FORTH — INSTITUTE OF COMPUTER SCIENCE

UNIVERSITY OF CRETE

Australian National University

Red Hat

DynaHeap: Dynamic Division of DRAM between Heterogeneous Managed Heaps

Iacovos G. Kolokasis, Shoaib Akram, Foivos S. Zakkak, Polyvios Pratikakis, and Angelos Bilas

Big Data Frameworks Need More Memory

- Data grow at an exponential rate, but DRAM scales slower than the data growth

- Existing works extend the managed heap over NVMe SSD, NVM, or remote memory
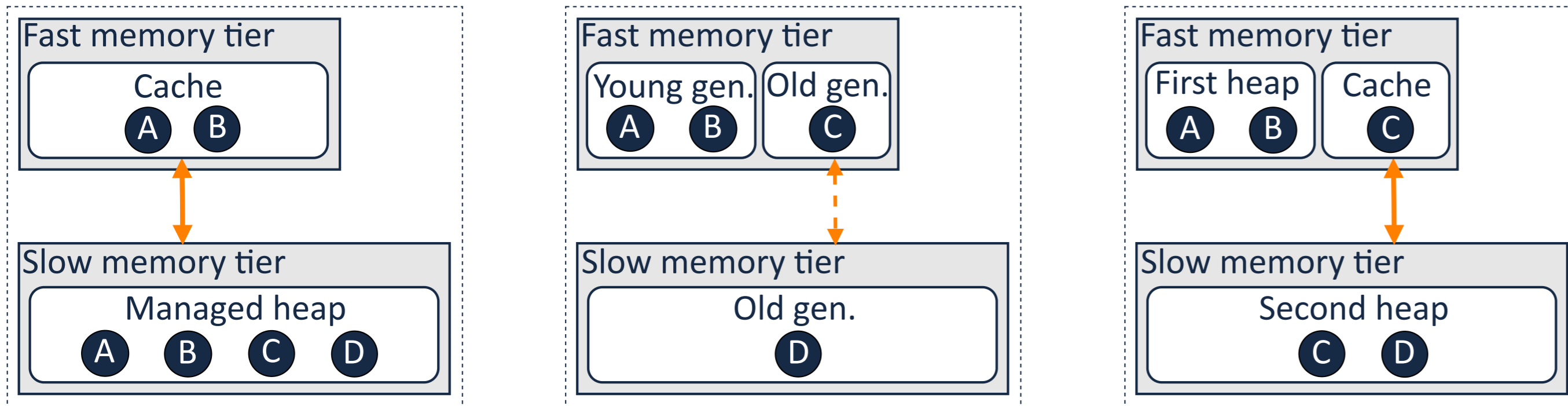
- + No object reference adjustment
- − GC scans over the slow tier

- + Reduce GC scans over the slow tier
- − Need object reference adjustment

- + Avoid GC scans over the slow tier
- + No object reference adjustment
- − Static DRAM division
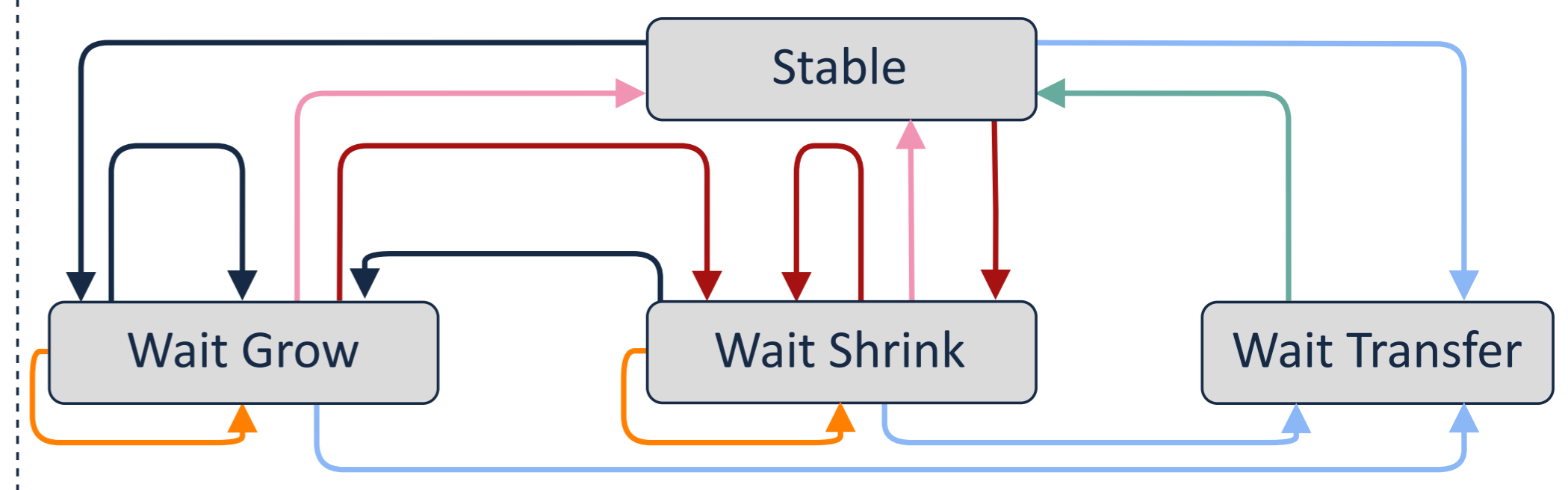
Static DRAM Division Limitation

SVM-Spark
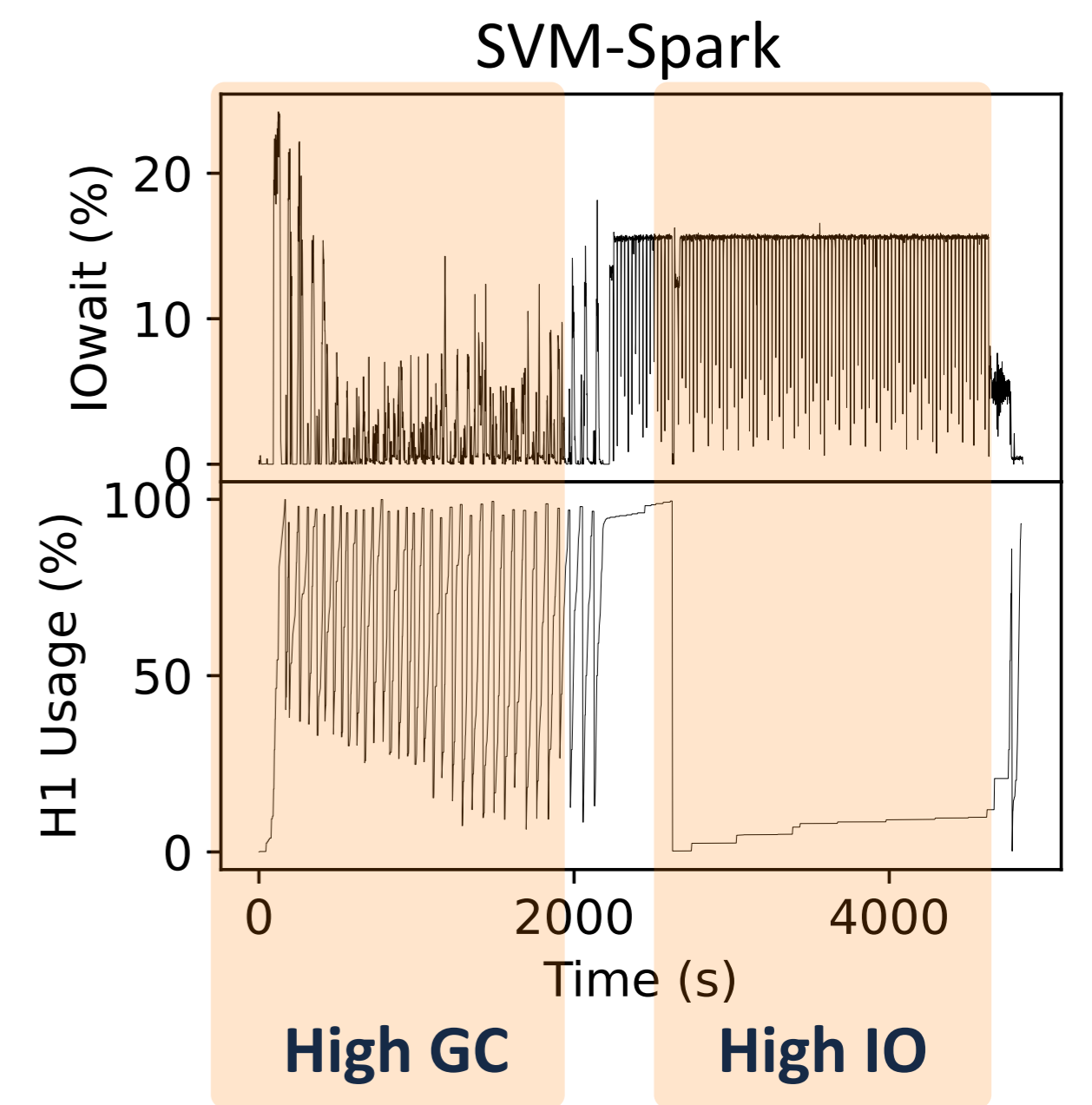
- Satic DRAM division **cannot cope with changing application behavior**
- High GC: need space for the first heap
- High IO: need space for cache
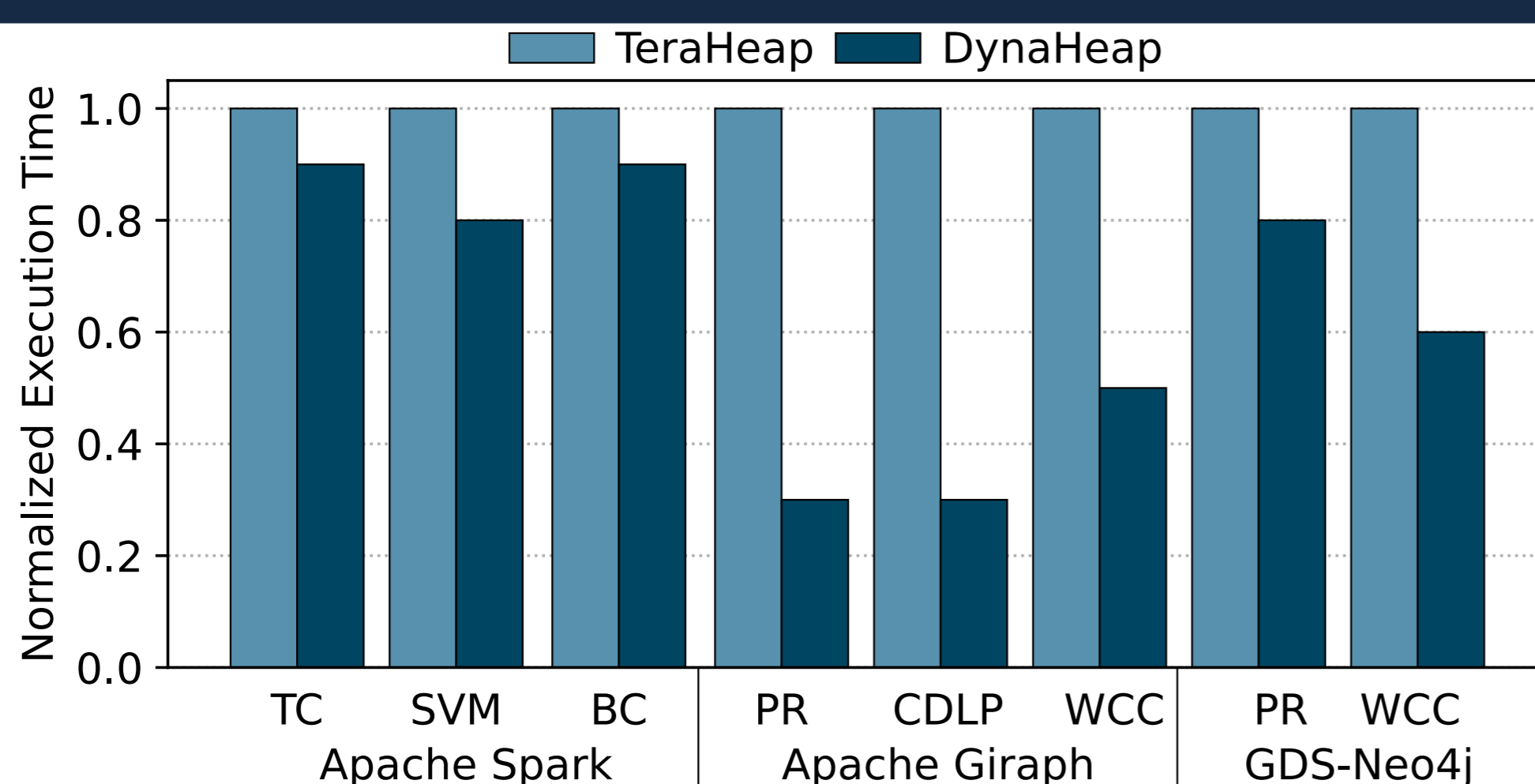
DynaHeap: Dynamic Division of DRAM
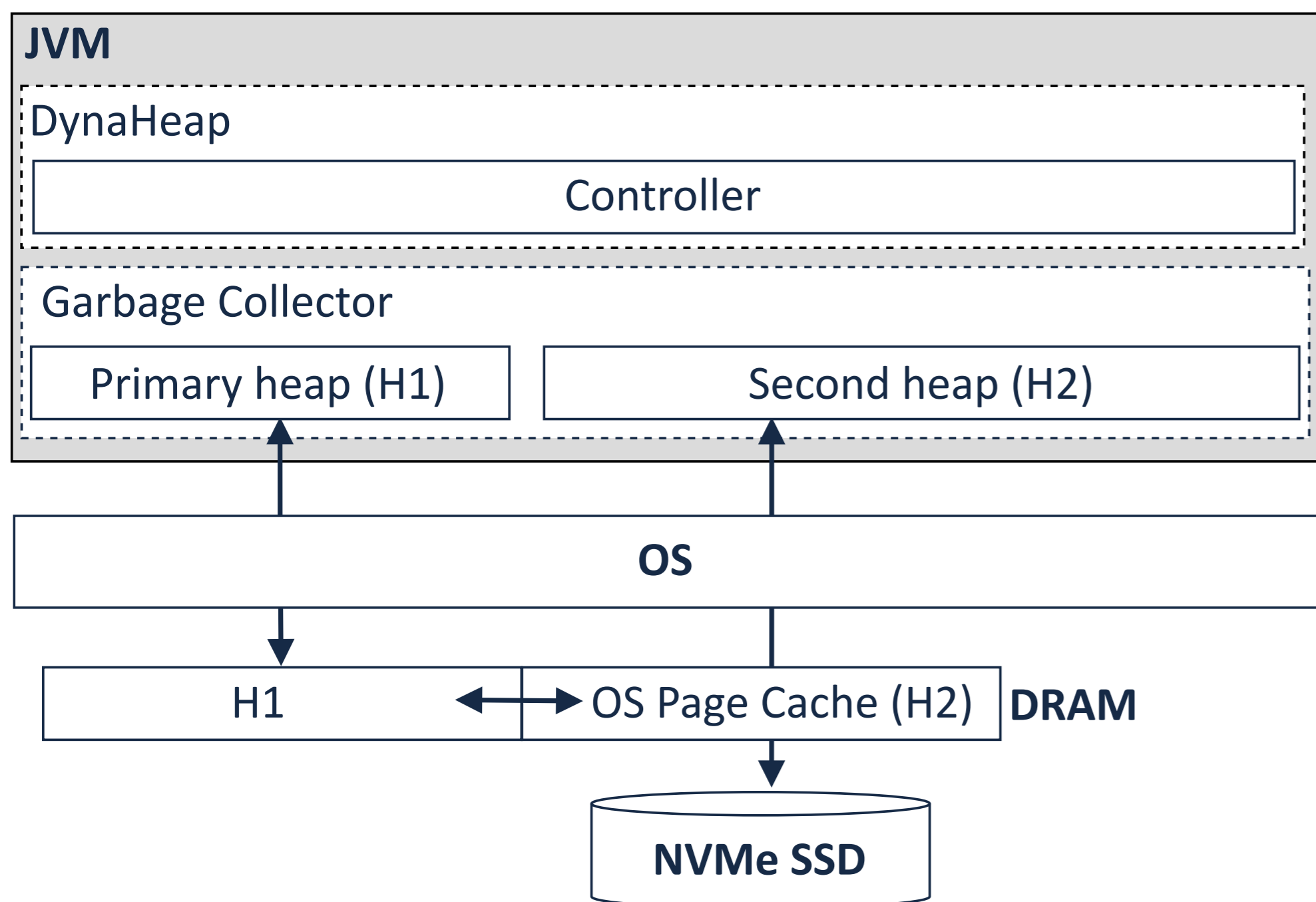
Controller: Mechanism to adjust memory between H1 and I/O cache

	Condition	Action
→ (black)	High GC	Grow H1
→ (light blue)	High GC and many objects can be moved to H2	Move H2
→ (red)	High I/O	Shrink H1
→ (orange)	Unused memory	-
→ (green)	Next GC event	-
→ (pink)	Reset actions	-

Preliminary Results

Legend: TeraHeap, DynaHeap

Key Takeaways

- Applications have different memory requirements at different periods
- Static division of DRAM between H1 and the cache for H2 cannot adatpt to dynamic changing application behavior
- DynaHeap is on average 70% better than TeraHeap

AERO