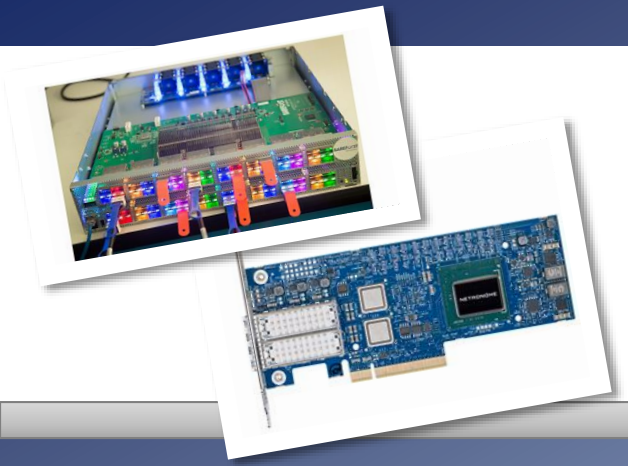
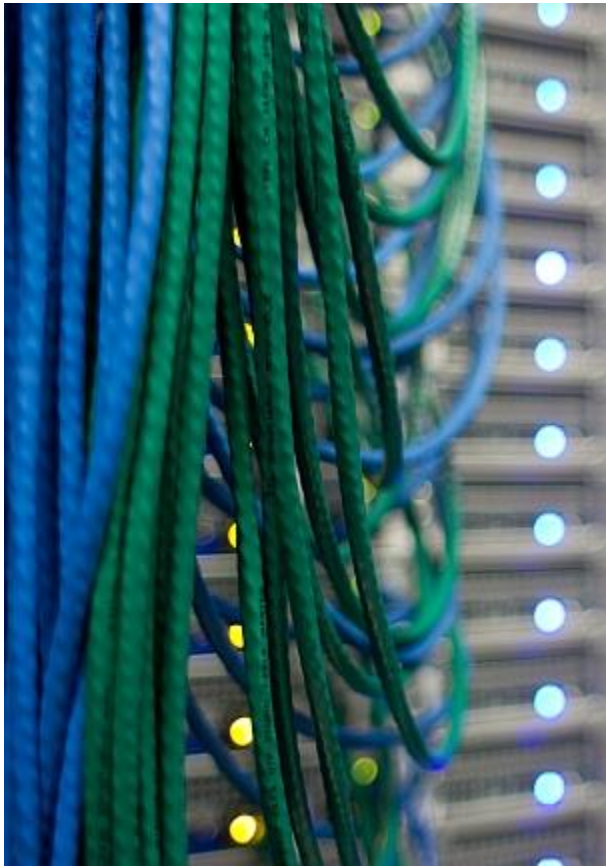


In-Network Computation and Processor-based SmartNICs

Lessons and suggestions from pushing the boundaries

René Glebke, Klaus Wehrle

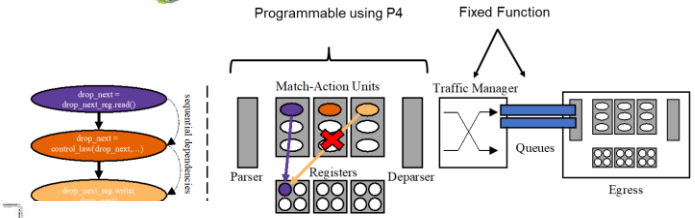
In-Network Computation: State-of-the-art applications

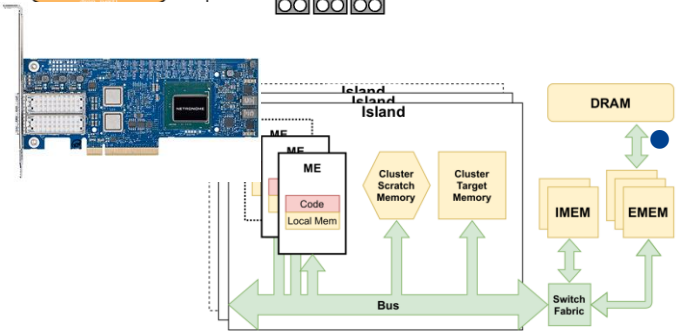


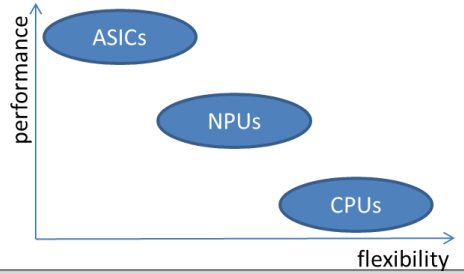
- **Programmable data planes enable scenarios that require low latencies & high bandwidths**
  - Network operation & management
    - AQM/load balanc.
    - Heavy-hitter handling
    - DDoS protection
  - Distributed algorithms & databases
    - Consensus
    - Key-value caching
    - Failure protection
  - Partial offloading of application logic
    - Data (pre-)processing for compute centers & sensors in CPS
    - Industrial feedback control
    - Energy network stabilization

Common pattern in most considered scenarios:
Few individual operations on **many small** items

In-Network Computation Platforms: ASICs vs. NPUs

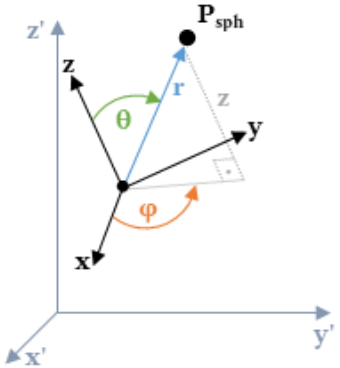






- **Programmable ASICs (e.g., Intel/Barefoot Tofino)**
  - Fixed-pipeline architecture, processing “stages”
  - Constant, predictable processing times, several Tbps
  - Limited arithmetic, limited memory (atomic r/w access)
  - More limited availability
- **Network Processor Units (e.g., Netronome Agilio)**
  - Many-core RISC architecture, thread-based processing
  - Shared buses require coordination → Timing hard to predict
  - Allow more complex operations, large stateful memory
  - Usually more affordable, less restrictive in application
- **Inherent trade-off between flexibility & performance**

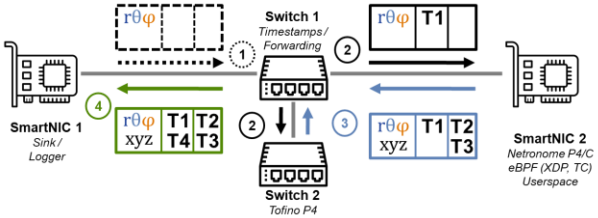
INC Example: Coordinate Transformation [ICPS 21]: Setting

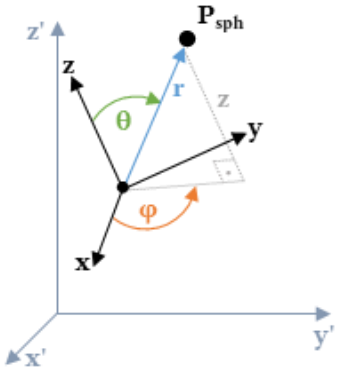


$$\begin{bmatrix} r\sin\theta\cos\varphi \\ r\sin\theta\sin\varphi \\ r\cos\theta \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

- **Problem setting: Coordinate tracking in industry**
  - Fast & accurate alignment of values from different systems
  - Problem: Calculation requires multiplication, trigonometry
    - Fixed point arithmetic, i.e., represent as $\pm$ [0...2^d].[0...2^$(31-d)$]
    - n x m matrix-vector multiplication
      - Variant 1: Long multiplication (school variant)
        - On ASICs: Recirculation → Throughput down by 1/(2mn)
        - On NPUs: Direct multiplication possible
      - Variant 2: Log-space mult. (a*b = exp(log(a) + log(b))) with LUTs
        - On ASICs: Large table sizes, but cannot reuse tables
        - On NPUs: Medium table sizes, tables often reusable
    - Trigonometry tables may grow large (2^{30} * 32 bit → 4 GB of space)
      - On ASICs: Split tables via $\sin a + b = \sin a \cdot \cos b + \cos a \cdot \sin b$
      - On NPUs: Also approximate via 6th degree Chebyshev poly.

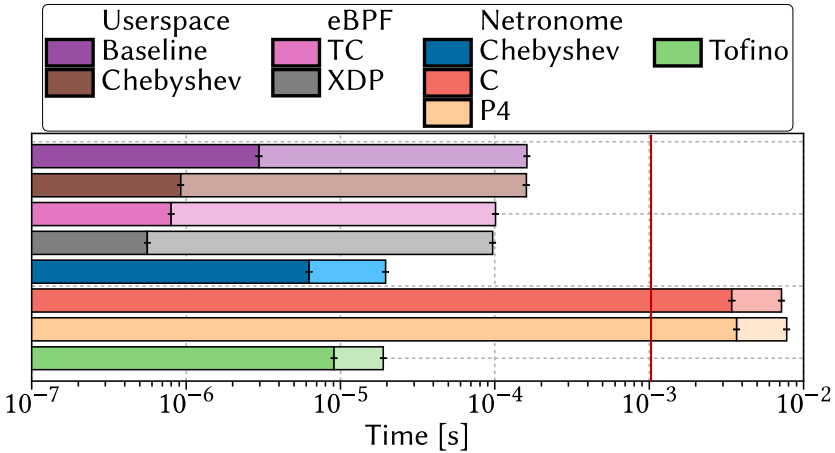
INC Example: Coordinate Transformation [ICPS 21]: Results





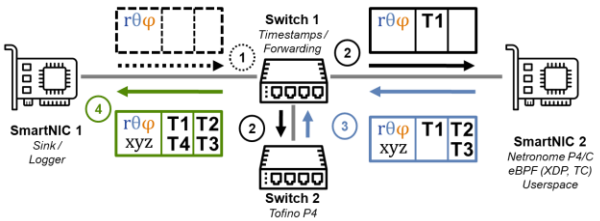
$$\begin{bmatrix} r\sin\theta\cos\varphi \\ r\sin\theta\sin\varphi \\ r\cos\theta \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

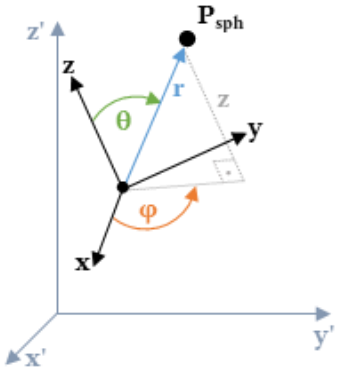
- **Evaluation 1: Calculation times in controlled setting**
  - 9000 calculations each



  - Raw calculation times (darker): CPUs perform best
  - Multiplication support on NPUs costly, optimizations help
  - Round-trip times (darker + lighter): ASICs/NPUs profit

INC Example: Coordinate Transformation [ICPS 21]: Results



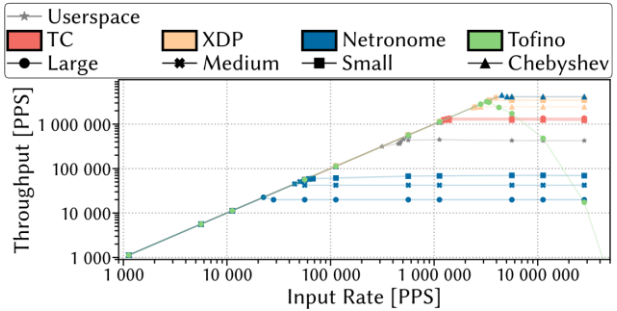


$$\begin{bmatrix} r\sin\theta\cos\varphi \\ r\sin\theta\sin\varphi \\ r\cos\theta \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

- **Evaluation 2: Accuracy**
  - Maximum tolerable error of 10 µm met by most ASIC/NPU configurations, NPU-based approximation approach fails

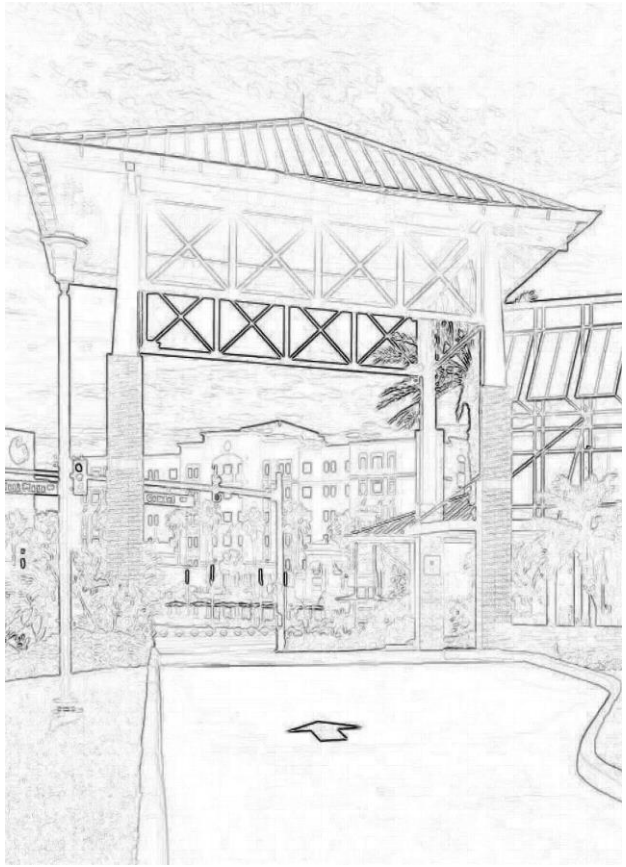
	Userspace	Chebyshev	Layout Large	Layout Medium	Layout Small
Eucl. Dist. [μm]	0.2	19.7	0.5	0.4	2.9
Violations [%]	0	48.5	0	0	0.7

- **Evaluation 3: Reliability**
  - ASIC/Tofino drops randomly after saturation point



→ **Need to prioritize recirculated traffic on this platform**

NPU-based INC Example: Computer Vision Offloading [ENCP 19]: Setting



- **Problem setting: In-network edge detection**
  - Given: Picture P (grayscale, $p \times q$ pixels)
  - Define: Filter F (grayscale or binary, $m \times n$ pixels)

Prewitt operator:

-1	0	1
-1	0	1
-1	0	1

F_{Δ_H}

-1	-1	-1
0	0	0
1	1	1

F_{Δ_V}

Scharr (symmetric Sobel) operator:

-47	0	47
-162	0	162
-47	0	47

F_{Δ_H}

-47	-162	-47
0	0	0
47	162	47

F_{Δ_V}

- Filter response: $R_{\Delta_{Dir}}(x,y) = \sum_{i=1}^{m}\sum_{j=1}^{n} P(x-i+a, y-j+a)\, F_{\Delta_{Dir}}(i,j)$
  - Maximum response $|M| = \sqrt{R_{\Delta_H}(x,y)^2 + R_{\Delta_V}(x,y)^2}$
  - Can be approximated: $|M| \propto \left|R_{\Delta_H}(x,y)\right| + \left|R_{\Delta_V}(x,y)\right|$

NPU-based INC Example: Computer Vision Offloading [ENCP 19]: Setting

Common pattern in most current scenarios:
Few individual operations on **many small** items

Independent of other pictures

Only local information needed (surroundings of a pixel)

Only addition/subtraction and multiplication of integers

Minimal global state (if any, maximum $|M|$ for normalization)

- **Problem setting: In-network edge detection**
  - Given: Picture P (grayscale, $p \times q$ pixels)
  - Define: Filter F (grayscale or binary, $m \times n$ pixels)
    - *Prewitt operator:*

-1	0	1
-1	0	1
-1	0	1

F_{Δ_H}

-1	-1	-1
0	0	0
1	1	1

F_{Δ_V}

    - *Scharr (symmetric Sobel) operator:*

-47	0	47
-162	0	162
-47	0	47

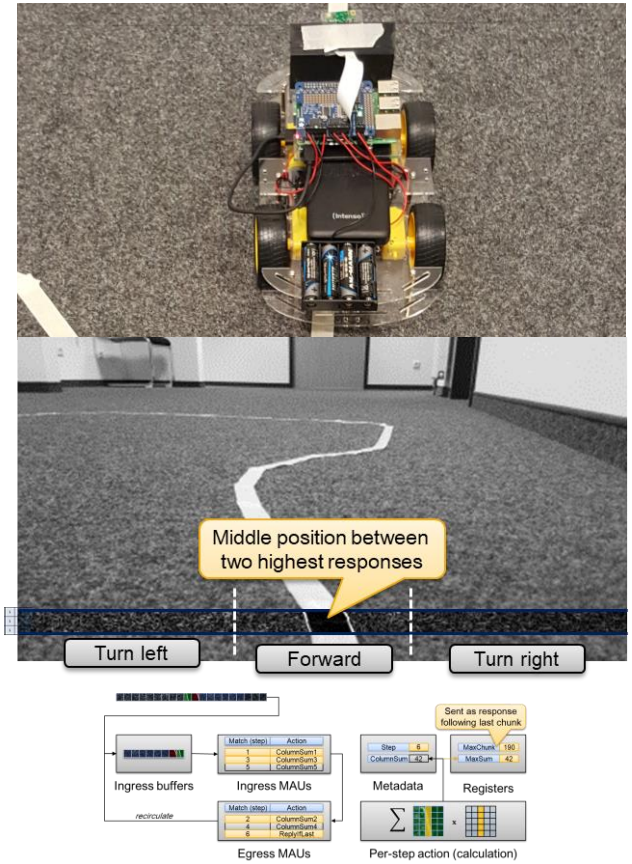
F_{Δ_H}

-47	-162	-47
0	0	0
47	162	47

F_{Δ_V}

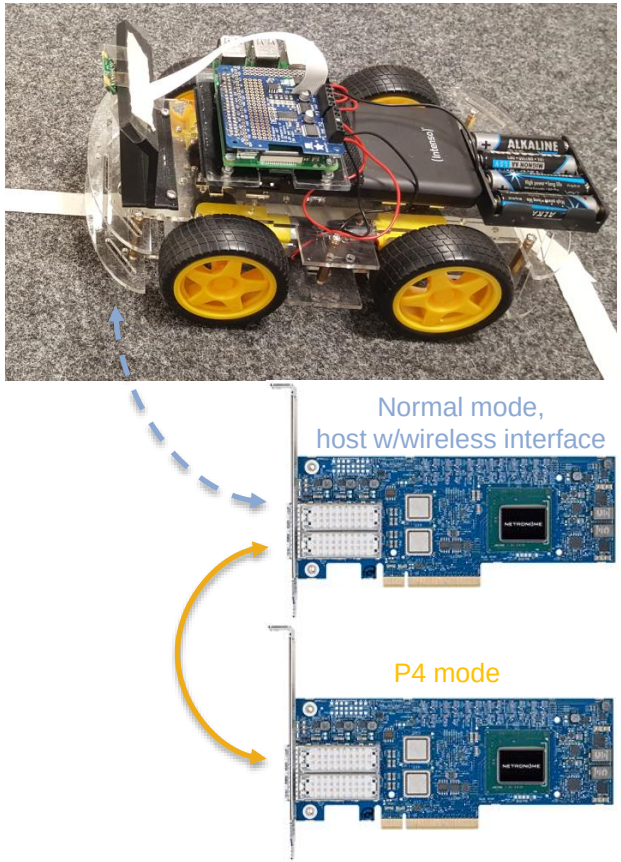
  - Filter response: $R_{\Delta_{Dir}}(x,y) = \sum_{i=1}^{m}\sum_{j=1}^{n} P(x-i+a, y-j+a)\, F_{\Delta_{Dir}}(i,j)$
    - Maximum response $|M| = \sqrt{R_{\Delta_H}(x,y)^2 + R_{\Delta_V}(x,y)^2}$
    - Can be approximated: $|M| \propto \left|R_{\Delta_H}(x,y)\right| + \left|R_{\Delta_V}(x,y)\right|$

NPU-based INC Example: Computer Vision Offloading [ENCP 19]: Implementation



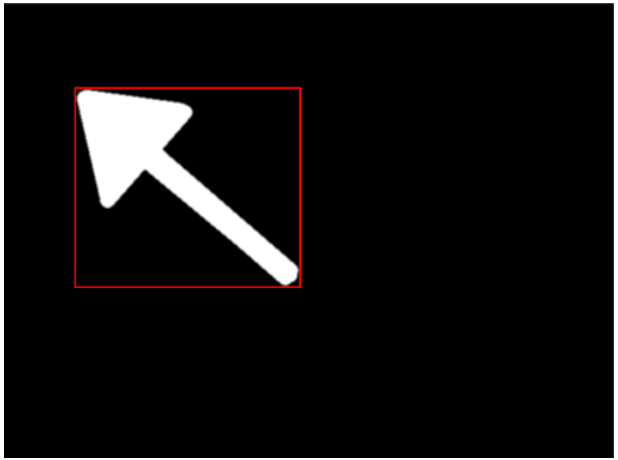
- **Application: Steering a toy car via P4 edge detection**
  - Captured & preprocessed on car (Python program), identification of middle of line in (pure) P4 program on NPU
- **Challenges**
  - **Large payload not accessible in P4** → Reduced chunk size → Messaging overhead
  - NPU's P4 pipeline too short for "complete" convolution → Use recirculations, split pipeline into multiple similar passes
  - NPU's P4 also has no associative memory (neither ASIC's)→ **Susceptible to re-ordering**
  - NPU's P4 programs are restricted in size → Maximum filter sizes

NPU-based INC Example: Computer Vision Offloading [ENCP 19]: Evaluation



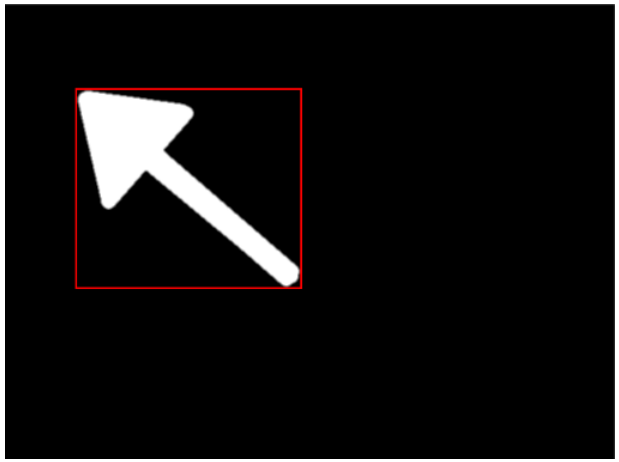
- **Real-world and synthetic benchmarks on Netronome Agilio CX 2x25GbE SmartNICs**
  - 2 connected NICs: 1 as car gateway/generator, 1 for P4
- **Filter- & chunk sizes: Up to 10x10 pixels**
  - O(1): Pipeline lengths (in-/egress)
  - O(n): Table entries, calculations per action
  - Good results at 5x5 already
- **Throughput: 19 fps (5x5); 77 fps (10x10)**
  - Processing of last chunk at 5x5: 150µs (stddev 1.3ms)
  - Processing of last chunk at 10x10: 187µs (stddev 0.6ms)
  - 13.7% drops at 5x5; none for 10x10 → buffering/recirculation

NPU-based INC Work-in-Progress: Extended CV Functionality I: Setting

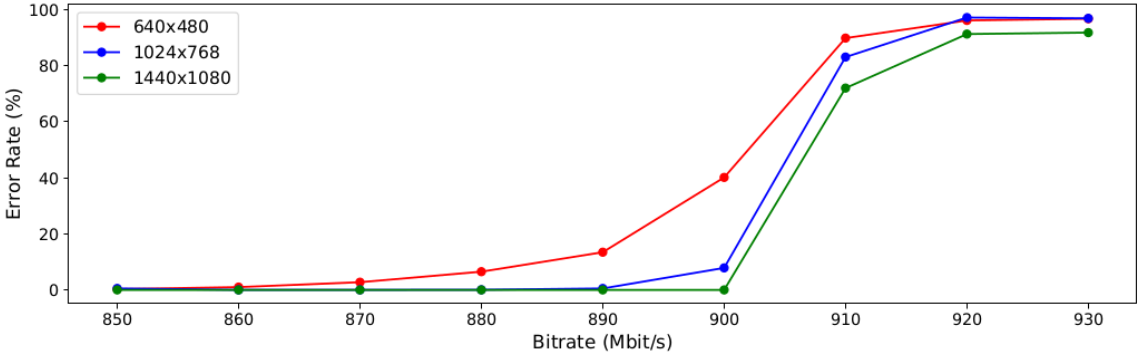


- **Problem setting: Data reduction via area-of-interest**
  - Only send packets containing region with pixels > threshold
- **Challenges**
  - Many-core architecture + memory hierarchy →
    Using too much shared memory incurs overhead →
    Split local (“for-each-packet”) / global (“for-trailer”) ops
    - Local: Save min/max x/y coordinates for pixel values > threshold
    - Global: Calculate min/max over “local” thresholds
  - Asynchronous thread operation →
    Trailer of images may arrive before threads finish local ops→
    Process n locally directly, globally when n+1 trailer arrives →
    Trade lag of >= 1 image for consistency

NPU-based INC Work-in-Progress: Extended CV Functionality I: Results

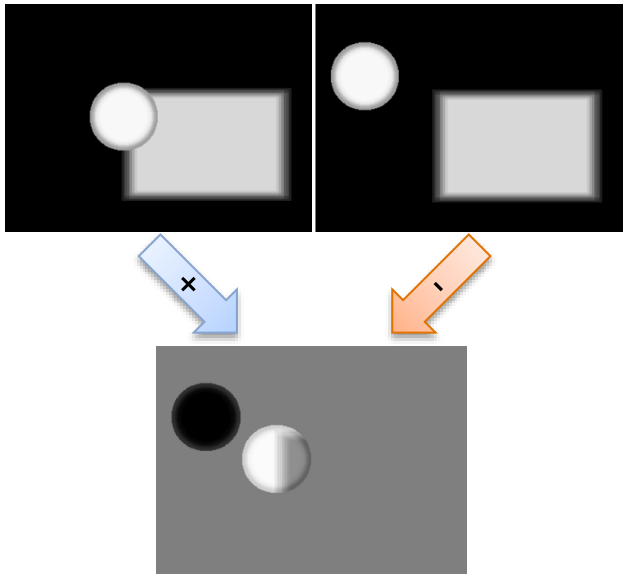


- **Evaluation: Introduced error levels / throughput**
  - Measured deviation from ground truth (exact are-of-interest)

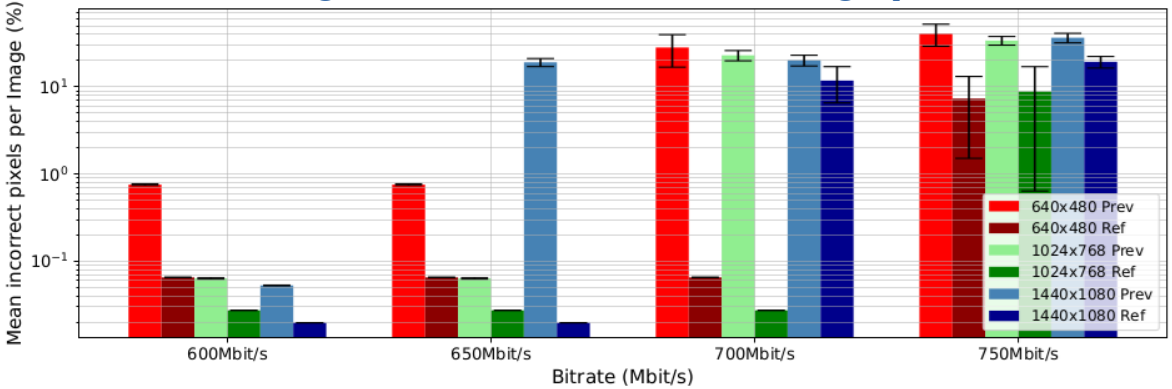


  - Sustainable throughput with < 1% error
    - 640 x 480 px: 341 FPS
    - 1024 x 768 pix: 138 FPS
    - 1440 x 1080 px: 70 FPS

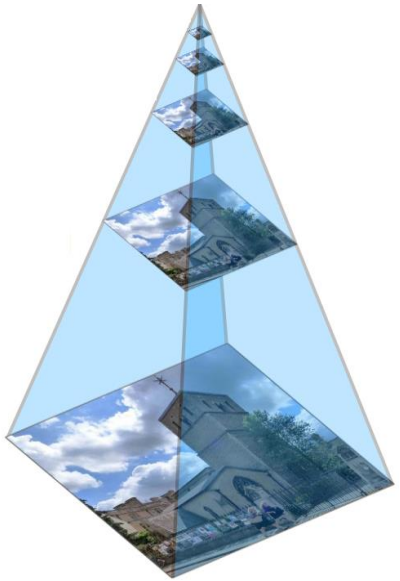
NPU-based INC Work-in-Progress: Extended CV Functionality II



- **Problem setting: Data reduction via image diff**
  - Only send packets when image differs significantly from last
- **Main challenge**
  - Need to save entire image to memory → #Streams limited
- **Evaluation: Again error levels, throughput**

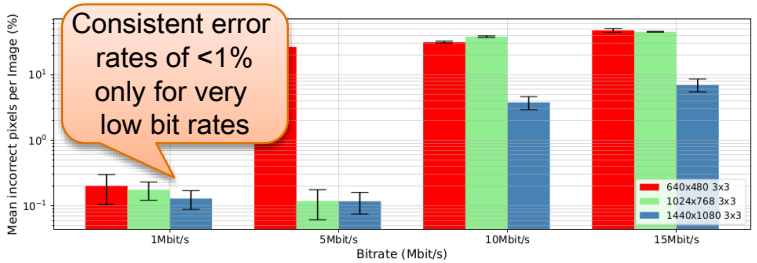


NPU-based INC Work-in-Progress: Extended CV Functionality III



0	1	2	3		1023
1024	1025	1026	1027		2047
2048	2049	2050	2051		3071
3072	3073	3074	3075		4096

- **Problem setting: On-path Gaussian blur for images**
  - Uses convolution operation from previous example
- **Main challenge**
  - Convolution costly + regions to convolute may cross packets
    - → Performing convolution on full packet too costly/slow
    - → Need to coordinate when to start which convolution (hard)
    - → **Alternative: Sliding-window approach with lag**
- **Evaluation: Again error levels, throughput**



COM SYS | RWTH AACHEN UNIVERSITY

In-Network Computation, ASICs and NPUs: Challenges and Possible Directions

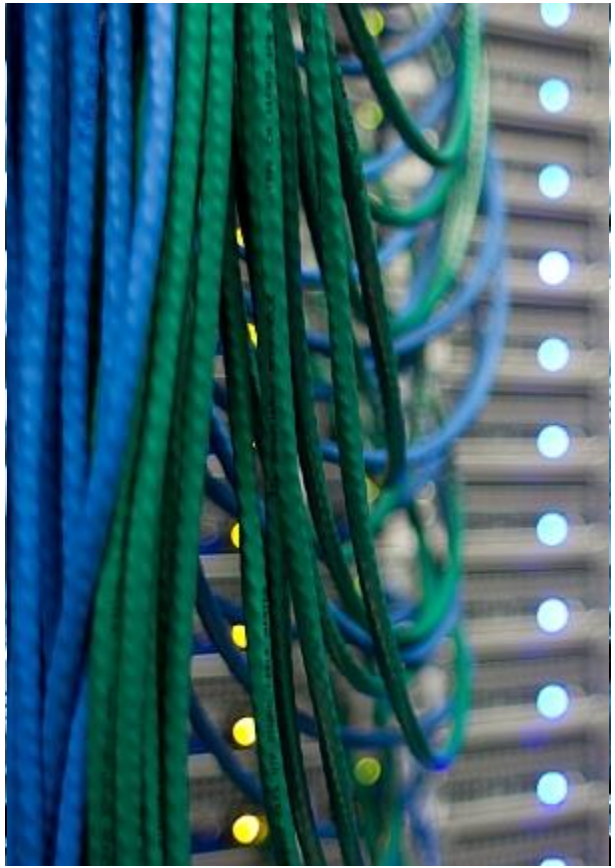
- **Mathematical functionalities**
  - ASICs: Need to **diligently design LUTs early on**
    - **Further research LUT-based calculations in general?**
  - Non-P4 NPUs: Few problems, but processing bounds unclear
  - P4 ASICs/NPUs: One-pass paradigm causes overhead
    - Recirculations required but reduce throughput
    - Recirculations also cause queueing problems (new vs. recirc'd pkts)
  - P4 ASICs/NPUs: One-stage-per-table paradigm
    - Recirculate packet (latency) vs. duplicate tables (memory)
    - **Introduce tables to read multiple entries from per pass?**
  - **Do we need "real" ALUs on Networking Hardware?**
    - Divisions hard but needed even in "core networking" scenarios…
    - At least some statistical functions such as (rolling) avg, stddev?

In-Network Computation, ASICs and NPUs: Challenges and Possible Directions

- **Architecture / memory organization**
  - P4 ASICs/NPUs: Cannot access full packet
    - Workaround 1: Define payload as “headers” → Parsers limited…
    - Workaround 2: Make packets smaller → Messaging overhead…
    - **Fundamental limitation?**
  - P4 ASICs/NPUs: Read-modify-write memory access
    - Required for pipelined execution
    - **Allow conditional admittance (“per-flow register locking”)?**
  - ASICs/NPUs: Re-ordering and multiple-packet data hard
    - **Re-formulate problems so that they are more “local”?**
    - **Consider “lagged” execution & “trigger packets”?**
    - **Introduce “accumulation memory” to compensate?**

- **Further challenges / thoughts**
  - ASICs: Limited packet generation on the data plane
    - Packet generator highly limited (1 fixed packet / pipeline) to rewrite
    - **Use templates that can replace ingress packets in egress?**
  - ASICs: Limited time-awareness of DP (CP packets & pkgten)
    - **Add further mechanism for time-triggred DP operations?**
  - ASICs/NPUs: Lack of cryptographic support
    - Big Netronome NICs have “crypto” modules; functionality unclear
    - Allow hashes / checksums beyond CRC/IP?
    - Message authentication code support?
    - **AES may map to lookup / match-action principle [Che20]**
    - Safe mechanism to share/deploy secrets/keys on-path?

In-Network Computation and Processor-based SmartNICs: Summary



- **Processing data on-path is still in its infancy**
- **Scenarios meant to show viability of the approach and test out boundaries**
- **Further work on our side**
  - Security/Reliability: SYMBIOSYS Project: Software Testing
  - Scalability: MAKI Project: Pipeline Multi-Tenancy
  - Standardization: IRTF COIN: Use Case Draft → RFC

Credits

- Parts of these slides are based on joint work with Johannes Krude, Ike Kunze, Jan Scheiper, Matthias Bodenbenner, Robert H. Schmitt (all RWTH Aachen University)

- Pictures on slides 1, 3, 10: Barefoot Networks (Routers) & Netronome (NICs)
- Picture on slides 2/18: Wikimedia / Victorgrigas
  (adapted (cropped) from https://commons.wikimedia.org/w/index.php?title=File:Wikimedia_Foundation_Servers-8055_22.jpg&oldid=218547099)
  CC BY-SA (https://creativecommons.org/licenses/by-sa/3.0/)
- Graphic on slide 3 bottom: Felix Senger, COMSYS, RWTH Aachen University
- Pictures on slides 7/8: University of Central Florida
  (adapted (cropped, filtered) from https://www.ucf.edu/pegasus/files/2015/10/HEADER_Fairwinds_PEGF15.jpg)
- Pictures on slides 11-13: Felix Senger, COMSYS, RWTH Aachen University
- Picture on slide 14: Wikimedia / Cmglee
  (adapted from https://commons.wikimedia.org/w/index.php?title=File:Image_pyramid.svg&oldid=474572310)
  CC BY-SA (https://creativecommons.org/licenses/by-sa/3.0)
- Picture on slide 15: Wikimedia / Mykl Roventine
  (adapted from https://commons.wikimedia.org/w/index.php?title=File:The_ladder_of_life_is_full_of_splinters.jpg&oldid=450746547)
  CC BY-SA (https://creativecommons.org/licenses/by-sa/3.0)