Pangolin: A Fault-tolerant Persistent Memory Programming Library

Lu Zhang, Steven Swanson
Non-Volatile Systems Laboratory
Department of Computer Science & Engineering
University of California, San Diego

Track I, 11:00am on Friday, July 12
Persistent memory modules finally arrive

- Working alongside DRAM
- New programming model
  - Byte addressability
  - Memory semantics
  - Direct access (DAX)
Challenges with programming

• Crash consistency
  – Volatile CPU caches
  – 8-byte store atomicity

• Fault tolerance
  – Media errors
  – Software bugs

No file system can protect DAX-mapped persistent memory data.
Pangolin design goals

- Ensure crash consistency
- Protect against media and software errors
- Require very low storage overhead (1%) for fault tolerance
Pangolin – Replication, parity, and checksums

- Combines replication and parity as redundancy
  - Similar performance compared to replication
  - Low space overhead (1% of gigabyte-sized object store)

| Metadata | Metadata | Object | Object | Object | Object |
|----------|----------|--------|--------|--------|--------|
|          |          |        |        |        |        |
| Object   | Object   | Object | Object |
|          |          |        |        |        |
| Object   | Object   | Object |
|          |          |        |
| Object   | Object   |
|          |          |
| Parity   |          |

- Checksums all metadata and object data
Pangolin – Transactions with micro-buffering

- Provides micro-buffering-based transactions
  - Atomically updates objects, checksums, and parity
  - Prevents programming errors from corrupting PMEM

Track I, 11:00am on Friday, July 12