Flash Memory SUMMIT

TUNA and HEAPO: Suite for Developing Software for NVM

Youjip Won

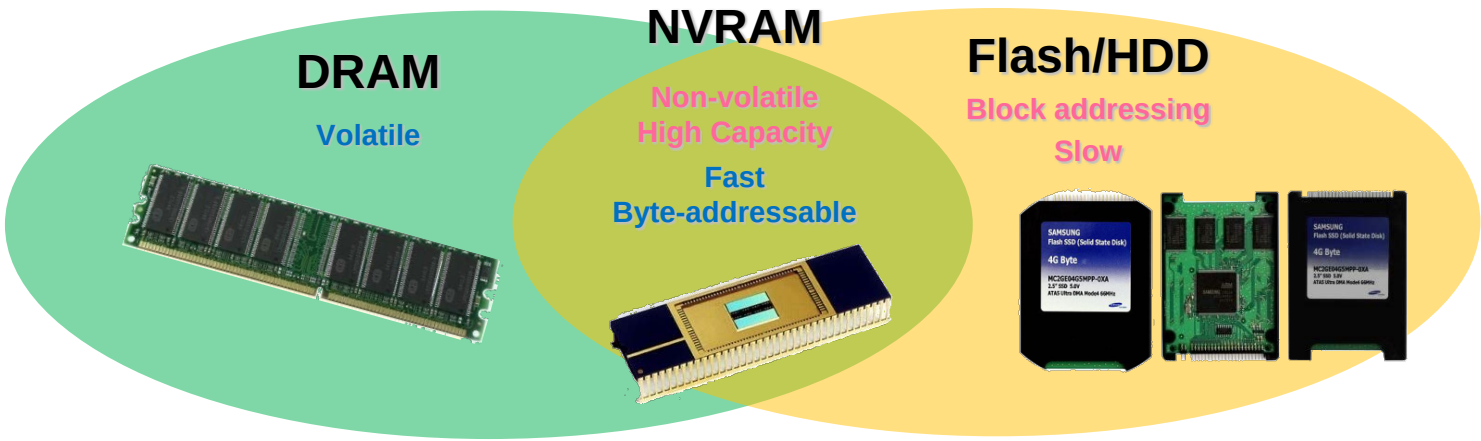
Hanyang University, Seoul, Korea

Outlines

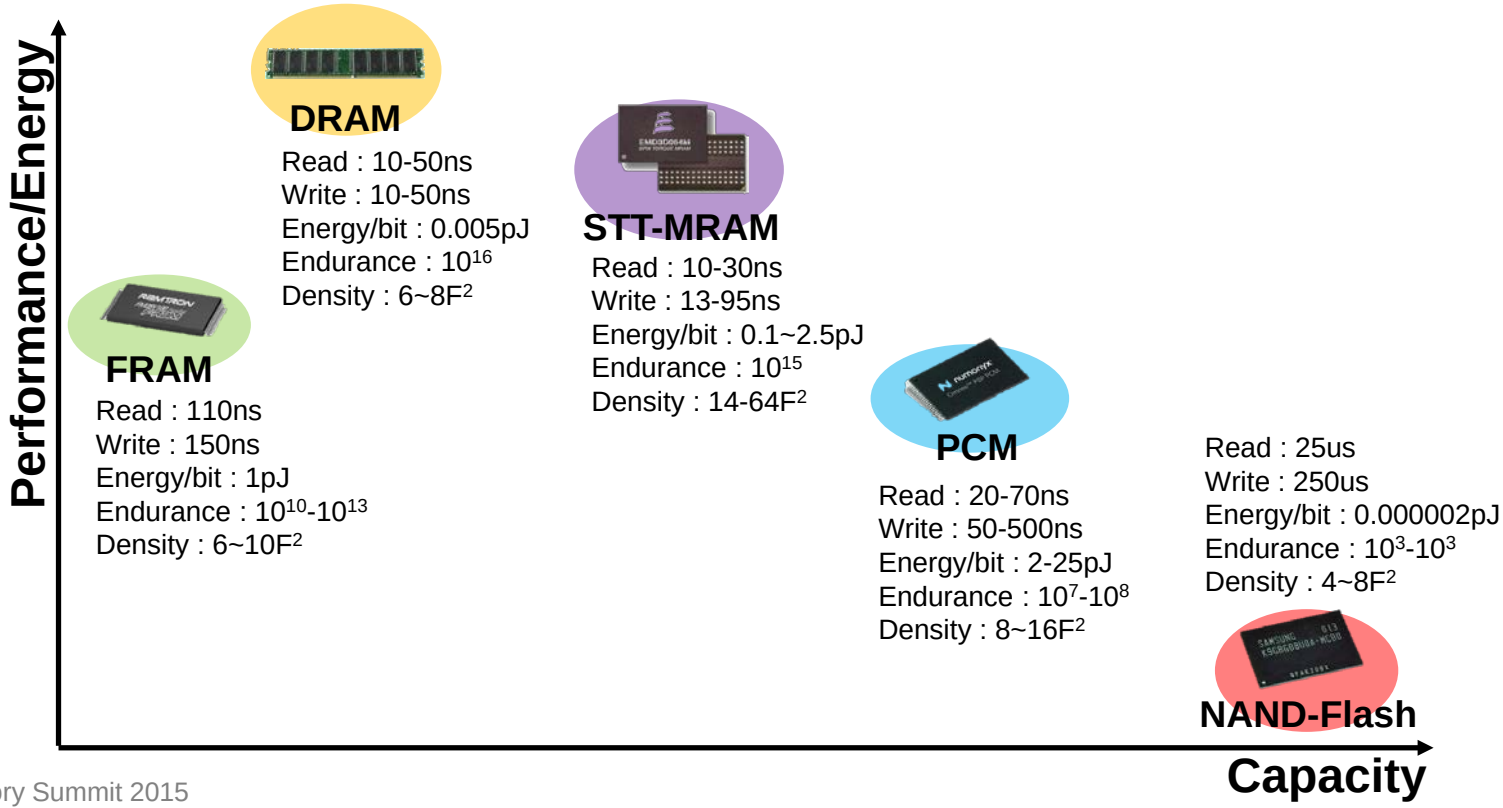
- TUNA: NVRAM Emulation Platform
- HEAPO: Persistent Heap Layer for Non-Volatile Memory

TUNA: The Platform for Emulating NVM

Non-volatile Memory



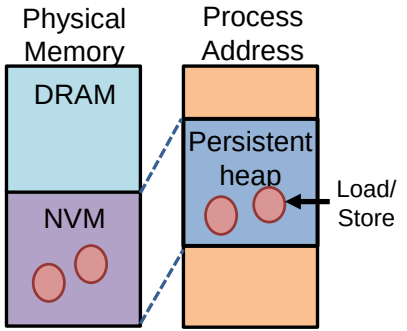
Characteristics of Memory Devices



Software Layers for NVM

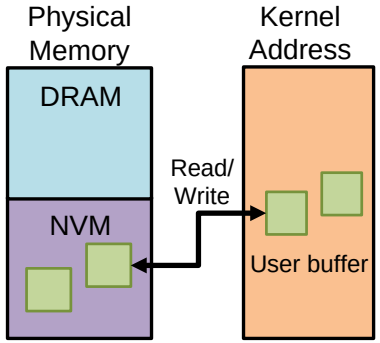
Persistent Heap

- NV-heap(Coburn, UCSD, ASPLOS'11)
- Mnemosyne(Volos, UW, ASPLOS'11)
- SoftPM (Guerra, FIU, Usenix ATC'12)
- WSP (Narayanan, MS, ASPLOS'12)
- HEAPO (Hwang, HYU, ACM TOS'14)
- NVM Duet (Liu, NTU, ASPLOS'14)



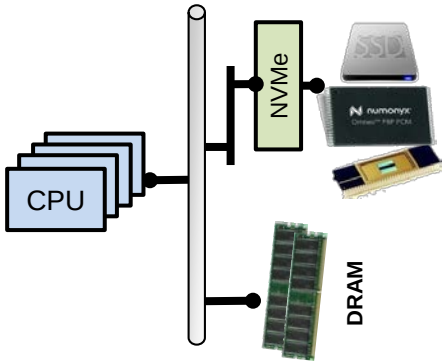
Byte-addressible Filesystem

- BPFS (Condit, MS, SOSP'09)
- FRASH (Jung, HYU, ACM TOS'12)
- SCMFS (Wu, Texas A&M, SC'11)
- PMFS (Dulloor, Intel, Eurosys'14)
- PMBD (Chen, LSU, MSST'14)



Fast Block Device

- Moneta (Caulfield, UCSD, MICRO'10)
- Onyx (Akel, UCSD, HotStorage'11)
- PCMSSD (Kim, IBM, FAST'14)
- DC Express (Vucinic, UCSD, FAST'14)
- nvramdisk (Jung, HYU)



Are the performance results credible?

Sources for inaccuracy

- **Real NVM device:** small scale and for dedicated purpose
- **Emulating NVM with DRAM:** DRAM is much faster.
- **Introducing software delay:** code overhead and subsequent cache pollution
- **Cycle-accurate simulation:** Lengthy simulation and cannot simulate realistic scenario
- **NVRAM emulation platform (Intel PMEP):** Only available in x86, cannot simulate the non-volatility.

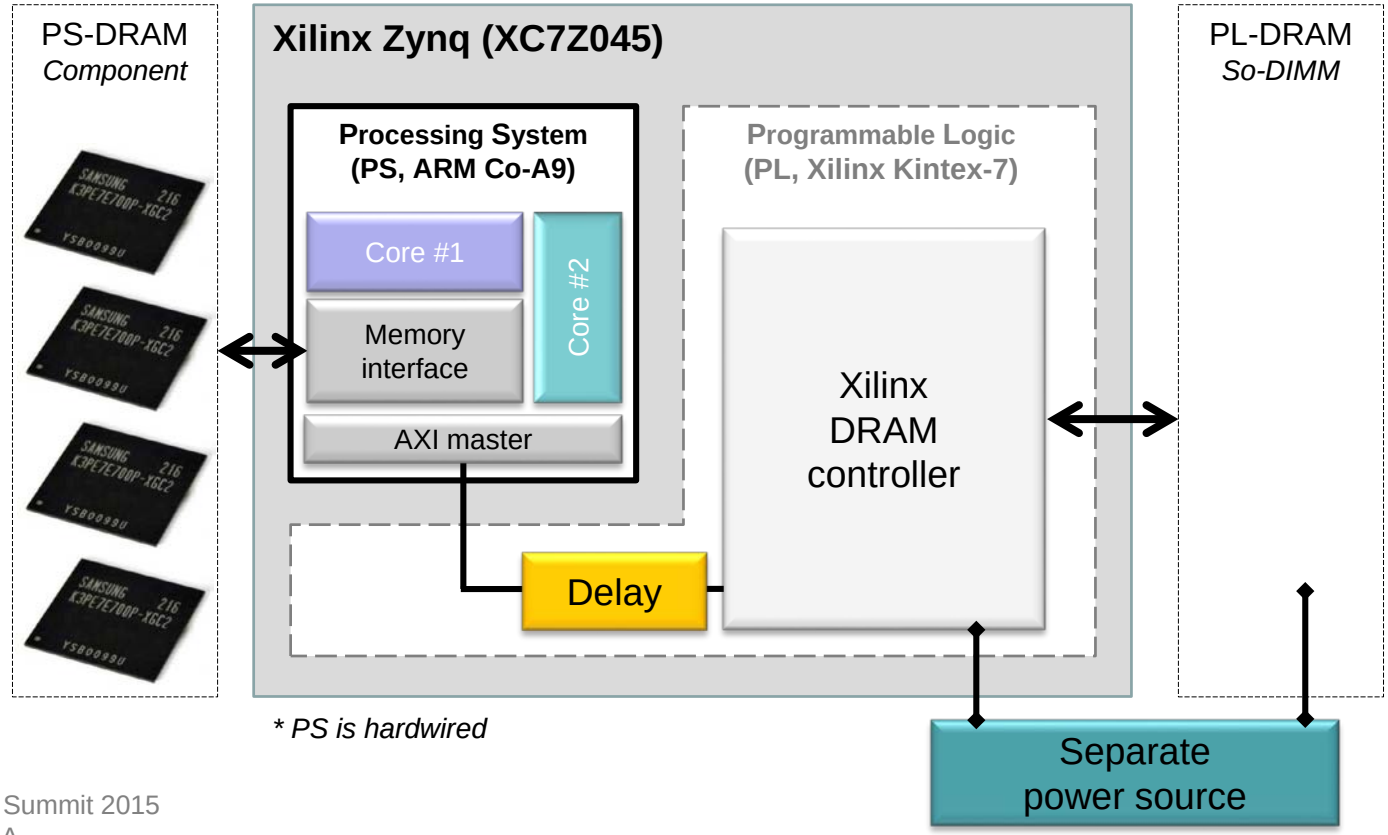
Hardware platform for NVM emulation for mobile device

- Variable Latency
  - PCM, STT-MRAM, and etc.
- Non-volatility emulation
- Mobile platform

TUNA: Hardware platform for NVM emulation

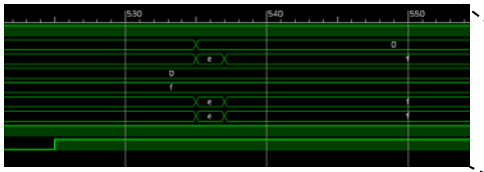
- Use DRAM to emulate NVRAM
- For Non-volatility
  - Separate power source for emulated NVRAM
- For Variable latency
  - API for adjusting the NVM latency
  - Separate latencies for `load` and `store`
- For Mobile platform
  - ARM
  - Android ported

TUNA Organization

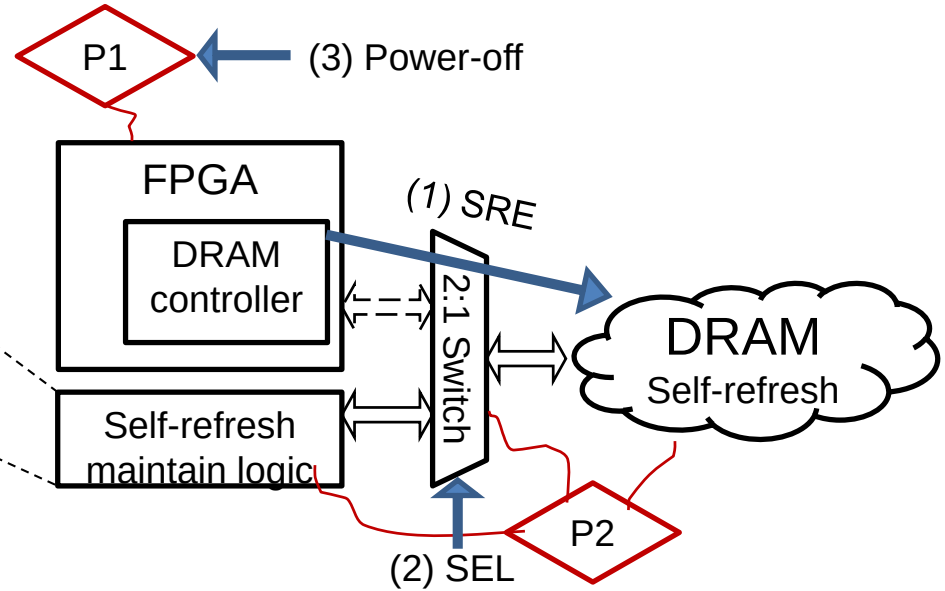


Emulating non-volatility with DRAM

- Separate power control
  - Self-refresh management circuit
  - PS-only reboot (NVM-DRAM under self refresh or auto-refresh)

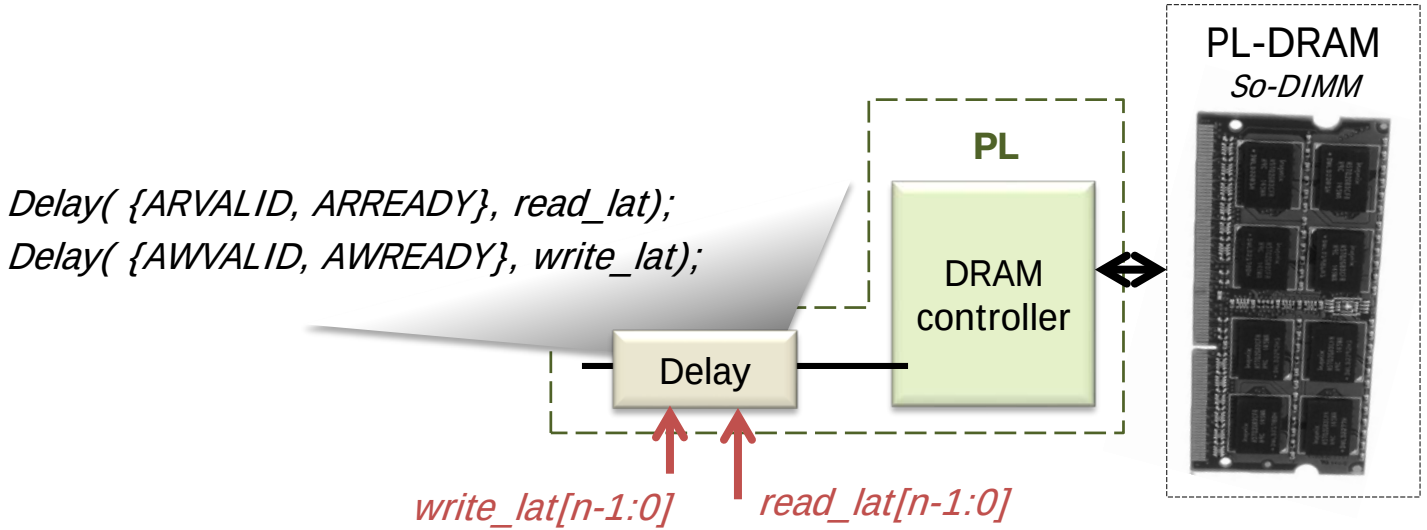


Based on real measurement on steady state logic value!

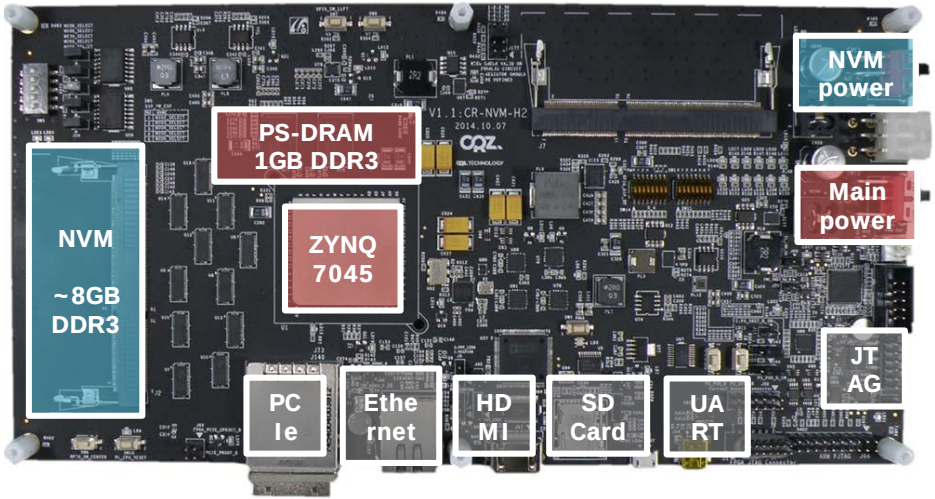


Support Variable Latency

- Independent RD/WR latency tuning
- Delay AXI packets from PS
  - Postpone handshake signals on AW/AR channel

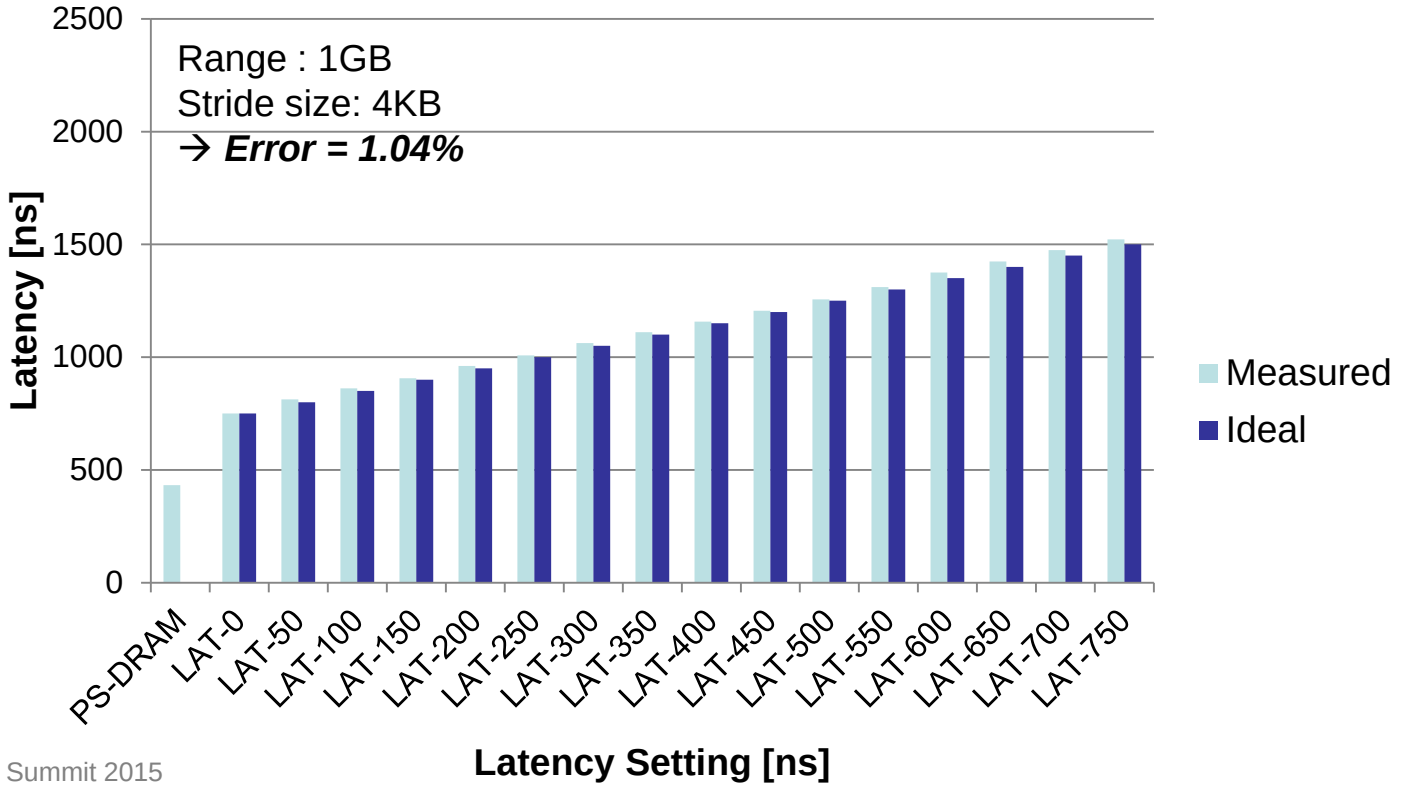


TUNA specification

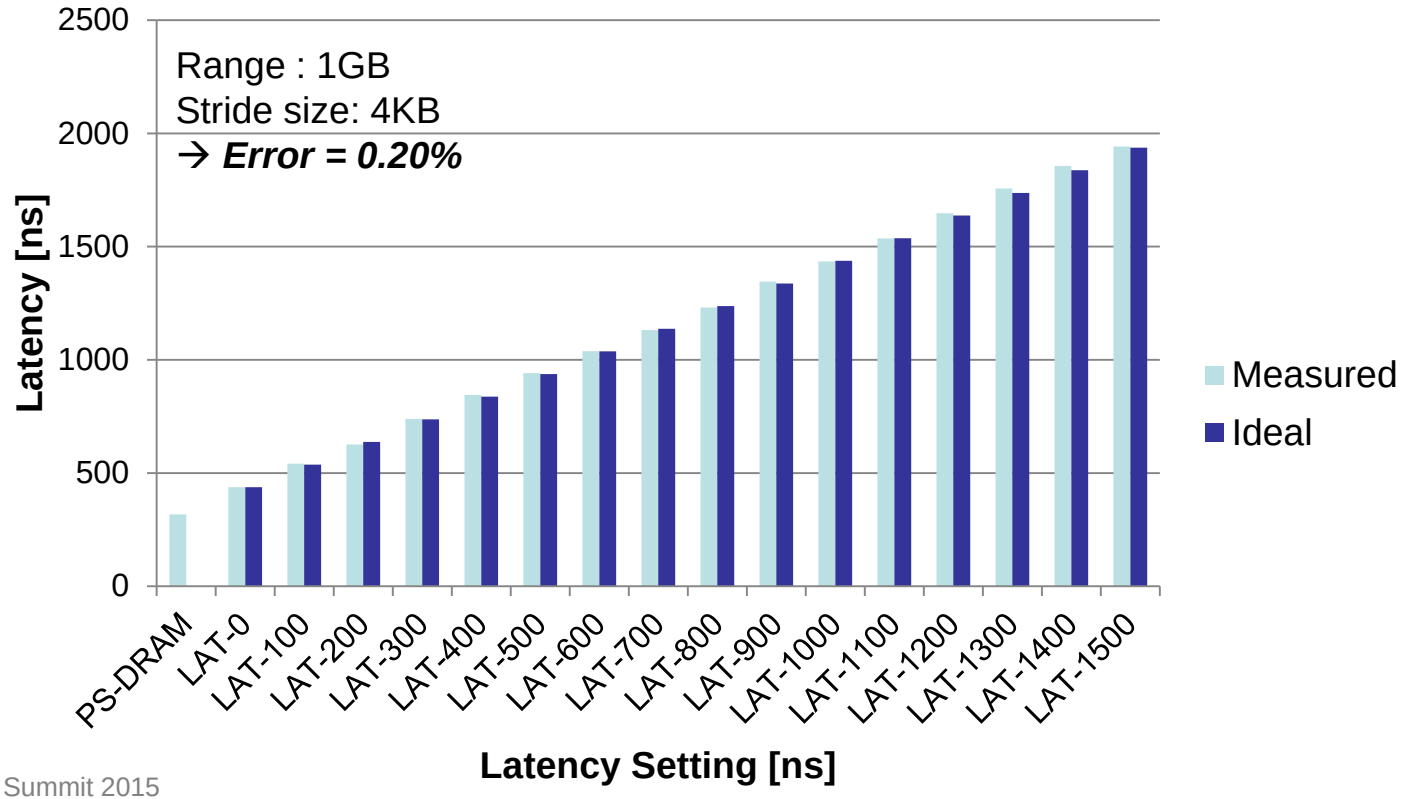


- **Xilinx Zynq SoC**
  - ARM Cortex-A9
  - Xilinx Kintex-7
- **Memory**
  - 1GB PS DDR3-SDRAM
  - For NVM: 8GB PL DDR3-SDRAM

Latency Emulation: *load*

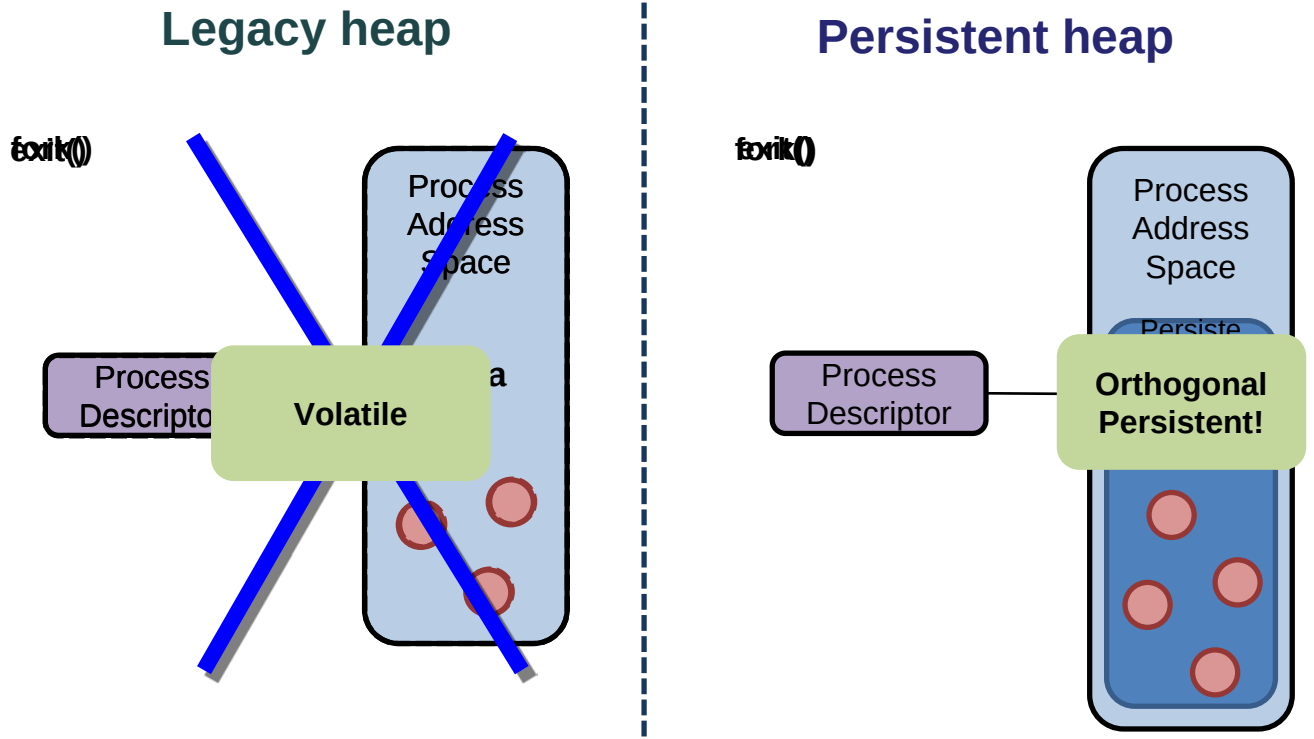


Latency Emulation: *store*

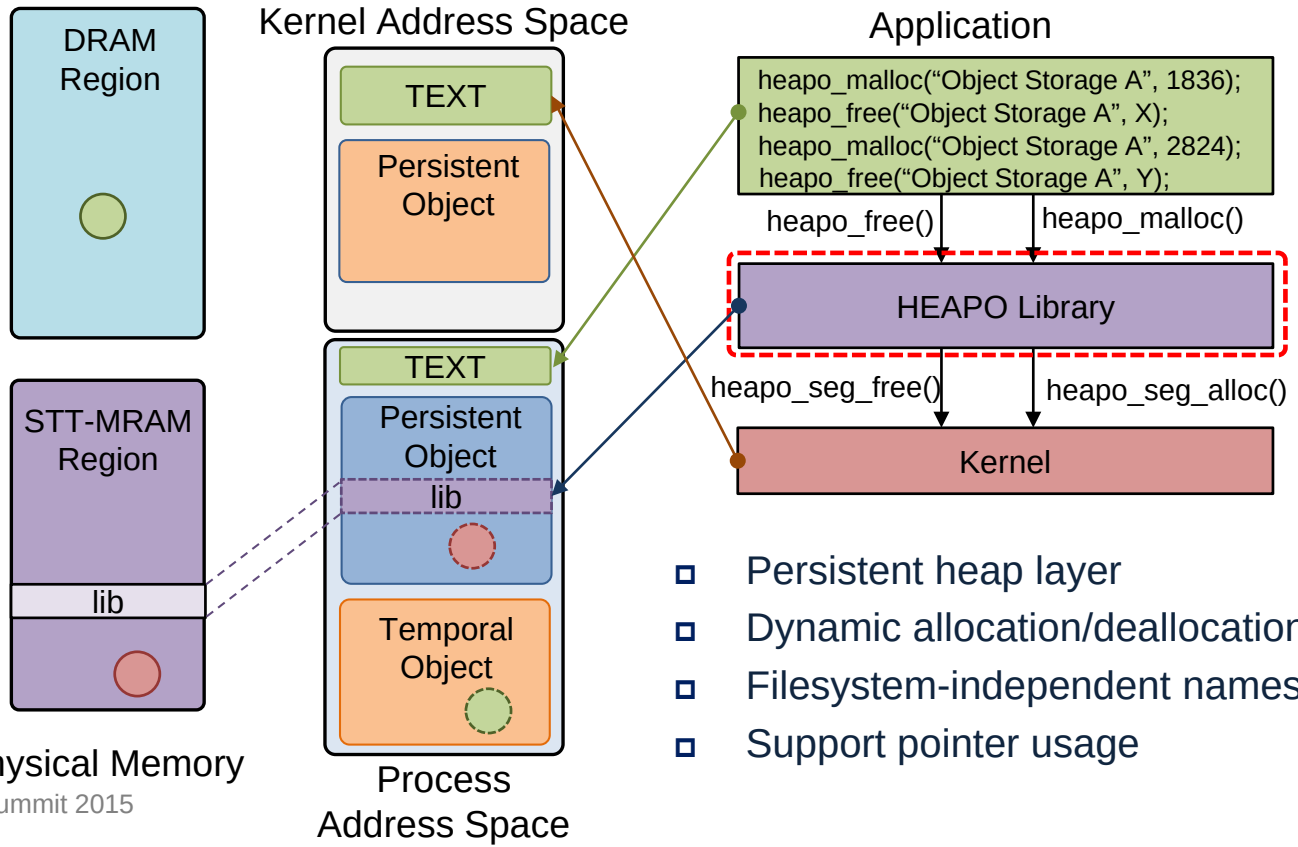


HEAPO: Persistent Heap for Linux

Persistent Heap

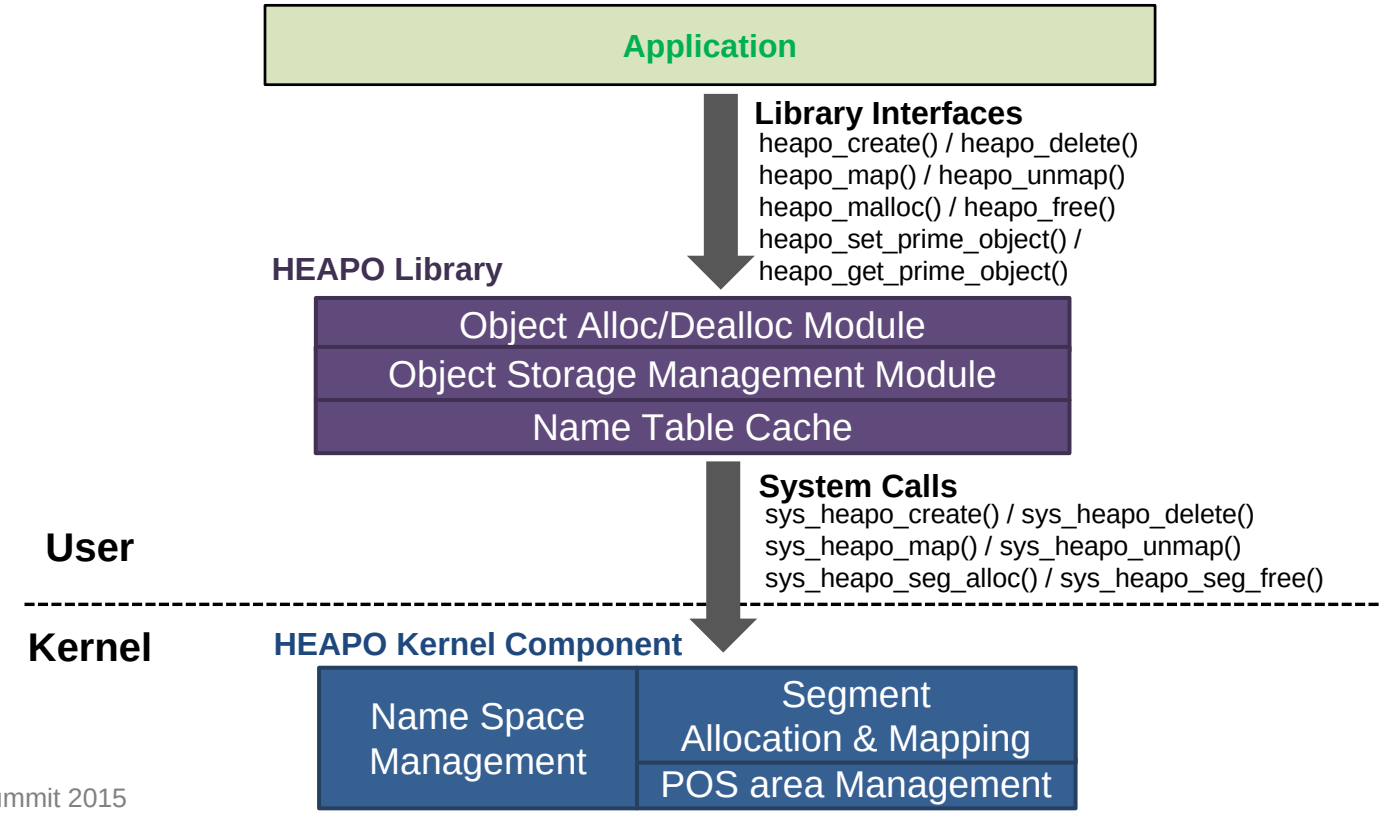


Overall Structure of HEAPO



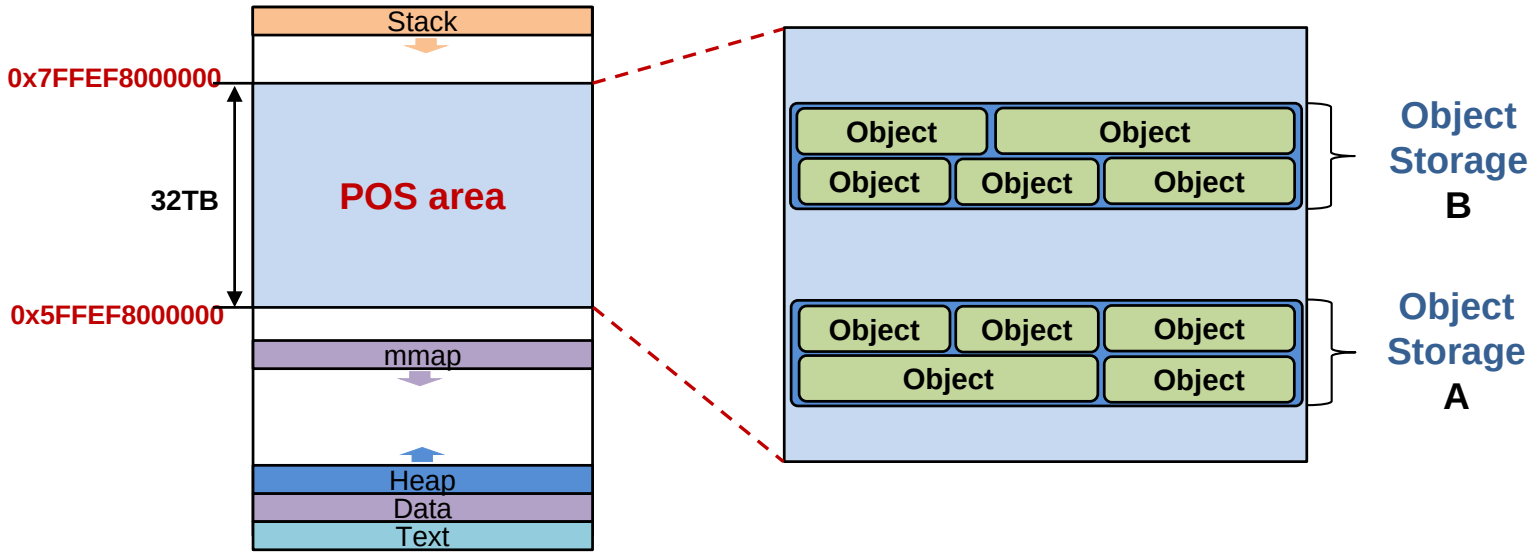
- Persistent heap layer
- Dynamic allocation/deallocation
- Filesystem-independent namespace
- Support pointer usage

Software Stack



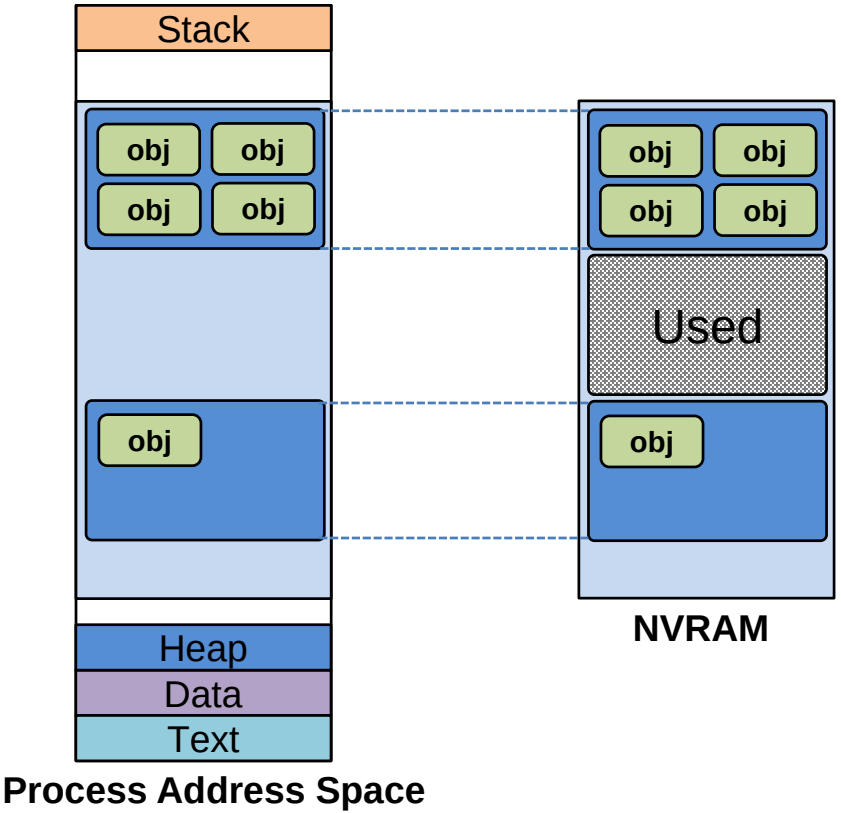
Object Storage & Object

- ***Object storage***: Persistent memory region with unique name
- ***Object***: Unit of memory allocation/deallocation



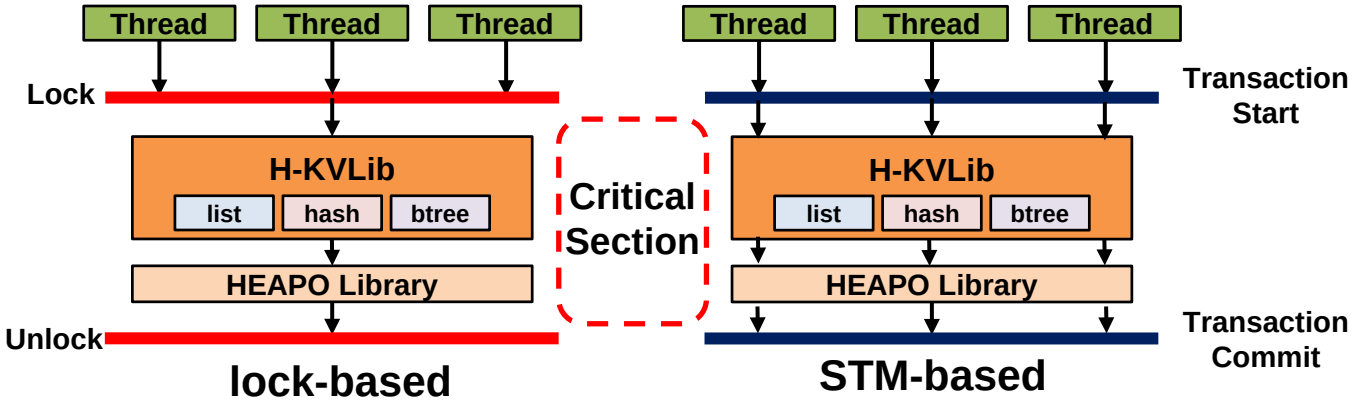
Usage Example

```
heapo_create("name")
heapo_malloc("name", size)
heapo_malloc("name", size)
heapo_malloc("name", size)
heapo_malloc("name", size)
heapo_malloc("name", size)
heapo_free(address)
heapo_free(address)
exit()
fork()
heapo_map("name")
heapo_unmap("name")
```

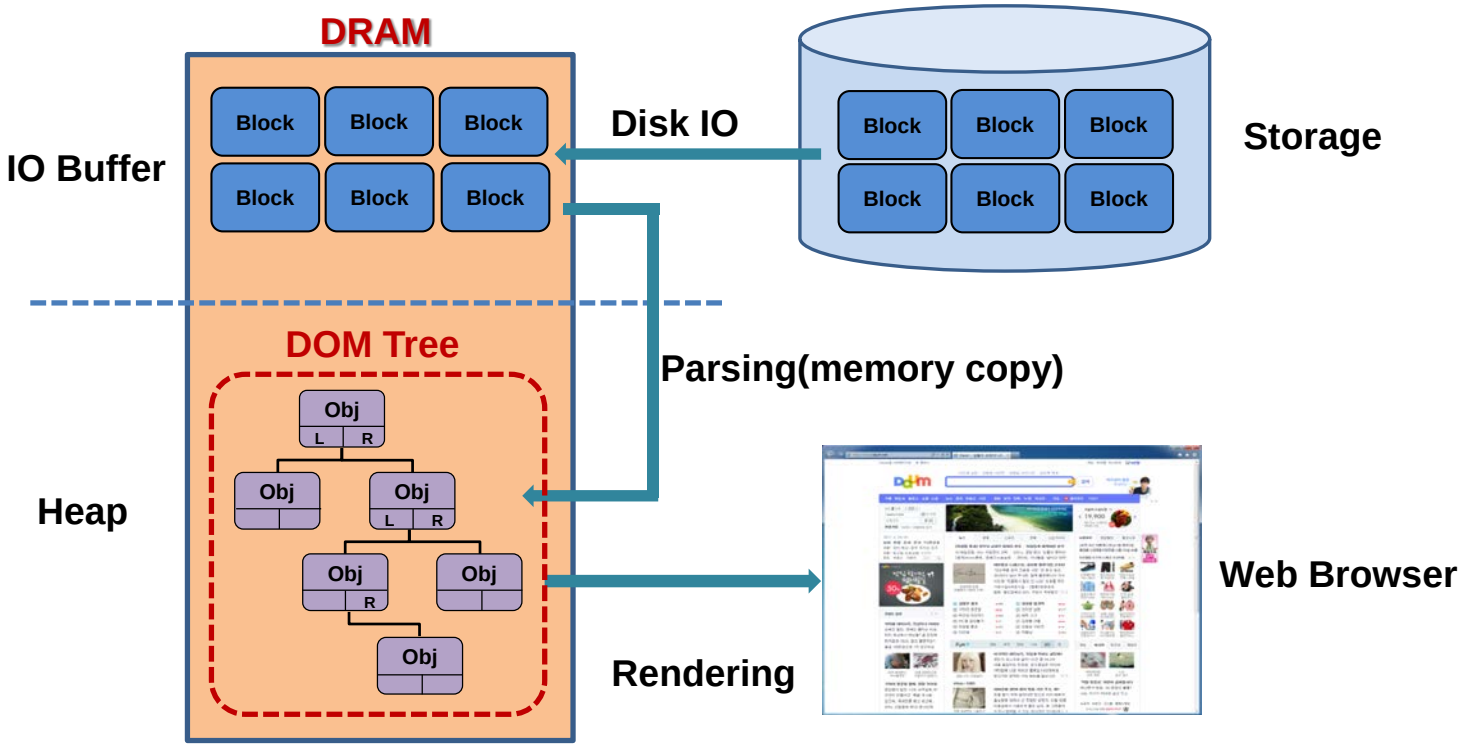


H-KVLib: HEAPO-based Key-value Library

- Internal data structure: B-tree, list, hash
- Guarantee fail-safe atomicity (minimal logging)
- Support lock-based and STM-based concurrency control

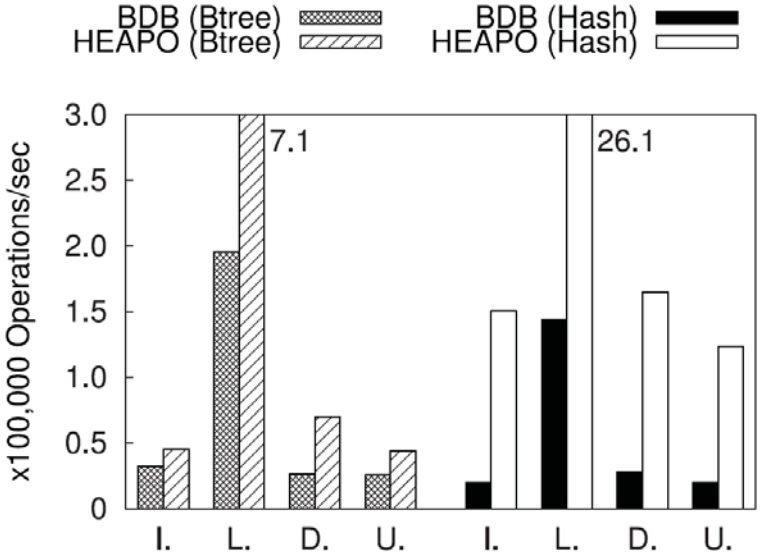


Use case: Web Application

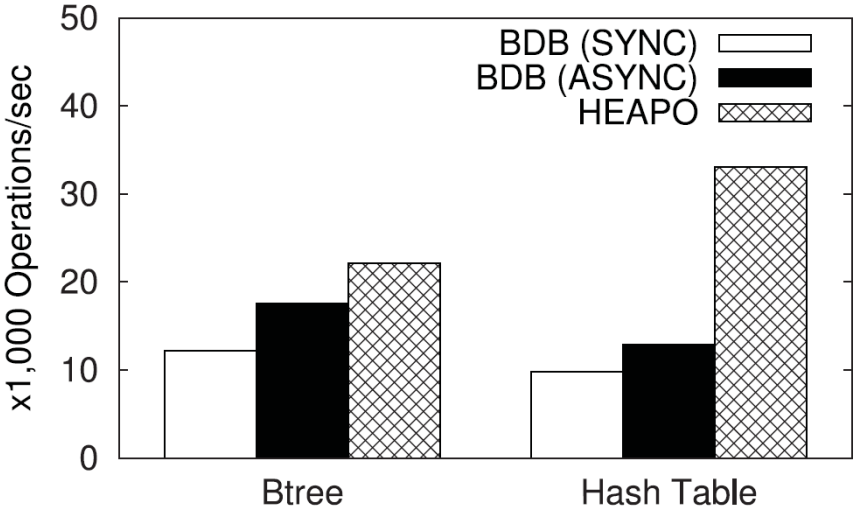


Launching Web Applications with HEAPO

HEAPO-based Key-value Store Performance



I: Insert, **L**: Lookup, **D**: Delete, **U**: Upate



Fail Safe Atomicity Support in HEAPO

- Logging for key-value operations in H-KVLIB
- For persistence and ordering
  - `clflush` and `mfence` in x86
  - `dccmvac` in ARM

TUNA & HEAPO: HW/SW for NVRAM

- TUNA platform (http://www.opennvram.org)
  - OS support: linux & android
  - Applications and Library : HEAPO and SQLite for NVM
- HEAPO (https://github.com/ESOS-Lab/HEAPO)
  - HW support: x86 & ARM
  - Provide dedicated key-value library

Summary

- HEAPO over TUNA
  - Comprehensive Platform for developing software for NVM based system
  - All open sourced
- TUNA platform (http://www.opennvram.org)
  - OS support: linux & android
- HEAPO (https://github.com/ESOS-Lab/HEAPO) for x86 and ARM
  - Provide dedicated key-value library

Collaborators

- Joint work with
  - CRZ Co.
  - Prof. Seungjoo Yoo at Seoul National University
- References
  - Hwang et.al. "HEAPO: Heap-based Persistent Object Store", ACM Trans. on Storage, 11(1), Dec. 2014
  - Lee et.al. "FPGA-based prototyping systems for emerging memory technologies," IEEE International Symposium on Rapid System Prototyping (RSP), 2014