MICA: A Holistic Approach to Fast In-Memory Key-Value Storage

Hyeontaek Lim (CMU), Dongsu Han (KAIST), David G. Andersen (CMU), Michael Kaminsky (Intel Labs)

Efficiency: Important and Hard

In-memory key-value store: All key-value items in DRAM
- Examples: Memcached, MemC3, RAMCloud, Masstree, ...
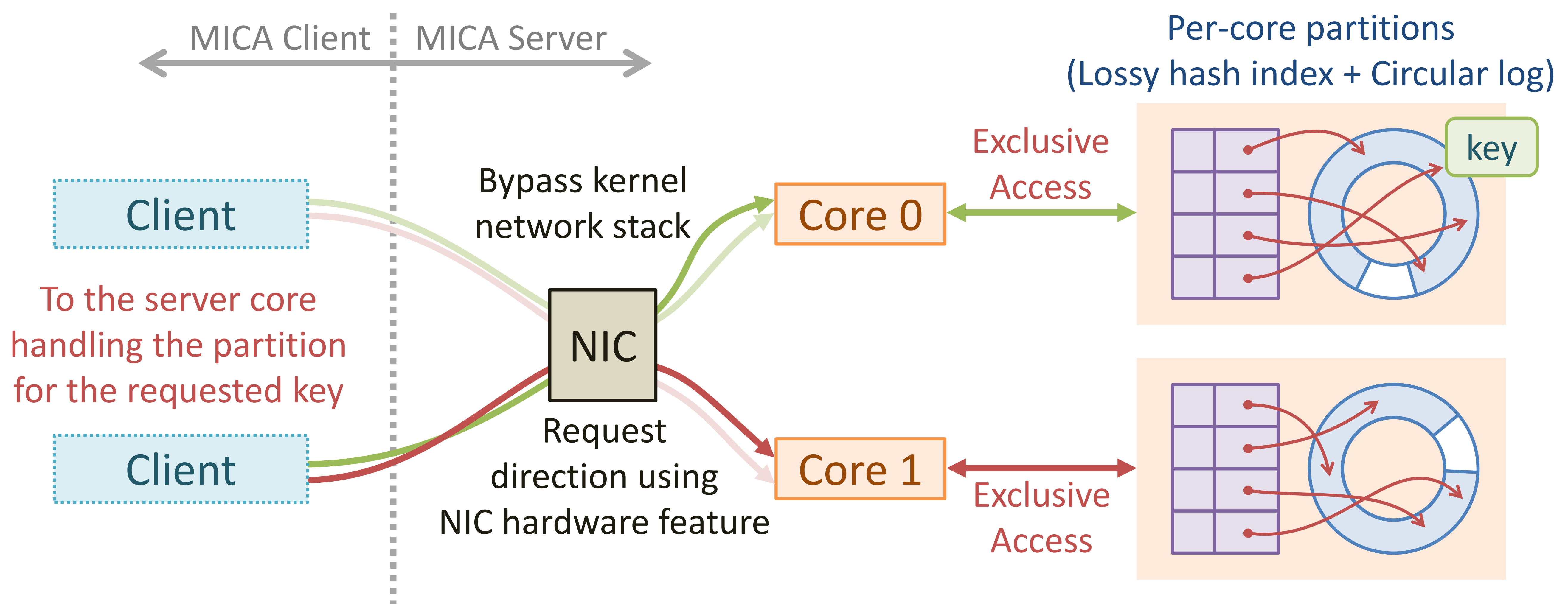- Goals: High throughput, low latency, low memory overhead

Main challenges: Performance and space efficiency
- Expensive concurrent writes
  - Existing solutions only for read-mostly workloads
- Network stack overhead
  - Partial solution (request batching) does not scale
- Memory fragmentation or expensive garbage collection
  - Low capacity and/or low performance
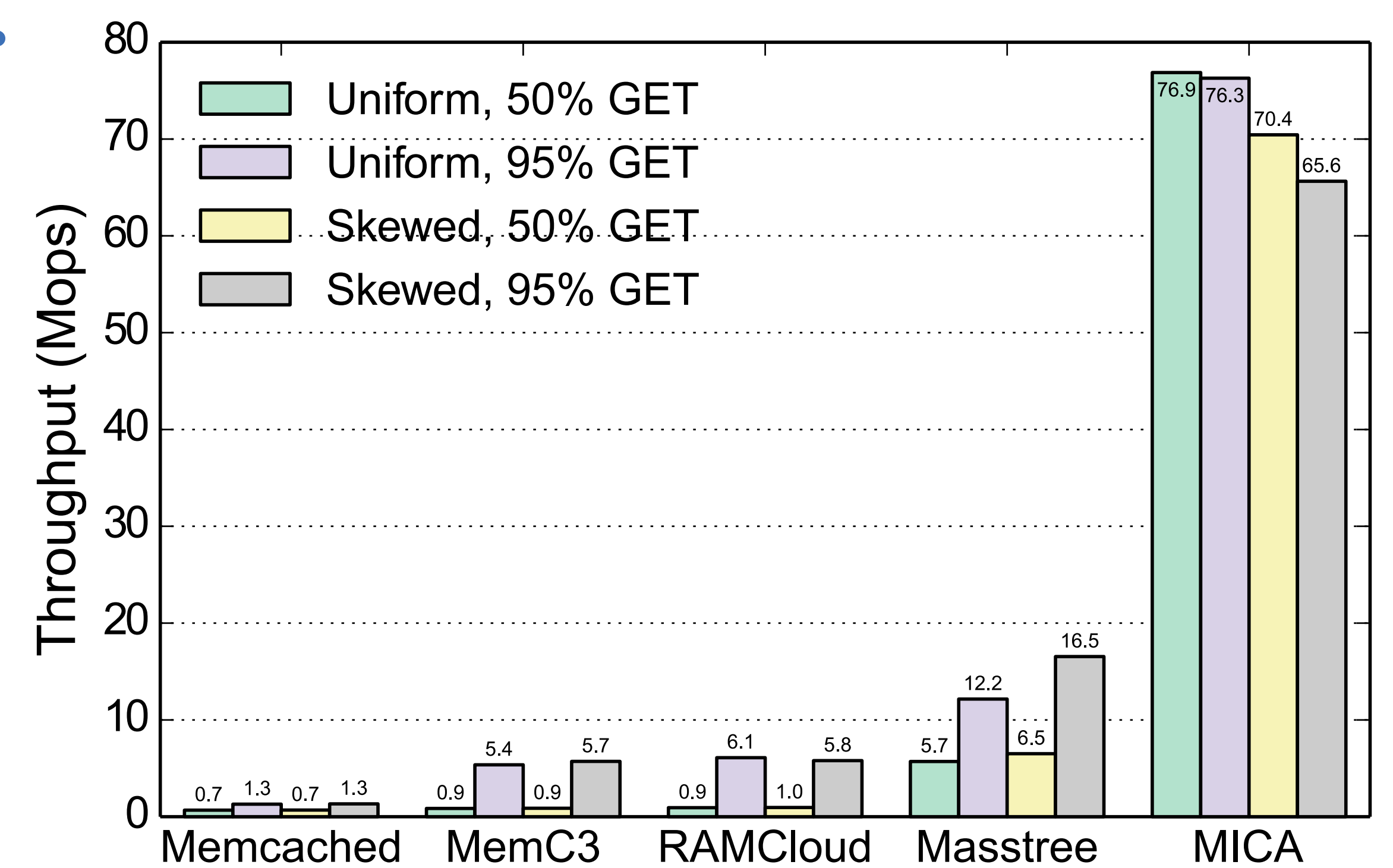
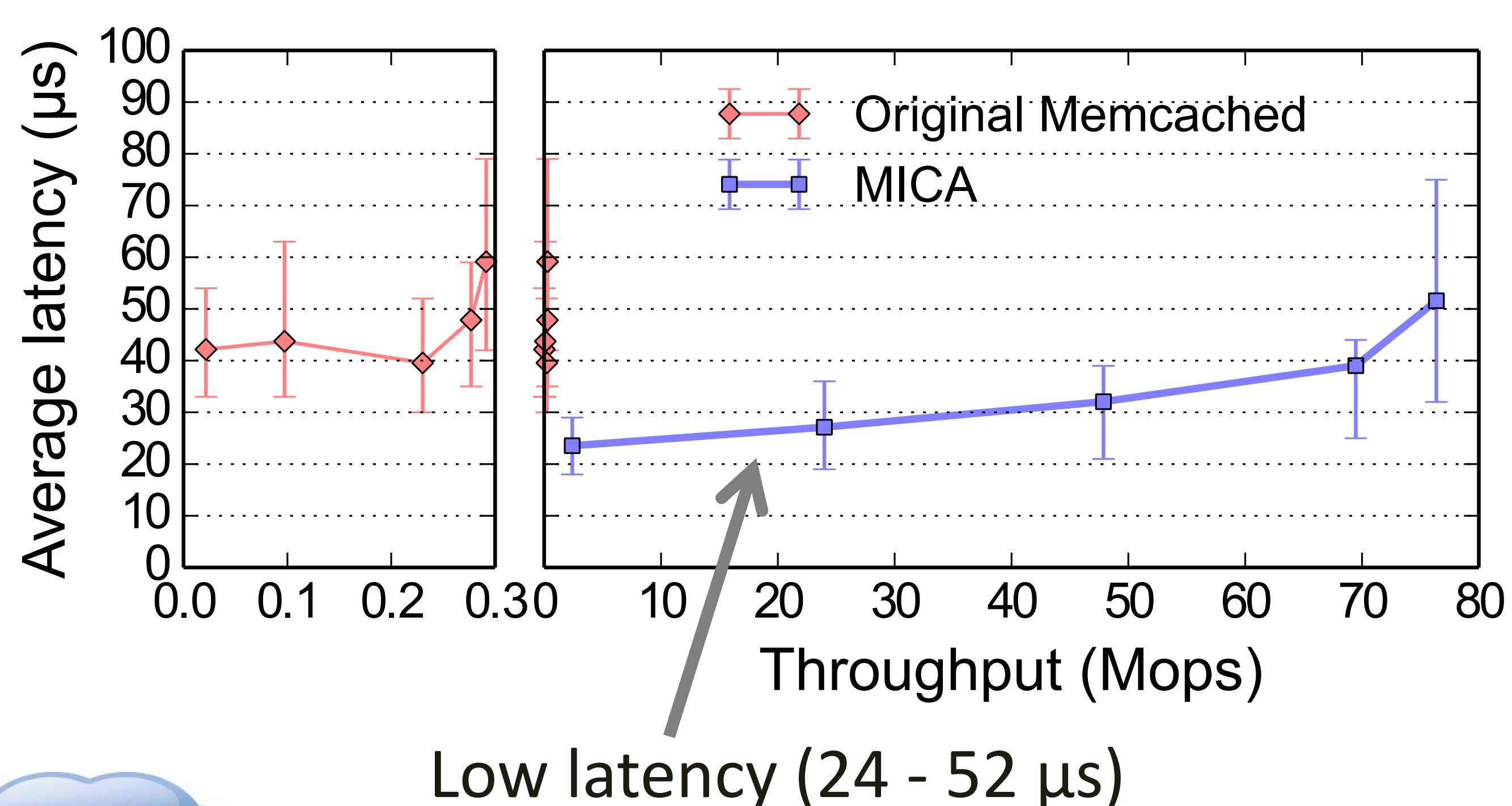**Full-system performance over the network**

All systems use our network stack   No batching

16-byte items   Zipf-distributed key popularity

MICA Design

MICA Client | MICA Server

To the server core handling the partition for the requested key

Bypass kernel network stack

Request direction using NIC hardware feature

Exclusive Access

Per-core partitions
(Lossy hash index + Circular log)

End-to-End Latency on Ethernet

Low latency (24 - 52 μs)

CPU Scalability

50% GET

Almost linear scalability