Karlsruhe Institute of Technology

100 Gbit/s UDP Data Acquisition on Linux using AF_XDP

The TRISTAN Detector

Jalal Mostafa, Denis Tcherniakhovski, Suren Chilingaryan, Matthias Balzer, Andreas Kopmann, and Juergen Becker

25 April 2024

Motivation

- Scientific detectors utilize **trigger** systems to extract **relevant scientific data** using **filtering algorithms**
- Trigger Systems employ **hardware accelerators** to respect the low-latency requirements of the filtering algorithms
- Hardware accelerators have **limited logic elements and memory resources** that should run resource-demanding filtering algorithms
    - Machine Learning-based Trigger Systems
    - High Memory Requirements Trigger Systems (e.g. histograms)

Data Acquisition with UDP

Karlsruhe Institute of Technology

- **Network-attached servers** of abundant computing resources can **augment** the resource-demanding **trigger** systems
- UDP is usually used to connect the trigger system to a computer server
    - UDP is easy to implement and is resource-efficient
- **UDP Networking** on servers is **inefficient**
- We propose “DQDK” a **novel 100+ Gbps UDP-based resource-efficient** DAQ **framework** on **Linux** using **AF_XDP** to augment hardware accelerators in trigger systems
- We test our framework for the TRISTAN detector use case

Efficient Networking on Linux

	Standard Sockets	DPDK	RDMA
Resource Efficiency on FPGA	+++ (UDP)	+++ (UDP)	-
Networking Performance	-	+++	+++
Ease of Programming	+++	-	-
Ease of Configuration	+++	-	-
API Long Term Support	Standardized	2 Years	Standardized

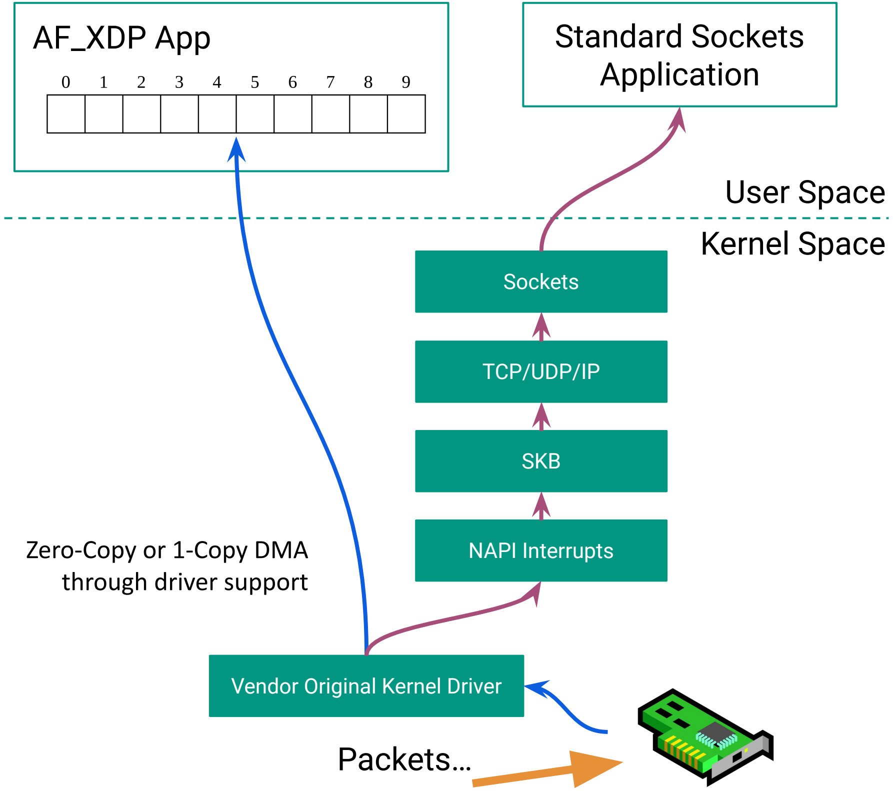
Efficient Networking on Linux

	Standard Sockets	DPDK	RDMA	AF_XDP Sockets
Resource Efficiency on FPGA	+++ (UDP)	+++ (UDP)	-	+++ (UDP)
Networking Performance	-	+++	+++	++
Ease of Programming	+++	-	-	++
Ease of Configuration	+++	-	-	+++
API Long Term Support	Standardized	2 Years	Standardized	Standardized

New Sockets type on Linux (> 4.18) for High Performance Networking

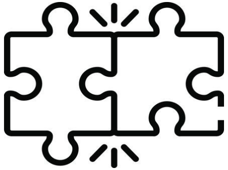
What is AF_XDP?

- Linux kernel 4.18
- Bypass Inefficient Kernel Networking
- Configure vendor’s kernel driver to use user-space memory
- Bind a NIC queue to a user memory
- 3 Modes:
    - Zero-Copy: best performance, need major driver support
    - 1-Copy: good performance, need minor driver support
    - SKB: compatibility mode, no performance gain, no driver changes
- Better performance than standard sockets



The DQDK Data Acquisition Framework

- We propose the DQDK DAQ Framework for multiple hundreds Gbps readout
- DQDK relies on AF_XDP to provide native high-performance networking on Linux
- DQDK exploits the best programming methods to achieve the best possible performance on multicore servers of modern and powerful NIC and hardware

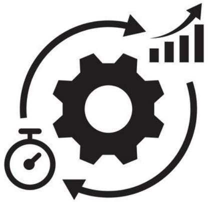


Universal

Using UDP/IPv4

High Performance

Up to 16 MPPS



Resource-efficient

~1.5% FPGA resources

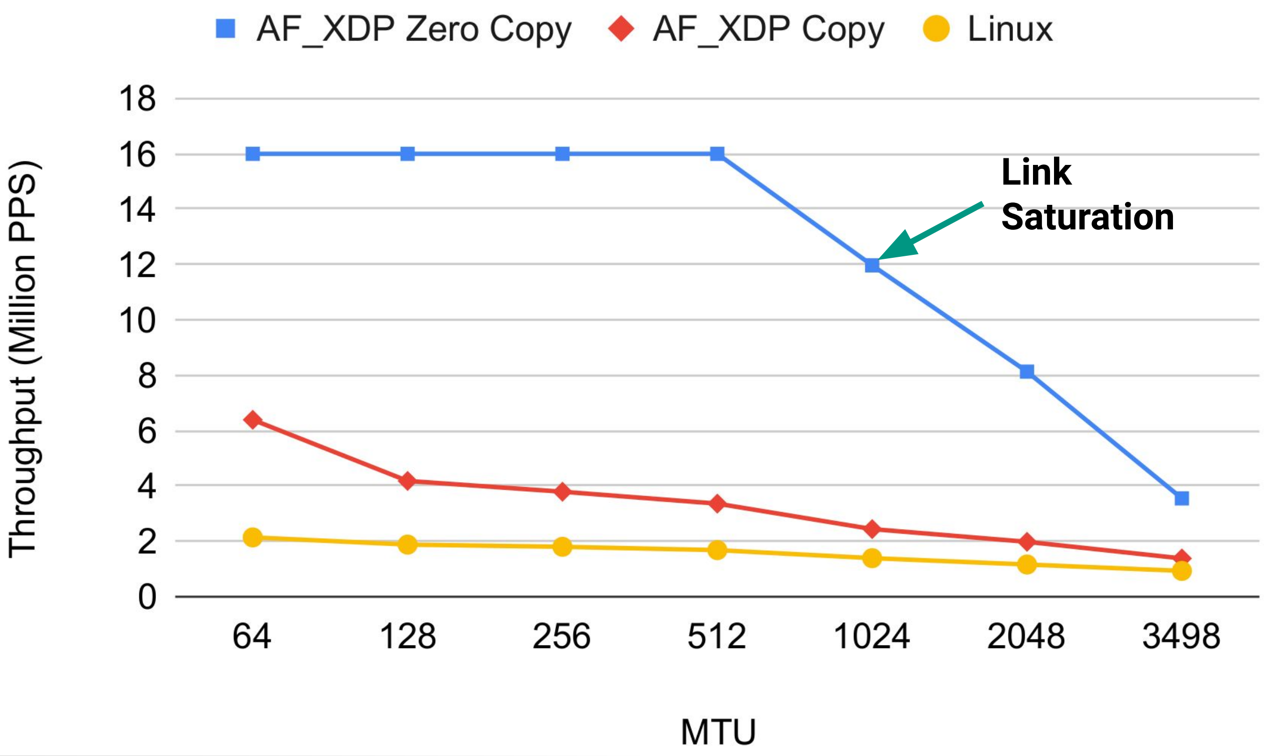
The DQDK Data Acquisition Framework

- CPU optimizations:
  - Lock-free Multi-threaded framework
  - Pin DQDK and interrupt handler threads to dedicated physical CPU cores
  - Cache coherence
  - Use inline keyword in the C language to inline heavily called functions
- Memory optimizations:
  - Prevent TLB Thrashing by using 2MB memory pages
  - Disable Swap memory to storage
  - Allocate memory on appropriate NUMA node
- Lightweight UDP/IPv4 Implementation

- Limitations
  - AF_XDP limitation: Jumbo frames no more than 4KB
  - No IPv6 yet
  - No IPv4 fragmentation

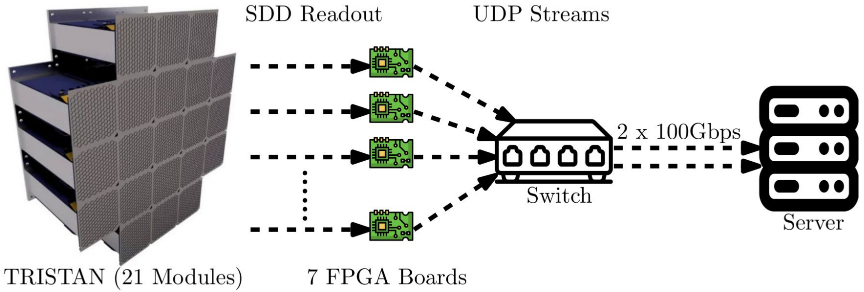
The DQDK Data Acquisition Framework

- 100 Gbps link
- Up to 8x better than Standard Sockets
- Up to 16 million packet per second
- Zero-copy performance is independent of packet size



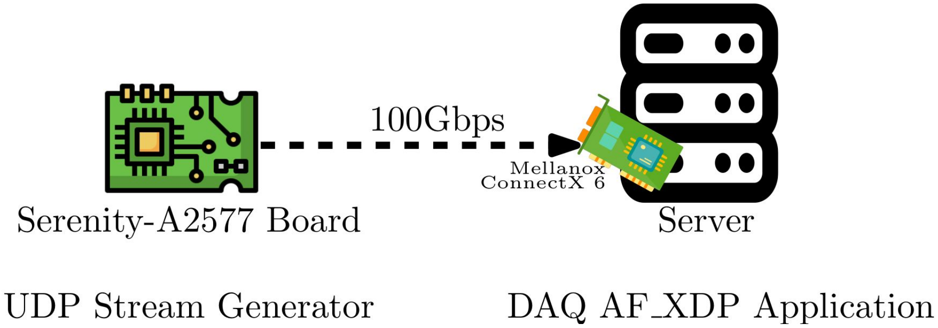
The TRISTAN Detector Use case

- TRISTAN detector wants to confirm the existence of neutrinos
- 21 independent tiles each has 166 channels
- Notable Modes of operation:
  - Waveform mode: 200 Gbit/s raw data for a certain duration
  - Histogram mode: build 4-10 histograms from 16.6 million energy events per second
  - Both modes should buffer the data before storing them on storage
  - High memory usage which is not available on FPGAs

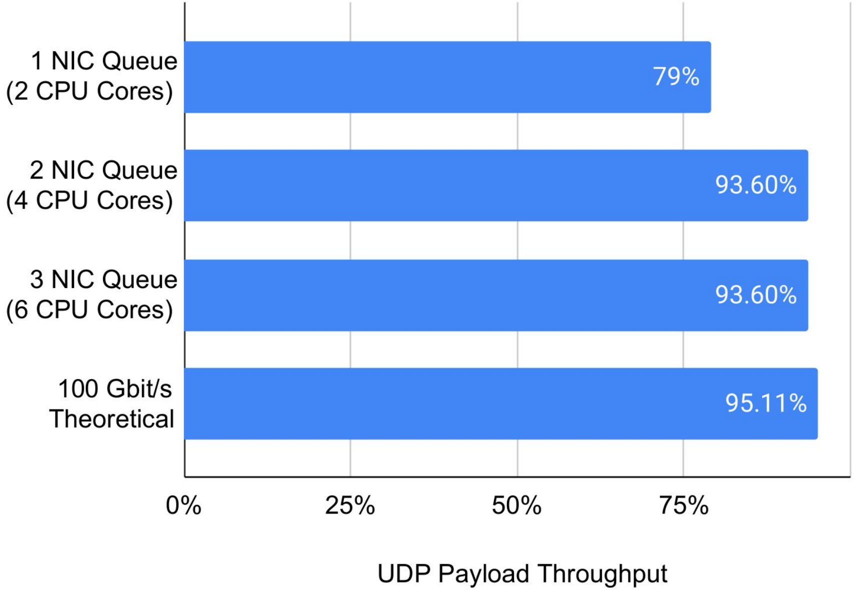


- 21 tiles managed by 7 FPGA-based boards
- Implement resource-efficient UDP on FPGA to transform raw data to UDP packets
- Aggregate all links in 2x 100 Gbit/s using a switch
- Use DQDK to handle 2x 100 Gbps on server

Proposed Setup & Experiments

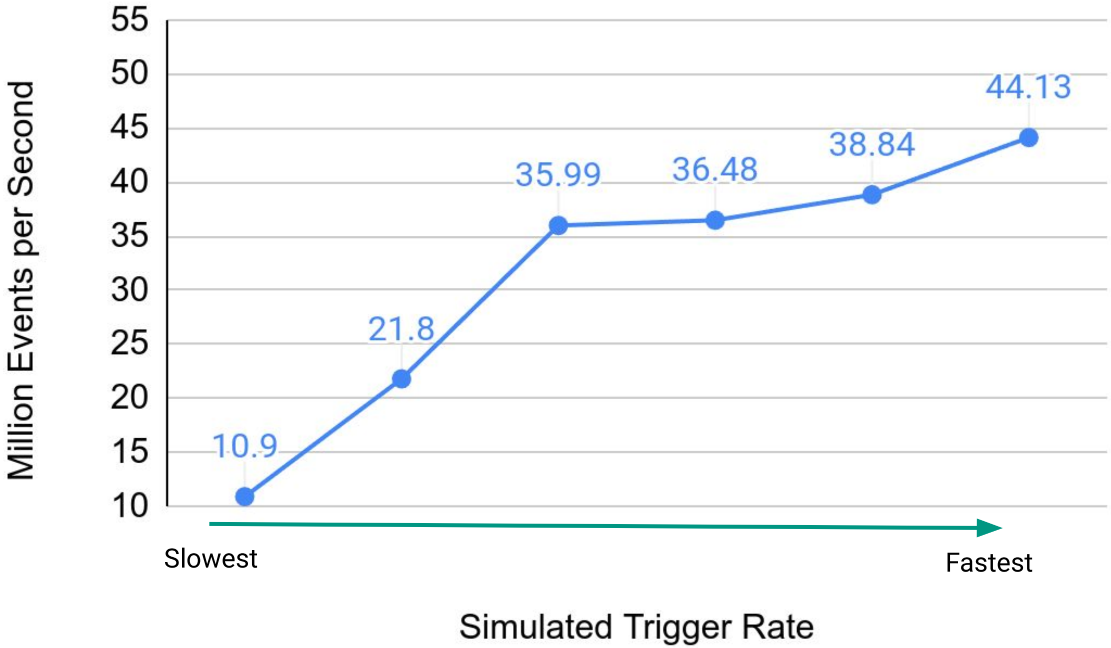


- **Universal** UDP/IPv4 protocol
- **Resource-efficient** UDP implementation on FPGA using only 1048 CLB (~1.5%)
- Targets:
  - Zero-loss data
  - Ability to process 16.6M energy events to construct the histograms for different trigger rates



- Simulation of the waveform mode
- UDP payload throughput is calculated without calculating protocol headers

Proposed Setup & Experiments



- Simulation of the histogram mode
- A trigger simulator for 1 TRISTAN Tile (166 channels) generates random energy values
- FPGA sends 3392 B UDP packets (212 triggered 16 B energy events)
- Receive packets, process each event, increment the corresponding histogram bin

Conclusion

- We propose the **DQDK** framework for a **universal**, **resource-efficient**, **high-performance** DAQ
- DQDK works on standard network protocols i.e. UDP/IPv4
  - It provides a resource-efficient UDP implementation for FPGAs (~1.5%)
- It uses the best programming practices to optimize CPU and memory usage on multicore and NUMA systems
- Our results show that DQDK can handle 100 Gbps with zero-loss using only 4 CPU cores
- We plan to extend our evaluation for the 21 tiles in the TRISTAN detector

Thank you!

jalal.mostafa@kit.edu