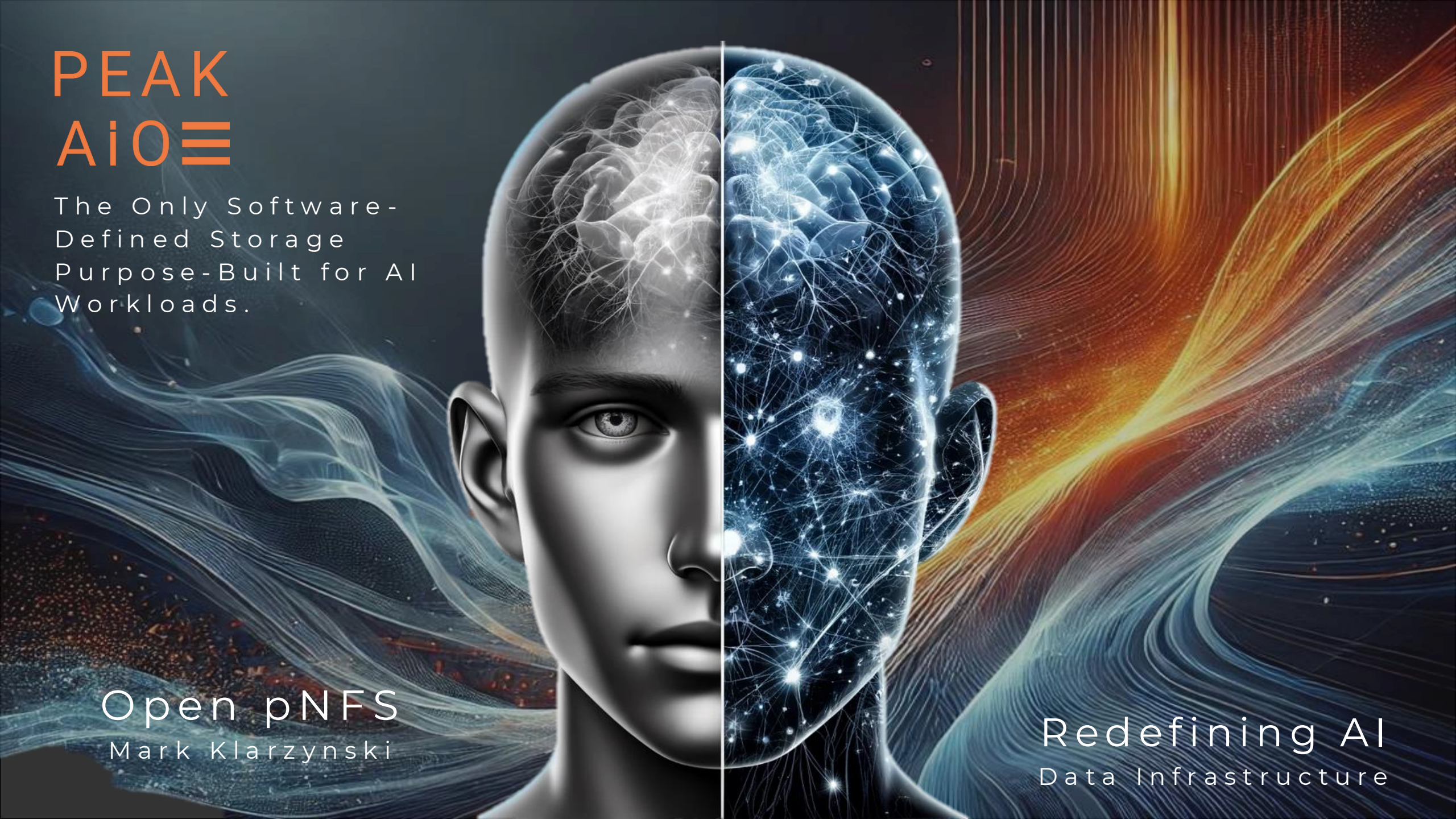

PEAK
AIOE
The Only Software-Defined Storage Purpose-Built for AI Workloads.
Open pNFS
Mark Klarzynski
Redefining AI
Data Infrastructure

Agenda

01	Who are PEAK:AIO, what we do today, why pNFS
02	Why we chose the path we have
03	What do we have today
04	Obvious additional features
05	Advanced features
06	Come join us

PEAK:AIO – Executive Summary

PEAK AIO

Decades in storage and software defined storage from SCSI to NVMeoF

Taught us the need to simplify complex solutions, prove predictable numbers, and keep pushing innovation

AI Created Change

AI put a rack of compute inside one server.

Teams needed speed, low latency, and low effort.

Not a science project.

Our Key Focus

Learn first.

Make it simple.

Focus on performance, simplicity, space and energy efficiency.

What we Built

A lean RDMA NFS stack that reaches 120GB/sec per 2U.

Stable tail latency.

Standard Linux clients.

How we Ship

Software only.
Click and go.
Hardware independent
Designed for none-IT users

A Data Server Rocket

This became our foundation – The Building Block.
Next, how we scale it.

PEAK:AIO Layers

Intelligent and data aware modules

A range of advanced use cases under development

Soon S3 RDMA / TCP

An emerging AI performance S3 with GPUDirect support and scale out

NVMe-oF RDMA /TCP

GPUDirect NVMe-oF, delivers the blisteringly fast block IO for intensive analytics workloads..

NFS RDMA / TCP

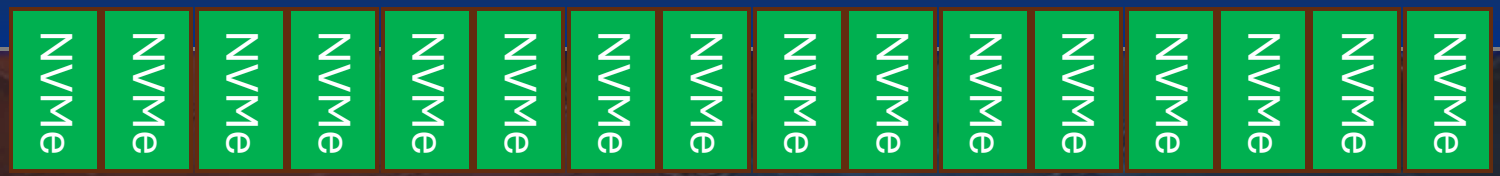
With GPUDirect, provides files based ultra-performance for shared datasets with full compatibility

Virtualisation Layer

Create volumes to suit

PEAK:PROTECT (RAID Layer)

Multi-stream Reed-Solomon erasure coding (N+2)



Pometry KVM / Docker

Direct user access to Pometry

Pometry

Micron Specific APIs – Redfish / Other

Trigger data clone via FamFS API

NVMe-oF RDMA /TCP

GPUDirect NVMe-oF, delivers the blisteringly fast block IO for intensive analytics workloads..

NFS RDMA / TCP

With GPUDirect, provides files based ultra-performance for shared datasets with full compatibility

Virtualisation Layer

Create volumes to suit

FamFS

XFS

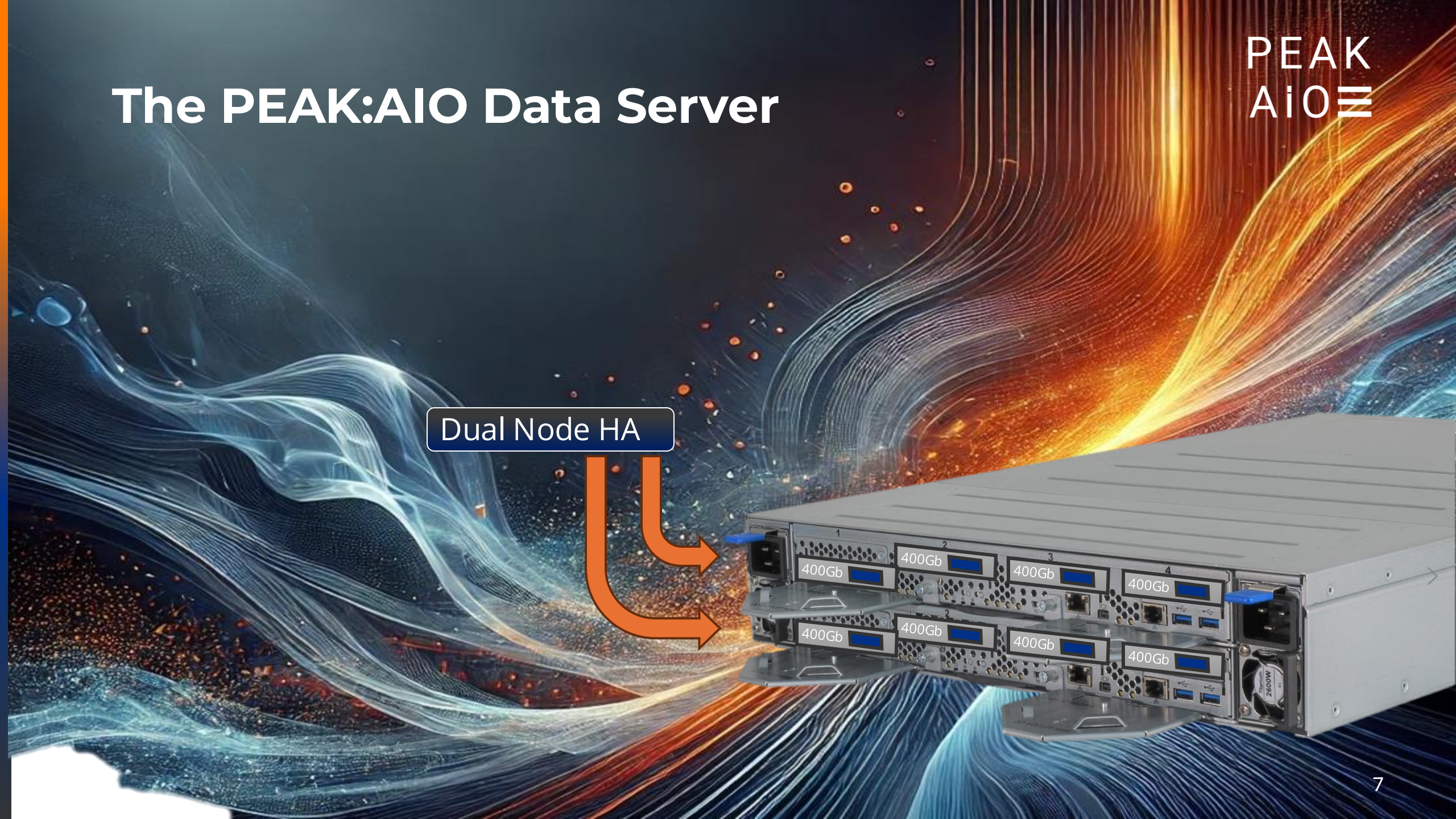
CXL CXL CXL CXL

Clone

NVMe NVMe NVMe NVMe NVMe NVMe NVMe NVMe NVMe NVMe NVMe NVMe NVMe NVMe NVMe NVMe

The PEAK:AIO Data Server

The PEAK:AIO Data Server



The PEAK:AIO Data Server

PEAK AIO

Perfect Flex Files Data Server

Active Active

160GB/sec 160GB/sec

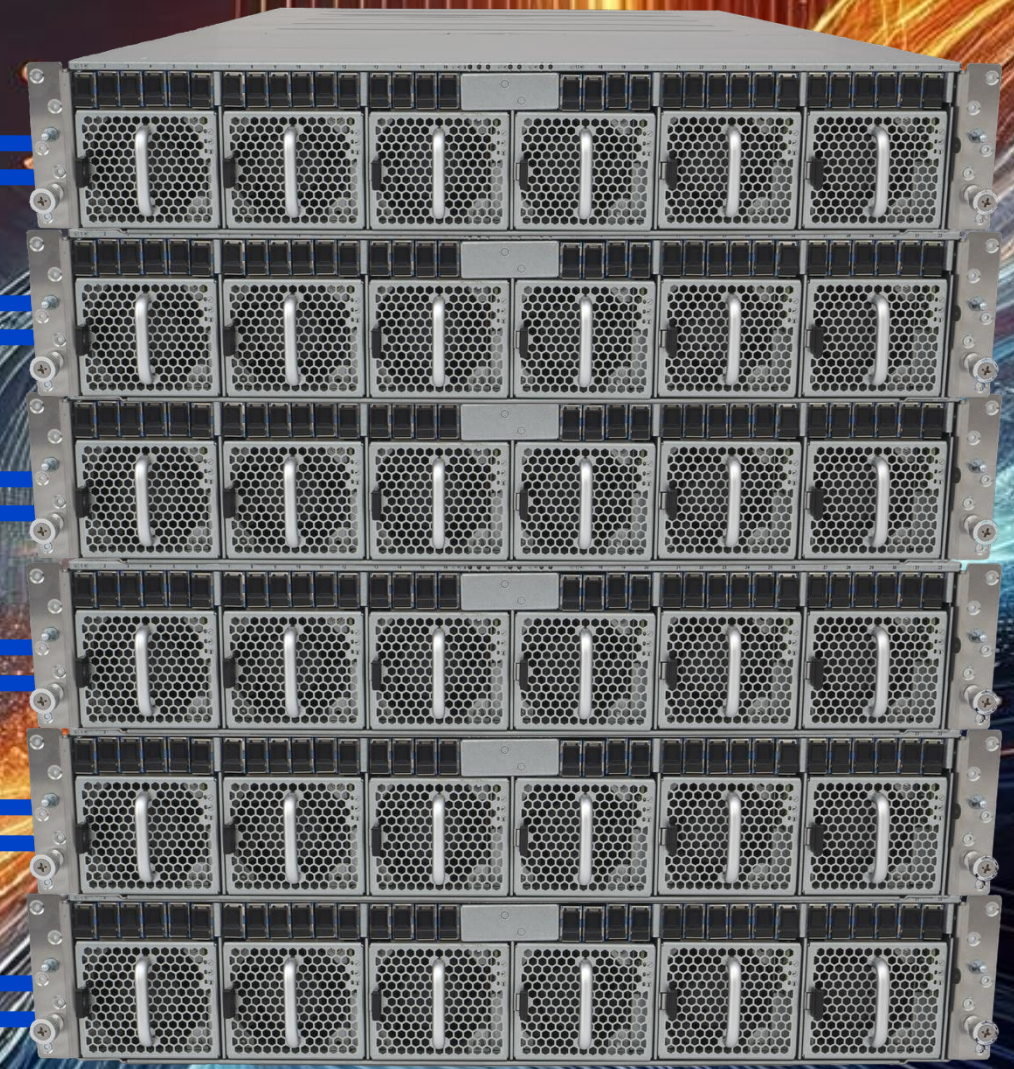
Perfect fit for pNFS – Flex Files

PEAK AIO

All about this!

pNFS Meta Data Server

Single Namespace



Workloads?

PEAK AIO

All about this!

pNFS Meta Data Server

Development Options

Challenges

- HPC Workloads
- AI Workloads
- Enterprise

Meta Data Server Choice

PEAK AIO

All about this!

pNFS Meta Data Server

Development Options

- Linux - NFSD
- Ganesha
- Others
- FreeBSD

Rick Macklem

FreeBSD – PEAK:AIO Changes

PEAK AIO

pNFS Meta Data Server

PEAK:AIO Changes

- Loosely Coupled
- NFSv3 MDS > DS
- Compatible with Linux
- Performance x-Fold
- RDMA DSs
- Miscellaneous / tunables
  - Force GETATTR
  - Return Layout on Close

PEAK:AIO Open pNFS MDS

PEAK AIO

PEAK:AIO Open pNFS

- Flex Files Support
- Scaled Bandwidth
- DS Grouping Support
- Replica / Mirror Support
- Trunking / LACP
- Low CPU use
- Physical / KVM on Linux host
- OPEN SOURCE!

pNFS Meta Data Server

PEAK:AIO Open pNFS MDS

PEAK AIO

PEAK:AIO Open pNFS

- TOOLS
  - Display xattrs
  - Mirror / Copy File Manager
  - DS / Mirror Kill

pNFS Meta Data Server

Extensions – to-do, community?

PEAK AIO

Larger scale testing LANL

pNFS Metadata Server

Extensions – to-do, community?

PEAK AIO

L: logical offset within the file

W: stripe width
W = number of elements in ffm_data_servers

S: number of bytes in a stripe
S = W * ffl_stripe_unit

N: stripe number N = L / S

Larger scale testing LANL

Striping – Flex Files

pNFS Meta Data Server

Extensions – to-do, community?

PEAK AIO

Larger scale testing LANL

Striping – Flex Files

Weighted Mirrors

ffds_efficiency

pNFS Meta Data Server

Extensions – to-do, community?

PEAK AIO

- Larger scale testing LANL
- Striping – Flex Files
- Weighted Mirrors
- RFC 9766 (with LAYOUT_WCC)

GETATTR

pNFS Meta Data Server

Extensions – to-do, community?

PEAK AIO

Files Zero Size

```
-rw-r--r--  1 nobody wheel 0 Sep 15 12:49 r0-f0
-rw-r--r--  1 nobody wheel 0 Sep 15 12:49 r0-f1
-rw-r--r--  1 nobody wheel 0 Sep 15 12:49 r0-f10
-rw-r--r--  1 nobody wheel 0 Sep 15 12:49 r0-f100
-rw-r--r--  1 nobody wheel 0 Sep 15 12:49 r0-f101
-rw-r--r--  1 nobody wheel 0 Sep 15 12:49 r0-f102
-rw-r--r--  1 nobody wheel 0 Sep 15 12:49 r0-f103
-rw-r--r--  1 nobody wheel 0 Sep 15 12:49 r0-f104
-rw-r--r--  1 nobody wheel 0 Sep 15 12:49 r0-f105
```

BIG ONES

KVS v FS with xattr

Extended attributes on the MDS (KV)

```
r0-f0    pnfsd.dsfile    pnfsd.dsattr
```

Raw binary xattr as printed by tool

```
r0-f0    ◆◆2◆◆◆Fk◆d◆}◆◆)p◆◆T◆"◆dzafb9c30fdec4d2030a00b81c00000000d89f010000000000000000
```

```
r0-f0:  192.168.100.122 ds27/afb9c30fdec4d2030a00b81c00000000d89f010000000000000000
```

DS location/name

pNFS Meta Data Server

Workload Dependent?

Read mostly? Heavy writes?

Extensions – to-do, community?

PEAK AIO

Files Zero Size

```
-rw-r--r--  1 nobody wheel 0 Sep 15 12:49 r0-f0
-rw-r--r--  1 nobody wheel 0 Sep 15 12:49 r0-f1
-rw-r--r--  1 nobody wheel 0 Sep 15 12:49 r0-f10
-rw-r--r--  1 nobody wheel 0 Sep 15 12:49 r0-f100
-rw-r--r--  1 nobody wheel 0 Sep 15 12:49 r0-f101
-rw-r--r--  1 nobody wheel 0 Sep 15 12:49 r0-f102
-rw-r--r--  1 nobody wheel 0 Sep 15 12:49 r0-f103
-rw-r--r--  1 nobody wheel 0 Sep 15 12:49 r0-f104
-rw-r--r--  1 nobody wheel 0 Sep 15 12:49 r0-f105
```

BIG ONES

KVS v FS with xattrs

Extended attributes on the MDS (KVs)

```
r0-f0    pnfsd.dsfile    pnfsd.dsattr
```

Raw binary xattr as printed by tool

```
r0-f0    ◆◆2◆◆◆Fk◆d◆}◆◆)p◆◆T◆"◆dzafb9c30fdec4d2030a00b81c00000000d89f010000000000000000
```

```
r0-f0:  192.168.100.122 ds27/afb9c30fdec4d2030a00b81c00000000d89f010000000000000000
```

DS location/name

pNFS Meta Data Server

Currently moving MD to memory

How much impact will this have
Will it move the problem from FS to CPU/RAM

Extensions – to-do, community?

PEAK AIO

What does tomorrows teach bring?

- **KVS - no rigid tree sharding**
- **Multiple 1.6Tb/sec per MDS**
- **Large CPU / GPU resource**
- **CXL**

BIG ONES

KVS v FS with xattr

Sharding MDS

Extensions – to-do, community?

PEAK AIO

What does tomorrow's teach bring?

- AI assisted metadata placement
  - Analyze live MD access patterns
  - Congestion
  - Predictively rebalance
  - Rather than waiting for hotspots

AI could drive dynamic sharding policies

BIG ONES

KVS v FS with xattr

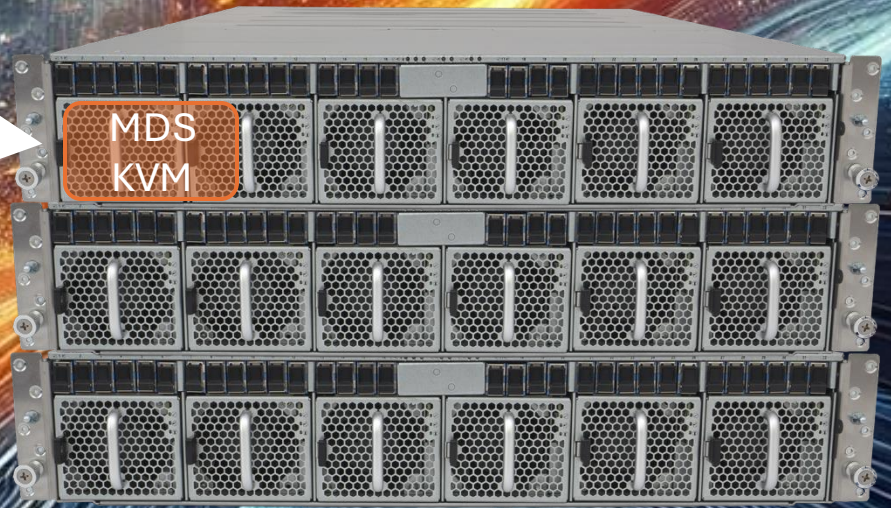
Sharding MDS

Extensions – to-do, community?

PEAK AIO

What if we stayed with a rigid tree strut

- BIG ONES
- KVS v FS with xattr
- Sharding MDS



Extensions – to-do, community?

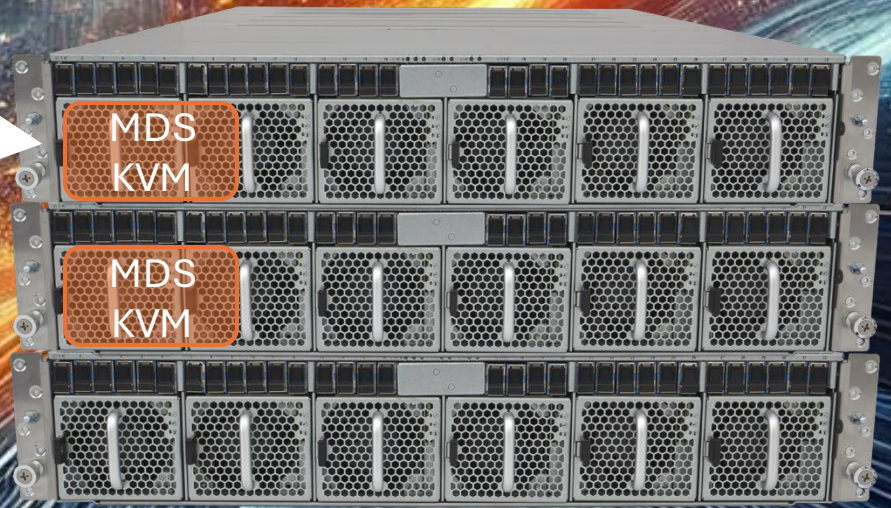
PEAK AIO

What if we stayed with a rigid tree strut

Can we use NFS4 referrals?

./c becomes hot

- BIG ONES
- KVS v FS with xattr
- Sharding MDS



RFC 8881 §11 (multi-server namespace: NFS4ERR_MOVED, fs_locations(_info)); §12 (pNFS: parallel data; **no striped metadata** — single MDS per filesystem).

Extensions – to-do, community?

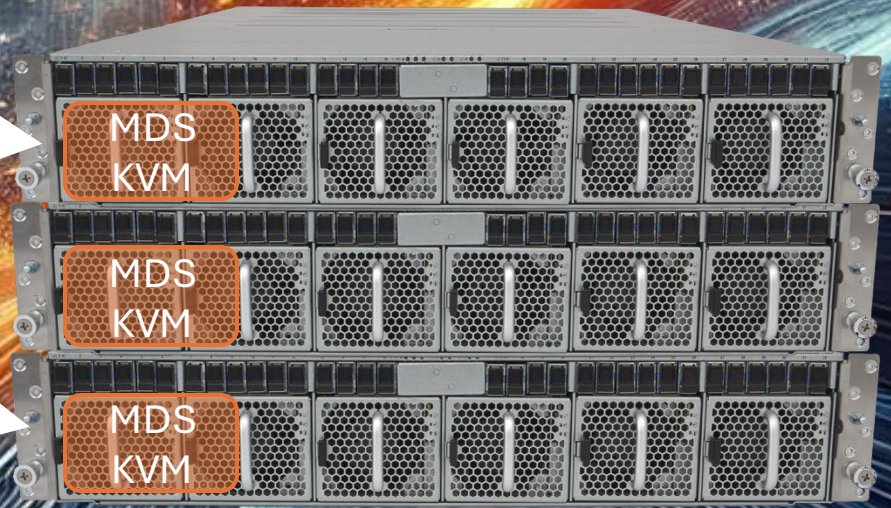
PEAK AIO

What if we stayed with a rigid tree strut

Can we use NFS4 referrals?

./c becomes hot

- BIG ONES
- KVS v FS with xattr
- Sharding MDS



RFC 8881 §11 (multi-server namespace: NFS4ERR_MOVED, fs_locations(_info)); §12 (pNFS: parallel data; **no striped metadata** — single MDS per filesystem).

Summary – Come Join Us

PEAK AIO

- Flex Files Proven
- Linearly scalable bandwidth
- Replica / Mirror Support
- Physical / KVM on Linux host
- Larger scale testing LANL
- OPEN

- Features – Striping
- KVS Ideas - Sharding Ideas
- Orchestration - Pushdown
- Help pNFS standardize
- Even if used as a Sandbox
- Open in return

Thank You

mark.klarzynski@peakaio.com