UniHeap : Managing Persistent Objects Across Managed Runtimes for Non-Volatile Memory

Daixuan Li Benjamin Reidys Jinghan Sun
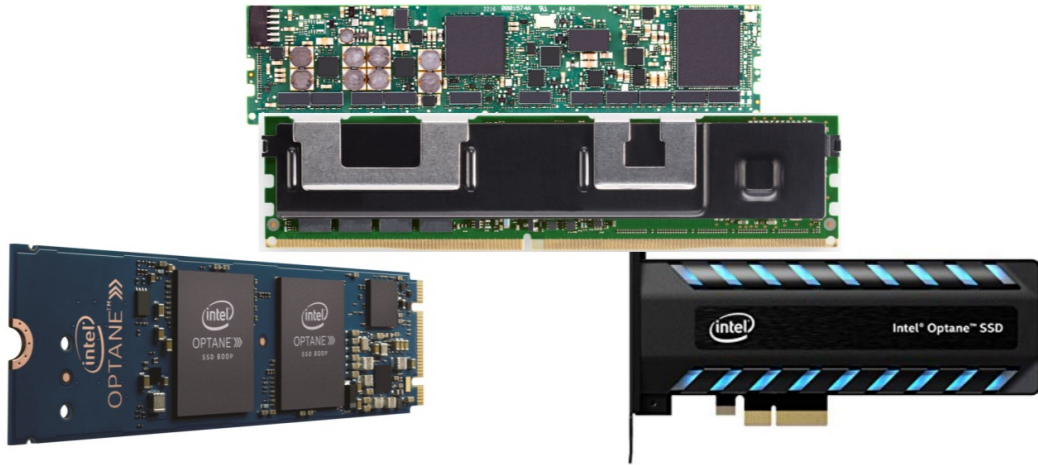
Thomas Shull Josep Torrellas Jian Huang

ECE ILLINOIS

Non-Volatile Memory: Opportunities & Challenges
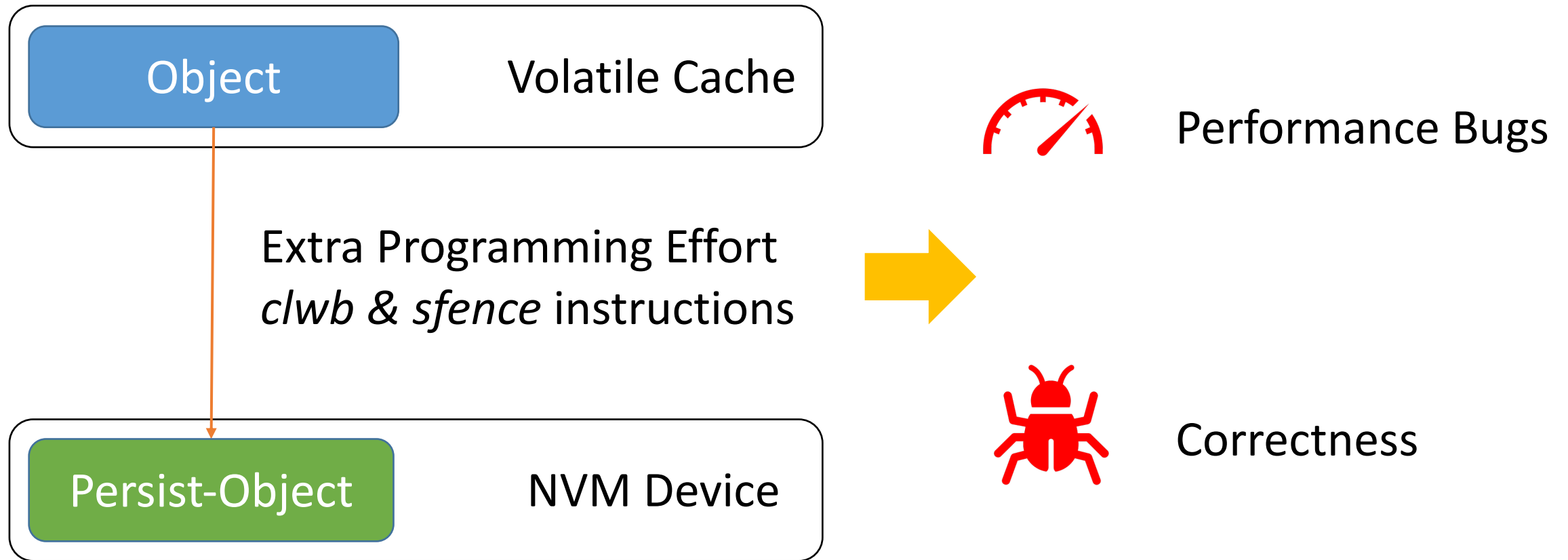
Performance & Persistency

Byte-Addressable
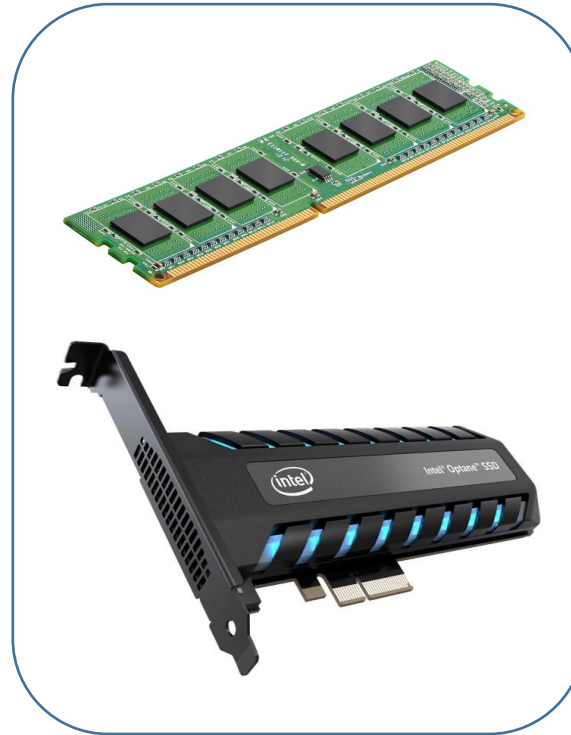
Data Durability

Memory Persistency

Programmability

Programmability Challenge of NVM

Object — Volatile Cache

Extra Programming Effort
clwb & sfence instructions

Persist-Object — NVM Device

Performance Bugs

Correctness

Leveraging Managed Runtime to Manage NVM
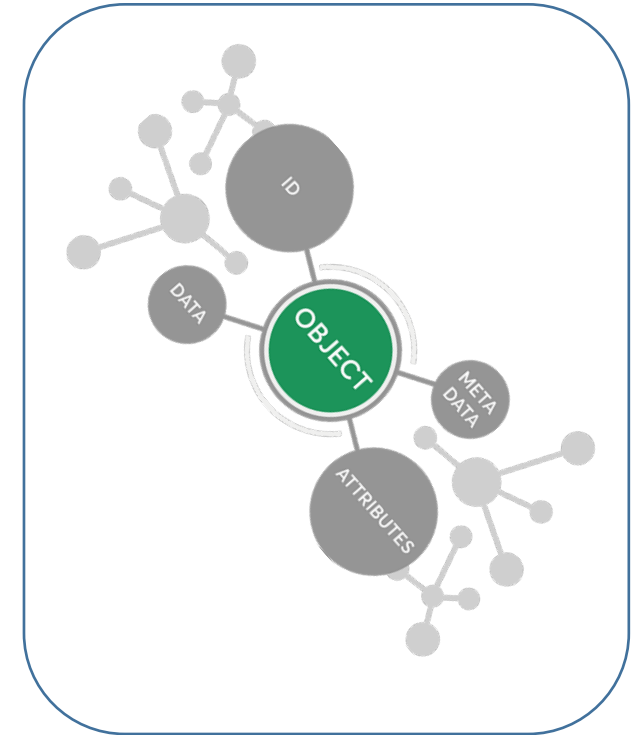

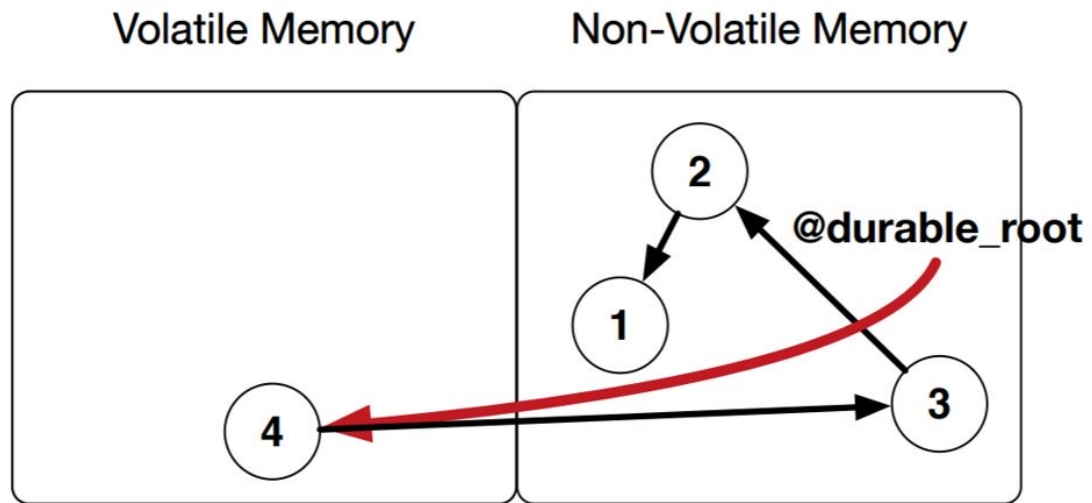**Popular Programming Models**

Hardware Complexity

Managed Data Objects

AutoPersist: An Easy-to-Use NVM Framework

Cross Managed Runtime?

Programmability Improvement

Managing Persistent Object Across Runtime is Desirable
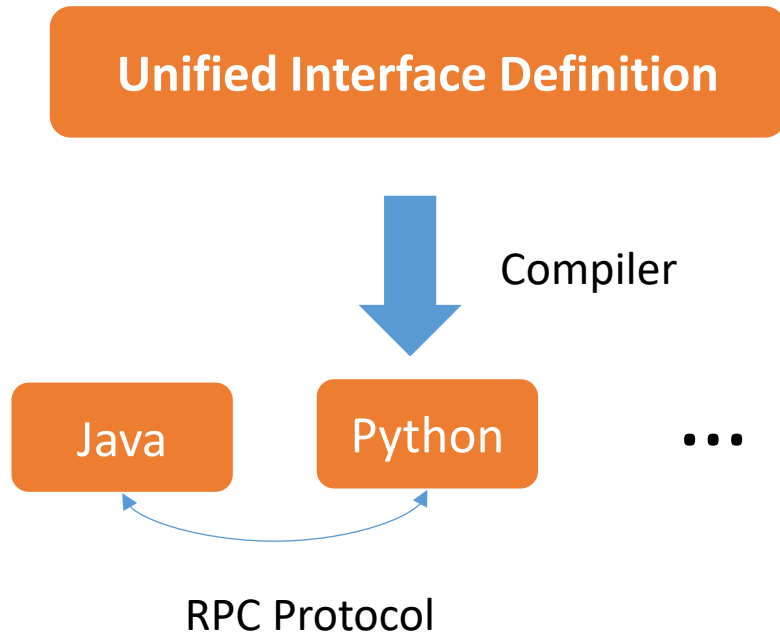
Web Service

Shared Libraries

Data Analytics

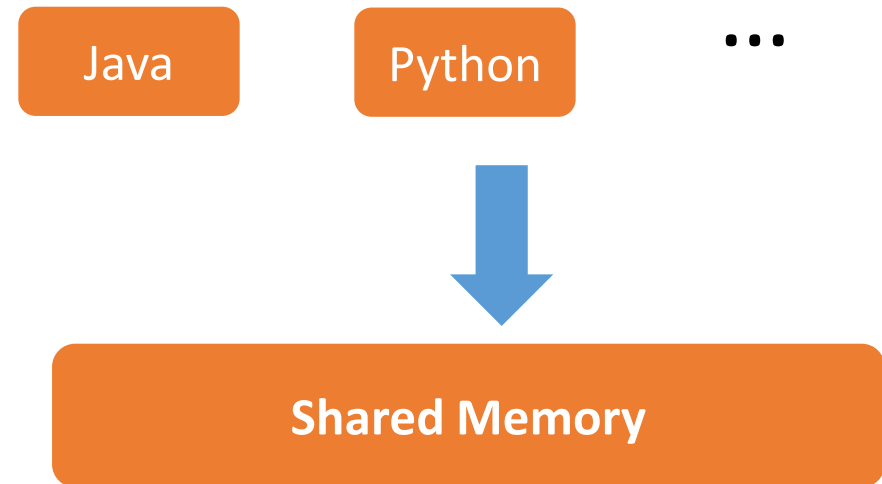**A Unified Interface** to Access Persistent Object is Needed.

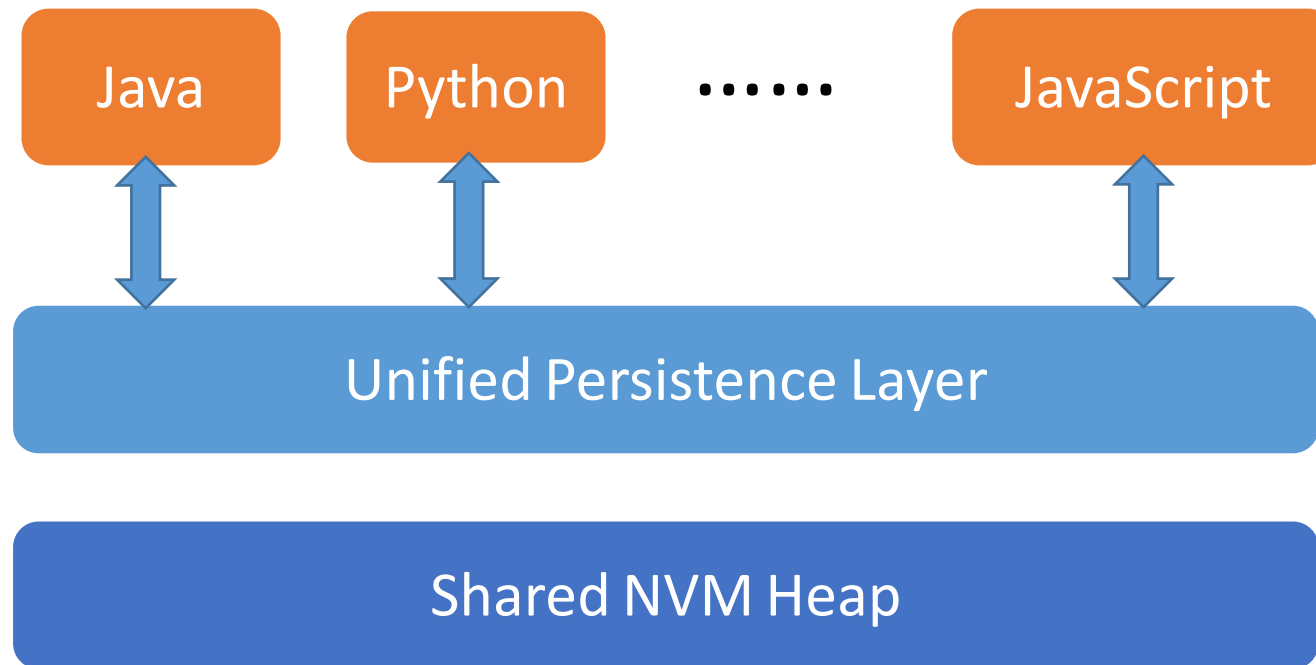# State-of-the-Art Object Sharing Approaches

- Thrift/Protobuf:

- Shared Memory:

UniHeap: Managing Persistent Objects Across Runtimes

Challenges of Persistent Object Management Across Runtimes

Unified Object Model
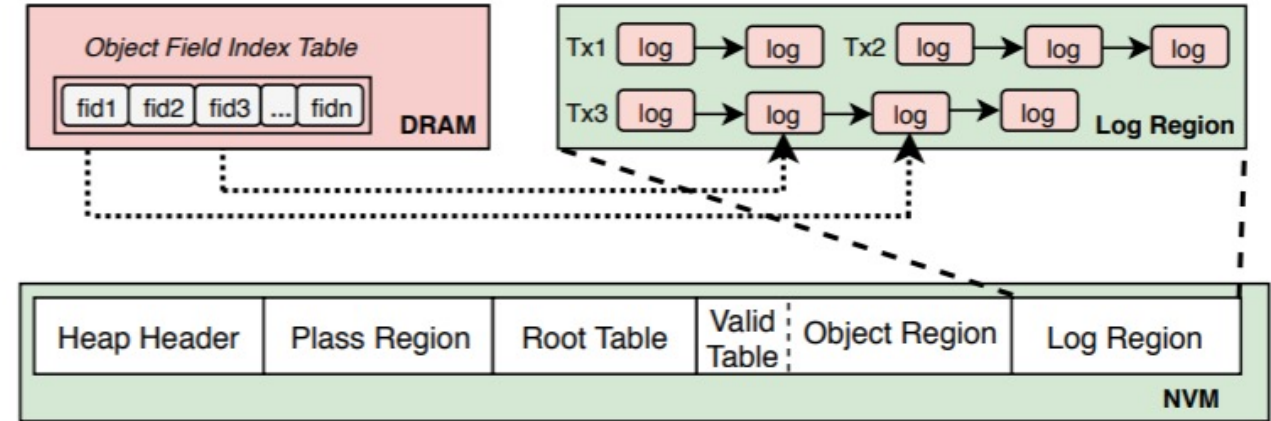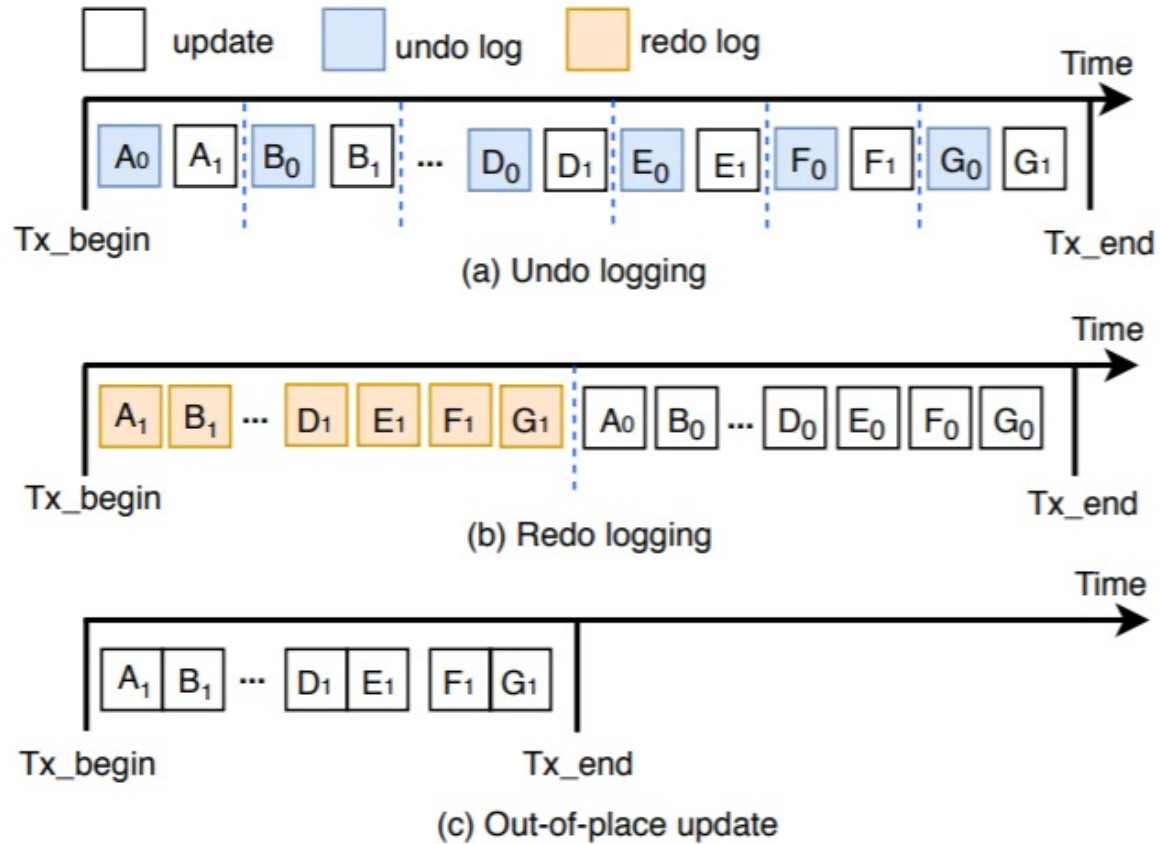
Persistent and Crash-Safe Implementation

Efficient and Correct GC

Unified Object Model and Type System

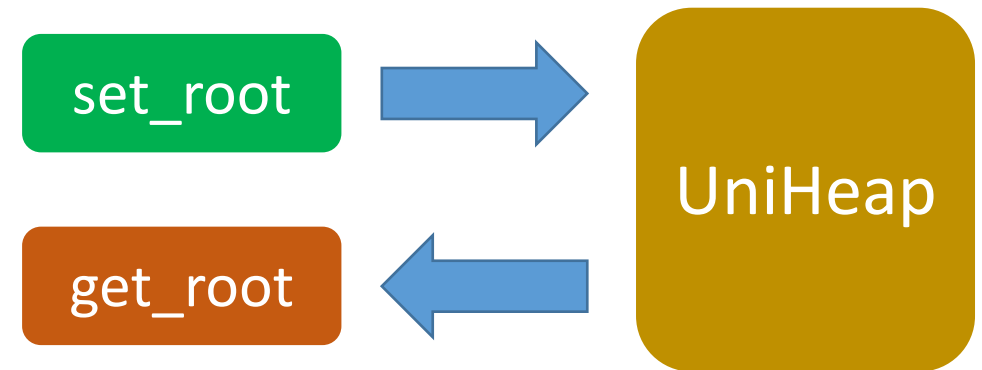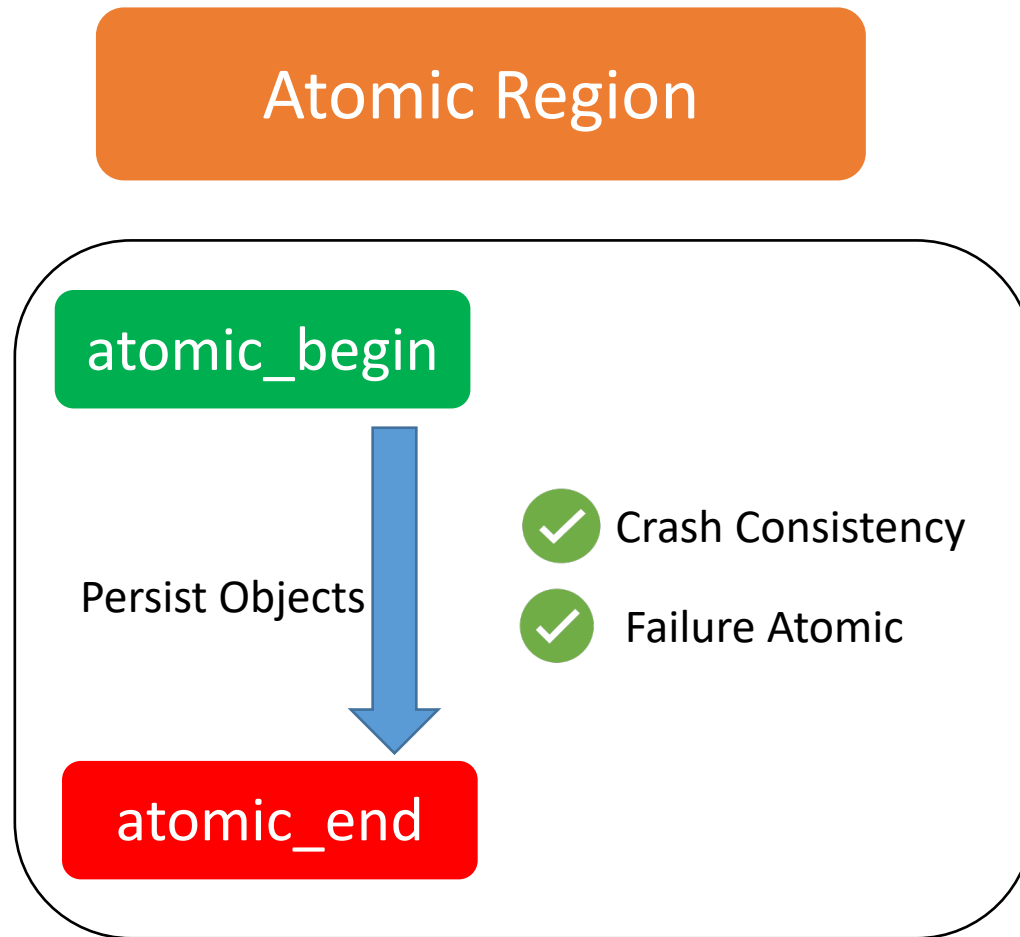| Uniheap	char	short	int	long	float	double	reference
Java	boolean, byte	char	int	long	float	double	reference, array
Python	-	-	int	long	float	-	list, dict, tuple
JavaScript	boolean	-	num	num	num	num	array

- Two kinds of built-in types: **numeral type** and **reference type**
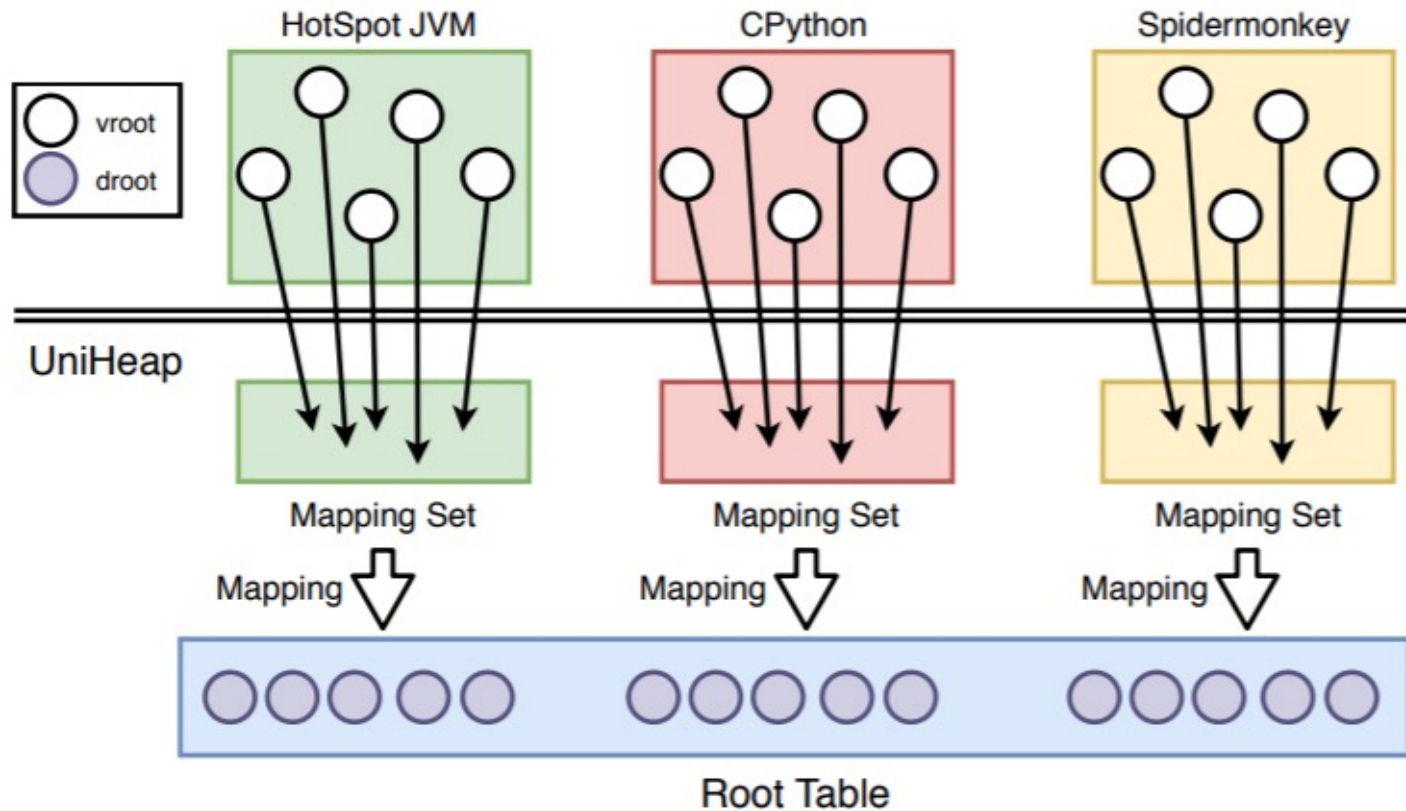
Managing Persistent Objects in A Log-Structured Manner

(a) Undo logging

(b) Redo logging

(c) Out-of-place update

Reduce Logging Overhead
with Out-of-Place Update

Enforce Memory Persistency with Simple API

Atomic Region

atomic_begin

Persist Objects

✅ Crash Consistency

✅ Failure Atomic

atomic_end

set_root → UniHeap

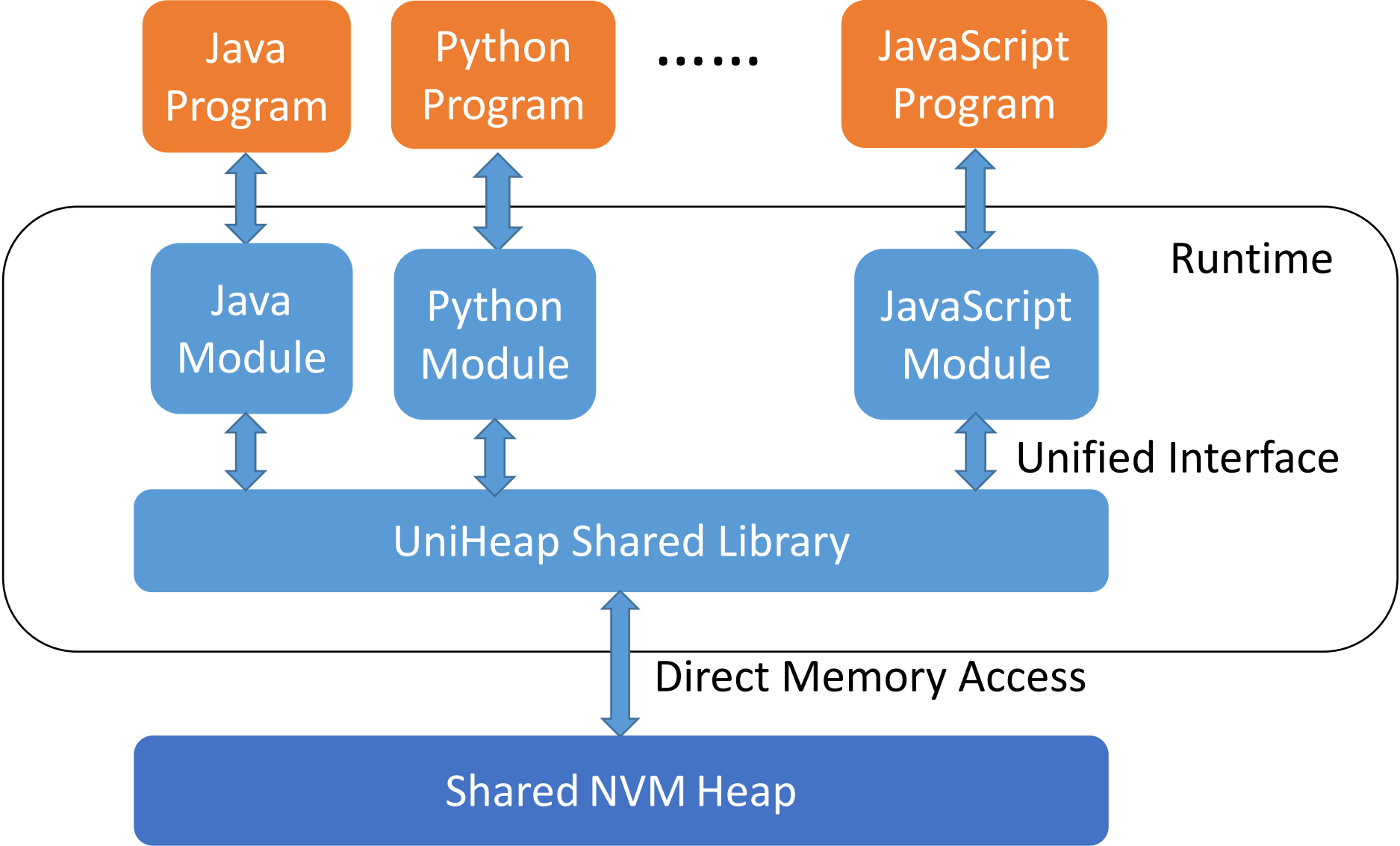get_root ← UniHeap

Durable Root

Coordinated GC Across Managed Runtimes

- Marking phase

- Relocation phase

- Compaction phase

- Clean-up phase
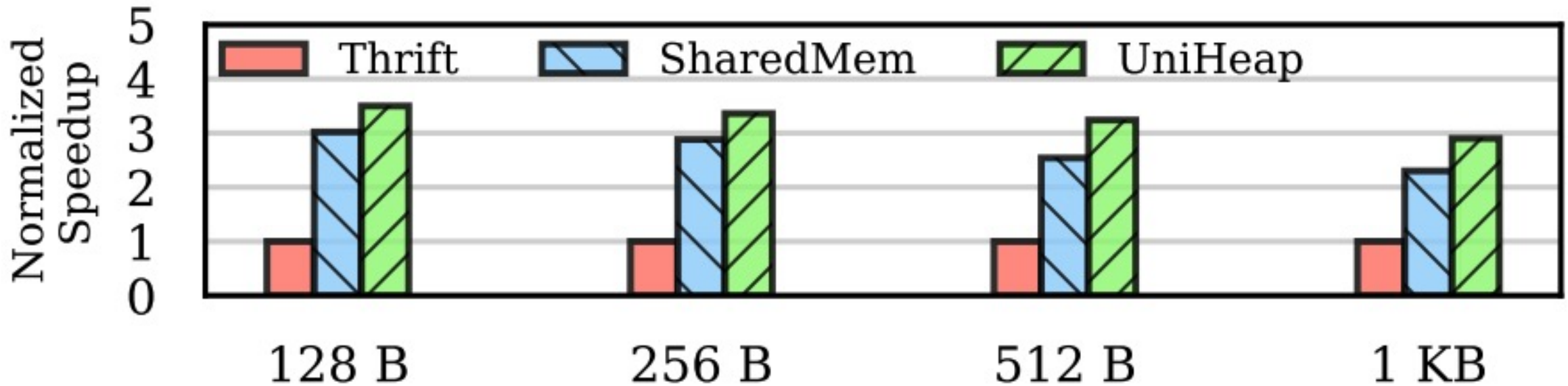
Put It All Together

Experiment Setup

- CPU: 24-core Intel 2nd Xeon
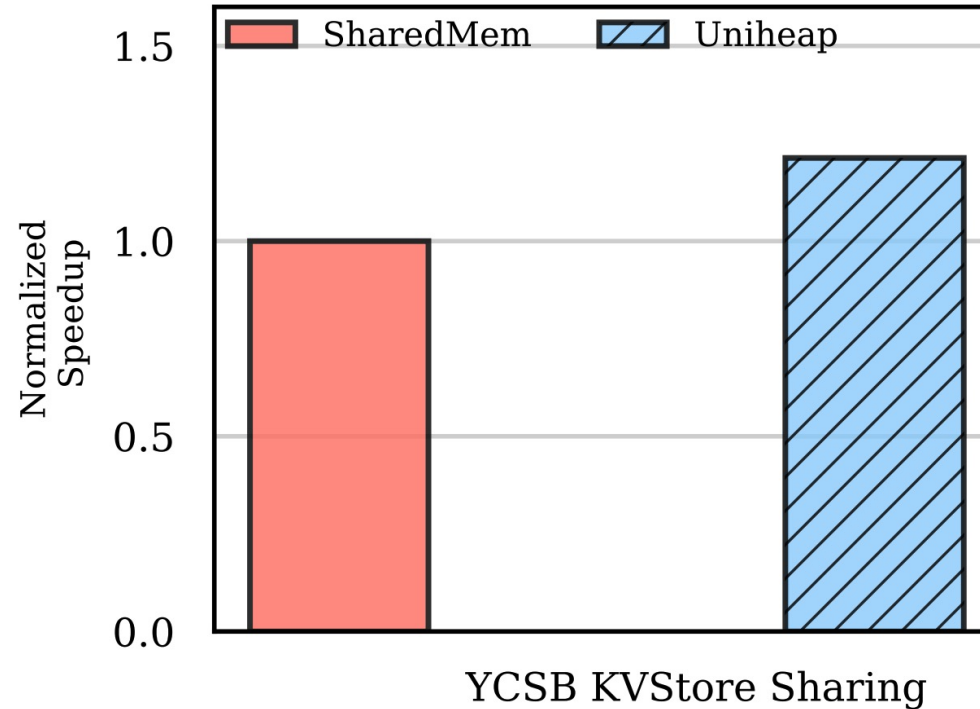- NVM: 128GB Intel Optane

Evaluation Benchmarks

- **Java:** YCSB
- **Python:** N Queens, N body
- **JavaScript:** Matrix multiplication
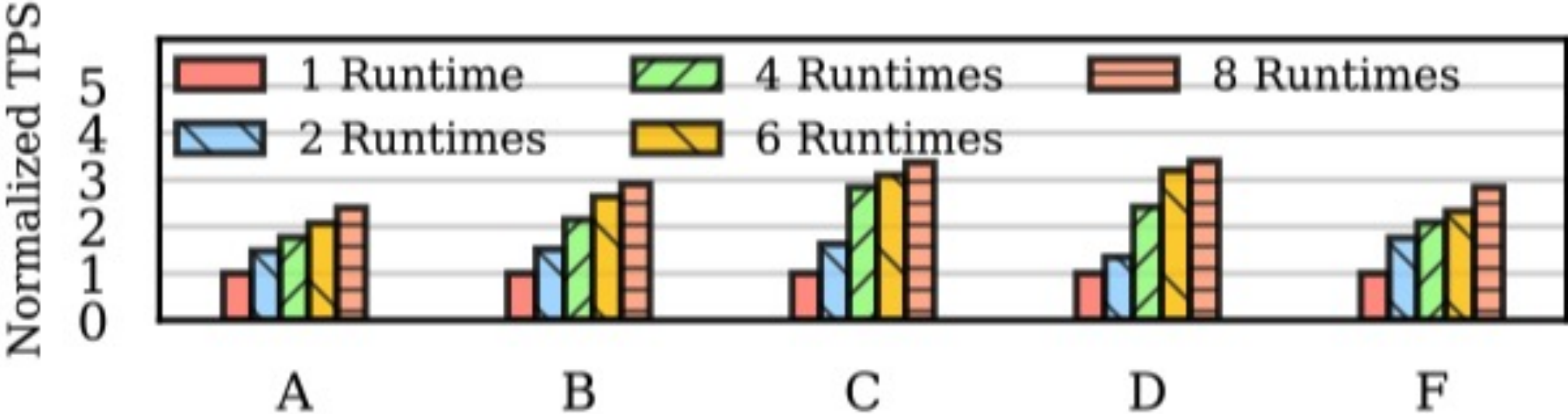
Performance of Persistent Object Sharing

UniHeap outperforms existing approach by 1.2x - 3.4x

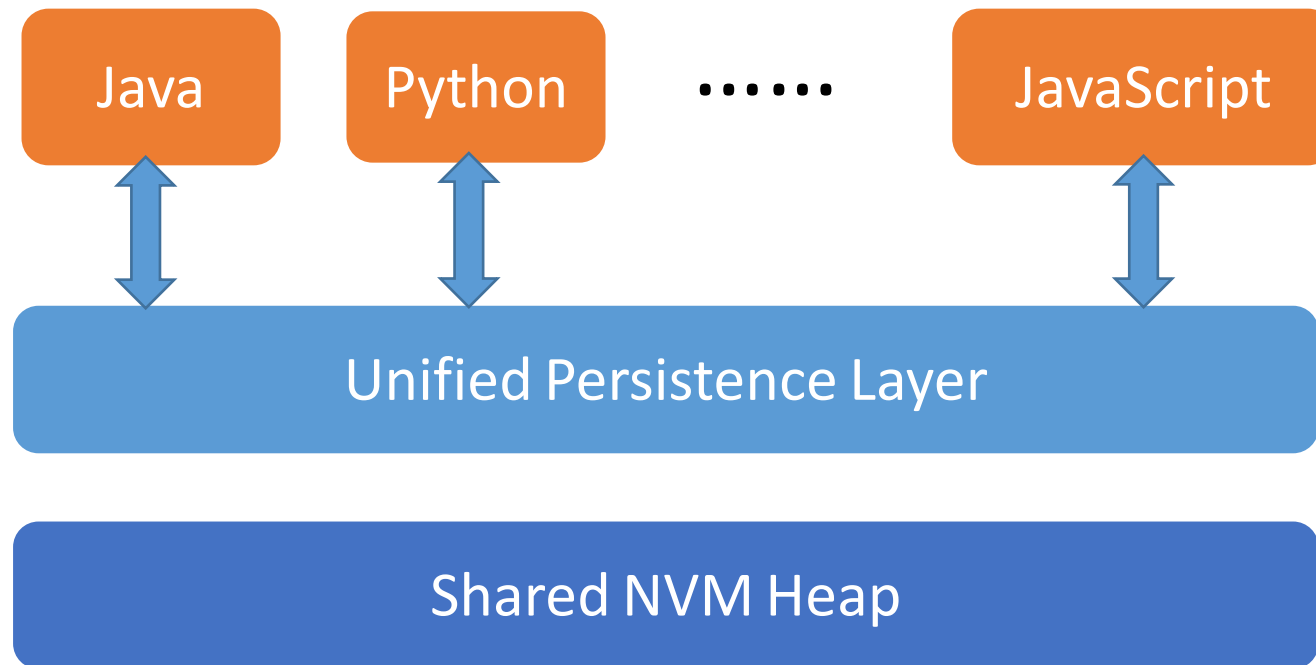# Performance of Persistent Object Sharing

UniHeap outperforms existing approach by 1.2x - 3.4x

Scalability of UniHeap

UniHeap can scale to support multiple managed runtimes.

UniHeap Summary

Thanks!

Daixuan Li   Benjamin Reidys   Jinghan Sun

Thomas Shull   Josep Torrellas   Jian Huang