Using RDMA Efficiently for Key-Value Services
Anuj Kalia (CMU), Michael Kaminsky (Intel Labs), David G. Andersen (CMU)

TODAY’S SYSTEMS USE > 1 RDMA READ TO ACCESS REMOTE HASH-TABLES

BACKGROUND
In-memory key-value services
- Interface: GET(key), PUT(key, value), DELETE(key)
- Data is stored in RAM
- A key is mapped to a pointer using an index (hash table, tree). Value is stored at the pointer
- Examples: Memcached, Redis, RAMCloud

OUR APPROACH: DON’T PAY MICROSECONDS TO SAVE NANOSECONDS

Let the server traverse the data structure
- Memory access latency (~ 100 ns) << RDMA read latency (~ 2-3 µs)
- Borrow lossy index and circular log data structures from MICA[3]
- Clients write requests to the appropriate server core using RDMA writes
- Server computes response and replies with a SEND message over a datagram connection

Evaluation: Comparison against stripped versions of Pilaf and FaRM-KV

Throughput:
- HERD delivers 26 Mops with 5 µs average latency
- Over 2X higher throughput than Pilaf and FaRM (with variable length keys)
- Average latency over 2X lower than Pilaf’s and FaRM-KV’s at their peak throughput

Paper and code:
[1] Using One-Sided RDMA Reads to Build a Fast, CPU-Efficient Key-Value Store. (Christopher Mitchell, Yifeng Geng, Jinyang Li), ATC 2013
[3] MICA: A Holistic Approach to Fast In-Memory Key-Value Storage (Hyeontaek Lim, Dongpu Han, David G. Andersen, Michael Kaminsky) NSDI 2014

Design:
- Borrow lossy index and circular log data structures from MICA[3]
- Clients write requests to the appropriate server core using RDMA writes
- Server computes response and replies with a SEND message over a datagram connection

Evaluation: Comparison against stripped versions of Pilaf and FaRM-KV

Throughput:
- HERD delivers 26 Mops with 5 µs average latency
- Over 2X higher throughput than Pilaf and FaRM (with variable length keys)
- Average latency over 2X lower than Pilaf’s and FaRM-KV’s at their peak throughput

Paper and code:
[1] Using One-Sided RDMA Reads to Build a Fast, CPU-Efficient Key-Value Store. (Christopher Mitchell, Yifeng Geng, Jinyang Li), ATC 2013
[3] MICA: A Holistic Approach to Fast In-Memory Key-Value Storage (Hyeontaek Lim, Dongpu Han, David G. Andersen, Michael Kaminsky) NSDI 2014