MemcachedGPU: Scaling-up Scale-out Key-value Stores

Tayler H. Hetherington
taylerh@ece.ubc.ca
The University of British Columbia

Mike O'Connor
moconnor@nvidia.com
NVIDIA / UT-Austin

Tor M. Aamodt
aamodt@ece.ubc.ca
The University of British Columbia

Paper: ece.ubc.ca/~taylerh/doc/MemcachedGPU_SoCC15.pdf

Code: github.com/tayler-hetherington/MemcachedGPU

Problem & Motivation

- **Data centers consume significant amounts of power to operate (e.g. 10's of Megawatts).**
- **There is a continuously growing demand for higher performance in the data center.**
  - Cannot easily trade performance for power.
- **Data center hardware needs to be general to support constantly changing workloads and requirements.**
- **Ideal properties for a data center accelerator:**
  - High performance.
  - High energy-efficiency.
  - High generality and programmability.
  - Low-cost commodity hardware.

Potential Solutions & Tradeoffs

"Wimpy" cores
+ Lowers power consumption.
− Reduces performance.

ASICs
+ Very high performance and energy-efficiency.
− Lacks generality.

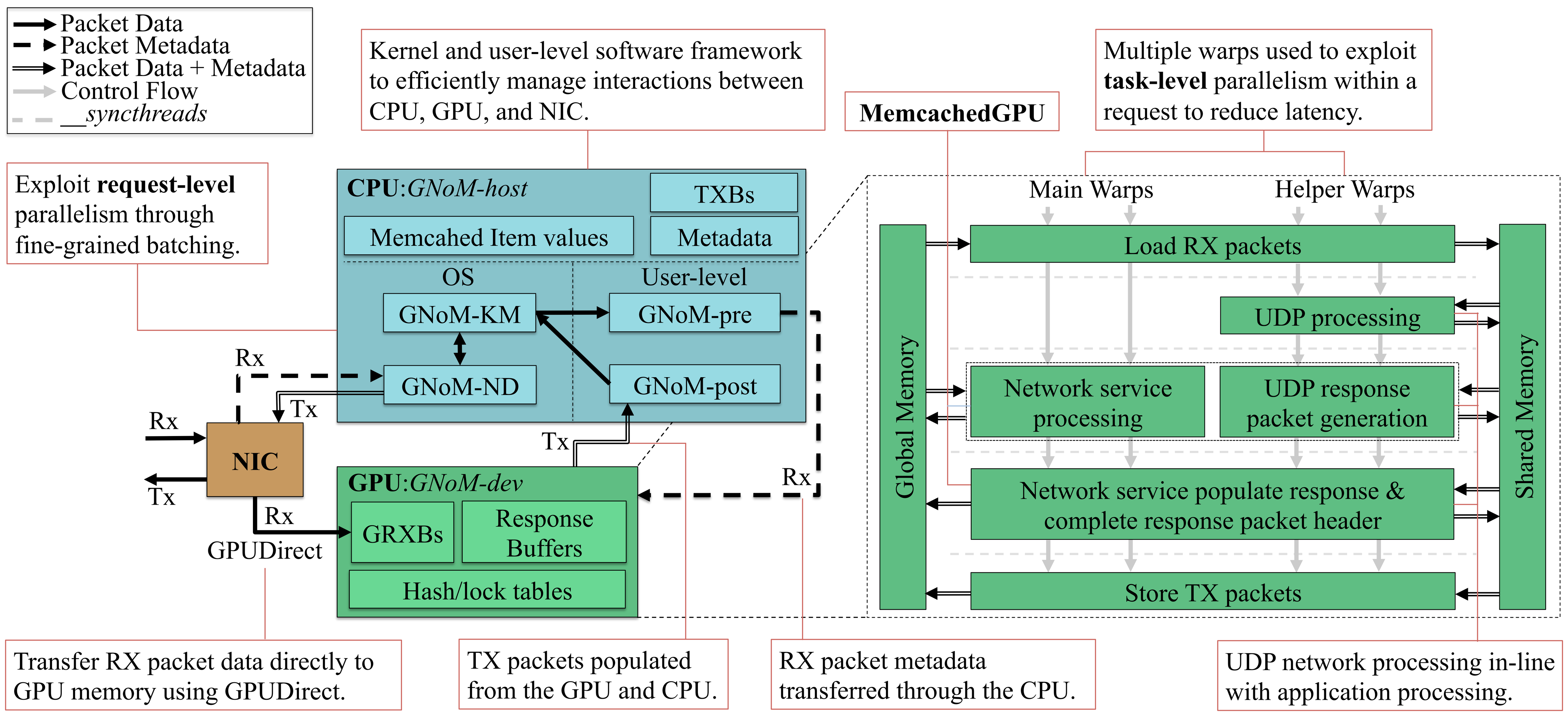
FPGAs
+ High performance and energy-efficiency.
+ Generality through reprogrammable hardware.
+ Used in Microsoft's data centers for Bing (Catapult).
− More difficult to program than general-purpose processors.
  - Programmability improving with high-level synthesis, but may trade off the quality of results.
− Reprogramming times may limit potential for fine-grained task switching to support multiple concurrent workloads.

GP-GPUs
+ High performance and energy-efficiency.
  - Improves throughput at the cost of latency.
+ General-purpose and highly programmable.
+ Positive impact on performance and energy-efficiency in supercomputing.
+ Used in Google's data centers for Machine Learning.
− SIMD architecture limits potential applications.
− Smaller main memory than CPUs.
  - Integrated GPUs may remove this limitation.

Main Goals and Contributions

- **Use GPUs as flexible, energy-efficient accelerators for general network services in the data center.**
  - Exploit request-level parallelism through request batching on the massively parallel GPU architecture.
    - Small batches (e.g., 512 requests) to improve latency.
    - Concurrent batches to improve throughput.
  - Perform both UDP network processing and application processing on the GPU.
- **GNoM: Achieve high-throughput, low-latency, and energy-efficient UDP network processing on commodity Ethernet and GPU hardware.**
- **MemcachedGPU: Design and evaluate a popular in-memory key-value store application, Memcached, on GPUs.**
  - Distributed look-aside cache to alleviate database load.
  - Requests: GET (read), SET/UPDATE/DELETE (modify).
  - **Goal:** Scale-up the GET performance of single server.

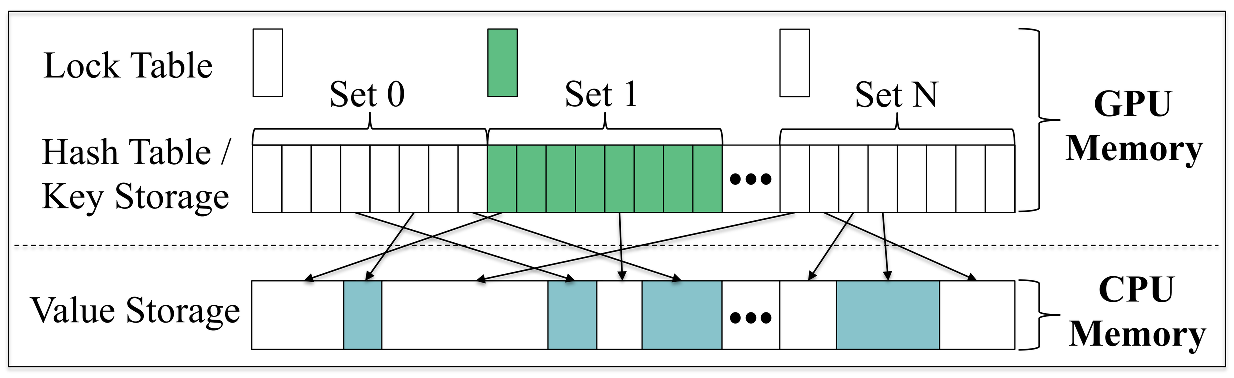
GPU Network Offload Manager (GNoM)



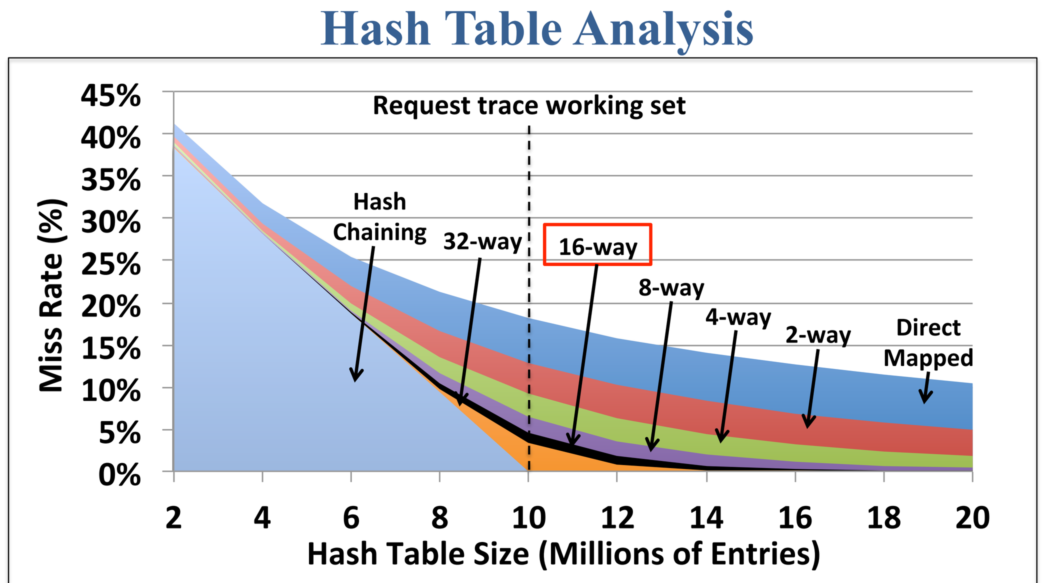
MemcachedGPU

- Focus on accelerating ***GET* requests** on the **GPU** - majority of *SET* request processing still on the CPU.
- Many changes required to the core Memcached data structures and operations to improve performance and scalability:
  - Partition key (GPU) and value storage (CPU).
  - Hash table: dynamic hash chaining → static set-associative.
  - Global LRU replacement → local per-set LRU replacement.
  - Global locking → per-set shared/exclusive locking.

Main Memcached Modifications



Hash Table Analysis



Miss-rate vs. hash table associativity and size compared to hash chaining for a request trace with a working set of 10 million requests following a Zipfian distribution. 16-ways closely matches hash chaining.

Evaluation

Peak GET Throughput Analysis

Key Size	16 B	64 B	128 B
Tesla drops @ server	0.002%	0.003%	0.006%
Tesla drops @ client	0.428%	0.043%	0.053%
Tesla MRPS/Gbps	12.9 / 9.9	8.7 / 10	6 / 10
Maxwell-NGD drops @ server	0.47%	0.05%	0.02%
Maxwell-NGD MRPS/Gbps	12.9 / 9.9	8.7 / 10	6 / 10

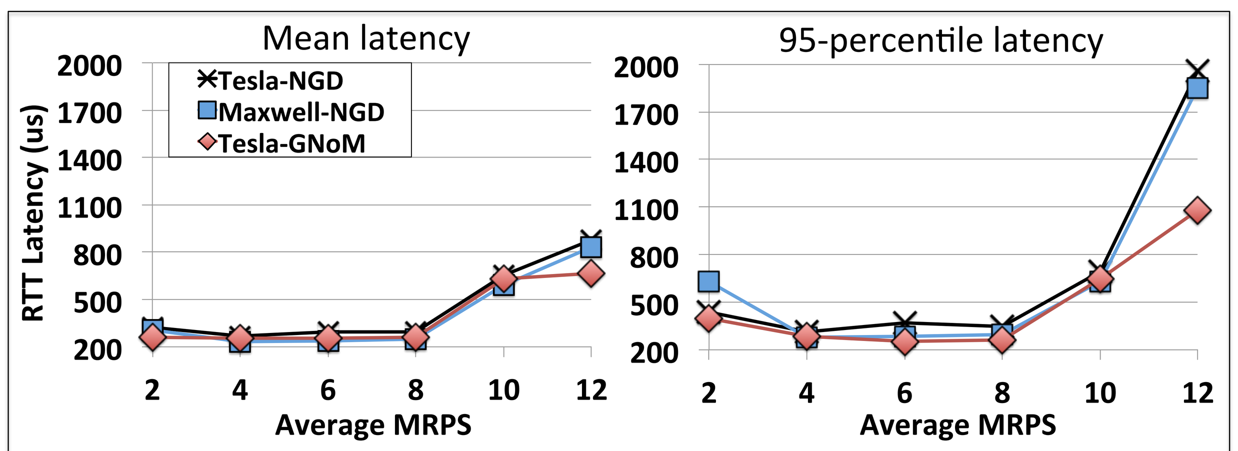
GNoM and MemcachedGPU achieve ~10 GbE processing at all key-value sizes. With varying key/value lengths, MemcachedGPU becomes network bound before compute bound.

GPUs

NVIDIA GPU	Tesla K20c	GTX 750Ti
Architecture	Kepler	Maxwell
# Cores/Freq.	2496 / 706 MHz	640 / 1020 MHz
Mem size / BW	5 GB / 208 GB/s	2 GB / 86.4 GB/s
TDP	225 W	60 W
Cost	$2,700	$150
RX mode	GPUDirect (GNoM)	Non-GPUDirect (NGD)

Evaluated a high-performance and low-power GPU. ***The low-power GPU has comparable performance with higher efficiency.***

RTT Latency Analysis



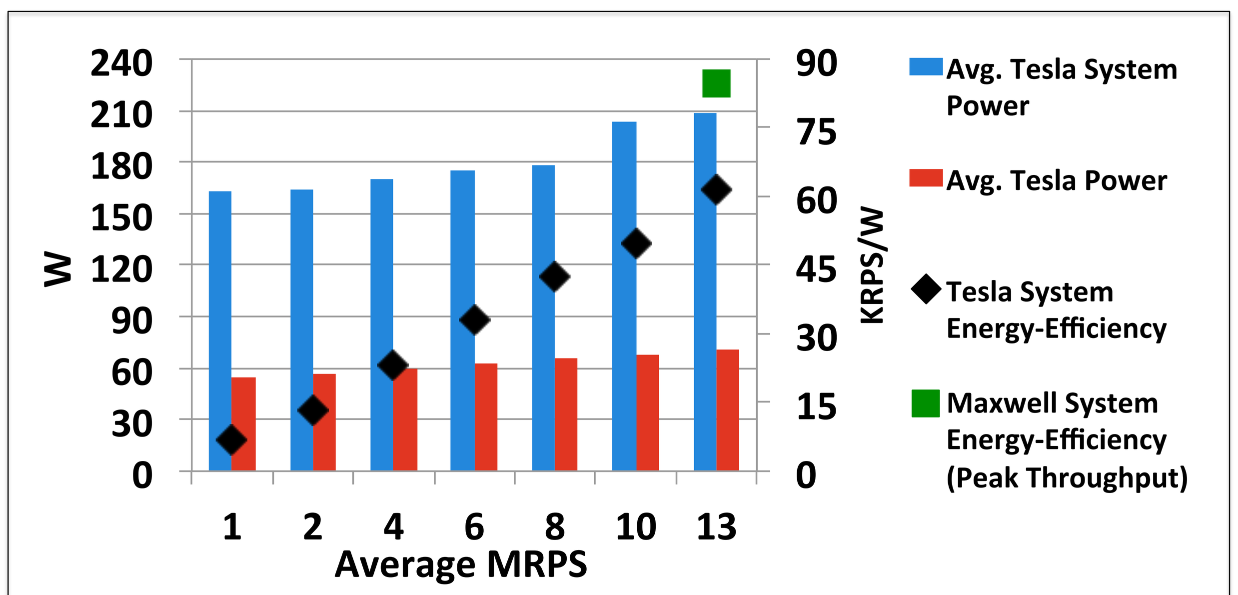
MemcachedGPU Mean and 95-percentile round-trip-time (RTT) latency with 512 requests/batch. GNoM reduces mean latency vs. NGD by 75%-96%.

Offline Analysis

NVIDIA GPU	Tesla K20c	GTX 750Ti
Throughput (MRPS)	27.5	28.3
Avg. Latency (us)	353.4	263.6
Energy-efficiency (KRPS/W)	100	127.3

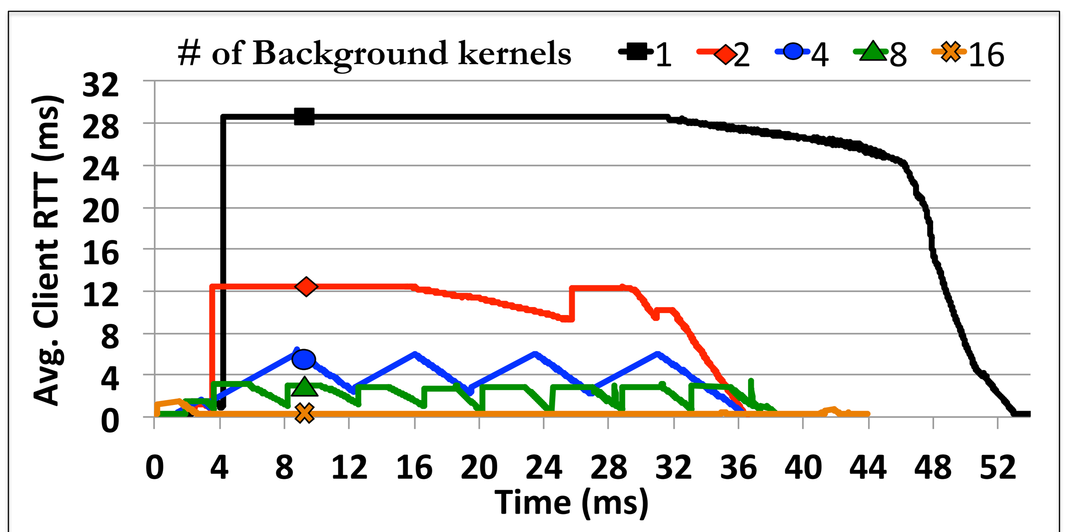
MemcachedGPU offline, in-memory limit-study without network transfers. Results show promise for even lower power integrated GPUs in the data center.

Power and Energy-Efficiency Analysis



System wall power and energy-efficiency of MemcachedGPU. The Tesla GPU only consumes 32% of peak TDP (underutilizing GPU resources). There are opportunities to further improve total system energy-efficiency through additional GPU I/O and system software support.

Workload Consolidation Analysis



Impact on RTT when running a low-priority background task (BGT) on the same GPU with MemcachedGPU at 4 MRPS. The BGT is split into smaller kernels with fewer CTAs (256 CTAs total). 16 CTAs per BGT kernel reduces the max client RTT by 18X, while increasing the BGT execution time by 50%.