ZUFS
Simplifying the Development of PM-based File Systems

Dr. Amit Golander, NetApp
Background

- **Plexistor (acquired by NetApp)**
  - PM-based FS pioneer since 2013
  - Contributing some of our IP

- **Our PM-based FS approach:**
  - Support legacy applications & Enable NPM (e.g. SPDK)
  - Feature rich
  - Integrate with NetApp product portfolio
Kernel Vs. User Space FS Implementation

<table>
<thead>
<tr>
<th><strong>Kernel</strong></th>
<th><strong>User space</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Fast (shortest path)</td>
<td>Portable</td>
</tr>
<tr>
<td>Resilient (contained)</td>
<td></td>
</tr>
<tr>
<td>Simpler to add functionality &amp; Debug</td>
<td></td>
</tr>
<tr>
<td>Fewer licensing restrictions</td>
<td></td>
</tr>
</tbody>
</table>

The desired missing link:
Near-memory speed Kernel-to-User bridge
## Why not extend FUSE to PM?

FUSE architecture is great for HDDs and ok(ish) for SSDs, but not suitable for PM.

### Typical medias
- **FUSE**: Built for HDDs & extended to Flash
- **ZUFS**: Built for PM/NVDIMMs and DRAM

### SW Perf. goals
- **FUSE**: Secondary (High latency media), Async I/O Throughput
- **ZUFS**: SW is the bottleneck, Latency is everything

### SW caching
- **FUSE**: Slow media -> Rely on OS Page Cache
- **ZUFS**: Near-memory speed media -> Bypass OS Page Cache

### Access method
- **FUSE**: I/O only
- **ZUFS**: I/O and mmap (DAX)

### Cost of redundant copy / context switch
- **FUSE**: Negligible
- **ZUFS**: The bottleneck -> Avoid copies, queues & remain on core

### Latency penalty under load
- **FUSE**: 100s of µs
- **ZUFS**: 3-4 µs
ZUFS Features & Architecture

- Low latency & Efficient
  - Core & L1 cache affinity
  - Zero data copy

- Manages devices
  - Optimal pmem access
  - NUMA aware
  - Data mover to lower tier devices

- Page table mapping supports
  - I/O & DAX semantics

- Misc
  - Async hook available
  - System service
Preliminary Results (for PM)

- Measured on
  - Dual socket, XEON 2650v4 (48HT)
  - DRAM-backed PMEM type
- Random 4KB DirectIO write access

FUSE Vs ZUFS Penalty (PM, DirectIO)
Summary

- **ZUFS is a Kernel-to-User bridge designed for PM**
  - Enables NetApp solutions

- **Being contributed upstream**
  - Hope to accelerate PM adoption and innovation
  - Link to Github (TBD)

- **You’re welcome to use, review and contribute code**
  - zufs@netapp.com
Thank you