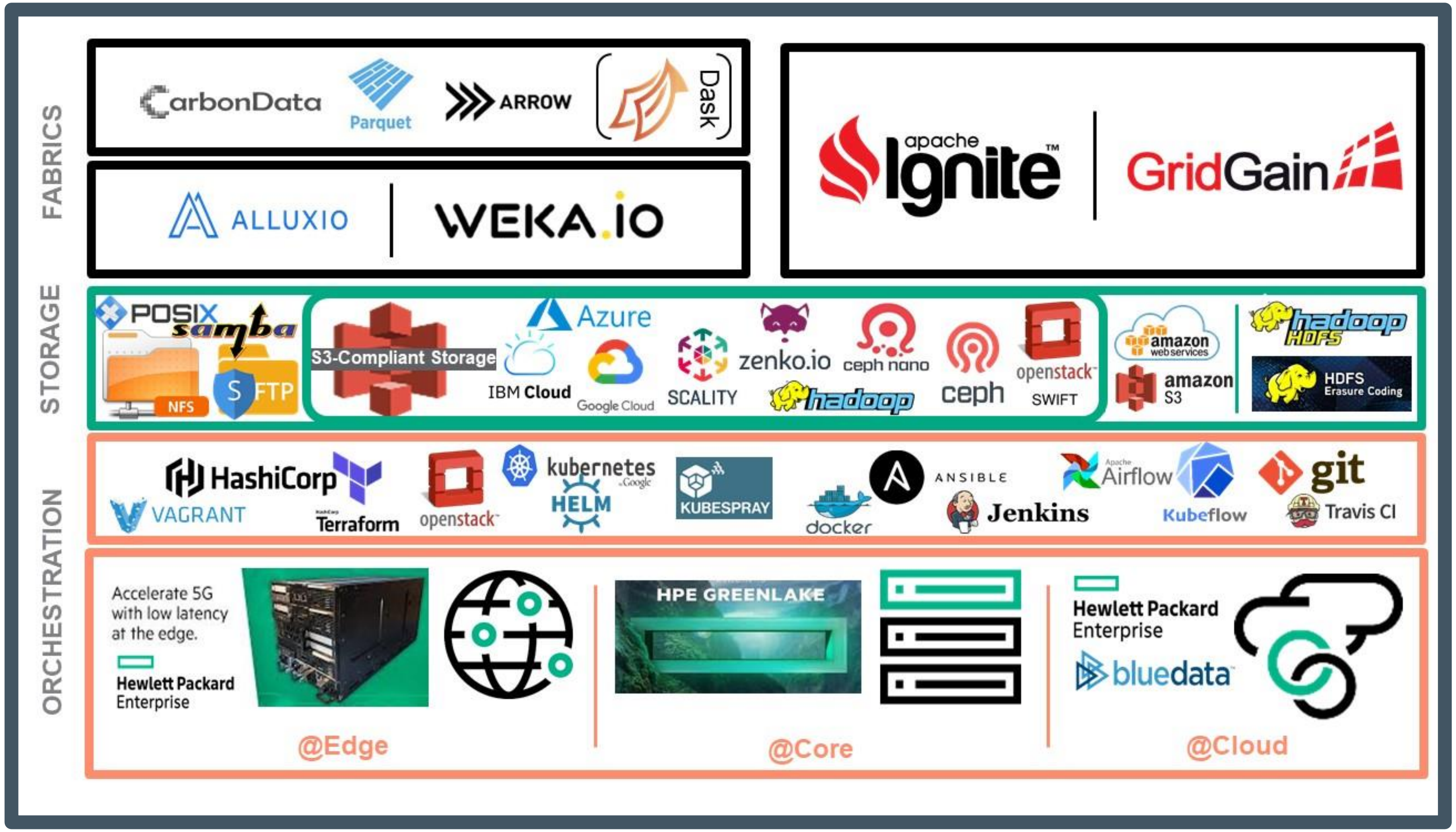
Core and Cloud Storage Fabric Layers

Core Data Centers

- Big Iron
- HPC Processing
- Deep Storage

Cloud Hosting

- Access Portals
- Transactional
- Bursting



Turning Data into an *Intelligent* Science

- **ML is Use Case Specific**
  - Astrophysics, Automotive, Finance, Trading, Healthcare, Aerospace, etc
- **Industry Specific Dev & Tools**
  - AstroML, OpenCV, Zipline, healthcareai-py, etc.
- **Domain Expertise needed for Fast ROI**
  - Astronomers, Physicists, Automotive & Aerospace Engineers, Bankers, Day-Traders, Doctors, Pilots, etc.
  - Developers – these best ones have worked in the field

Model

Training, Testing, Validation & Deployment

1

2

262	9	60	2	2	0	0	1	4	20
263	9	60	2	2	1	0	1	4	20
264	9	60	2	0	0	0	1	4	20
265	9	60	2	2	0	0	1	4	20
266	9	60	2	2	1	0	1	4	20

Astronomy Example: Self-Organizing Map (SOM) Neural Network

3

Image from: https://link.springer.com/article/10.1007/s11214-018-0489-2

Hewlett Packard Enterprise

AEROSPIKE SUMMIT '19

Now, let's talk SLAs and Solutions for our Real-Time Predictions

Requirements for Fronting Ingest Tier:

- **THROUGHPUT:**
  - 1TB/sec
    - Equals 1,000,000,000,000 bytes
- **PAYLOAD:**
  - Message Size = 1,000 bytes
- **LATENCY:**
  - Ack in 10ms

Everything matters here … the hardware, the software, the networking, the code, and everything in-between

Storage	Latency (ns)
HDD	10M
SSD (SAS)	100K
PCI NVMe	10K
PMEM	100 (+/-)
DRAM	10+
CPU Cache	0 (+/-)

Hewlett Packard Enterprise

AEROSPIKE SUMMIT '19

Intel Optane DC Persistent Memory

- **Co-exists with conventional DDR4 DRAM DIMMs**
  - DCPMM sits in Server DIMM Slots
- **Data persists after power-cycle**
  - Store indexes in pmem, allows for a warm database restart
- **Software can be modified to take advantage of this new tier in the memory hierarchy**
  - PMEM-aware filesystem manages access to persistent memory device
  - No buffering in DRAM
  - Kernel maps persistent memory to application address space
    - App now has direct access to persistent memory, so it can load and store data without the kernels involvement

- **Direct Access (DAX) Implementations**
  - FSDAX (/mnt/mem0)
  - DEVDAX (/dev/pmem0)
- **App-Direct Implementations**
  - C
  - C++
  - Java
  - Python

The Tested HPE Prototype Server Hardware (x3)

- **CPU**
  - 2 x 28-core **Cascade Lake** Procs
    - *CPU 0000%@ (fam: 06, model: 55, stepping: 05)*
    - CLX SP 28c 2.5GHz 205W
    - L1 = 1792 KB, L2 = 28672 KB, L3 = 39424 KB
- **Storage**
  - NVMe Controller
    - 3 x 1600GB SSD
  - HPE Smart Array P408i-a SR Gen10
    - 2 x 400 GB SSD
    - 1 x 480 GB SSD
- **Memory**
  - 2.6 GHz
  - Total = 1.75 TB (2x12 Slots)
    - RDIMM = 16.00 GB x 6
      - 96 GB * 2 = 192 GB Total
    - **DCPMM** = 126.38 GB x 6
      - 758.16 * 2 = 1516.32 GB

HPE Proliant DL380 Gen10 Server

HPE Persistent Memory for 2nd generation Intel® Xeon® Scalable processors

- **Network**
  - Adapter 1 / LOM
    - HPE Eth 10/25Gb 2p 640FLR-SFP28
  - Adapter 2 / 1GbE
    - HPE Ethernet 1Gb 4-port 331i Adapter
  - Adapter 3 / 100GbE
    - HPE InfiniBand EDR/Ethernet 100Gb 2-port
  - Adapter 4 / 10GbE
    - HPE Eth 10/25Gb 2p 621SFP28

Hewlett Packard Enterprise

AEROSPIKE SUMMIT '19

The Tested “PMEM-aware” NoSQL Database Implementations

- **DEVDAX**
  - Cassandra (Java)
    https://github.com/shyla226/cassandra/tree/13981_llpl_engine
- **FSDAX**
  - Aerospike Enterprise Server (C)
    Version 4.5.0.5-1
  - RocksDB (C++)
    https://github.com/pmem/rocksdb
  - Redis (C)
    https://github.com/pmem/pmem-redis
  - *Memcached (C)*
    https://github.com/lenovo/memcached-pmem
  - *MongoDB (C++)*
    https://github.com/pmem/pmse

MongoDB is a document store and Memcached is not used for persistence (in my work), so I excluded these two from the K/V store comparison tests

 | Proprietary & Confidential |

Hewlett Packard Enterprise

AEROSPIKE SUMMIT ‘19

The Test Dataset was *Yahoo! Cloud System Benchmark* (YCSB)

- **Core Workload A:**
  - **Update Heavy** Workload
    *This workload has a mix of **50/50 reads and writes**.*
    An application example is a session store recording recent actions.
    - **Load** Command (100% Inserts)

```
./bin/ycsb load [database] -s -threads 112 -P workloads/workloada \
-p "[database].hosts=[ip_address]" -p recordcount=500000000 \
> outputs/workloada_load_[database]_500m-112t.out \
2> outputs/workloada_load_[database]_500m-112t.err
```

    - **Run** Command (50% Read / 50% Updates)

```
./bin/ycsb run [database] -s -threads 112 -P workloads/workloada-bench \
-p "[database].hosts=[ip_address]" -p target=[X] -p maxexecutiontime=14400 \
> outputs/workloada_run_[database]_500m-112t.out \
2> outputs/workloada_run_[database]_500m-112t.err
```

      - workloada-bench file

```
recordcount=500000000
operationcount=500000000
```


Hewlett Packard Enterprise

AEROSPIKE SUMMIT '19

Actual Workloads Had to Vary for each Database

Aerospike

```
./bin/ycsb run \
aerospike -s \
-threads 112 \
-P workloads/workloada-bench \
-p as.host=10.20.100.65 \
-p as.user=admin \
-p as.user=admin \
-p as.namespace=ycsb \
-p target=250000 \
-p maxexecutiontime=14400 \
 > outputs/workloada-bench_4hr-run_aerospike_500_150.out \
2> outputs/workloada-bench_4hr-run_aerospike_500_150.err
```

***Reduced threads to 112 (1/cpu) to decrease the excessive server load 200 threads caused**

RocksDB

```
./bin/ycsb run \
rocksdb -s \
-threads 10 \
-P workloads/workloada-bench \
-p target=80000 \
-p maxexecutiontime=14400 \
-p rocksdb.dir=/mnt/mem/ycsb-rocksdb-data \
> outputs/workloada-bench_4hr-run_rocksdb_500_10.out \
2> outputs/workloada-bench_4hr-run_rocks_500_10.err
```

***Total threads could not exceed ~10, else ops/sec would decrease considerably**

Cassandra

```
./bin/ycsb run \
cassandra2-cql -s \
-threads 8 \
-P workloads/workloada-bench \
-p target=92000 \
-p hosts=10.20.100.66 \
-p user=cassandra \
-p password=cassandra \
-p as.namespace=ycsb \
-p maxexecutiontime=14400 \
> outputs/workloada_run_cassandra__4hr_500m-8t-15k.out \
2> outputs/workloada_run_cassandra_4hr_500m-8t-15k.err
```

***4 Hour Mark Reached, job automatically killed, only 251M of 500M Records Processed**

Redis

```
./bin/ycsb run \
redis -s \
-threads 10 \
-P workloads/workloada-bench \
-p redis.host=10.20.100.67  \
-p redis.port=6379 \
-p target=10000 \
-p maxexecutiontime=14400 \
> outputs/workloada-bench_4hr-run_redis_50_10.out \
2> outputs/workloada-bench_4hr-run_redis_50_10.err
```

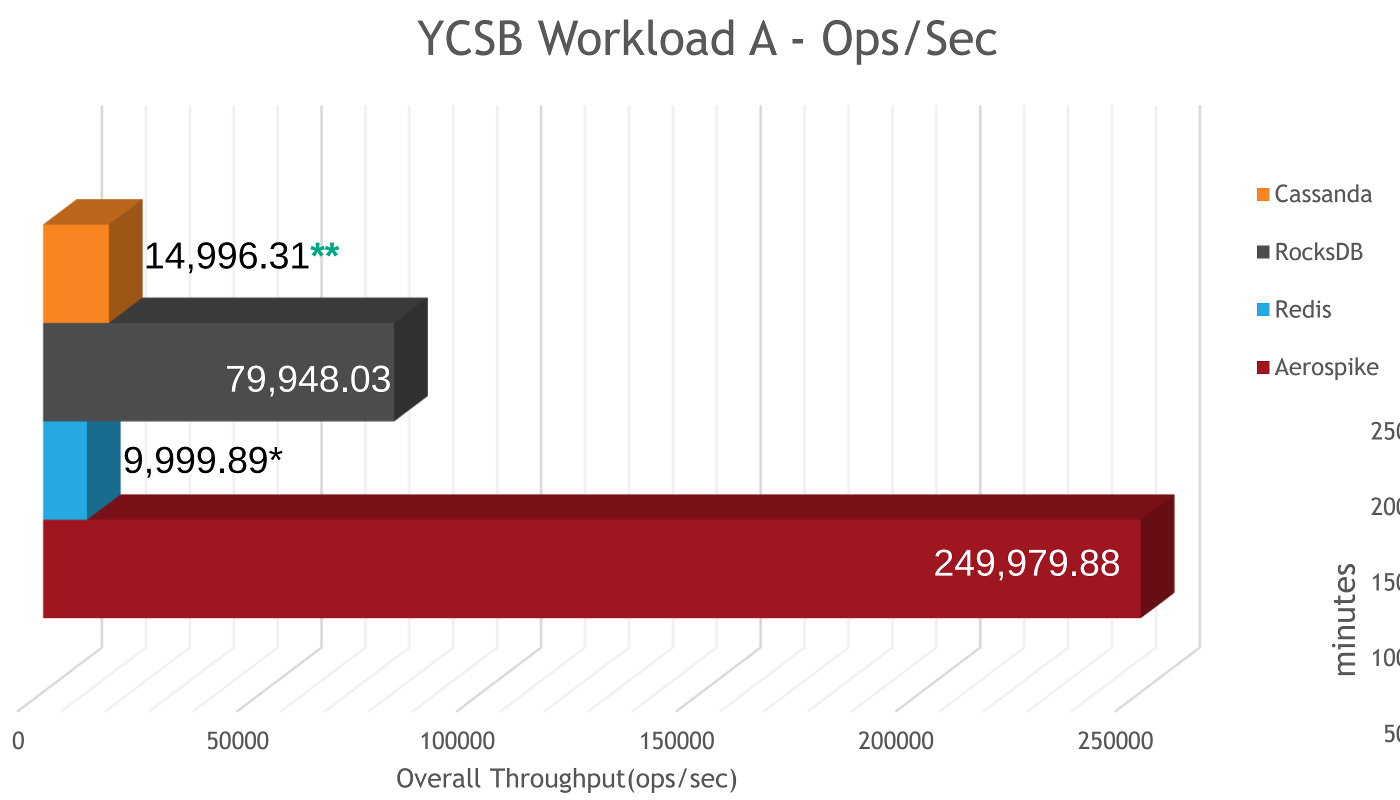
***Would have taken >2.5 days to load 500M records, reduced YCSB benchmark test to 50M records for Redis (only)**

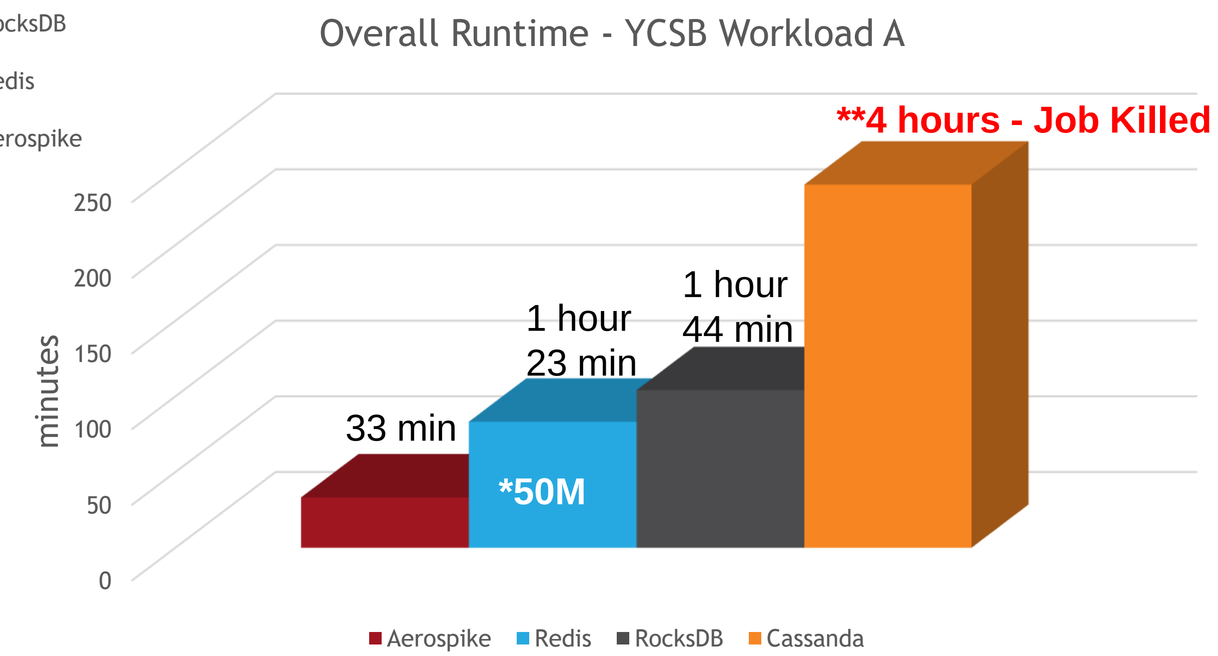
Workload A – Overall Runtime Results

500 Million 1 Kilobyte Records Processed (50r|50w)

**Redis could not 'load' 500M records in the required timeframe so 50M records was used for its test*

***Cassandra performance dropped 'off a cliff' on reads, writes could be sustained at >90K ops/sec*





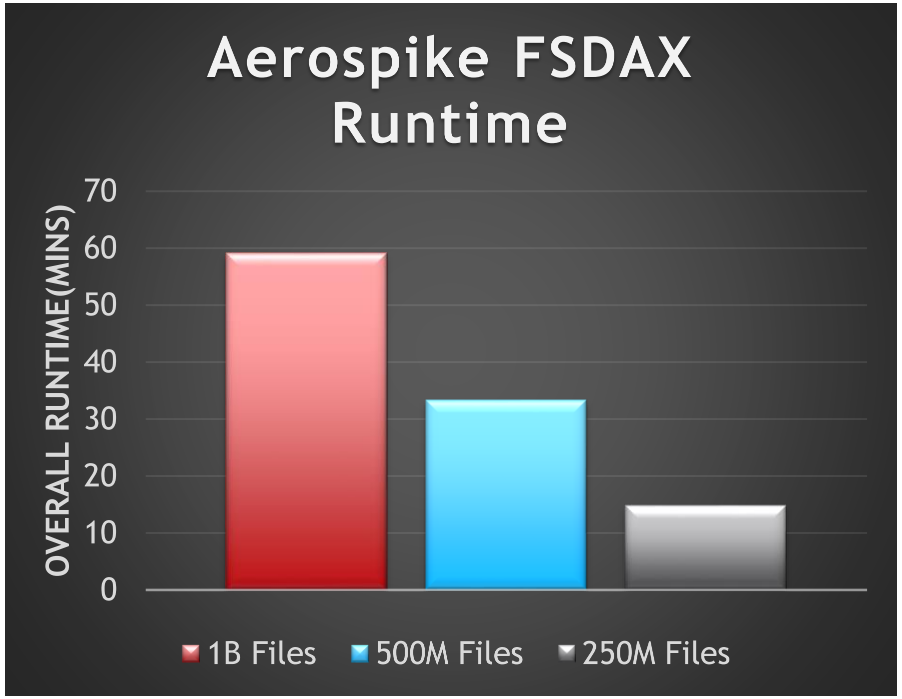
Hewlett Packard Enterprise

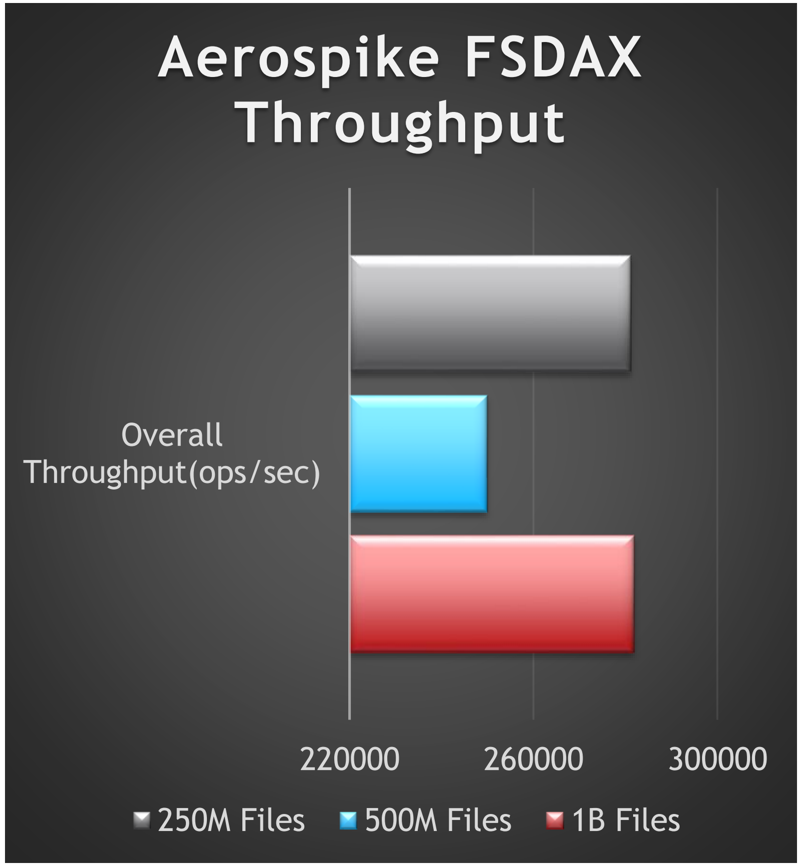
AEROSPIKE SUMMIT '19

Aerospike PMEM-Aware Throughput Testing

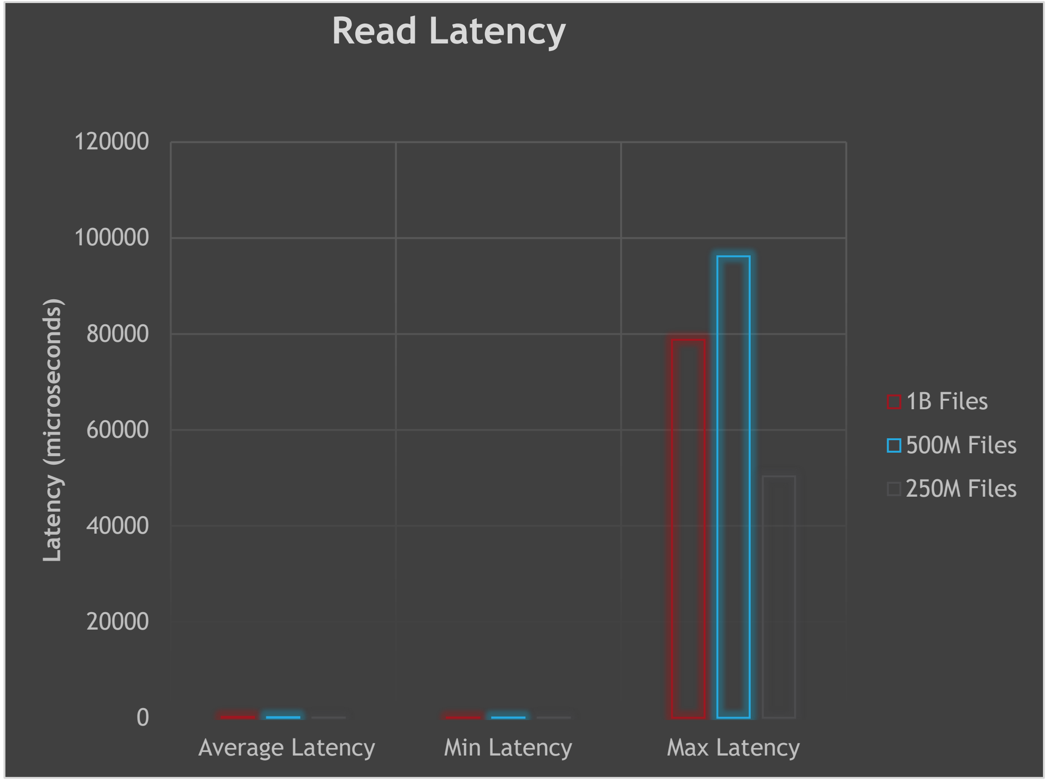
- **500M Records**
  - Threads were originally set to 200, but none of the other DBs in the Comparison Test could come close to this number so thread count was decreased to **112** for Aerospike, or 1 thread per CPU, server load was still ‘off the charts’
  - **Ops/Sec target=250000**
  - 172GB RAM for Aerospike
  - PMEM-devs used for Indexes Only
  - 3 x NVMe SSD disks used for data
- **1B Records**
  - Same as above, just doubled the load to try to make aerospike fail
  - **Ops/Sec target=282000**
  - Reduced thread count to **64** to reduce server load
- **250M Records**
  - Wrote to a single PMEM-dev (/mnt/mem0)
  - No data in memory
  - **Ops/Sec target=286000**
  - OOM errors with any thread count above **32**

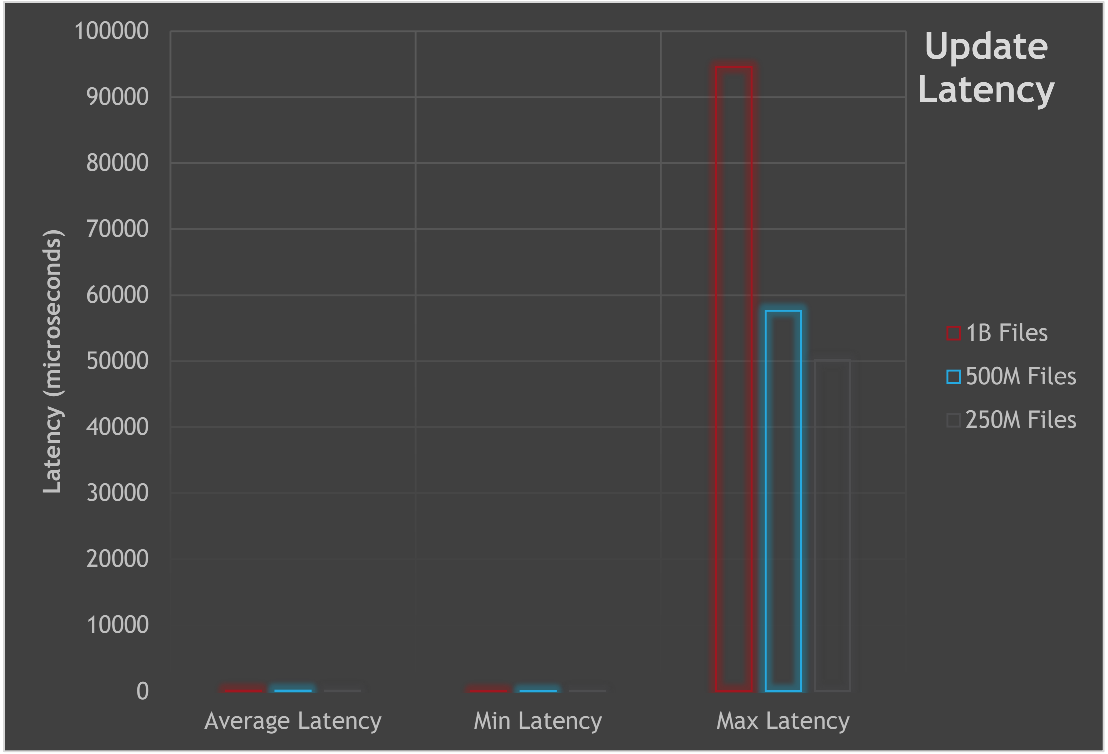
```
PMEM Devices (indexes)
/dev/pmem0        /mnt/mem0
/dev/pmem1        /mnt/mem1
NVMe Devices (data):
/dev/nvme0n1
/dev/nvme1n1
/dev/nvme2n1
```





Aerospike PMEM-Aware Latency Testing





Take Away

- **Aerospike's pmem-aware performance for a single node**
  - 300% increase over RockDB
  - Between 270% and 1667% over Cassandra's
    - 270% if Cassandra was able to sustain its 92K ops/sec
    - 1667% per Cassandra's current 15K sustained r/u performance
- **Aerospike's Cluster Footprint**
  - 2-nodes required for Strong Consistency
    - 33% footprint reduction over nearly any other K/V store
- **@250K msg/sec and with average latency @158µs it would take ~4K Aerospike instances to meet the throughput requirement of 1TB/s**
  - Excluding replication, compression and server load

Hewlett Packard Enterprise

AEROSPIKE SUMMIT '19

Thank you!

theresa.melvin@hpe.com

Hewlett Packard Enterprise

