**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

**MICA Design**

- **Client**
  - To the server core handling the partition for the requested key

- **NIC**
  - Bypass kernel network stack
  - Request direction using NIC hardware feature

- **Core 0**
  - Exclusive Access

- **Core 1**
  - Exclusive Access

**End-to-End Latency on Ethernet**

- **Low latency (19 - 51 µs)**

**CPU Scalability**

- **Almost linear scalability**

**Full-system performance over the network**

- All systems use our network stack
- No batching
- 16-byte items
- Zipf-distributed key popularity

**Per-core partitions**

- (Lossy hash index + Circular log)