

ALGORITHMS IN LOGIC



[HTTP://ALGO-LOGIC.COM](http://ALGO-LOGIC.COM)

*Implementing Ultra Low Latency Data Center
Services with Programmable Logic*

John W. Lockwood, CEO: Algo-Logic Systems, Inc.

<http://Algo-Logic.com> • Solutions@Algo-Logic.com • (408) 707-3740 • 2255-D Martin Ave., Santa Clara, CA 95050

Why the Move to Programmable Logic?

“There are large challenges in scaling the performance of software now. The question is: ‘What’s next?’ We took a bet on programmable hardware.”
- Doug Burger, Microsoft

• Driving Metrics in the Data Center

– Latency:

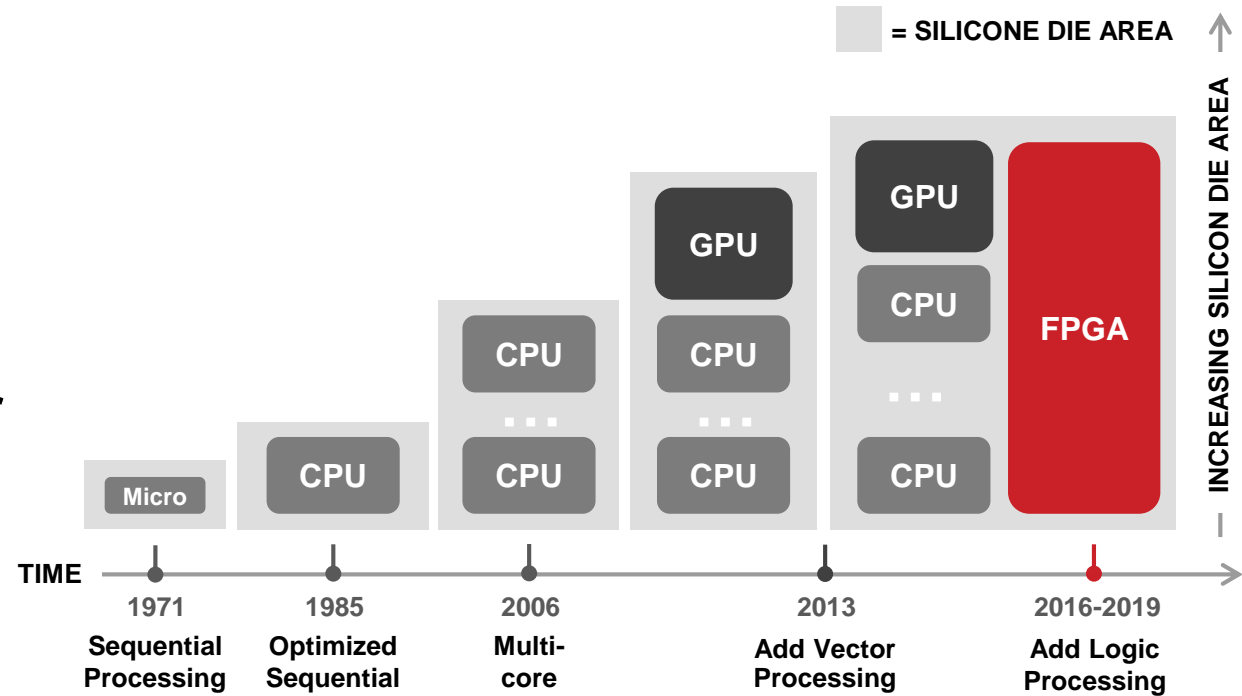
- Reduce delay
- Avoid jitter

– Throughput

- Processing packets at line rate
- Handle 10G, 25G, 40G, and 100G

– Power:

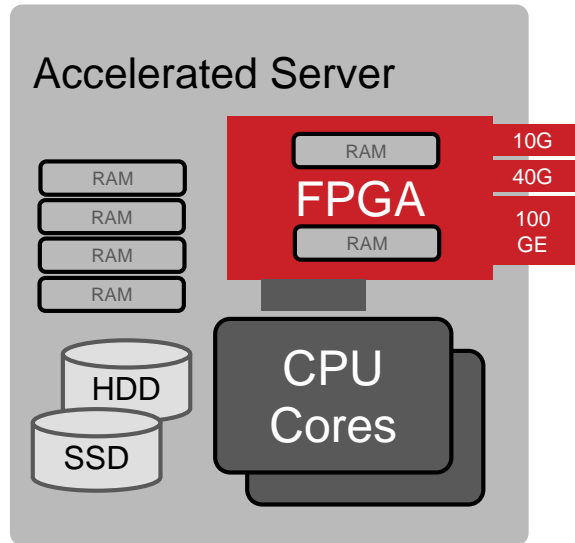
- Driving cost of OpEx



- Field Programmable Gate Array (FPGA) logic moves into the CPU
- Microsoft accelerates BING search with FPGA
- Intel acquires Altera

Servers Accelerated with FPGA Gateway

- **FPGA Augments Existing Servers**
 - Can run on an expansion card (same size as a GPU)
 - Or may be integrated into the CPU socket
- **GDN Applications run on FPGA**
 - Implements low-latency, low-power, high-throughput data processing



Example of Low Latency Service: Key/Value Store



Examples:	Key	Value
Directory	Company Algo-Logic	Phone # (408) 707-3740
Forwarding Tables	IP Address 204.2.34.5	Interface : MAC Address Eth6 : 02:33:29:F2:AB:CC
Data De-duplication	Content Hash XYZ	Storage Block ID 948830038411
Stock Trading	Order ID ATY11217911101	Symbol, Side, Price AAPL, B, 126.75
Graph Search	Virtex v140	Edge List v201, v206, v225

- **Key/Value Store (KVS)**

- Simplifies implementation of large-scale distributed computation algorithms
- Data Center Servers exchanges data over standard Ethernet

- **Challenges**

- Operating System delays packets and limits throughput
- Per-core processing inefficient at high-speed packet processing

- **Solutions**

- Bypass kernel bypass with DPDK
- Offload of packet processing with FPGA

Mobile Application Servers Need Fast Key/Value Stores

- **Scalable backend services to share data**
 - Sensors { location, bio, movement, .. }
 - Social { status, dating, updates, multi-player games .. }
 - Media { video/security, audio/music, .. }
 - Communication { network status, handoff, short messages .. }
 - Database { users, providers, payments, travel, authentication, .. }
- **Must be able to scale**
 - as the number of users grows
 - Scale up to provide the best latency, throughput, and power
 - Scale out to increase storage capacity, throughput, and redundancy
- **Example:**
 - Mobile location sharing



Case Study: Implementing Uber with KVS

- **Uber in 2014**

- 162,037 drivers in the US completed 4 or more trips
- New drivers doubled every 6 months for past 2 years
- Number of Uber Users = 8M
- Number of Cities = 290
- Total trips = 140M
- Daily Trips = 1M

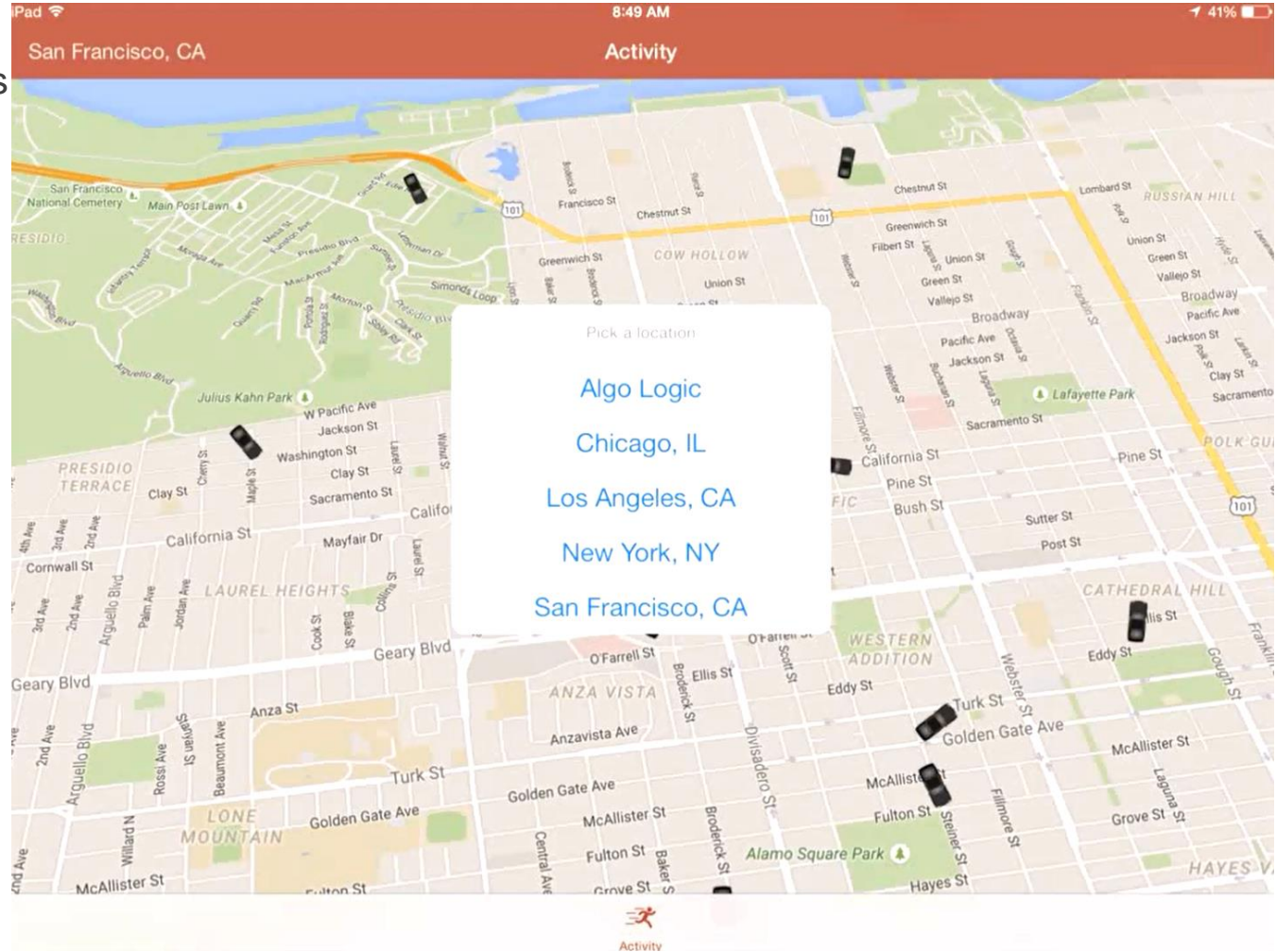
- **Analysis and Assumptions**

- Assuming 25% of the drivers are active
- 25% of 160k drivers = 40k active cars (<48k)
- Drivers update position once per second = 40k IOW

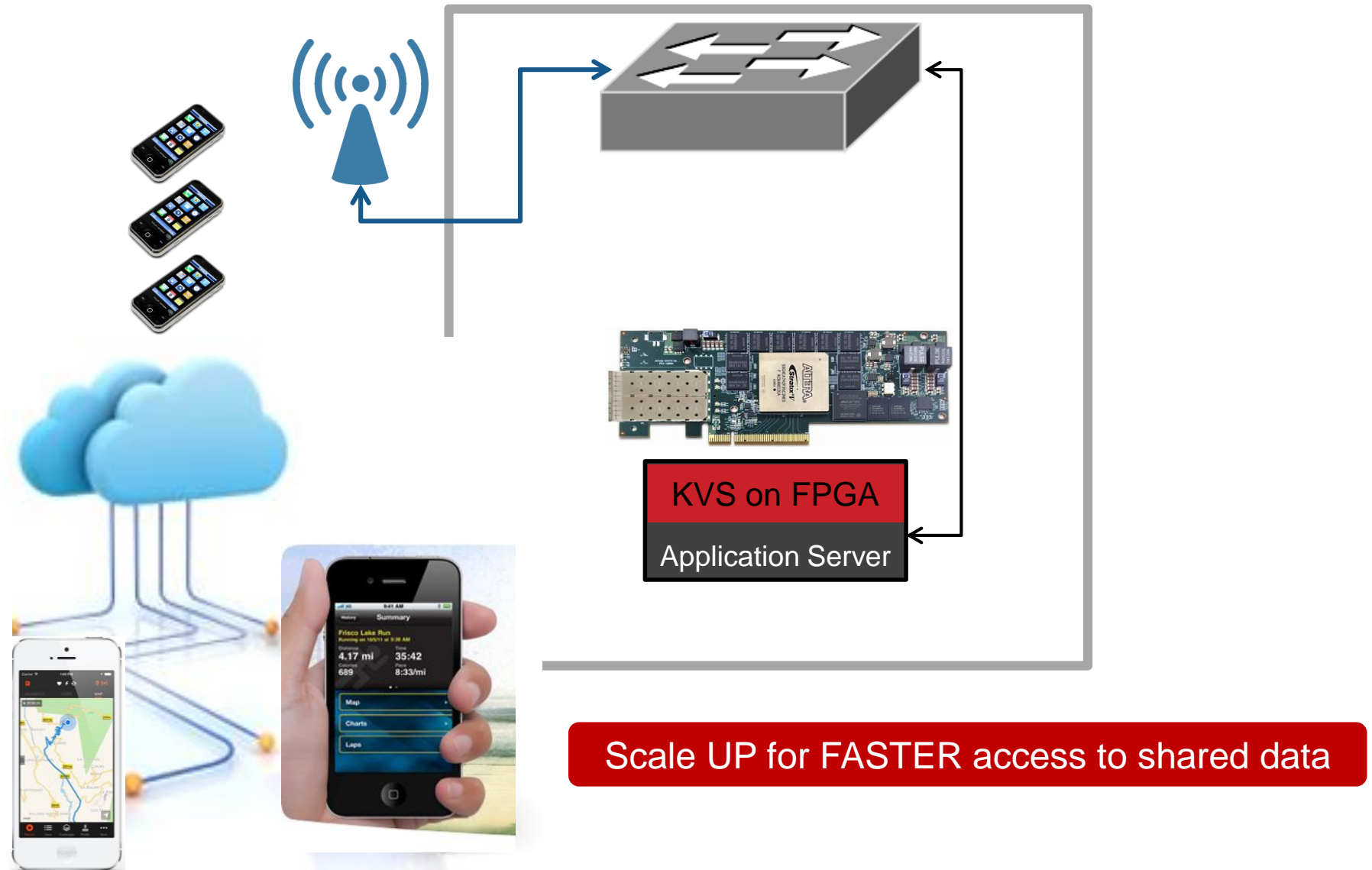
- **Implementation**

- Uber with on an Algo-Logic KVS card

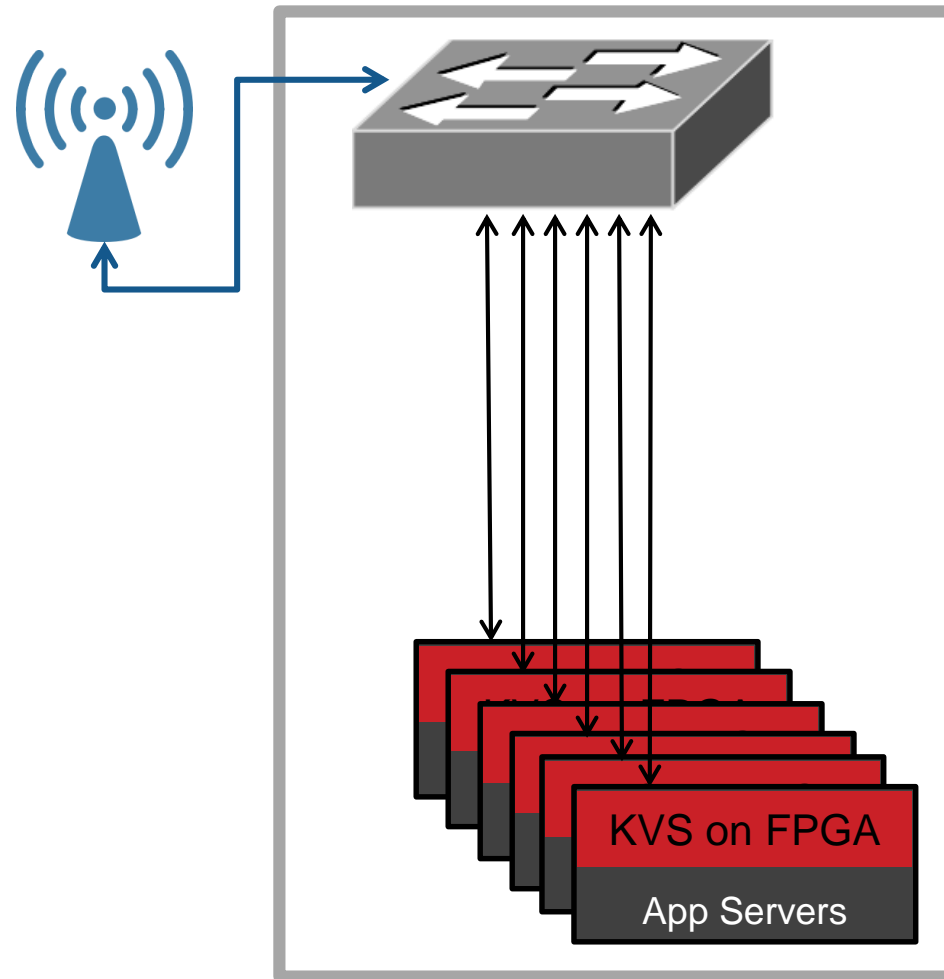
Washington Post, Dec. 2014



Algo-Logic's KVS solution for Mobile Applications

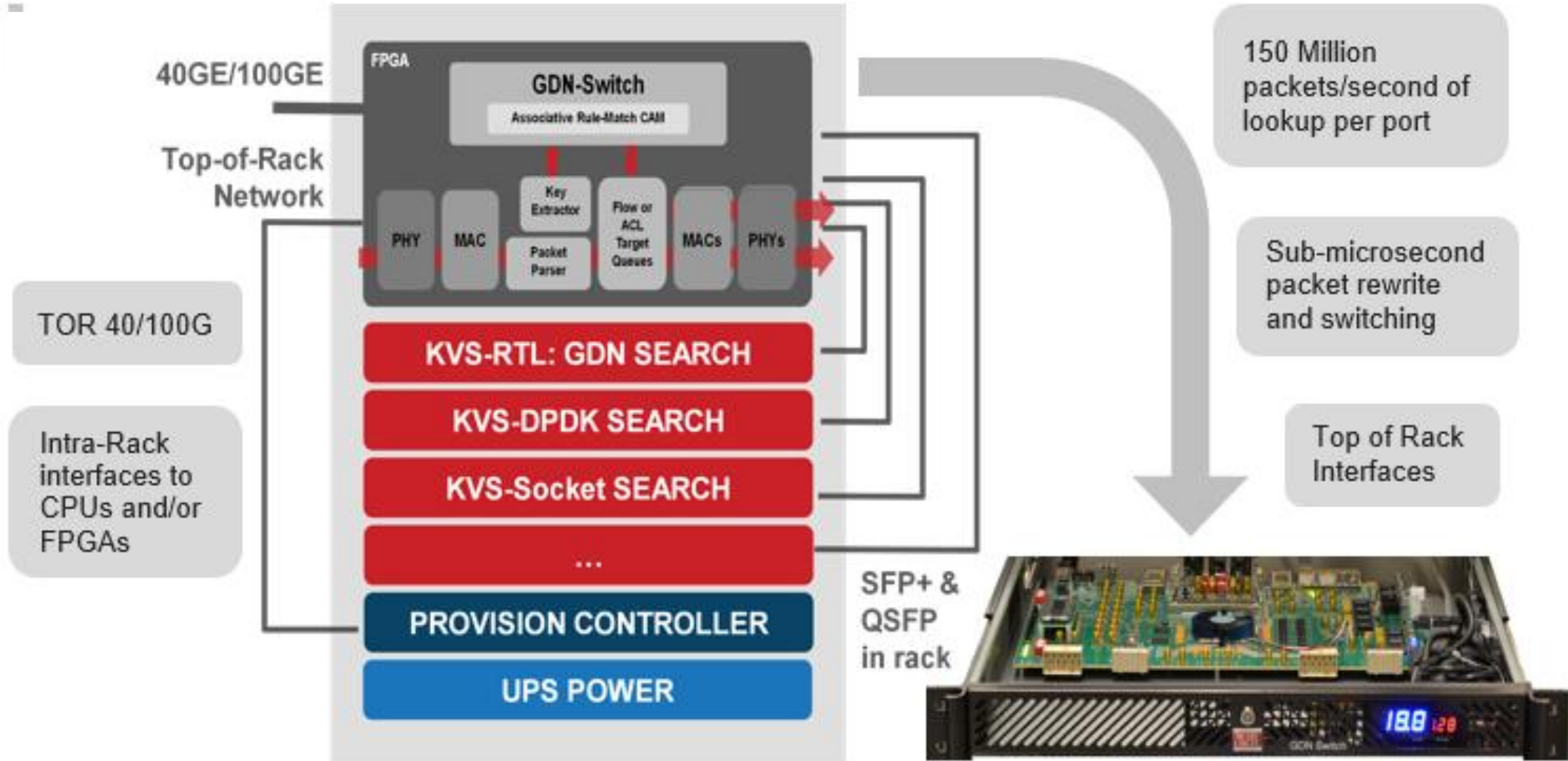


Algo-Logic's KVS solution for Mobile Applications

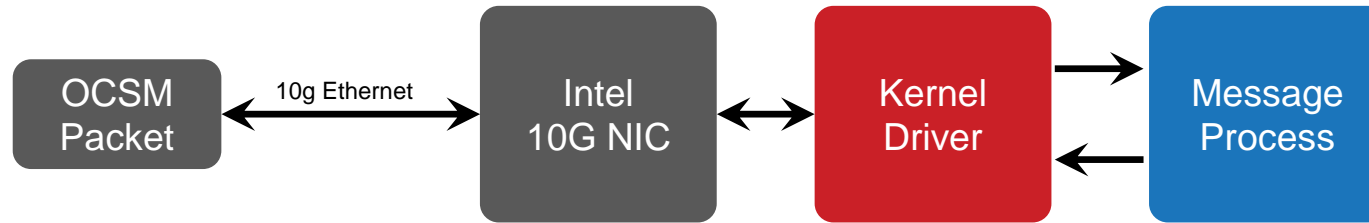


And SCALE-OUT quickly to increase storage capacity and throughput


Provisioning and Measurements with GDN-Switch



Linux Software Socket KVS



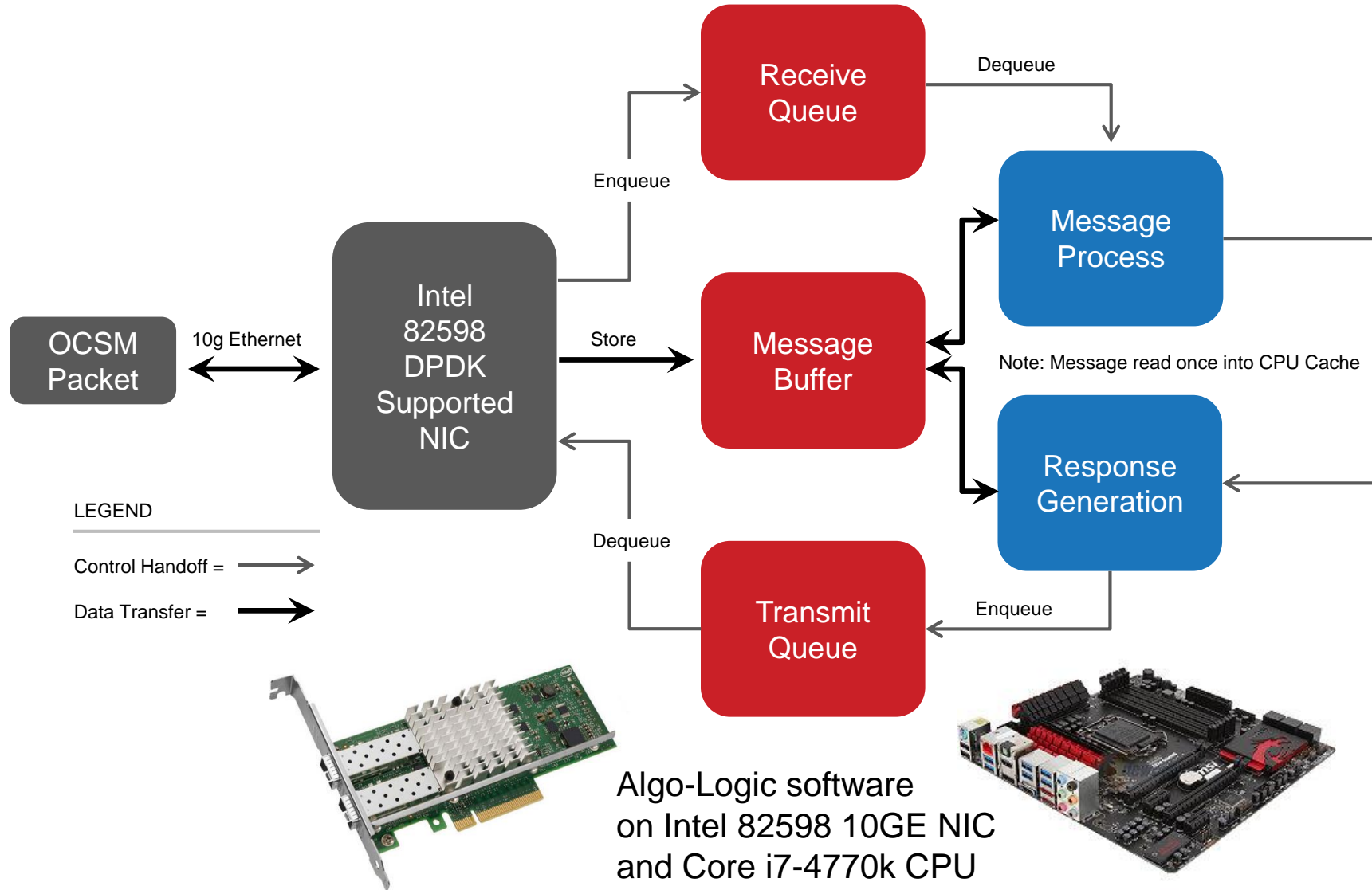
LEGEND

Data Transfer = 

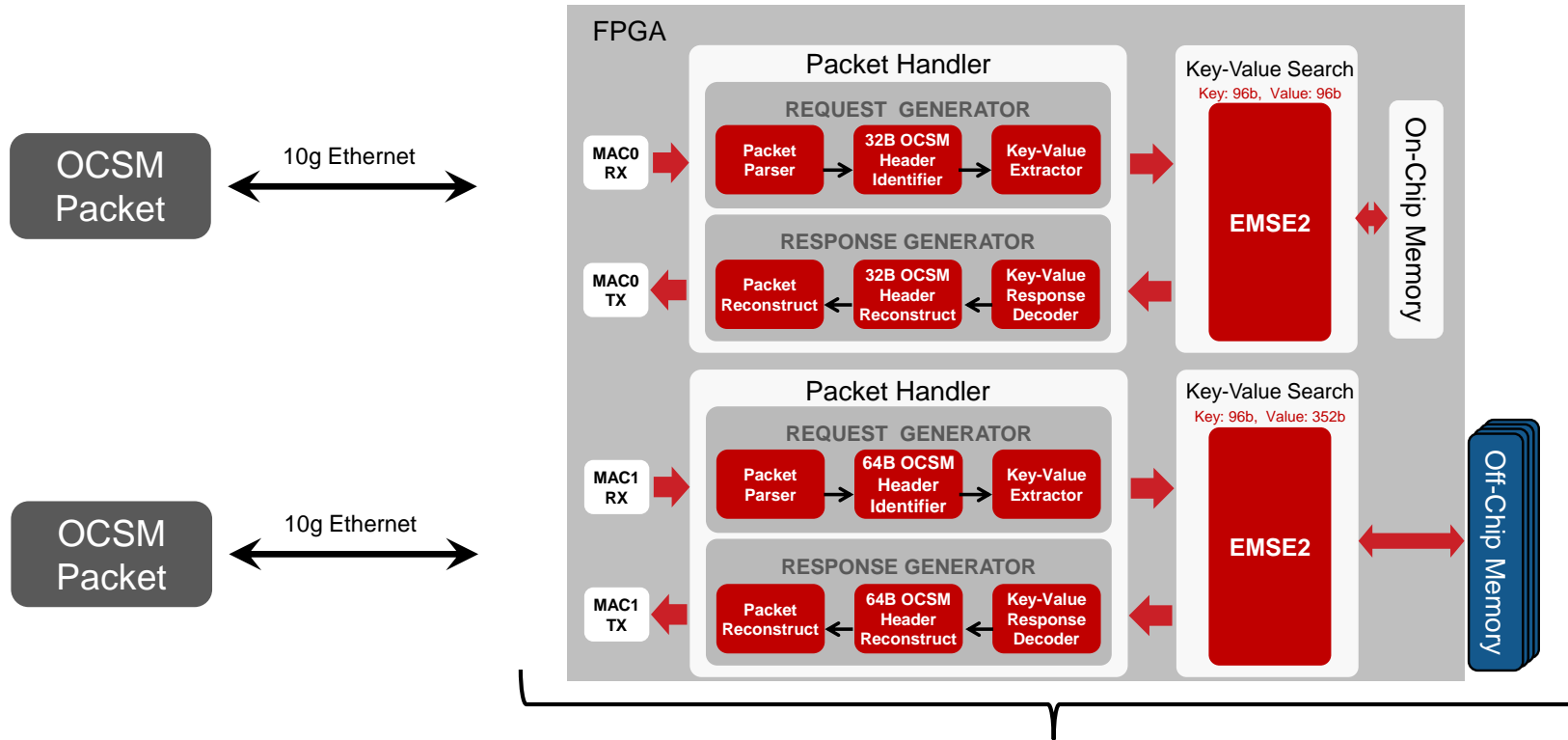


Algo-Logic software
on Intel 82598 10GE NIC
and Core i7-4770k CPU

Algo-Logic KVS with DPDK: Bypass the Kernel



Algo-Logic GDN-Search: KVS in FPGA



Algo-Logic gateway on Nallatech P385 with Altera Stratix V A7 FPGA

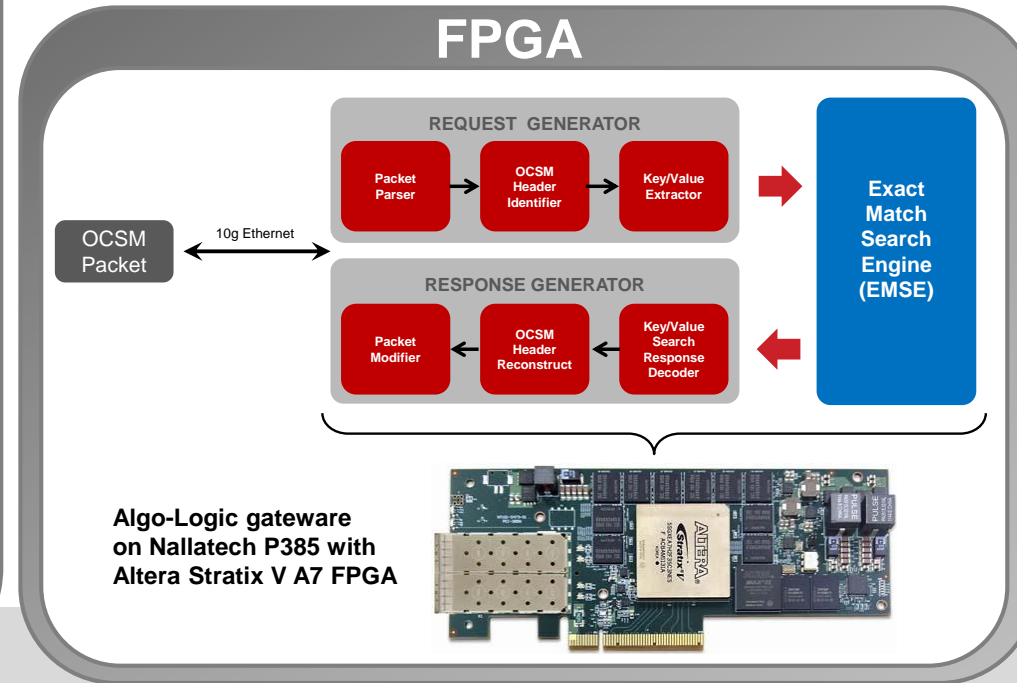
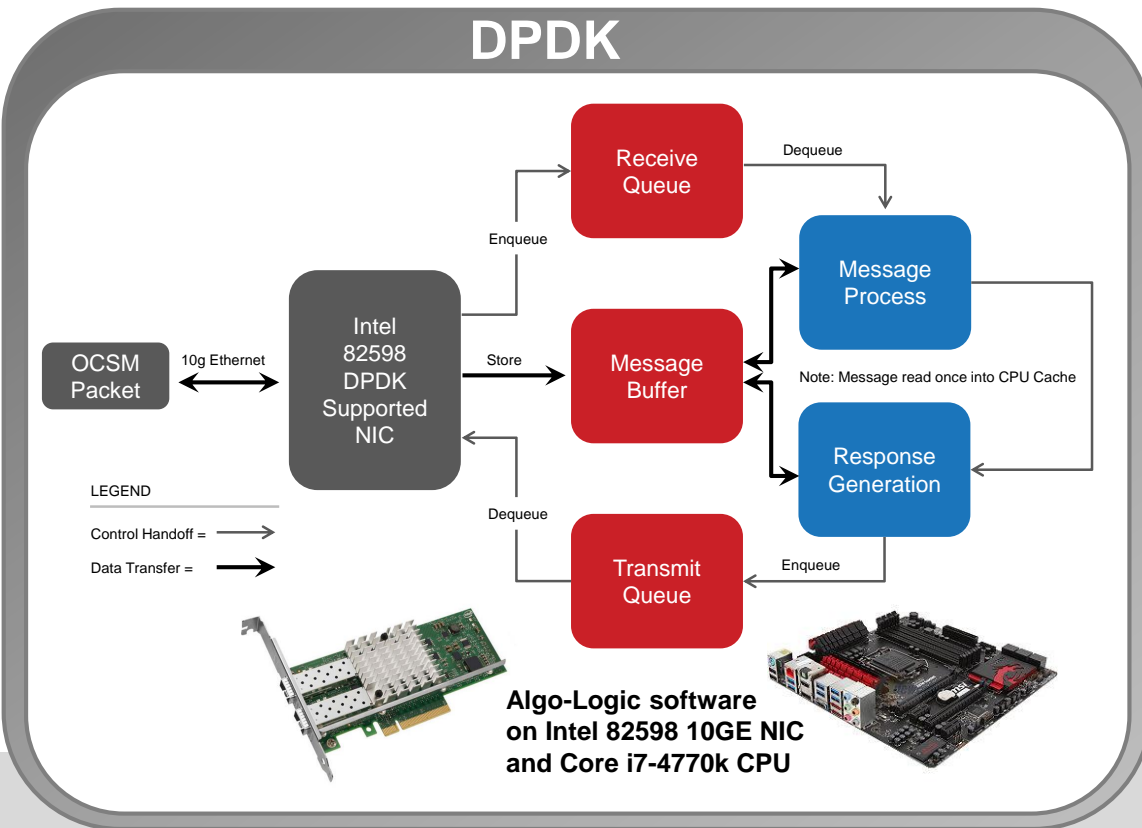
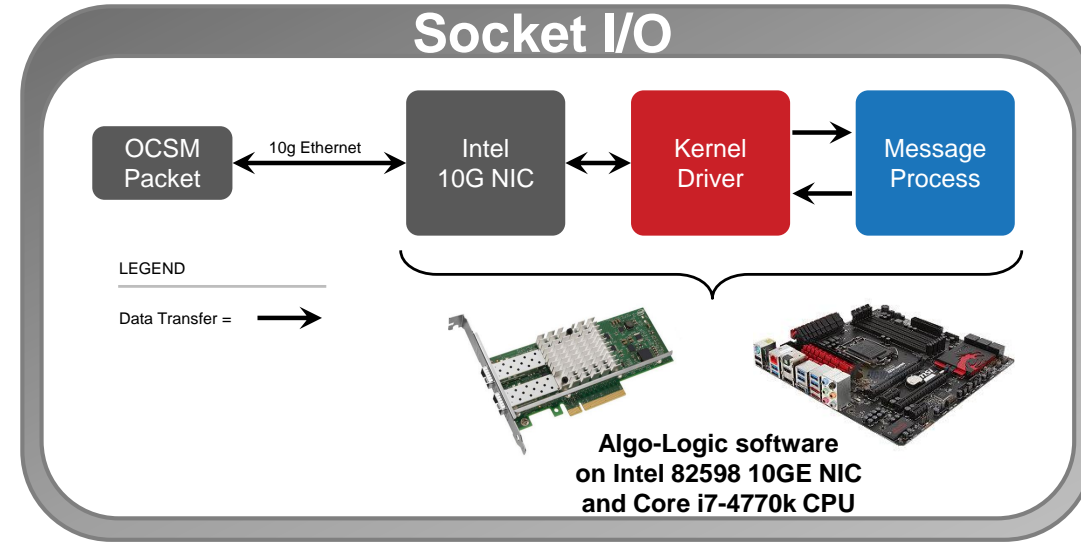


Trends for Adding Storage around FPGAs

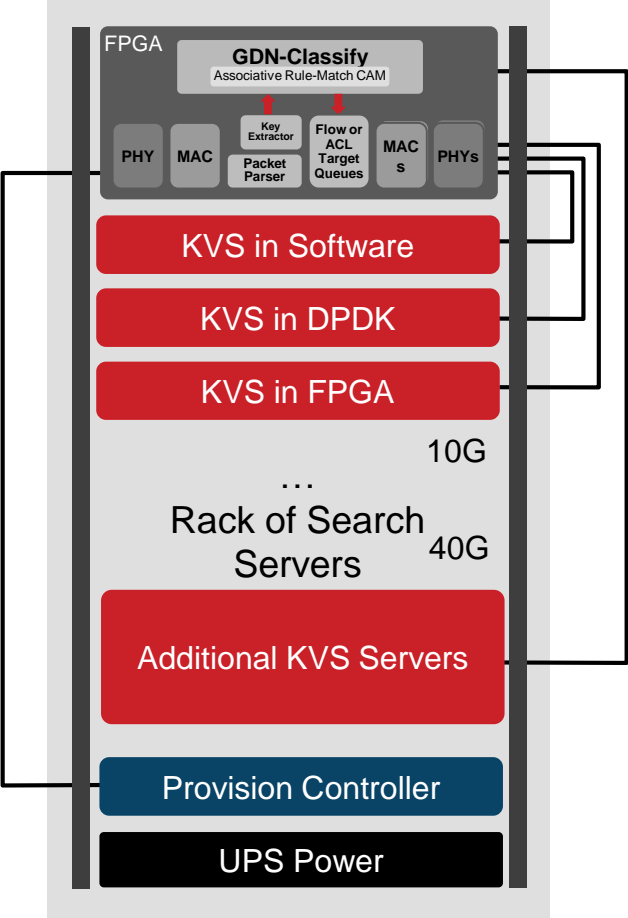
- **Six banks of memory controllers**
 - QDR SRAM
 - RLDRAM
 - DDR3, DDR4
- **64 lanes of SERDES**
 - SATA disk and Flash
 - Serial memories
 - Hybrid Memory Cube (HMC)
 - Mosys Bandwidth Engine (BE2, BE3)
- **New Memories**
 - 3D Xpoint with DDR4 Interface
 - Potential for Terabytes of Memory on each card

Implementation of KVS with Socket I/O, DPDK, and FPGA

- **Benchmark same application**
 - Key/Value Store (KVS)
- **Running on the same PC**
 - Intel i7-4770k CPU, 82598 NIC, and Altera Stratix V A7 FPGA
- **With three different implementations**
 - Socket I/O, DPDK, FPGA



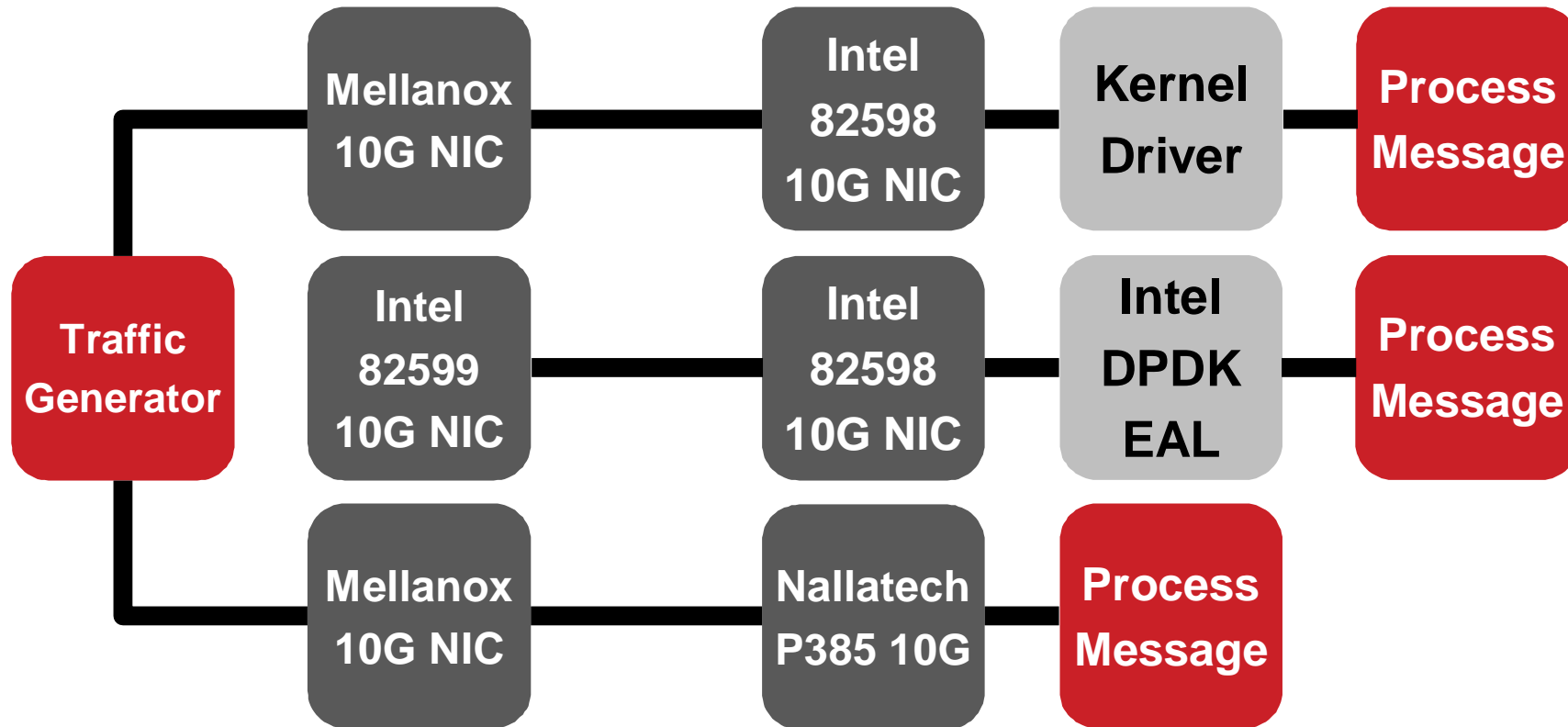
KVS Hardware in Data Center Rack



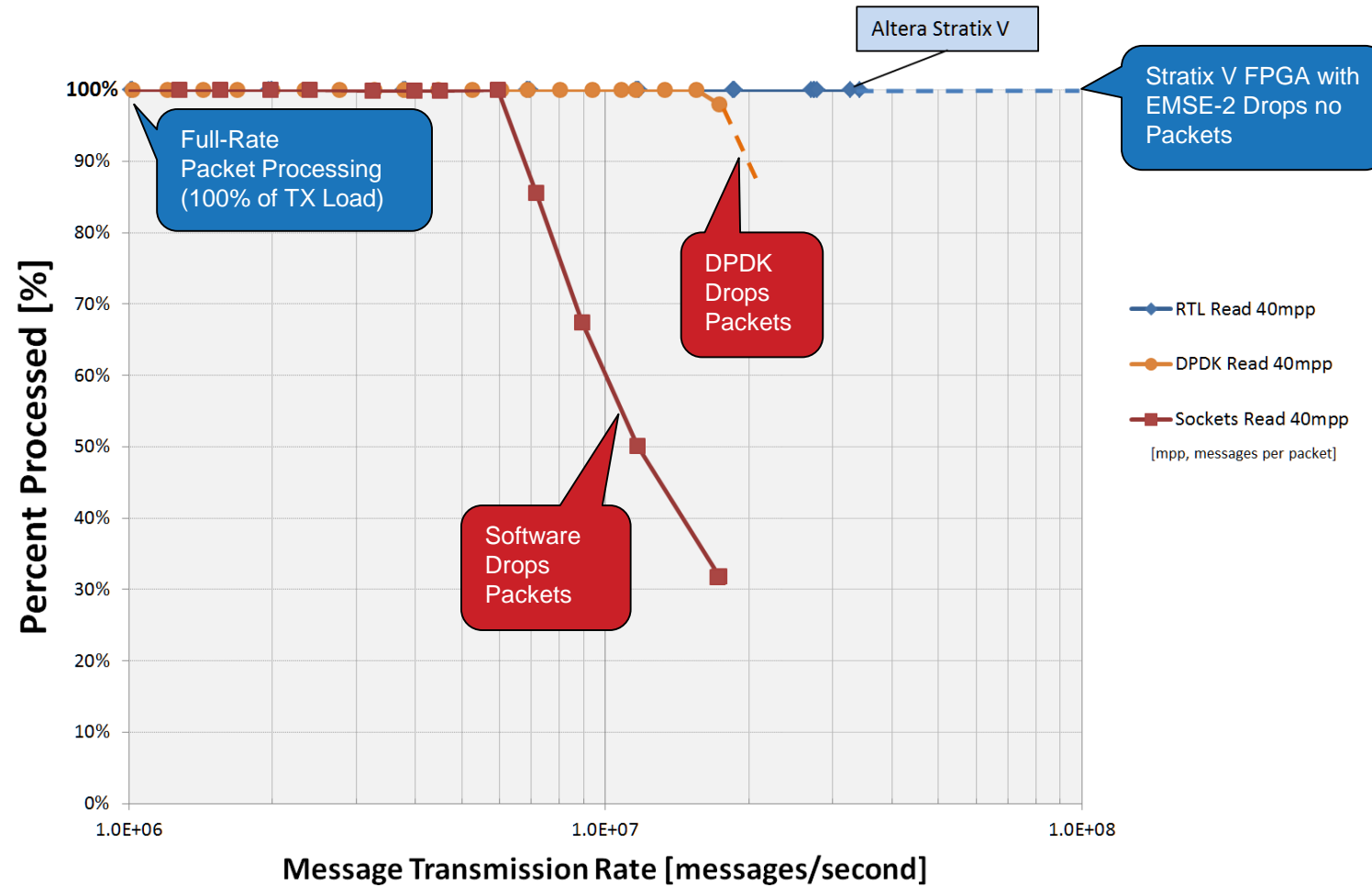
Load Testing off the KVS Implementations

GEN: Traffic Generator

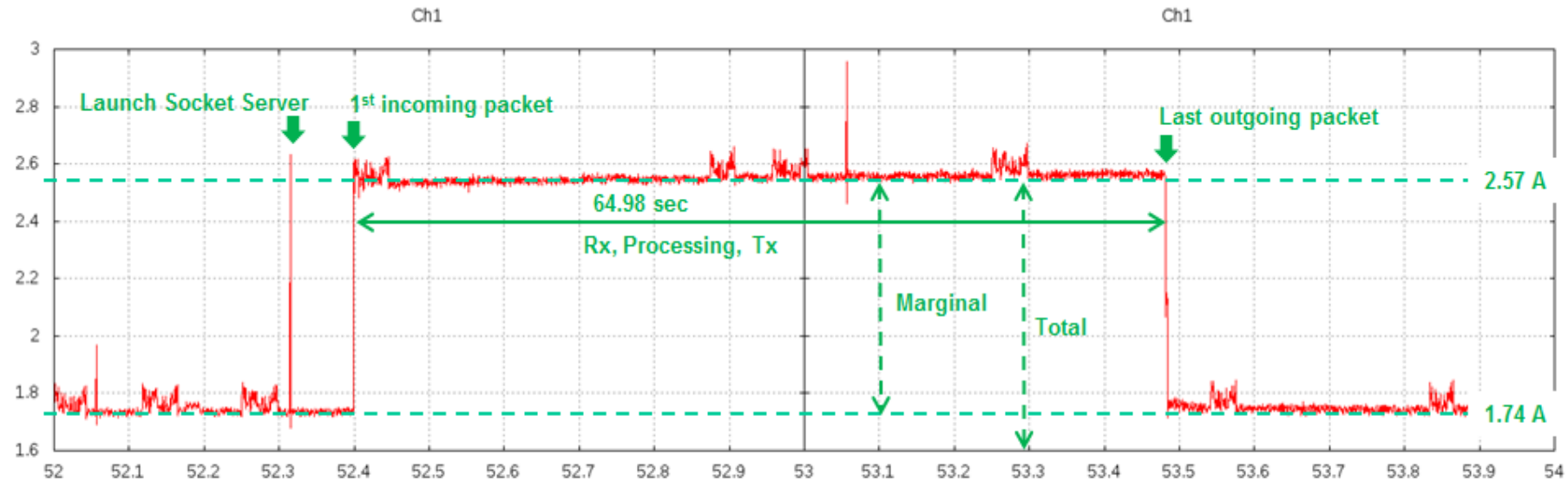
KVS Implementations



Percent Processed vs. Message Transmission Rate



Sockets Power Consumption Profile (10M Packets with 40 CSM Messages/Packet)

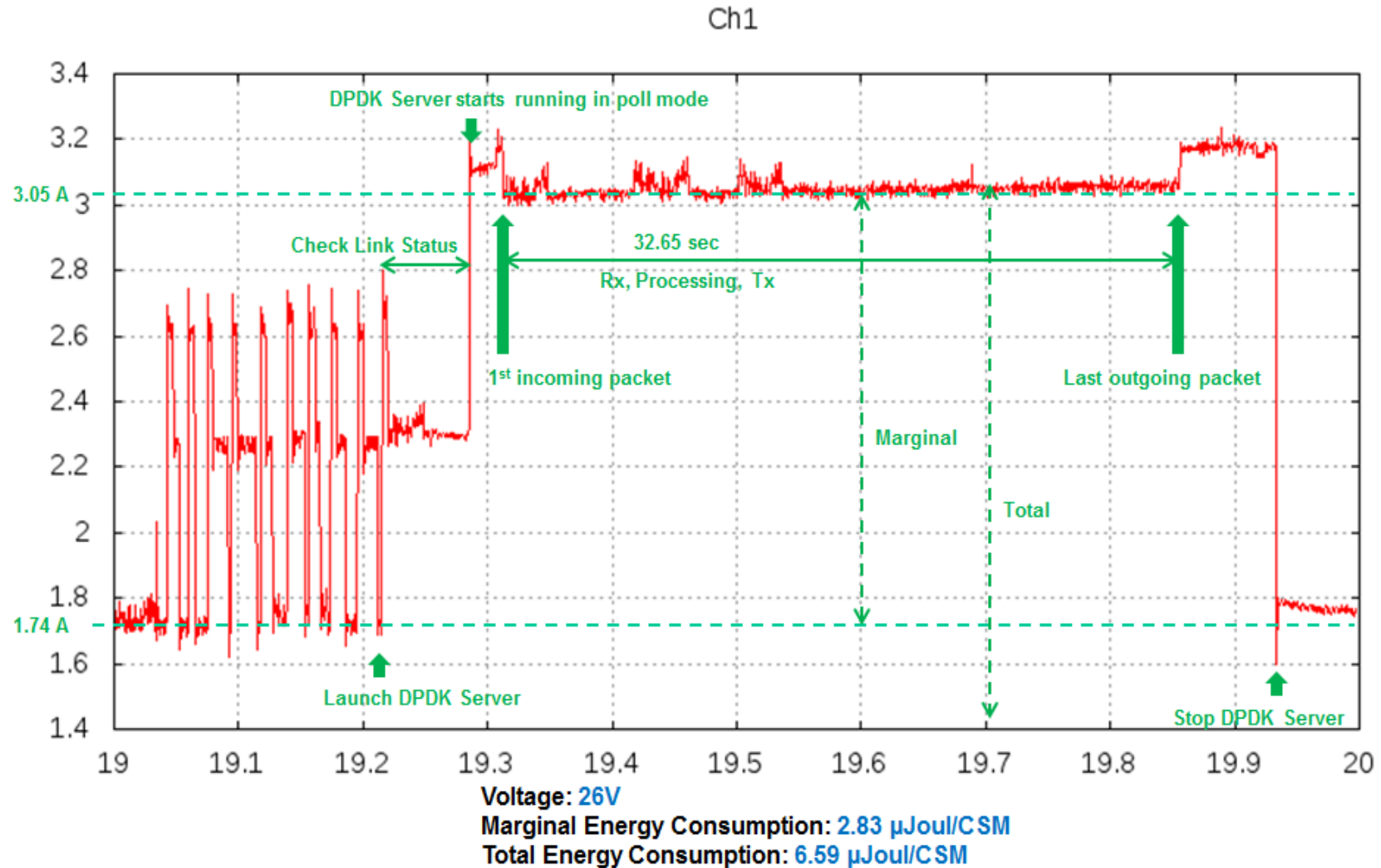


Voltage: 26V

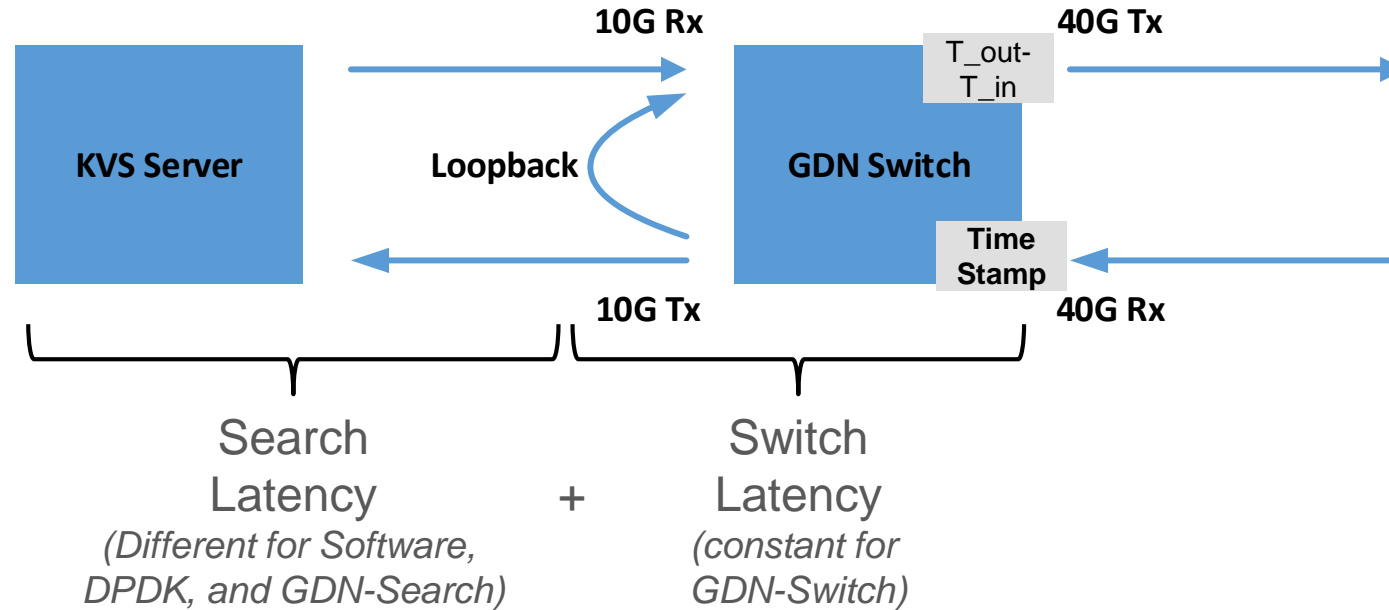
Marginal Energy Consumption: 3.53 μ Joul/CSM

Total Energy Consumption: 11.07 μ Joul/CSM

DPDK Power Consumption Profile (10M Packets with 40 CSM Messages/Packet)



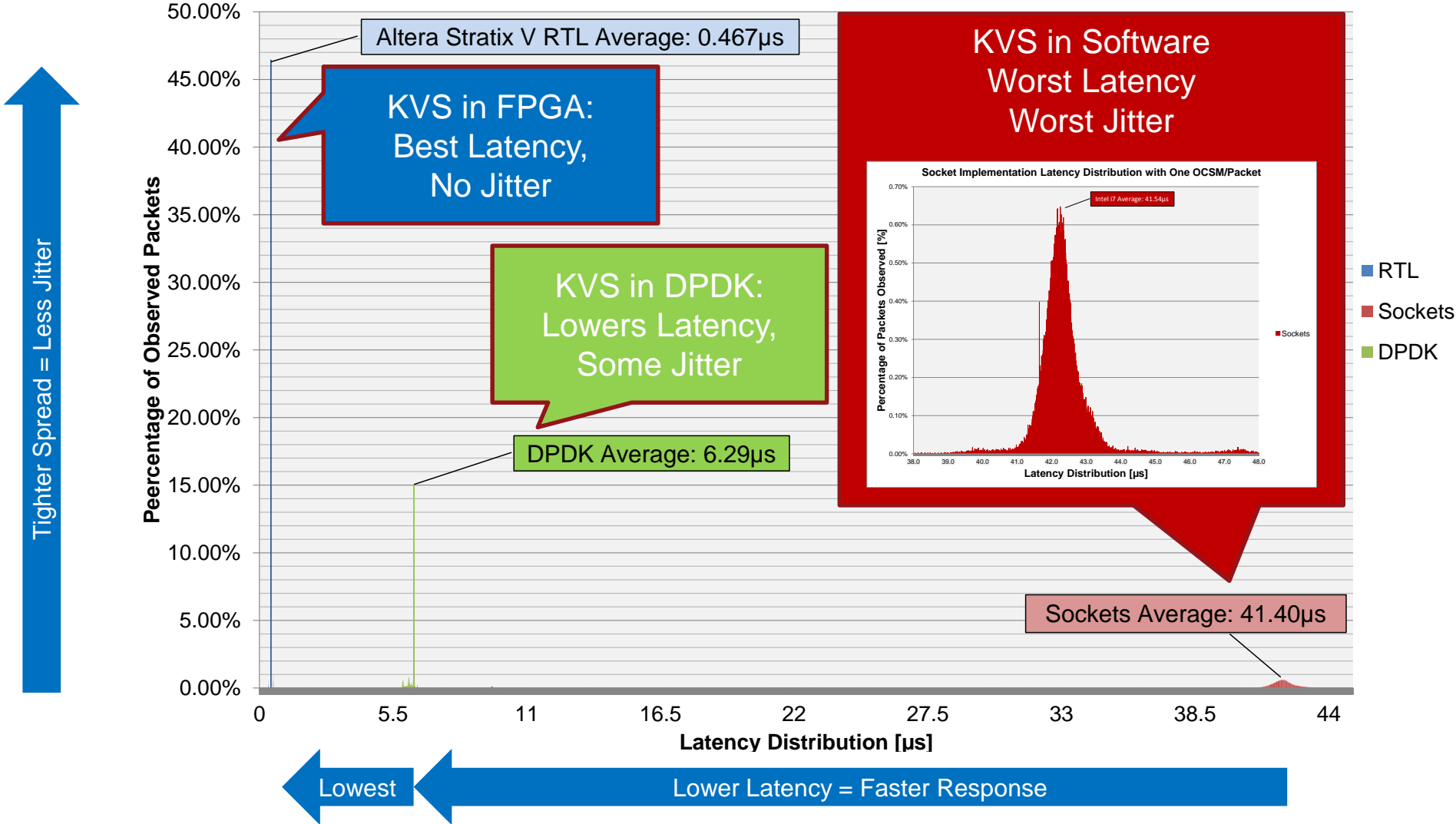
Latency Measurement with GDN-Classify



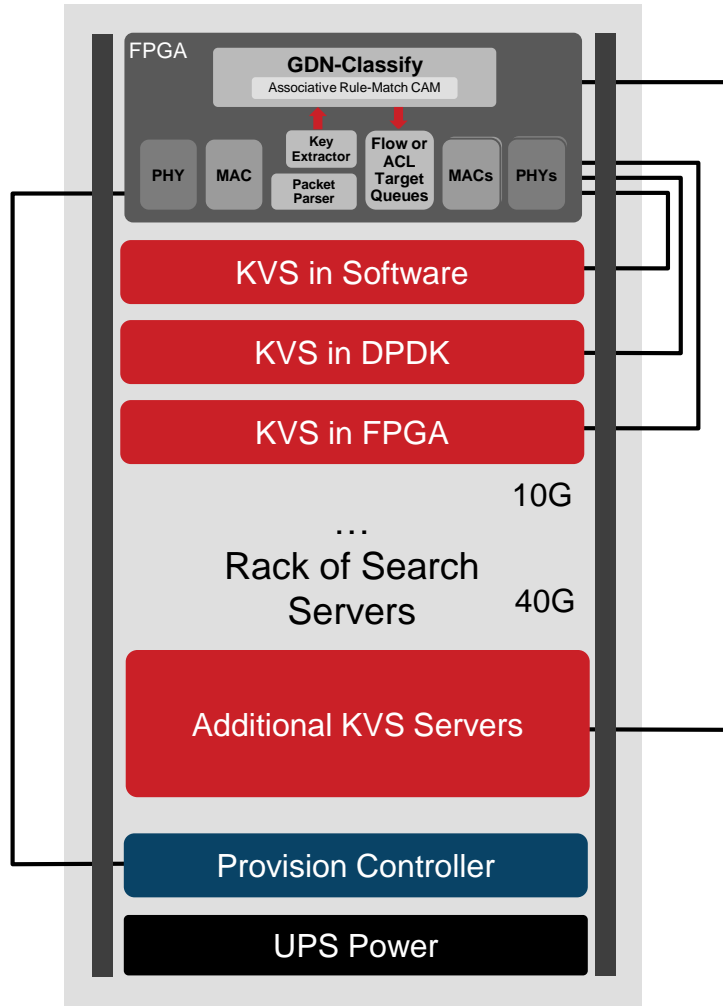
- Round trip latency of GDN Switch is deterministic (constant)
- Round trip latency of KVS Sever = Total Round trip Time – Round trip time with 10G loopback on GDN Switch

KVS Latency in FPGA, DPDK, and Sockets

Latency Comparison 100k packets, 1 OCSM per packet, 1k pps



Measured Latency, Throughput, and Power Results



All Datapaths Summary	Latency (μseconds)	Tested Throughput (CSMs/sec)	Power (μJoules/CSM)
Sockets	41.54	4.0	11
DPDK	6.434	16	6.6
RTL	0.467	15	0.52

All Datapaths Summary	Latency (μseconds)	Maximum Throughput (CSMs/sec)	Power (μJoules/CSM)
GDN vs. Sockets	88x less	13x	21x less
GDN vs. DPDK	14x less	3.2x	13x less

Conclusions: Programmable Hardware in the Data Center

- **Lowers Latency**
 - 88x faster than Linux networking sockets
 - 14x faster than optimized DPDK (kernel bypass)
- **Increases Throughput (IOPs)**
 - 3x to 13x improvement in throughput
 - Lowers Capital Expenditures (CapEx)
- **Reduces Power**
 - 13x to 21x reduction in power
 - Reduces Operating Expenditures (OpEx)