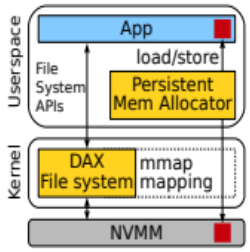
POSEIDON: Safe, Fast and Scalable Persistent Memory Allocator

Wook-Hee Kim, Anthony Demeri, R. Madhava Krishnan, Jaeho Kim*, Mohannad Ismail, Changwoo Min
Virginia Tech, *Gyeongsang National University

12TH ANNUAL NON-VOLATILE MEMORIES WORKSHOP

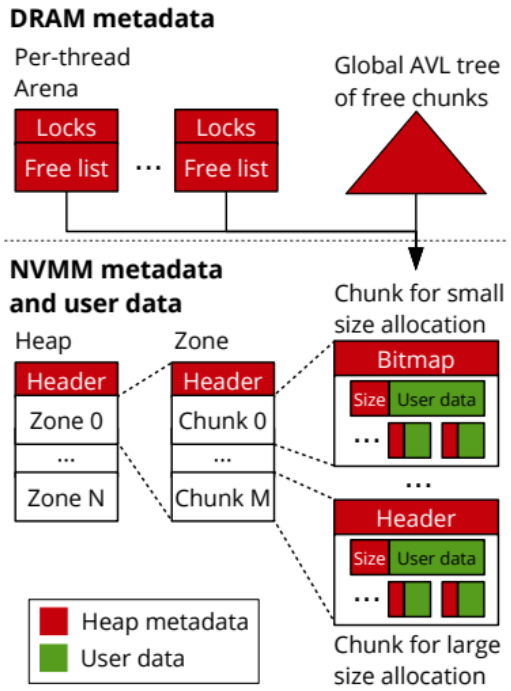
Background

- **Design Requirement of PM Allocator**
  - Transactional allocation
  - Crash consistency of heap metadata.
  - Protection of heap metadata from program errors.
  - Scalability



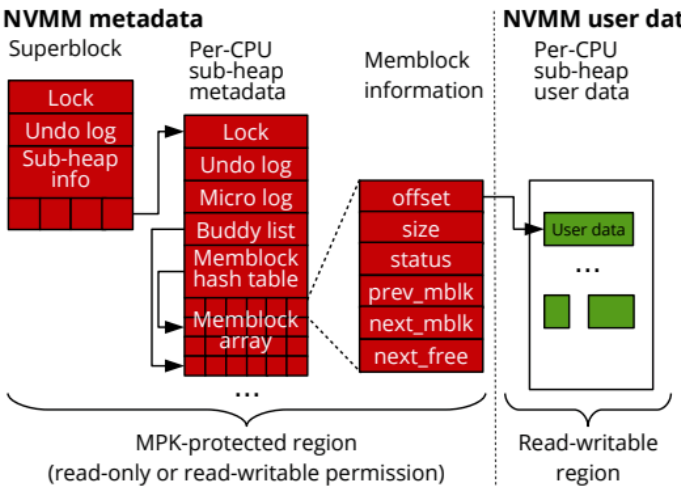
Motivation : Problems in existing allocators

- **The Case of PMDK**



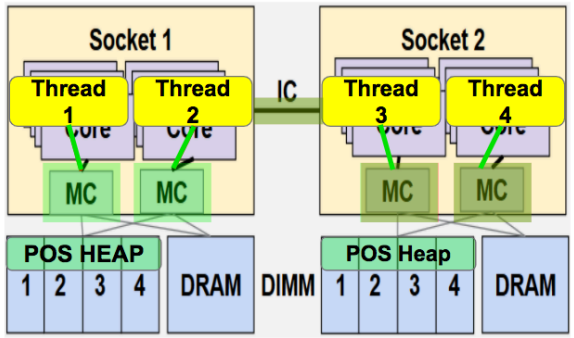
- In-place Metadata corruption
- Direct Metadata corruption
- Volatile Metadata corruption
- Non scalable performance due to the global index

Architecture of POSEIDON



- Per-CPU Sub-Heaps
- Metadata Management Using Hash Table
- Fully segregated Metadata
- Metadata Protection with Intel MPK
- Crash Consistency

Per-CPU Heap with Hash table based Metadata

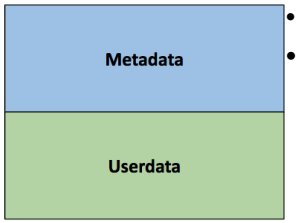


- Guarantees constant lookup time
- Minimizes inter-socket memory accesses
- Maximizes use of memory controllers
- Eliminates global system bottlenecks

Metadata Protection using Intel MPK

Intel Memory Protection Keys (MPK)

Binning metadata storage is critical

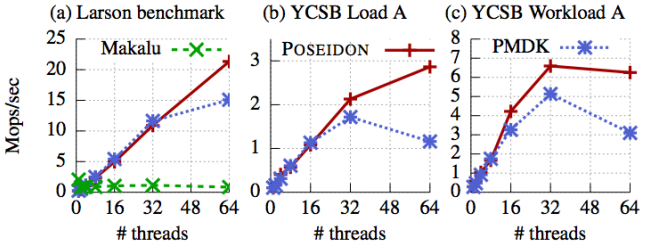


- Lightweight
- Page size granularity
  - In-place metadata design cannot leverage MPK

Crash Consistency

- **Undo Logging**
  - Guarantees failure-atomicity of the metadata update.
- **Micro Logging**
  - Keeps history of memory allocations to prevent memory leaks

Evaluation



- POSEIDON significantly outperforms other persistent memory allocators up to 4×.
- POSEIDON shows better performance in YCSB evaluation (read-world evaluation).

Conclusion

- Poseidon guarantees safety using Intel MPK
- Poseidon's per-CPU sub-heap design show better scalability