VTA-NIC: Deep Learning Inference Serving in Network Interface Cards

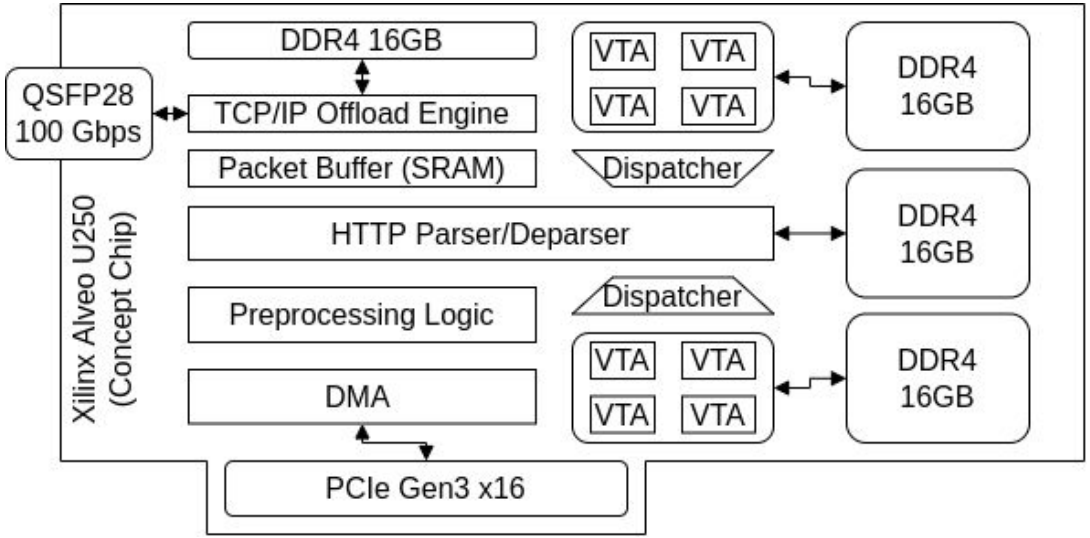
Kenji Tanaka[1], Yuki Arikawa[1], Kazutaka Morita[2], Tsuyoshi Ito[1],
Takashi Uchida[3], Natsuko Saito[3], Shinya Kaji[3], Takeshi Sakamoto[1]

[1]NTT Device Technology Labs, [2]NTT Software Innovation Center, [3]Fixstars Corporation

Abstract: VTA-NIC Chip Architecture

We aim to achieve DL inference serving (DLIS) without CPU interference.

We integrate hardware data paths as a NIC (Network Interface Card),
a REST API parser/deparser and multiple VTAs (Versatile Tensor Accelerators).



Configuration	VTA-NIC
Process node	16 nm FinFET @Xilinx FPGA
Number of Cores	8 VTA Cores
Core Frequency	213 MHz
MACs per core	169
Memory Throughput	19.2GB/s (DDR4-2400)
Number precision	INT8

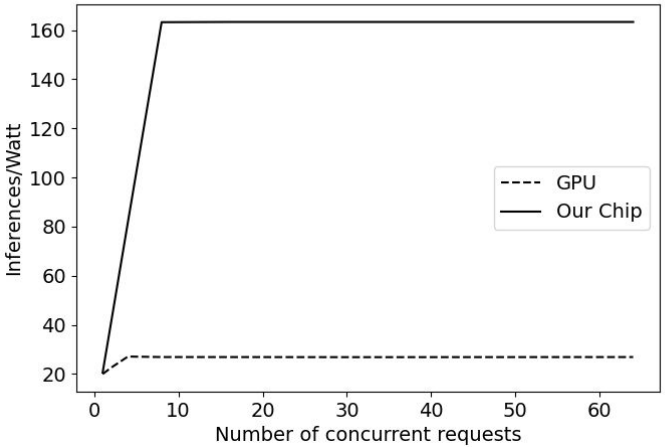
Abstract: Performance

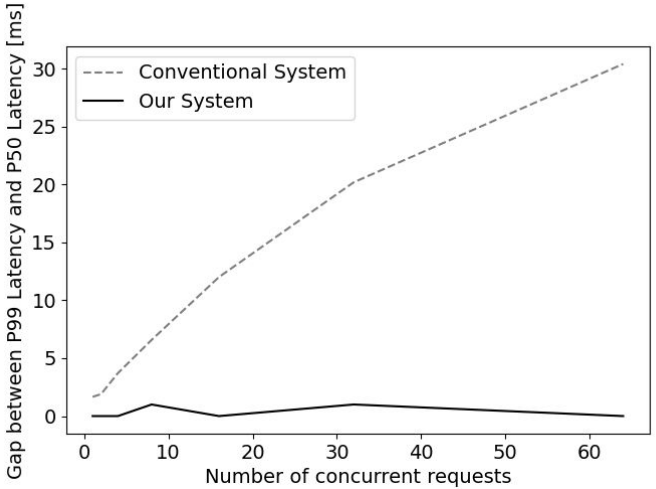
Power Efficiency

The DLIS power efficiency of VTA-NIC is 6.1x better than that of GPU (Nvidia V100).

Tail Latency

At high load, the tail latency of heterogeneous systems unexpectedly increases.
With our chip, the tail latency is predictable since it is proportional to the load.





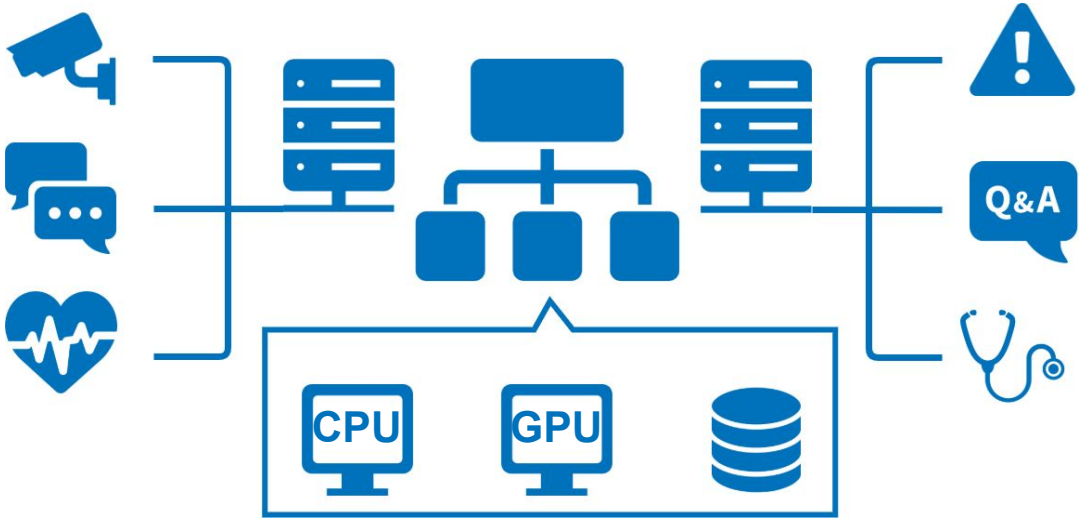
Background

Recently, web applications are often built on microservices.

DL Inference Serving (DLIS) is one of those microservices[1].

DLIS is provisioned with a special accelerator instance[2].

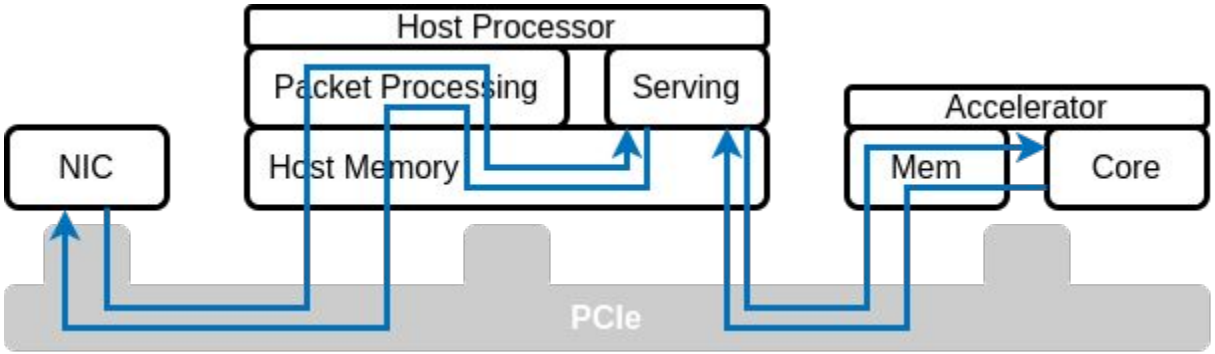
The microservices/instances are loosely coupled via APIs.

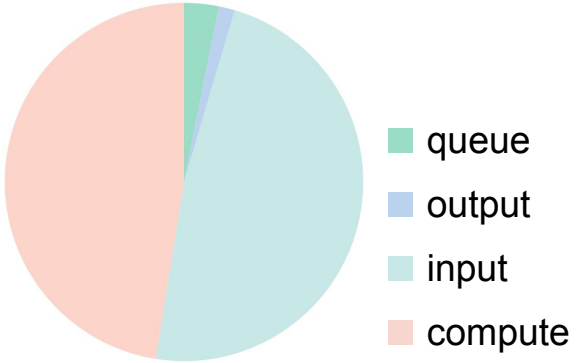


Background

Accelerator instances risk inefficient data movement.

1. Moving data via host processors decreases the accelerator's utilization.[3]
   a. In our preliminary experiments, half of the DLIS latency was caused by moving data.
2. Under high-load conditions, the interference of host processors degrades DLIS tail latency by up to 100 times[4].
   a. In the real cloud, 9% of light DLIS tasks suffer server tail latency, and half of the serving time is waiting time.[5]





Model: ResNet-18@TensorRT
Precision: INT8
System: Triton Inference Server
Accelerator: Nvidia V100

Introduction

Objective

- By integrating NN processing cores in the NIC,
  we eliminate redundant data movement in DLIS.

Challenges

- An architecture to serve inference requests to NN processing engines.
- An architecture to bridge the inter-service communication protocol and the instructions of NN processing engines.

Solution

- A VTA-NIC architecture that integrates an open DL inference engine, a VTA (Versatile Tensor Accelerator), into the NIC.
- Offload host processing to the NIC to achieve DLIS without CPU processing.
- Integrate a circuit that bridges the VTA's instructions and the web API.

Solution

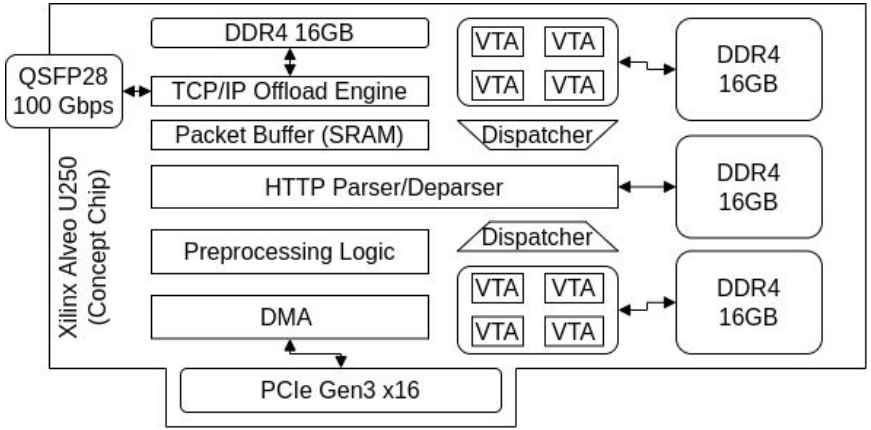
A VTA-NIC architecture that integrates VTAs into the NIC.

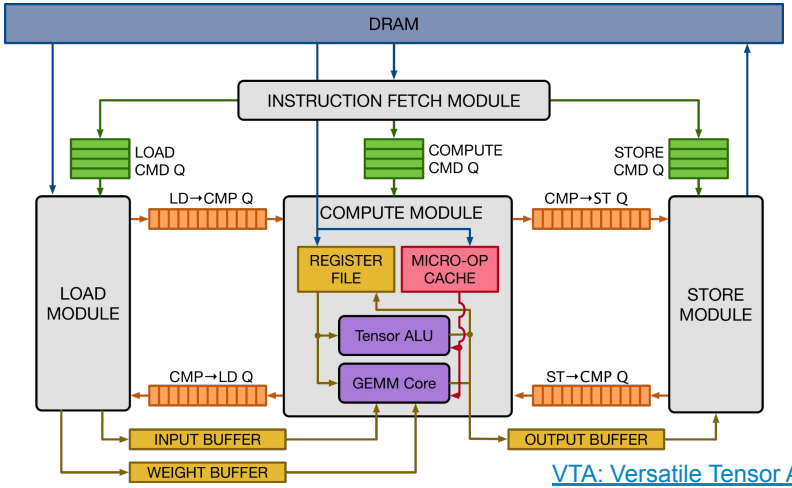
Motivation

- Minimize data movement for inference requests.

Key Points

- We choose TVM-VTA[6] for its open ecosystem.
- The VTA-NIC maximizes throughput by integrating multiple VTAs that operate asynchronously.





VTA: Versatile Tensor Accelerator

Solution

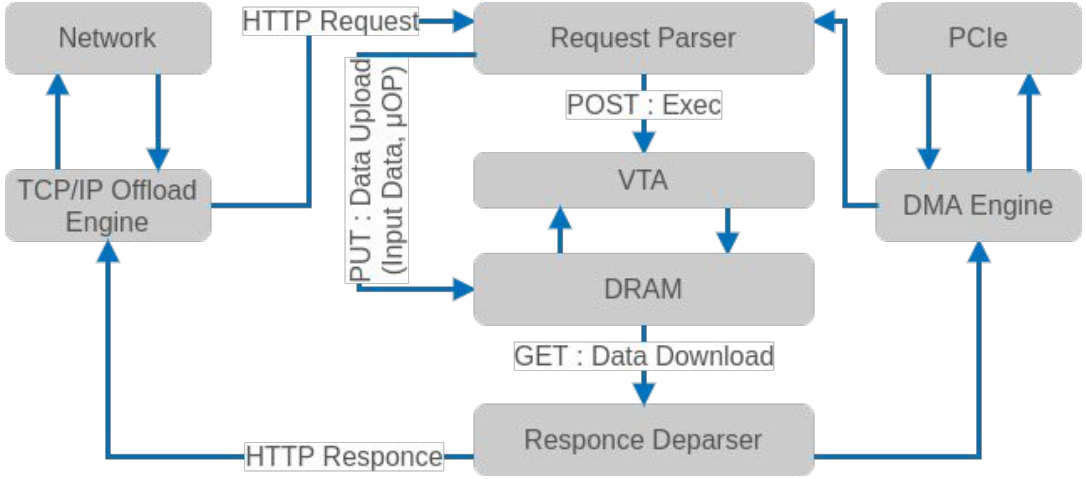
Offload host processing to the NIC to achieve DLIS without CPU processing.

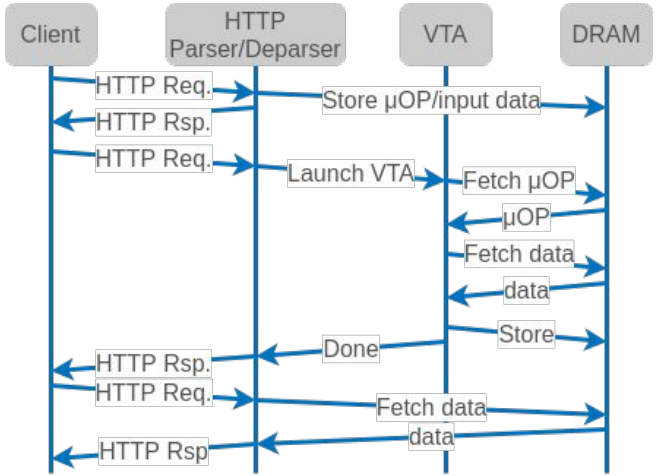
Motivation

- Execute DLIS without CPU processing.

Key Points

- Packet processing (TCP/IP) is offloaded to the VTA-NIC.
- Requests can also be sent from the DMA Engine.





Solution

Integrate a circuit that bridges the VTA's instruction and the web API.

Motivation

- Include protocols for controlling VTAs in the inter-service communication.

Key Points

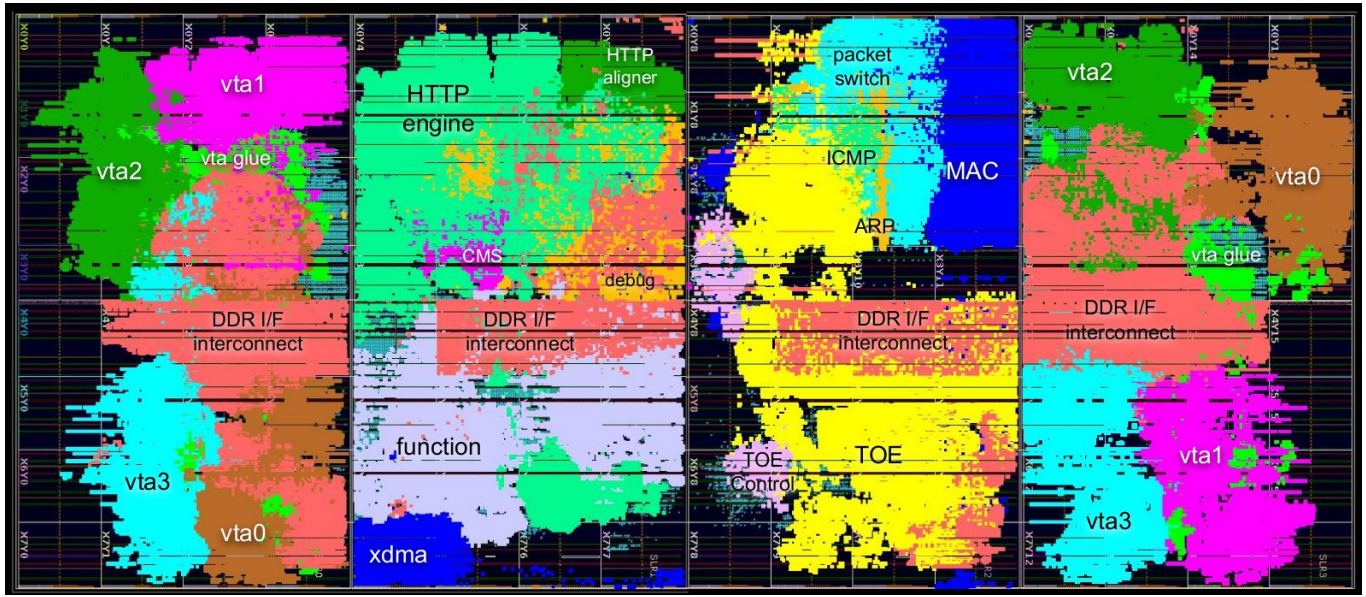
- Develop a circuit to convert REST API to VTA instructions.
- Send HTTP requests to FPGA IP Address/Port with the URN (Uniform Resource Name).

API	URN	VTA-NIC Behavior
GET	/memory/{memory_address}?size=	DRAM Data Download
PUT	/memory/{memory_address}	DRAM Data Upload
POST	/device?insn_addr={memory_address}&insn_count={offset}	VTA Louching
POST	/{processing_logic}	Processing Logic Louching

Implementation

- NoC and DDR I/F that connects between FPGA dies.
- Uses approximately 42% of Alveo U250's resources.
- In TOE - HTTP Parser - VTA core, pass-through latency is 468 nsec (sim.).

Configuration	VTA-NIC
Process node	16 nm FinFET @Xilinx FPGA
Core Frequency	213 MHz
SRAM	40 MB
LUT	0.5 M
DSP	3.3 K slice



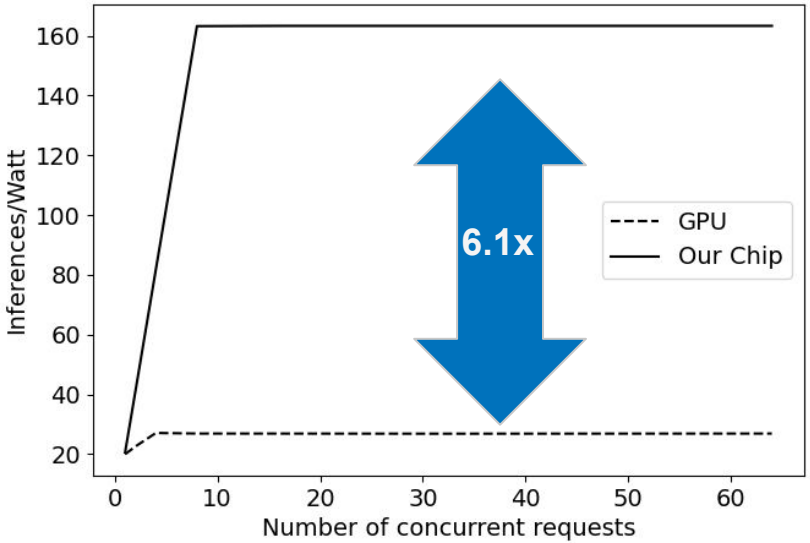
Evaluation 1

Motivation

- Evaluate the power efficiency of the VTA-NIC.

Key Points

- DLIS for ResNet-18@INT8.
- Reference system was served with Triton Inference Server[7] with an Nvidia V100 GPU.
- The process node of the Nvidia V100 GPU is more advanced than that of our chip.
- Our chip achieved about 6.1x better power efficiency than the GPU.



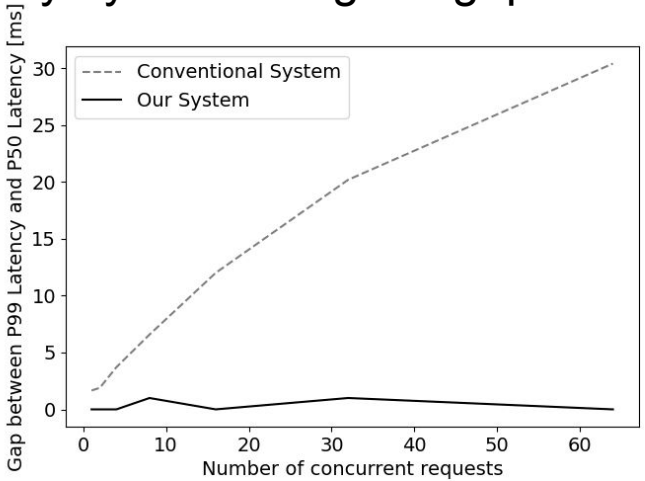
Evaluation 2

Motivation

- Evaluate the unpredictable tail latency under heavy load.

Key Points

- Generally, P50 latency increases in proportion to the load.
- We observe that there is no unpredictable tail latency by evaluating the gap between P99 latency and P50 latency.
- In the Triton inference server, the unpredictable gap between P50 latency and P99 latency increases in proportion to the load.
- In Our System, the gap between P50 latency and P99 latency is constant and unpredictable tail latency isn't observed.



Future Work

NTT

FIXSTARS
Speed up your Business

Low latency inference

- Our chip performs about 26x worse than the GPU system in terms of P50 latency.
  - The clock speed is 7.15x slower than V100 GPU's clock speed.
  - Memory bandwidth is 23.4x narrower than that of V100 GPU.
  - The size of the matrix arithmetic unit is 30x smaller than for V100 GPU.
- Because many of the performance limitations are due to the FPGA, future chips will likely improve performance.

VTA micro architecture

- Need to modify VTA Micro Architecture in the future.
  - Parallel operation of GEMM and ALU
  - Cache for data reuse

Cooperative inference of multiple VTA cores to handle large models.

- Shared memory for VTA cores is required.

Related Work

NSDI ‘22
“Re-architecting Traffic Analysis with Neural Network Interface Cards” [8]

Closely related work that demonstrates the advantages of BNN inference in the NIC.

- Small power overhead for inference in the NIC
- Reduces tail latency

Our chip realizes these advantages, plus the following benefits.

- Possible to execute inference on more general models, including BNNs.
- Hardware-integrated serving functions and inter-service communication.
- No need to install special communication protocols, and easy to connect to other services.
- No need to use special programming languages or compilers, and can benefit from the OSS ecosystem.

Conclusions

- DLIS has been facing a challenge caused by redundant data movement in heterogeneous server architectures, which degrades the performance of the accelerator and the serving system.
- In this work, we proposed an architecture that enables DLIS on NICs.
  - A multi-core DL Engine that can directly serve data from Network Clients.
  - TVM-VTA is used as the DL Engine to exploit the large ecosystem.
  - The circuit that bridges the TVM-VTA's instruction and the web API was integrated into a chip.
- The advantages of our chip compared to conventional systems are as follows.
  - 6.1x power efficient
  - No unpredictable tail latency under heavy load

References

1. Simple steps to create scalable processes to deploy ML models as microservices
2. Applied Machine Learning at Facebook: A Datacenter Infrastructure Perspective
3. Solros: a data-centric operating system architecture for heterogeneous computing
4. Serving DNNs like Clockwork: Performance Predictability from the Bottom Up
5. MLaaS in the Wild: Workload Analysis and Scheduling in Large-Scale Heterogeneous GPU Clusters
6. VTA: Versatile Tensor Accelerator
7. Triton Inference Server
8. Re-architecting Traffic Analysis with Neural Network Interface Cards