Determining a Cache Hit/Miss over RDMA

A NetCAT Replication

Emerson Ford Calvin Lee

CS 6465 - Fall 2019

NetCAT Overview

Claim

Using RDMA over Infiniband, a remote host can measure if a remote memory access is served from LLC or from DRAM on a target host with DDIO enabled.

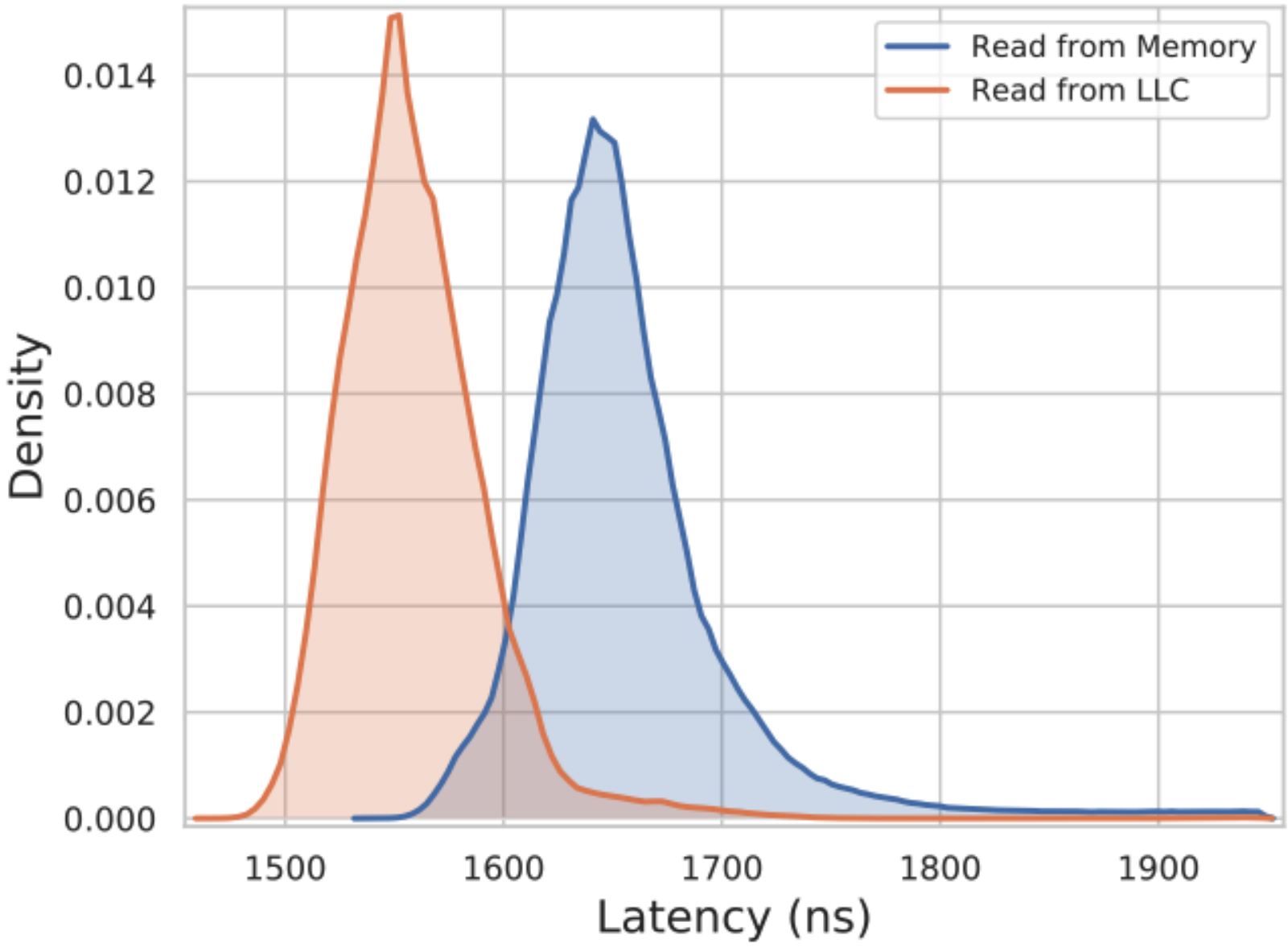
Impact

Enables cache-timing based attacks (such as PRIME+PROBE) over the network which then enables attacks like SSH keystroke timing attacks.

Key Replication Questions

1. Is it actually possible to measure cache hit/cache hit on a remote memory access?
2. If so, can we replicate their method of building a remote eviction set?

Key Graph to Replicate



RDMA Overview

1. Server and client both register memory to be used for RDMA.

Reads

2. Client specifies a remote address and fires off 'READ' verb.
3. Client NIC communicates with remote NIC to read remote memory address (no CPU involvement).
4. Client NIC places remote memory contents into client's registered memory.

Writes

2. Client alters local registered memory.
3. Client specifies a remote address and fires off 'WRITE' verb.
4. Client NIC communicates with remote NIC to write local memory contents at remote address (no CPU involvement).

Other Key Facts

DDIO

- Reads can be served from LLC or DRAM. If served from DRAM, the memory is **not** loaded into LLC.
- Writes will load memory into the LLC if not already present.
- DDIO is "restricted to 10% of the last-level cache".

Infiniband

- DRAM access and LLC access for an Infiniband NIC should take longer than a CPU's access due to PCIe communication?
- Infiniband RDMA reads (on `apt080` and `apt083`) take 1900ns on average with 50ns standard deviation.

Timing Code

Read $\rightarrow$ Write $\rightarrow$ Read a remote address.

Problems

- NUMA
- RDMA-enabled nodes are likely to be network-traffic intensive
- Prefetchers