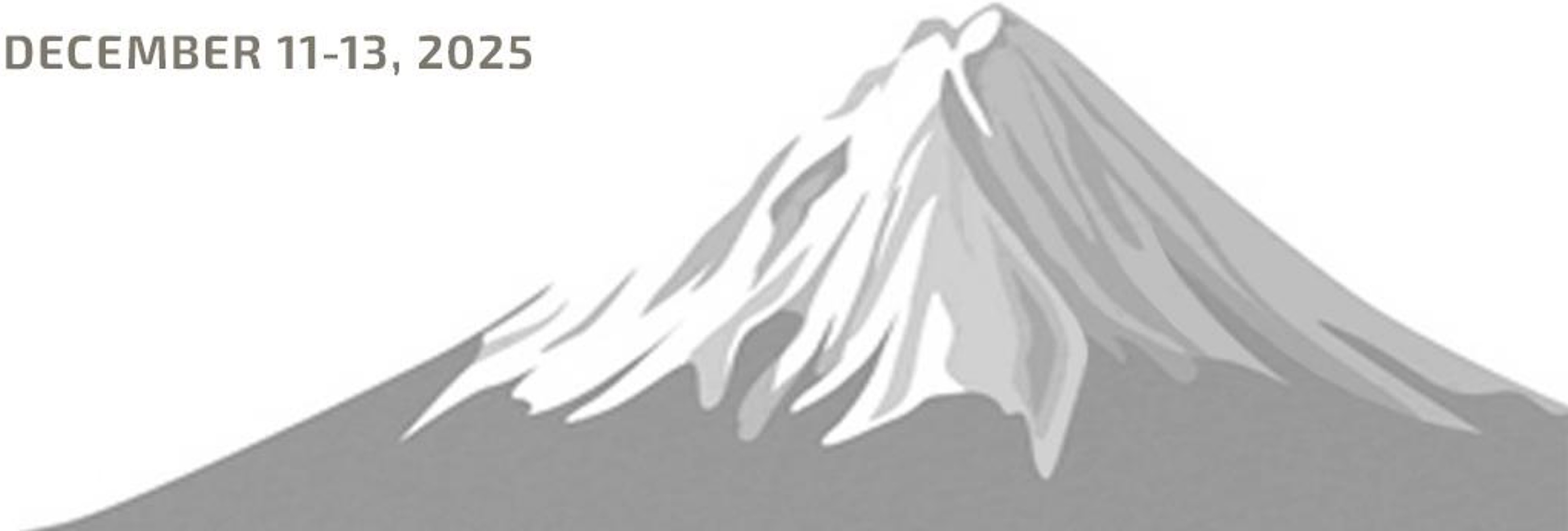
東京 2025

LINUX PLUMBERS CONFERENCE

TOKYO, JAPAN / DECEMBER 11-13, 2025



Famfs Update: Status, DAX Challenges & Use Cases

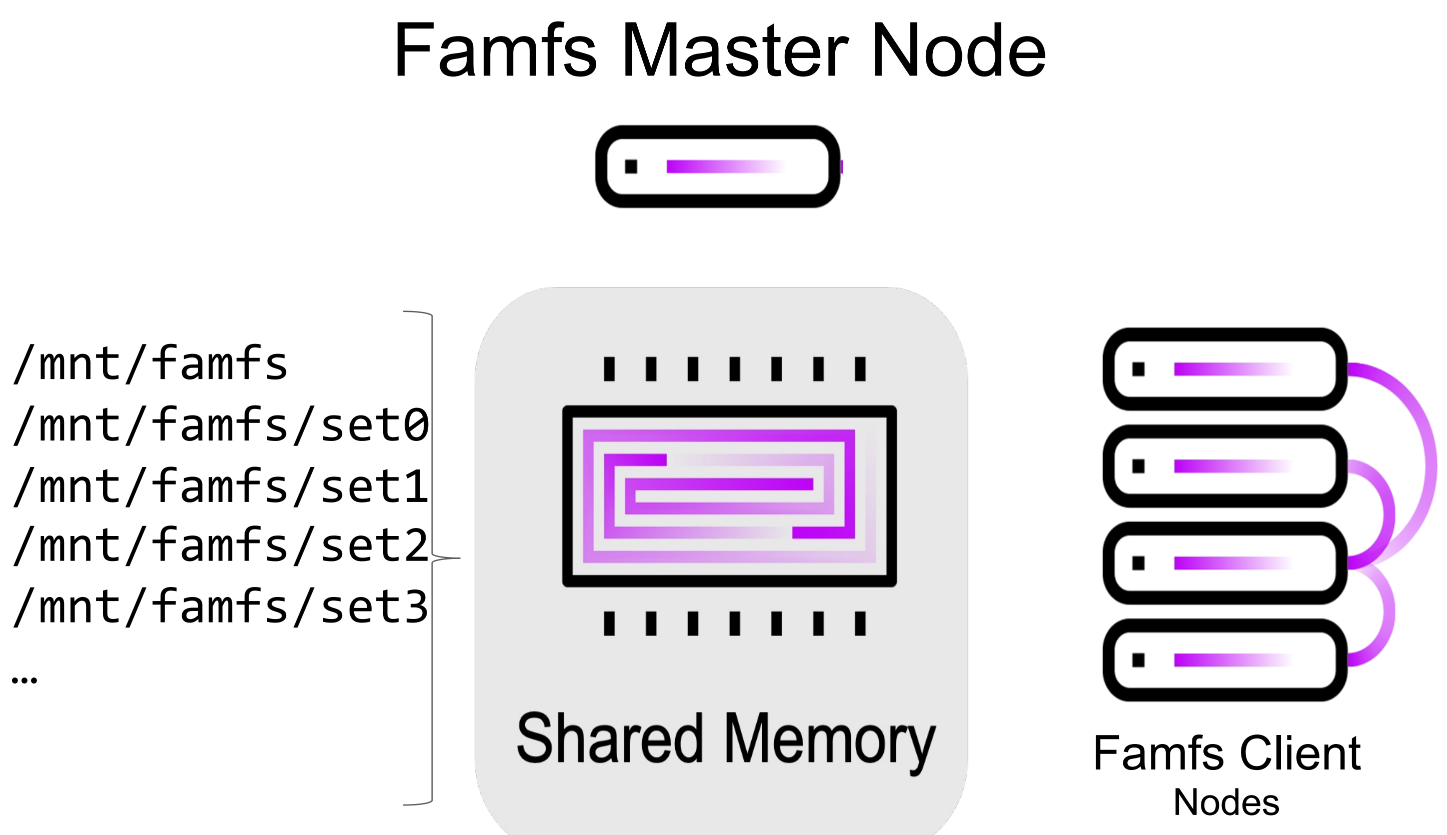
John Groves, Micron
Co-chair, CXL Consortium Software & Systems WG
jgroves@micron.com / john@groves.net

東京 2025
LINUX
PLUMBERS CONFERENCE
TOKYO, JAPAN / DEC. 11-13, 2025

Famfs: Background

The Fabric-Attached Memory File System

- Disaggregated shared memory needs an access method
  - Linux can’t online memory that is shared with other linux
  - Shared memory surfaces as DAX devices
- Famfs formats /dev/dax devices as file systems
  - Multiple hosts can mount a single famfs instance
  - Memory-mapped files provide byte-level access
  - Read/Write are `memcpy()`
- Core insight: files are a natural abstraction for data in shared memory
- First patches published February 2024



Famfs Master Node

```
/mnt/famfs
/mnt/famfs/set0
/mnt/famfs/set1
/mnt/famfs/set2
/mnt/famfs/set3
…
```

Shared Memory

Famfs Client
Nodes

```
mkfs.famfs /dev/dax0.0
famfs mount /dev/dax0.0 /mnt/famfs
famfs cp [-r] <src> <dest>
famfs creat -s <size> <dest>
```

Famfs: Upstreaming Status

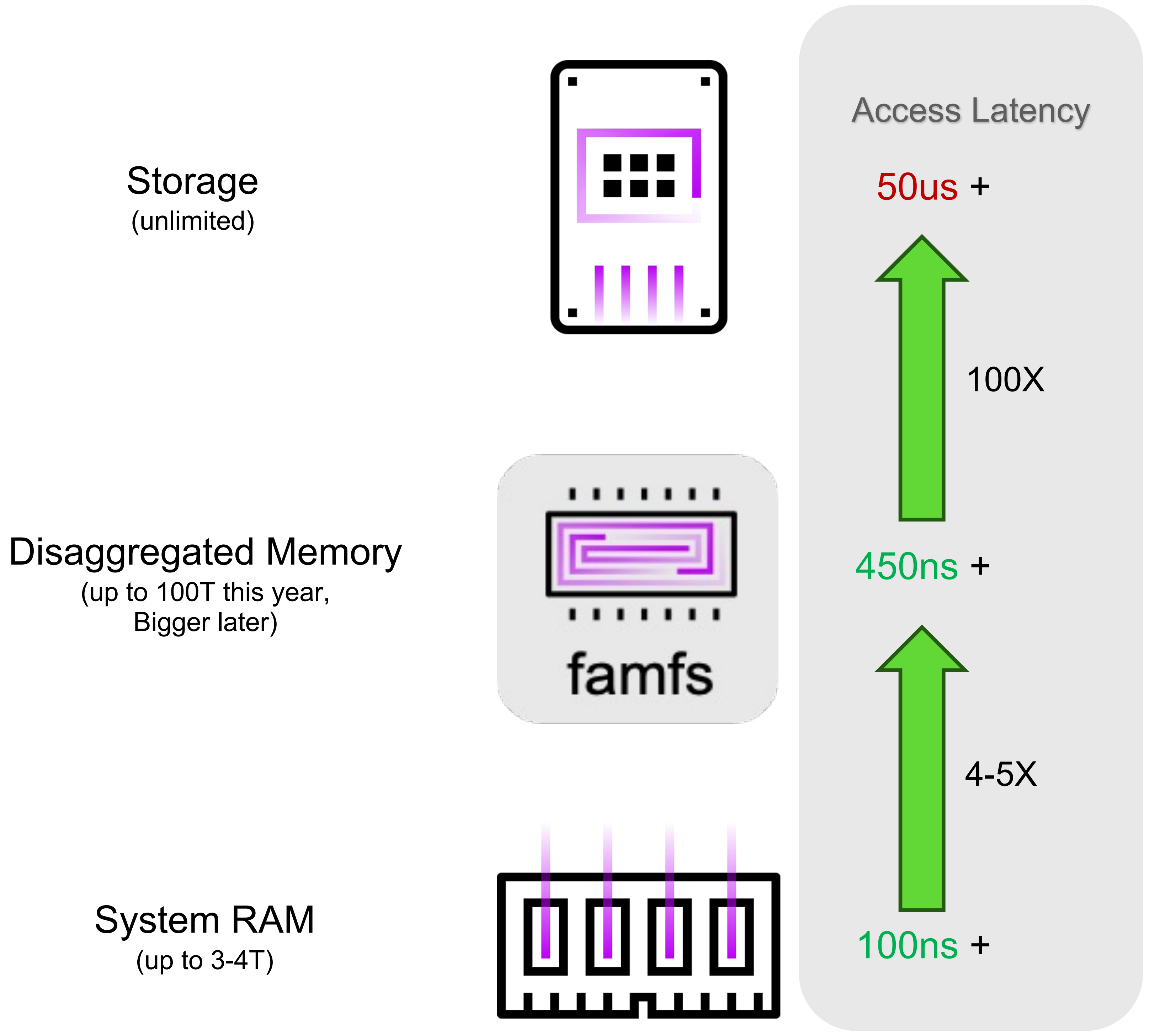
Goal: Upstream in early-ish 2026

- Famfs has been ported into fuse
  - Entire file-to-dax maps are cached in-kernel for all open files
- Dax challenges
  - Famfs is the first file system to reside on devdax (i.e. non-pmem dax)
  - Alistair tightened up dax exception checking starting in 6.15, stalling famfs
  - Famfs is now (finally!) fully working in 6.18
  - Next patch set will introduce a new /dev/dax mode: ‘famfs’ (or maybe ‘fsdev’)
  - Will comply with `dax_break_layout_final()`
- Next version: should drop the “RFC” tag…

Famfs: Use Cases

Random-access data (e.g. graphs) that is too big for regular memory

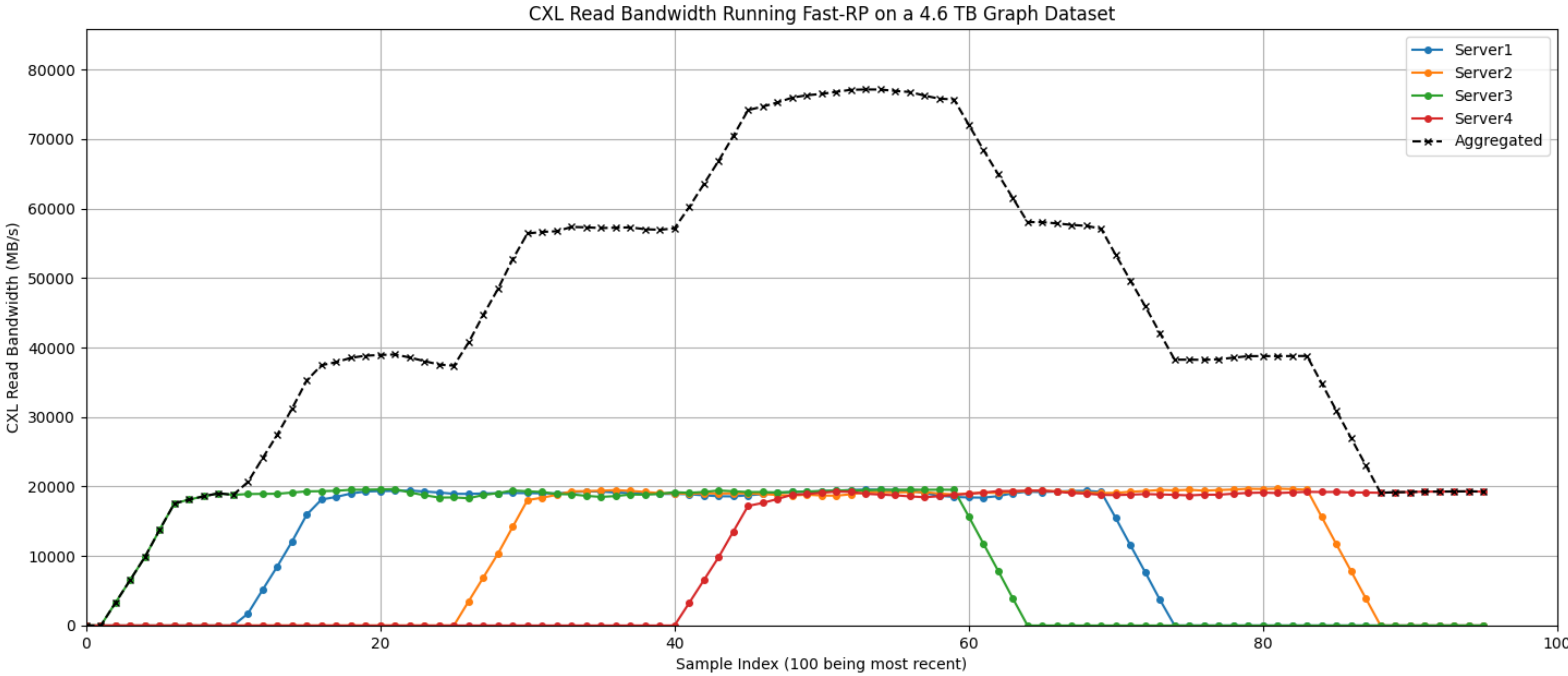
- The superpower of Memory is low-latency random access (non-prefetchable)
- Compare disaggregated memory to storage, not system-ram
- Data that doesn’t fit in System-RAM can be random-accessed in disaggregated memory 100x faster than storage



Famfs: Graph Analysis

Fast-RP Graph Analysis, 4-node cluster, 4.6TB graph in shared famfs

- Random access is cache-miss-bound – not bandwidth-bound
- Famfs interleaved across 22 CXL devices behind an Xconn switch
- Same workload with staggered start: no performance interference
- Alternatives are demand-paging or sharding



Large CXL memory appliances and famfs make intractable problems solvable

Famfs: Roadmap

Most of famfs is in user space

- Kernel, libfuse and daxctl upstream!
- Interleaved file support
  - ✓ Already supported on switches that concatenate back-end daxdevs as current switches do
- DCD support and file systems that span multiple DCD allocations (aka tagged capacity instances)
  - Dax devices are tagged capacity allocations
  - Interleave across separate memory devices: kernel support done, user space in 2026
- Proper SW-based cache coherency library
- Persistent config options via system and per-mount config files
  - (e.g. interleave width, copy thread counts, metadata timeouts, etc.)
- Relax limitations on file creation and metadata mutation
- pNFS integration

Composable Mem and Interleaving

Interleaving is critical to how we use memory

- DCD is the device type for dynamically-composable memory

- In CXL 3 and onward, switches don’t participate in interleaving – they expose memory devs directly

- You won’t normally want memory from just one CXL device
  - ➢ Allocate from many devices and interleave

- CXL can program interleaving via HDM decoders…
  - …but CXL interleaving requires the same DPA range on every device
  - DPA space fragmentation (due to alloc/free) will make this difficult
  - Famfs doesn’t care about DPA/HPA, but is limited to interleaving page-size chunks (4K, 2M…)

Famfs: Links

- famfs.org – Documentation and user space repo
- Famfs at LSFMM (2025, 2024)
- Famfs at past LPC conferences (2023, 2024)
- Famfs talk at SNIA SDC 2025

東京 2025

LINUX PLUMBERS CONFERENCE

TOKYO, JAPAN / DECEMBER 11-13, 2025