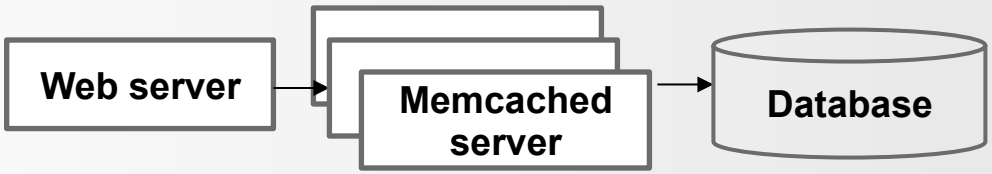
Achieving 10Gbps Line-rate Key-value Stores with FPGAs

Michaela Blott, Kimon Karras, Ling Liu, Kees Vissers - Xilinx Research
Jeremia Baer, Zsolt Istvan - ETH Zurich

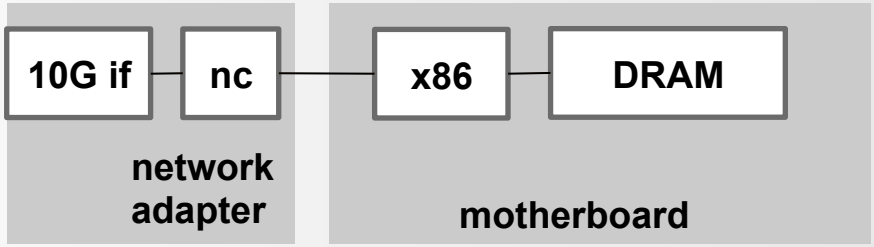
Introduction

- Common middleware application running on x86s to alleviate bottlenecks on databases



- Agenda
  - State of the art
  - FPGAs
  - Proposed architecture
  - Results
  - Limitations

Typical Memcached Implementations



- Best published numbers:

Platform	RPS [M]	Latency [us]	RPS/W [K]
Intel Xeon (8 cores)	1.34	200-300	7
Intel Xeon (2 sockets, 4 cores)	3.15	200-300	11.2
Memcached & Intel Xeon (2 sockets, 16cores)	1.8	12	Unknown
TilePRO (64 cores)	0.34	200-400	3.6
TilePRO (4x64 cores)	1.34	200-400	5.8
Chalamalasetti (FPGA)	0.27	2.4-12	30.04

1.4MRPS

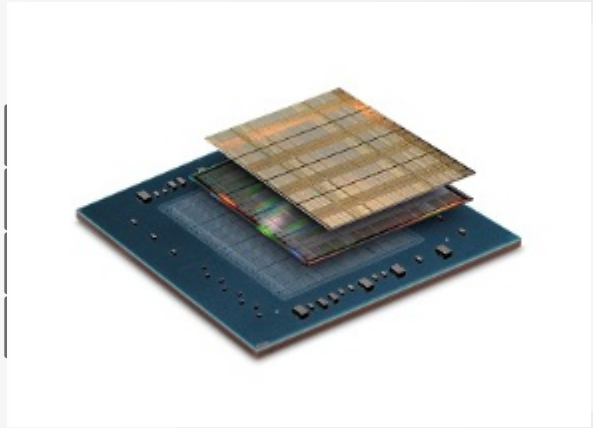
200us latency

7K

Source: see paper

What are FPGAs?

- Semiconductor devices that consist of a matrix of configurable logic blocks (CLBs) connected via programmable interconnects.

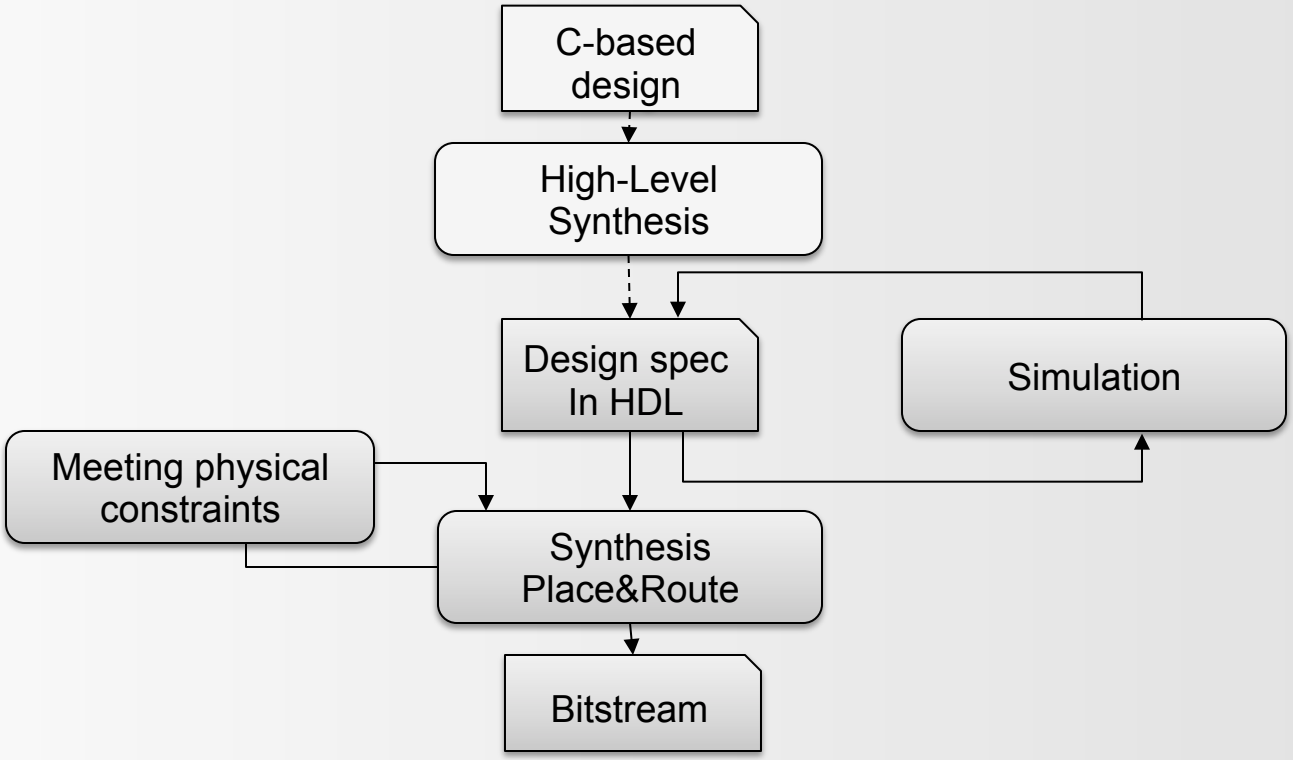


- On an extremely large scale – highly integrated (28nm, 2million logic cells)
- Nowadays, FPGAs integrate many other blocks
  - Up to 3600 DSPs, 68Mb SRAM, 96 high-speed serial IO, ARM processors (Cortex-A9)
- They can be programmed to become ANY custom circuit
  - Ideal for architectural exploration

We use FPGAs to implement a new dataflow-based architecture for memcached

Programming FPGAs?

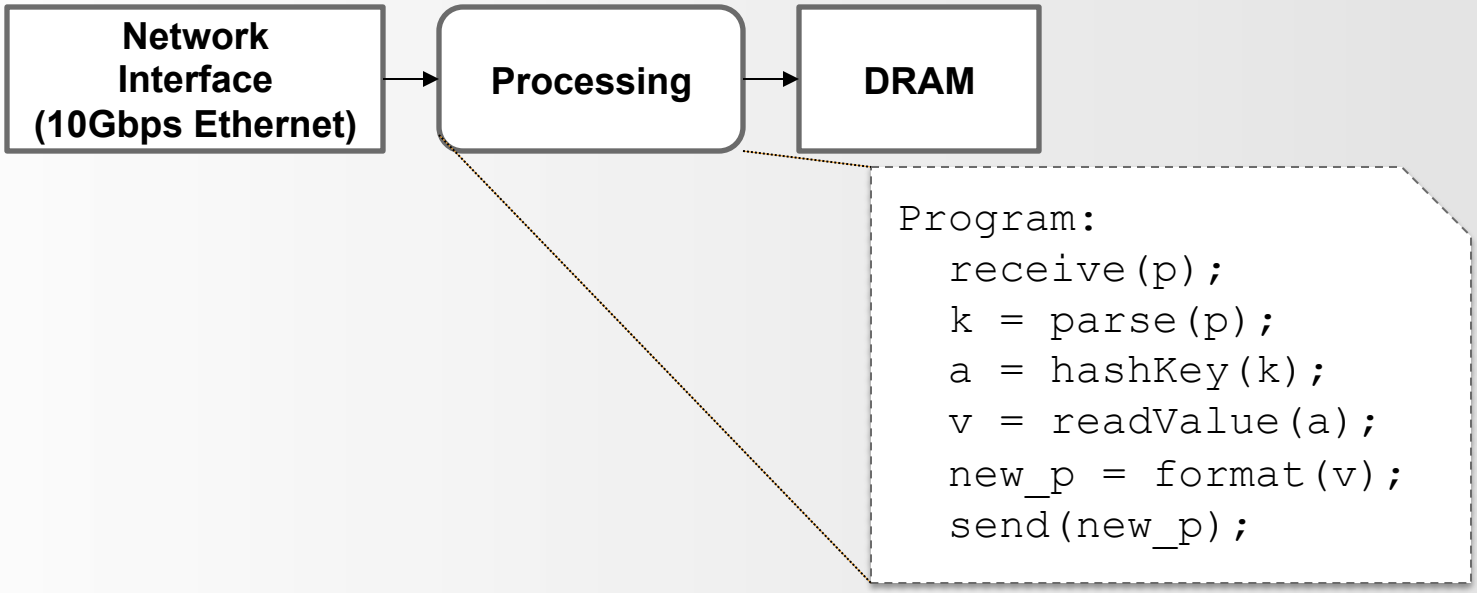
- FPGAs can be reprogrammed after manufacturing through hardware-based design flow



Hardware design exposes a greater complexity to the user and requires therefore more engineering effort

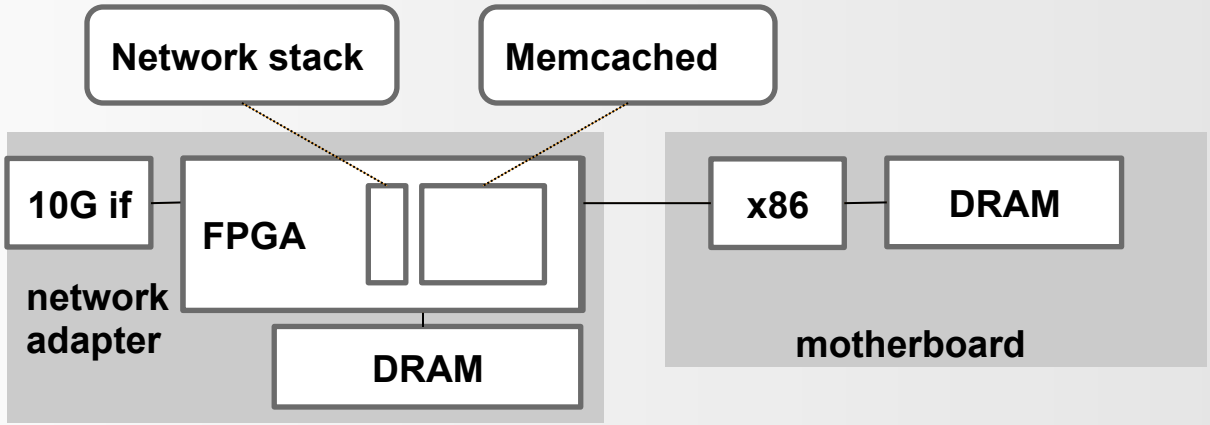
Why Dataflow Architectures?

- **Memcached is fundamentally a streaming problem**
  - Data is moved from network to memory and back with little compute



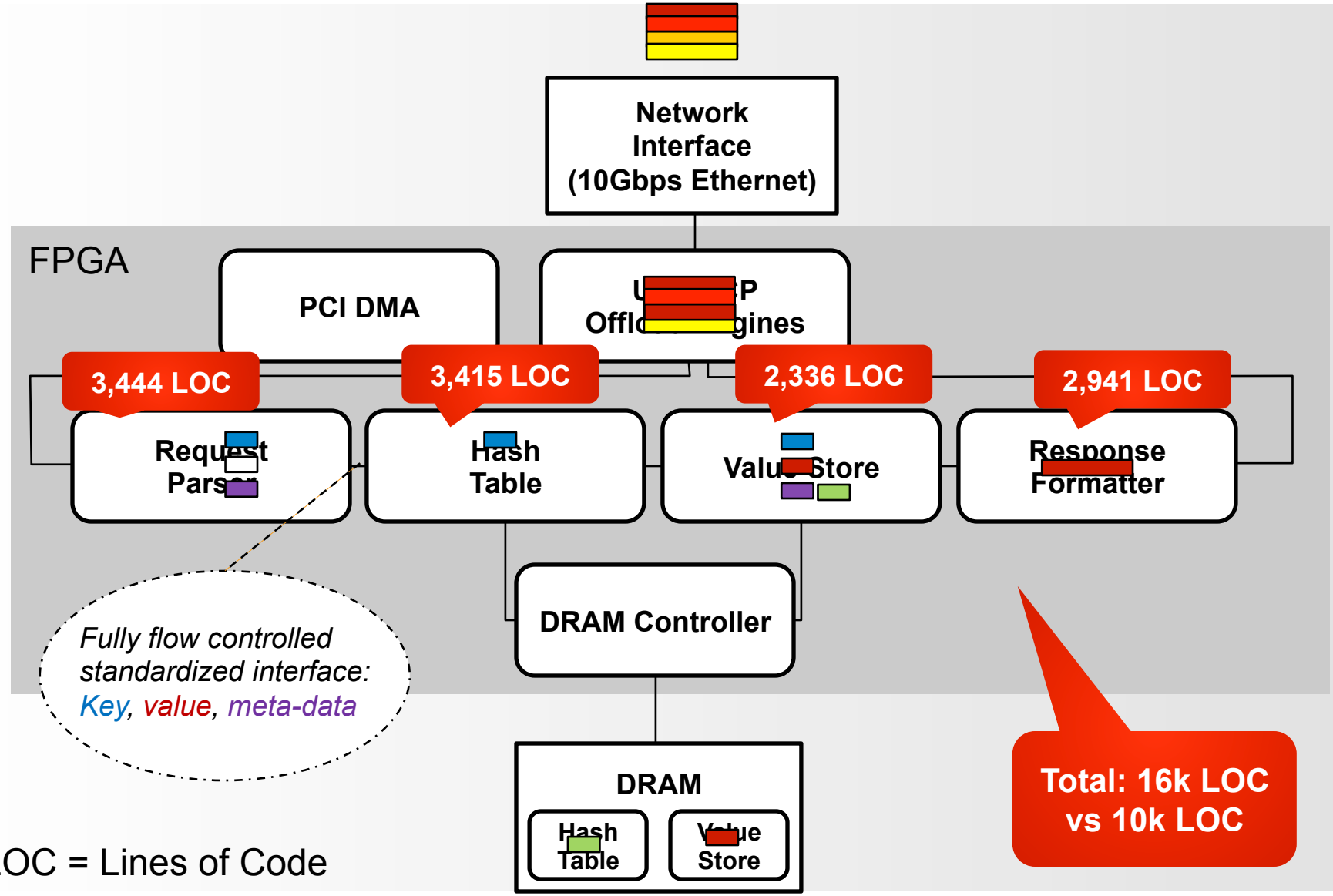
- **Dataflow architectures, frequently used for network processing, should be well suited towards the application**

System-level Architecture



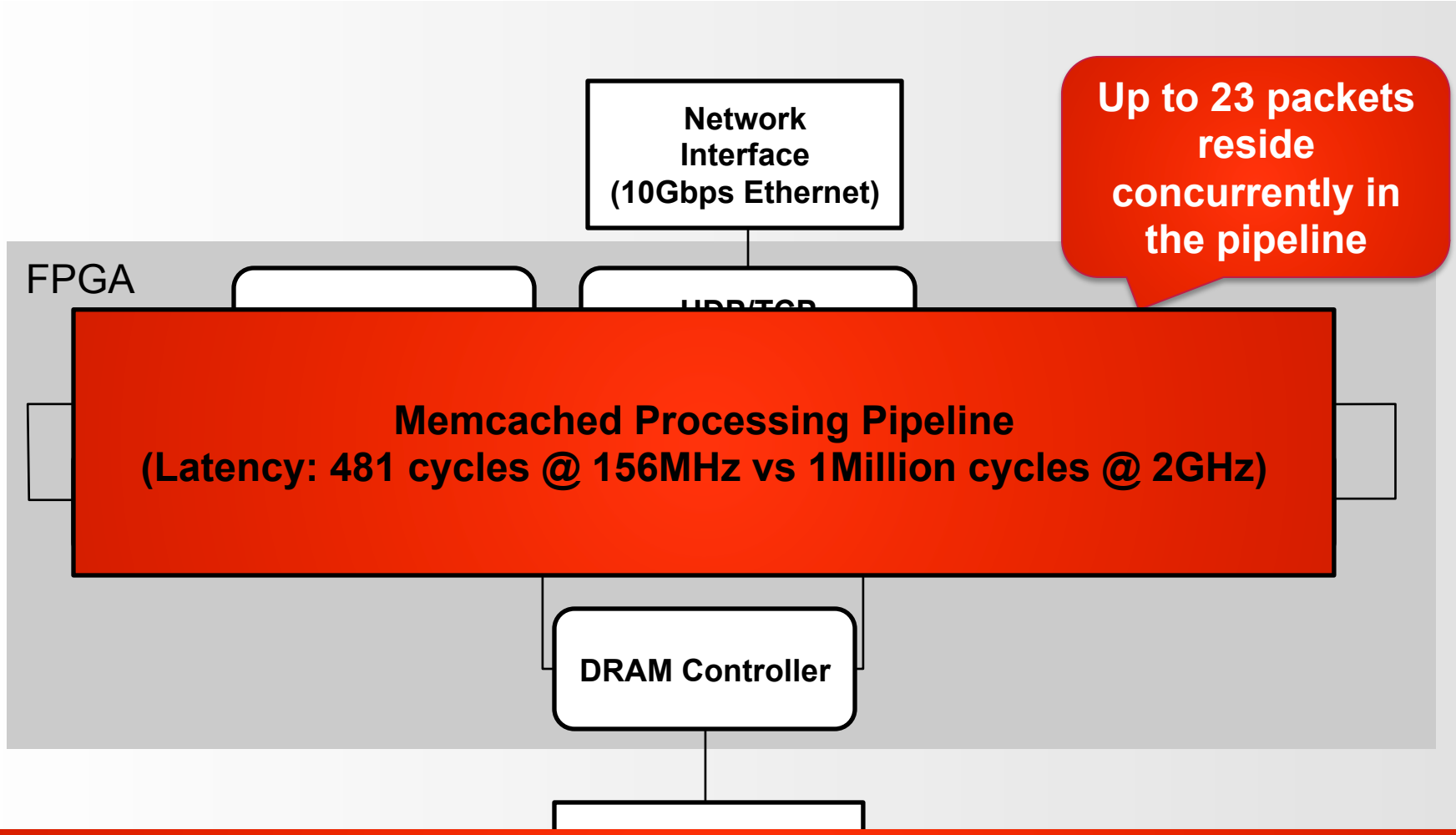
- High throughput through data-flow architecture
- Low latency through tight integration of network, compute and memory
- Hash table to support 2 million entries and 24GB of storage

FPGA-based Dataflow Architecture



LOC = Lines of Code

FPGA-based Dataflow Architecture

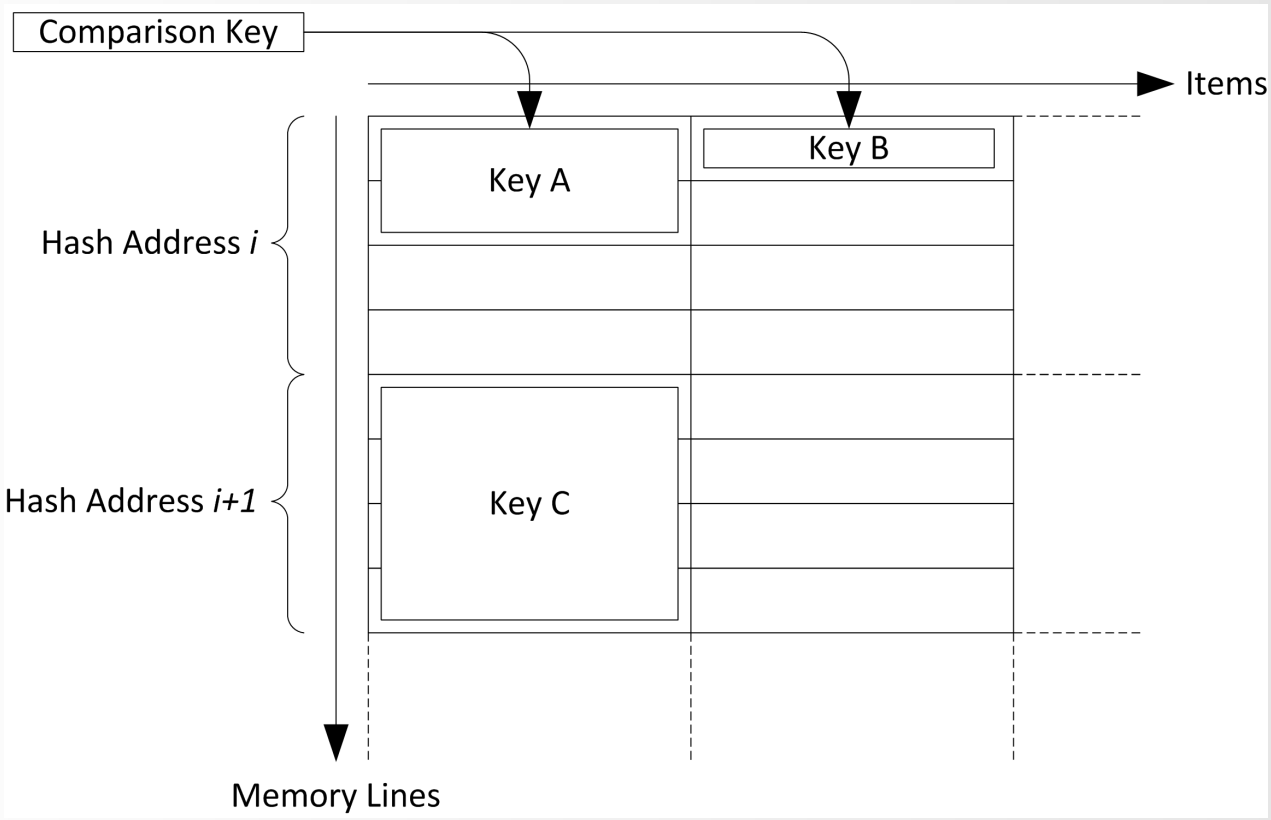


Exploiting task and instruction-level parallelism increases throughput and is more power efficient

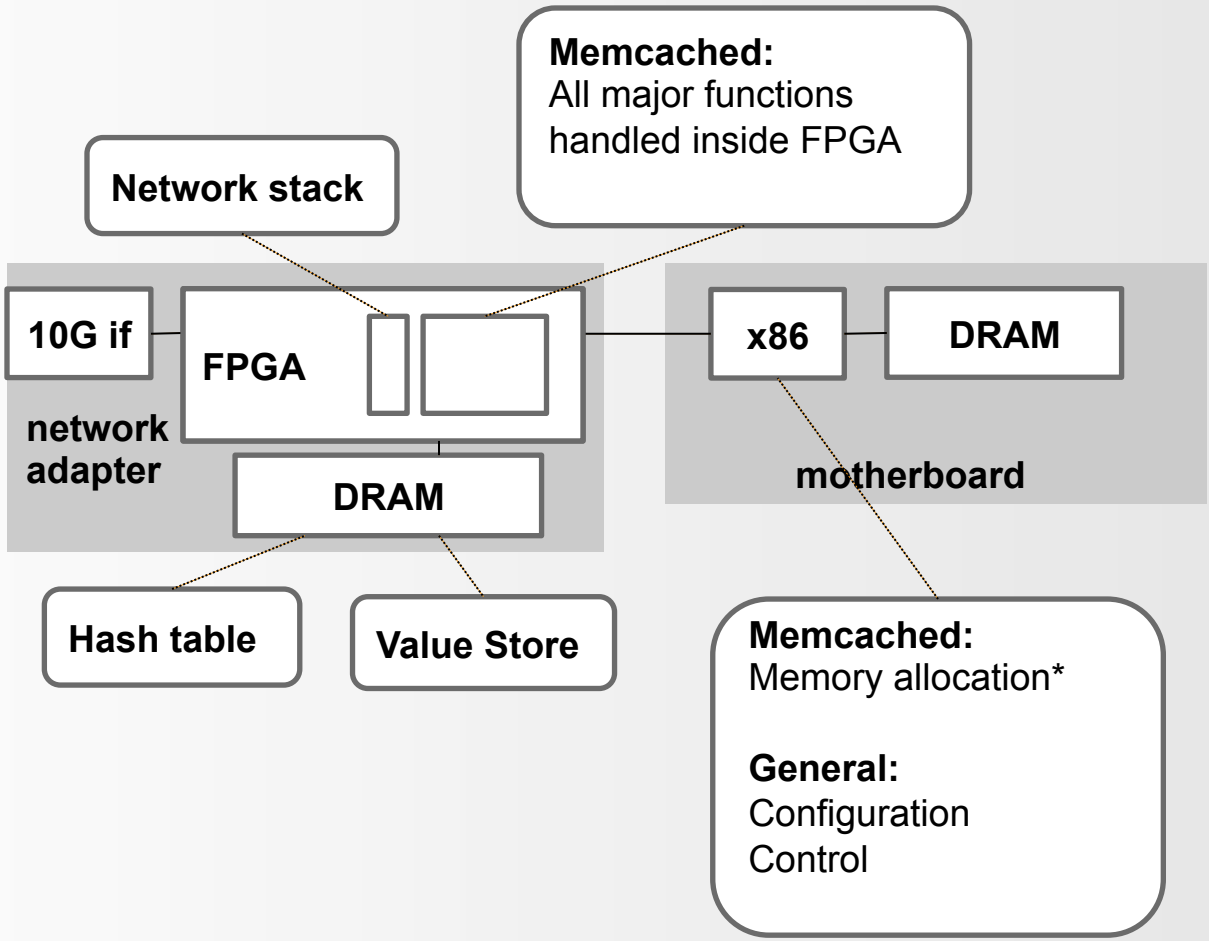
Inherently scalable

Hash Table architecture

- Bob Jenkins lookup3 implemented in FPGA
- Collision handling through parallel lookup (8-way)
- Flexible key handling through striping

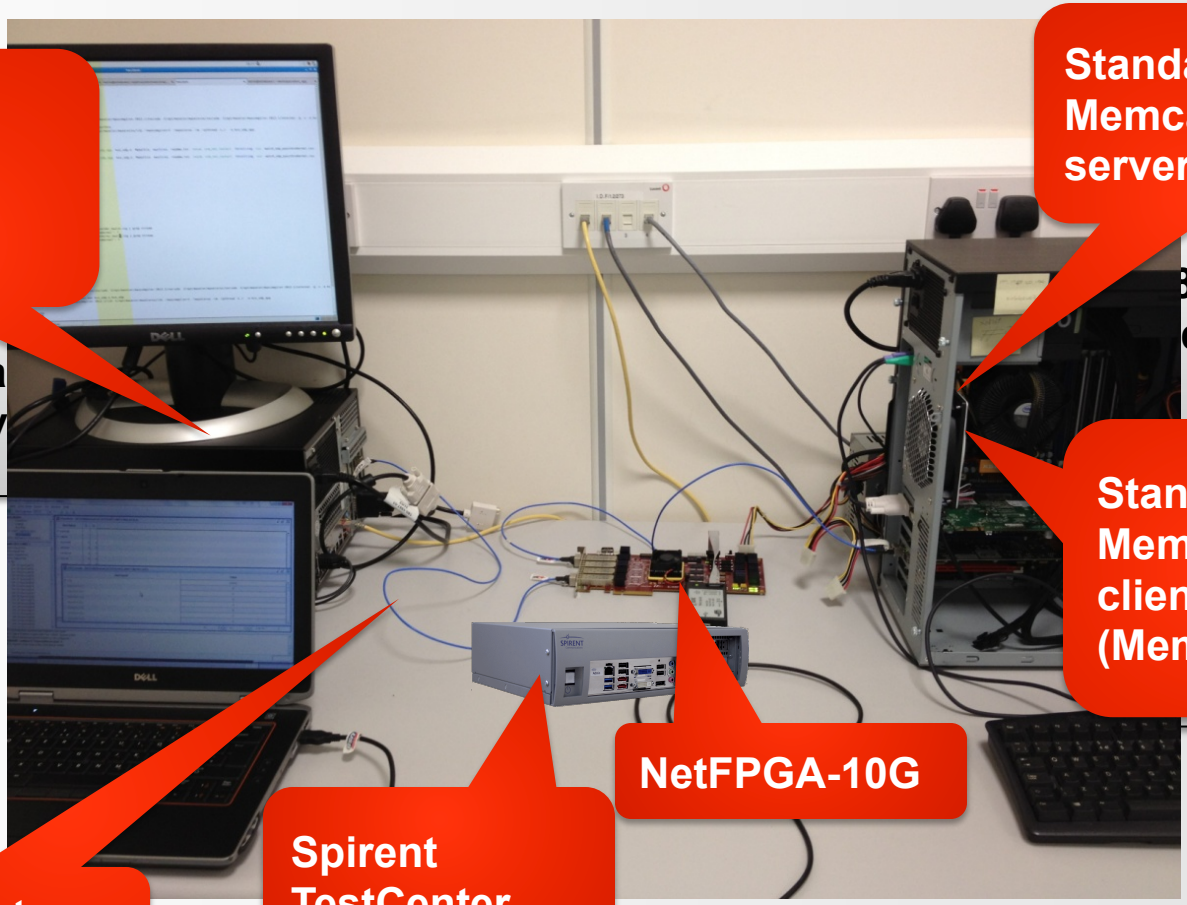


System-Level Architecture



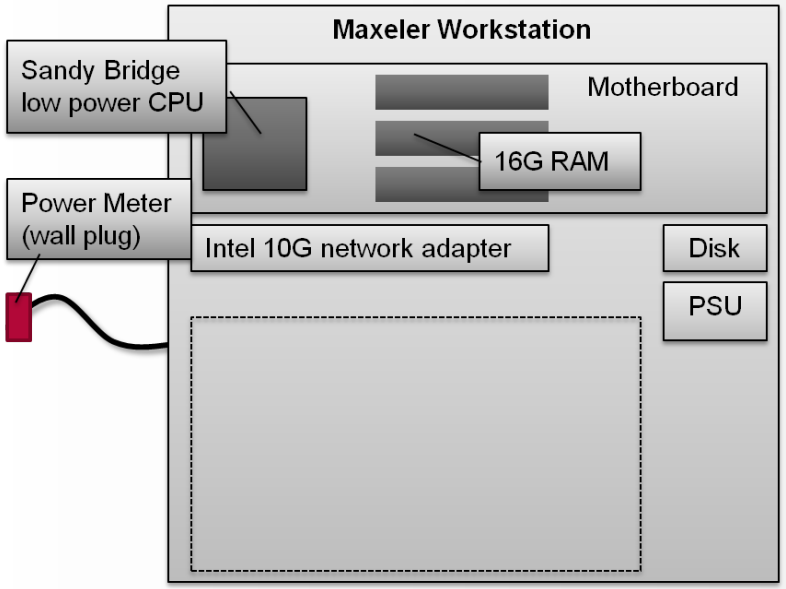
*below 3% of 1 core for 10% SET operations
*limited memory access bandwidth on platform

System Test Setup

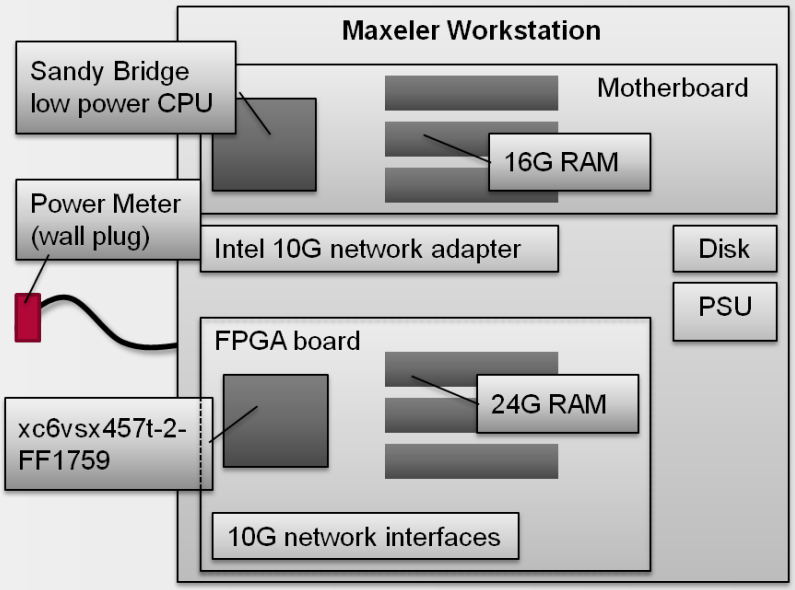


Power - Test Setup & Results

Test system 1: without FPGA board

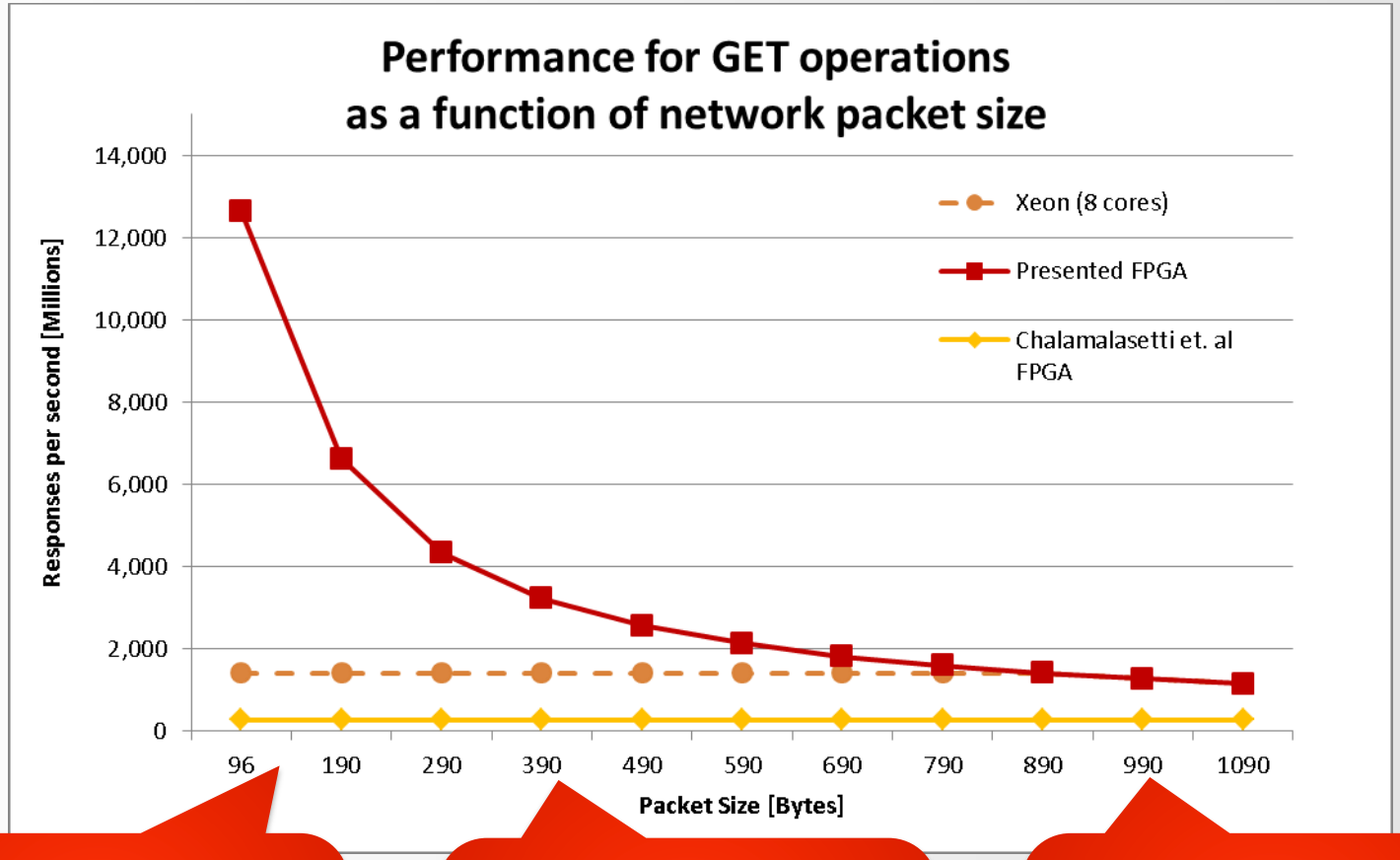


Test system 2: with FPGA board



*(Power sourced from: power plug meter, xpower, data sheets and power regulator readings)
**(UDP, binary protocol)
***(includes FPGA and host system)

Results - Performance



FPGA delivers constant 10Gbps performance – network becomes the bottleneck

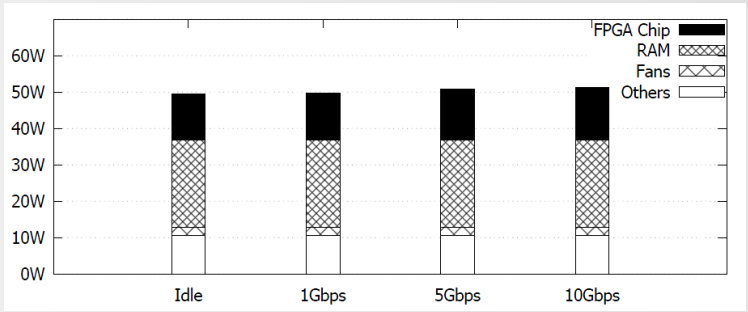
X86 performance limited by a per packet overhead

Set performance saturates network as well

*** UDP, GET, BINARY**

First Results of Memcached Evaluation

- **Sustained line rate processing for 10GE – 13MTps possible**
  - Significant improvement over latest x86 numbers
- **Lower power**
- **Combined: 36x in RPS/Watt with low variation**
- **Cutting edge latency**
  - microseconds instead of 100s of microseconds

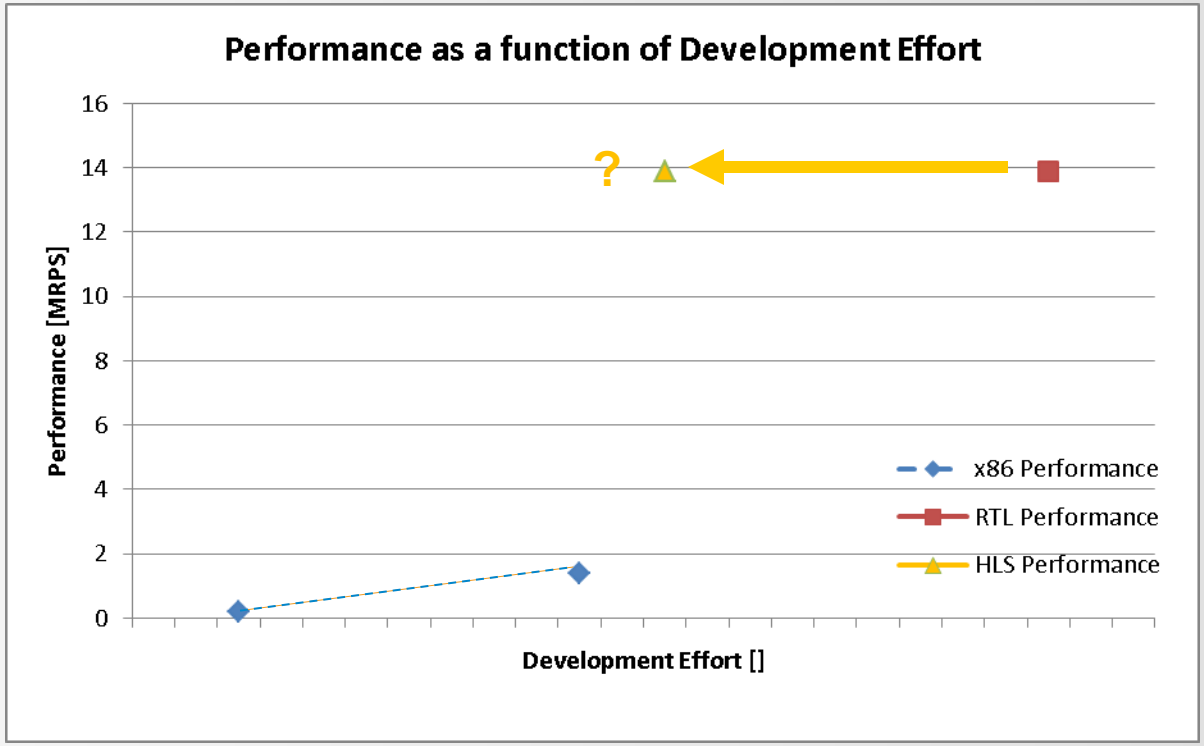


Platform	RPS [M]	Latency [us]	RPS/W [K]
Intel Xeon (8 cores)	1.34	200-300	7
TilePRO (64 cores)	0.34	200-400	3.6
FPGA (board only)	13.02	3.5-4.5	254.8
FPGA (with host)	13.02	3.5-4.5	106.7

Source: see paper

Current Platform Limitations

- **Programmability**
  - HLS investigation
- **TCP offload**
  - Future work
- **Storage capacity**
  - SSD
- **Memory allocation & cache management**
  - Current platform limitation
- **Limited protocol support**



Qualitative graph only

Summary & Next Steps

- **Dataflow architecture delivers 10Gbps line-rate performance and scalability to higher rates**
- **Significantly higher RPS/Watt, with that lower TCO**
- **Minimal latency**
- **Next Steps:**
  - Addressing limitations
  - Trials with real use cases
  - Exploration of other applications

Thank You.

mblott@xilinx.com