TVM Streamer - Accelerating multimedia framework with TVM

Cecilia Albertsson[1]

Hiroki Endoh[2]

Shinya Kaji[1]

[1] Fixstars Corporation, [2]NTT TechnoCross Corporation

Agenda

- Our Challenges
    - Business needs for a high-performance video streaming system

- TVM Streamer Overview
    - Accelerating inference for video stream processing

- Benchmark Results
    - Performance comparison for 4K and HD video streams

- Future Work

Our Challenges

- Background
  - Growing demand for intelligence video analytics.
  - NTT needs a high-performance video streaming system that can efficiently process large amounts of data such as 4K video.
  - We employ a variety of vendor-neutral and high-performance devices.

- Motivation
  - TVM has the versatility and potential to satisfy our requirements, but we know of no video streaming system with TVM that meets the above expectations.

- Proposal
  - Implement an inference application using TVM in GStreamer, a framework for multimedia processing.

TVM Streamer Overview

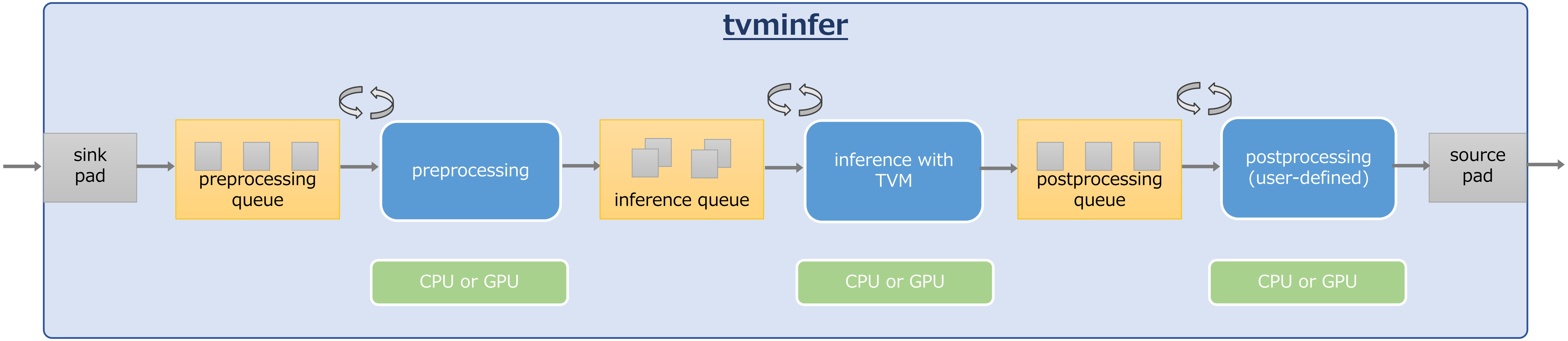
- TVM Streamer is implemented as a filter-type GStreamer plugin called **tvminfer**
- tvminfer implements image processing including **inference with TVM**
- tvminfer executes image processing **on CPU and GPU**
- Current support:
  - x86_64 and ARM64 CPUs
  - NVIDIA Jetson TX2, NVIDIA Tesla T4, and NVIDIA A100 GPUs
  - Single input layer DNN models in pre-compiled TVM format

TVM Streamer Processing

- TVM Streamer applies the following processing to images in a video stream:
  - **Preprocessing**: resizing, batching etc.
  - **Inference**: loads and runs pre-compiled model in TVM
  - **Postprocessing**: can be anything, supplied as a function that receives an image and the associated inference results
- Parameters for preprocessing and inference may be tweaked via properties passed to tvminfer

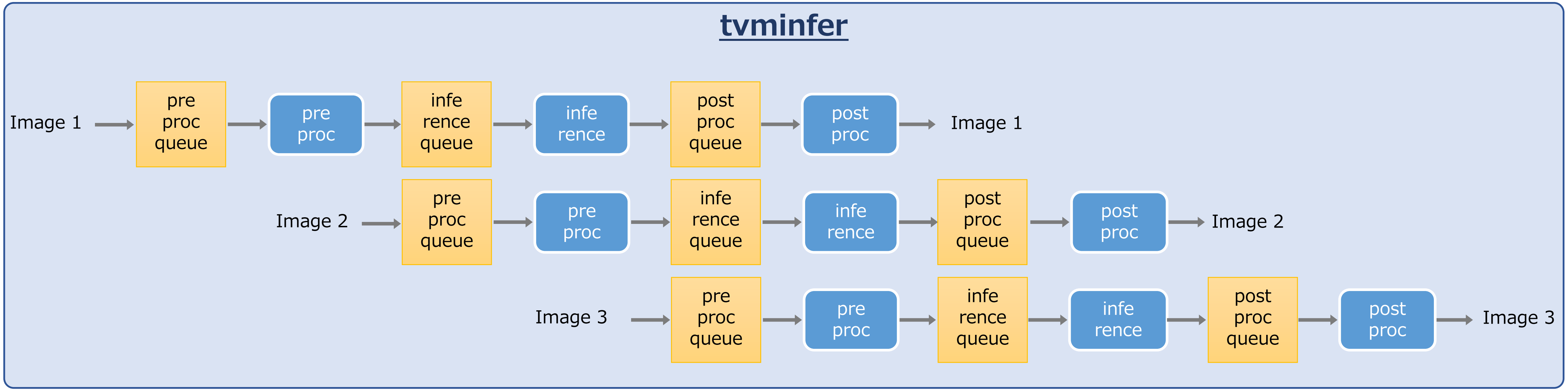
TVM Streamer Structure

- TVM Streamer employs a system of **queues** to pass images between processing stages
- Each processing stage runs in a separate **thread**



TVM Streamer Concurrency

- TVM Streamer exploits CPU-side multithreading and GPU-side CUDA streams to optimize **concurrency**
  - Processing of each image overlaps with that of the previous image, saving time



Benchmark Results

- We compared performance results of TVM Streamer to those of DL Streamer (for CPU) and DeepStream SDK (for GPU)
  - DL Streamer: inference with Intel OpenVINO
  - DeepStream SDK: inference with NVIDIA TensorRT
- We used GStreamer pipelines that reproduce, as closely as possible, the same processing for each framework
- We measured latency, throughput, and power efficiency
- We used AutoTVM to tune models for the TVM Streamer benchmarks

- Result:

  TVM Streamer exhibits significantly higher performance than DeepStream SDK in some cases

Benchmarks x86_64 CPU

- Comparison between TVM Streamer and DL Streamer on x86_64 CPU

Model	Resolution	Latency (msec)		Throughput (FPS)		Power efficiency (FPS/average Watts)	
		TVM Streamer	DL Streamer	TVM Streamer	DL Streamer	TVM Streamer	DL Streamer
mobilenetv3_large (224x224)	4K	9.33	10.41	107.24	96.06	0.27	0.35
	HD	1.98	1.19	506.29	841.90	1.55	3.54
yolo3_darknet53_coco (416x416)	4K	57.70	20.27	17.33	49.33	0.05	0.09
	HD	57.96	18.60	17.25	53.77	0.05	0.09

Benchmarks NVIDIA Tesla T4

- Comparison between TVM Streamer and DeepStream SDK on NVIDIA Tesla T4

Model	Resolution	Latency (msec)		Throughput (FPS)		Power efficiency (FPS/average Watts)	
		TVM Streamer	DeepStream SDK	TVM Streamer	DeepStream SDK	TVM Streamer	DeepStream SDK
mobilenetv3_large (224x224)	4K	5.60	11.36	178.55	88.01	0.59	0.30
	HD	0.98	1.75	1016.42	573.05	3.76	2.23
yolo3_darknet53_coco (416x416)	4K	19.99	18.72	50.03	53.41	0.17	0.18
	HD	20.24	19.18	49.41	52.15	0.16	0.20

Benchmarks NVIDIA Jetson TX2

- Comparison between TVM Streamer and DeepStream SDK on NVIDIA Jetson TX2
  - We did not measure power efficiency on Jetson TX2

Model	Resolution	Latency (msec)		Throughput (FPS)	
		TVM Streamer	DeepStream SDK	TVM Streamer	DeepStream SDK
mobilenetv3_large (224x224)	4K (*1)	20.30	N/A	49.25	N/A
	HD	7.18	7.81	139.22	128.02
yolo3_darknet53_coco (416x416)	4K	225.64	175.15	4.43	5.71
	HD	222.41	166.86	4.50	5.99

*1: DeepStream SDK does not support input/model resolution ratios in excess of a factor of 16 on NVIDIA Jetson TX2

Future Work

- Additional benchmarks
- Support for edge devices
  - E.g., Google TPU, Qualcomm Snapdragon
- Adding useful functions related to inference processing

Thank you!