Research Summary

Compiler Dataflow Accelerator Architect

Jianming Tong
Advisor: Tushar Krishna
Georgia Institute of Technology
(+1) 4703578082, jianmingtong.github.io
jianming.tong@gatech.edu

About Me

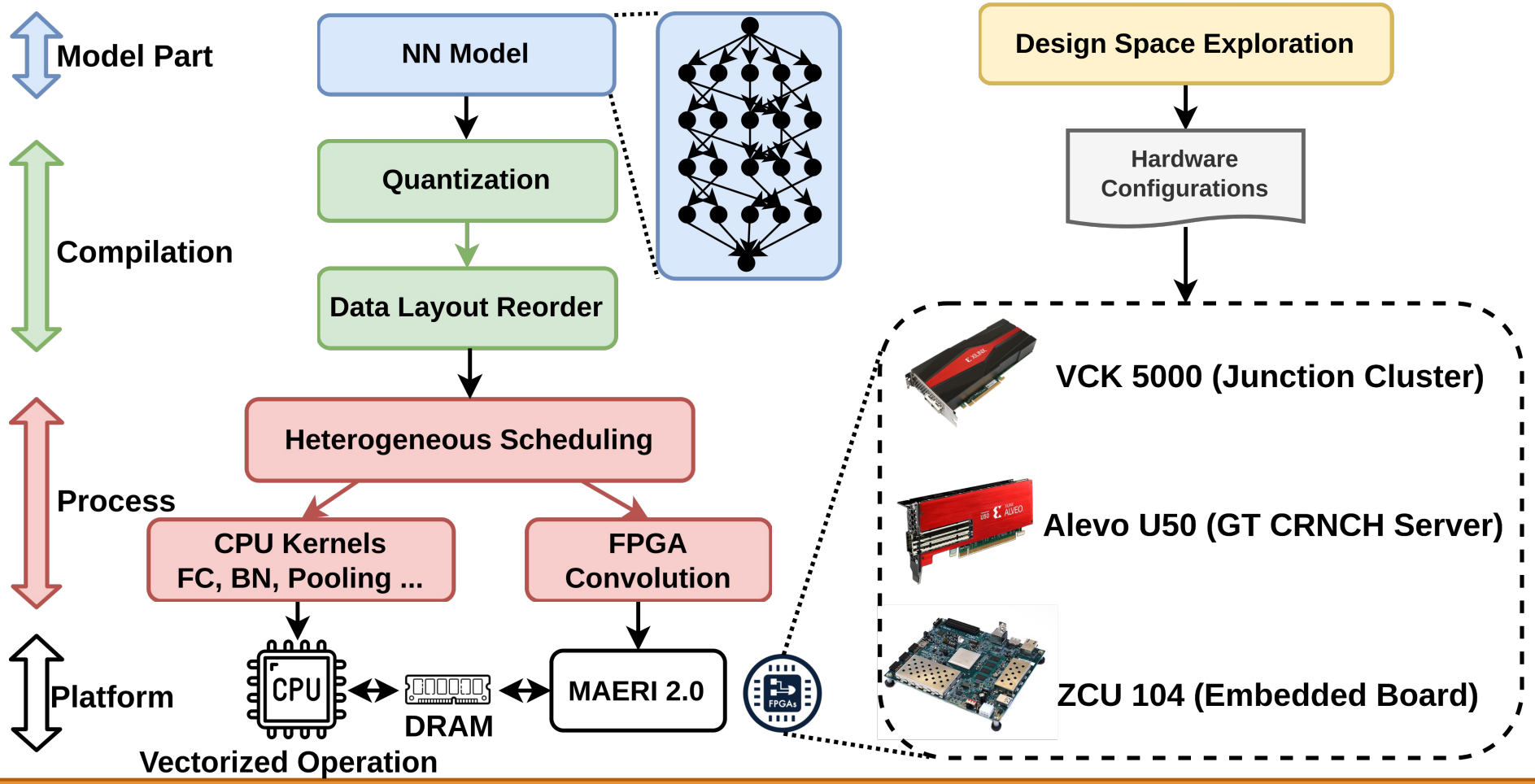
- FPGA Acceleration Leader @Synergy Lab in Georgia Tech
- Advisor: Tushar Krishna
- Finalist in Qualcomm Fellowship 2022
- MAERI 2.0 Main developer.

MAERI 2.0

End-to-end framework for inference on heterogeneous clusters

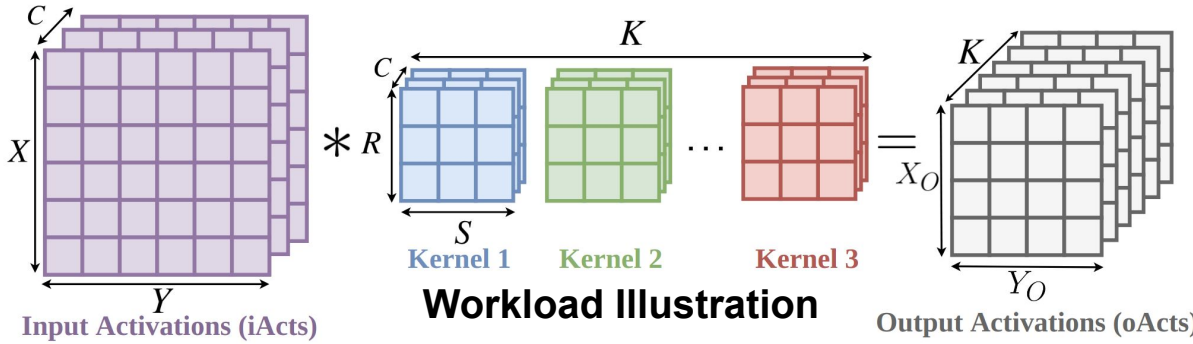
Jianming Tong, Yangyu Chen, Yue Pan, Abhimanyu Bambhaniya, Tushar Krishna
Georgia Institute of Technology
jianming.tong@gatech.edu

MAERI 2.0 Overview

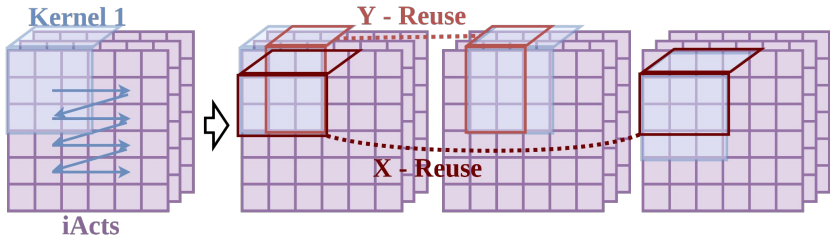


Overview: end2end framework enable NN inference on Heterogeneous Cluster

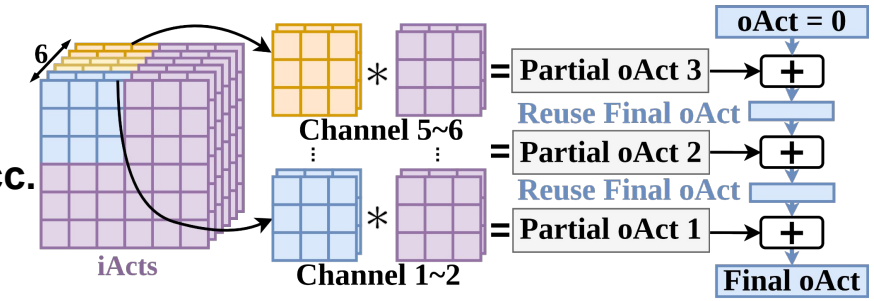
Challenge: How to Leverage Various Data Reuse



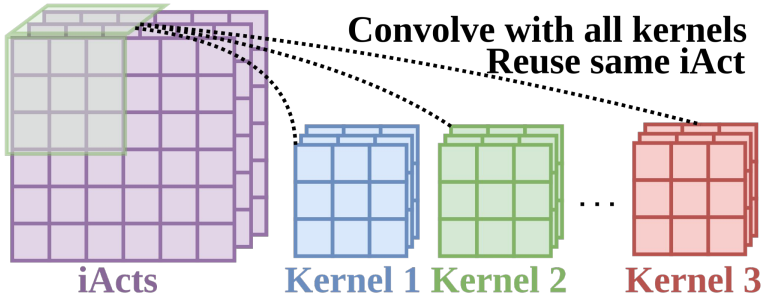
iAct Reuse
Sliding Windows
Overlap



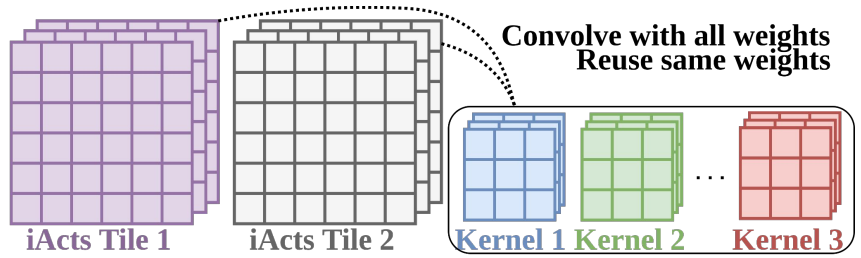
oAct Reuse
Partial Sum Acc.



iAct Reuse
Multi-Kernel

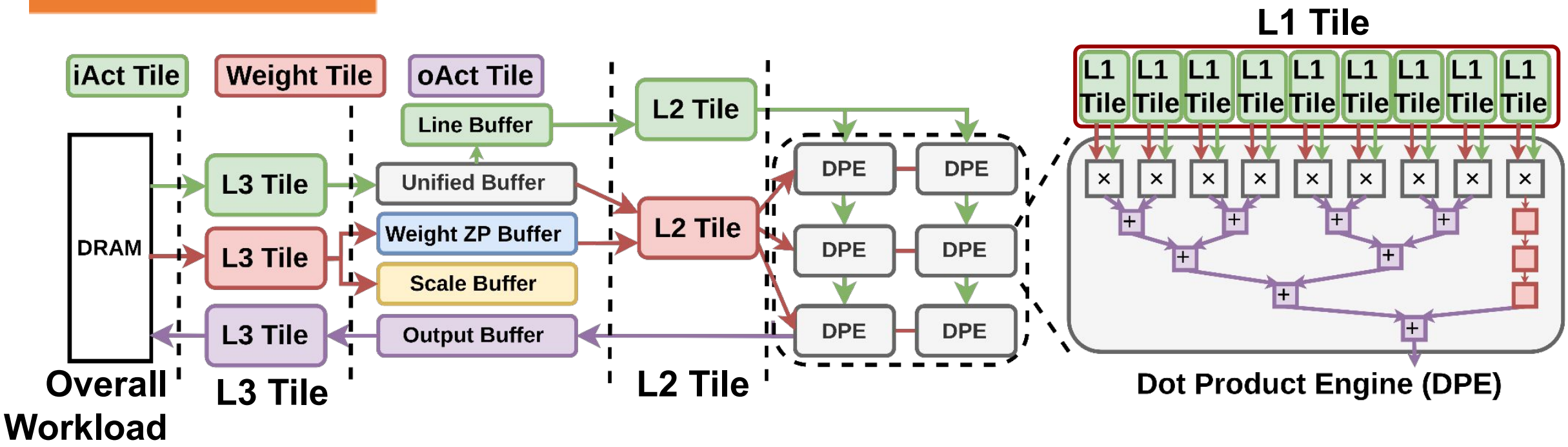


Weights Reuse
Multi-iAct Tiling



Insight: need on-chip buffer to store data for leveraging reuse.

MAERI 2.0 μarch - Overview



- L3 Tile: Transferred Data from DRAM to achieve continuous data access.
- L2 Tile: Data the entire DPE Array requires every cycle.
- L1 Tile: The data each single PE requires.

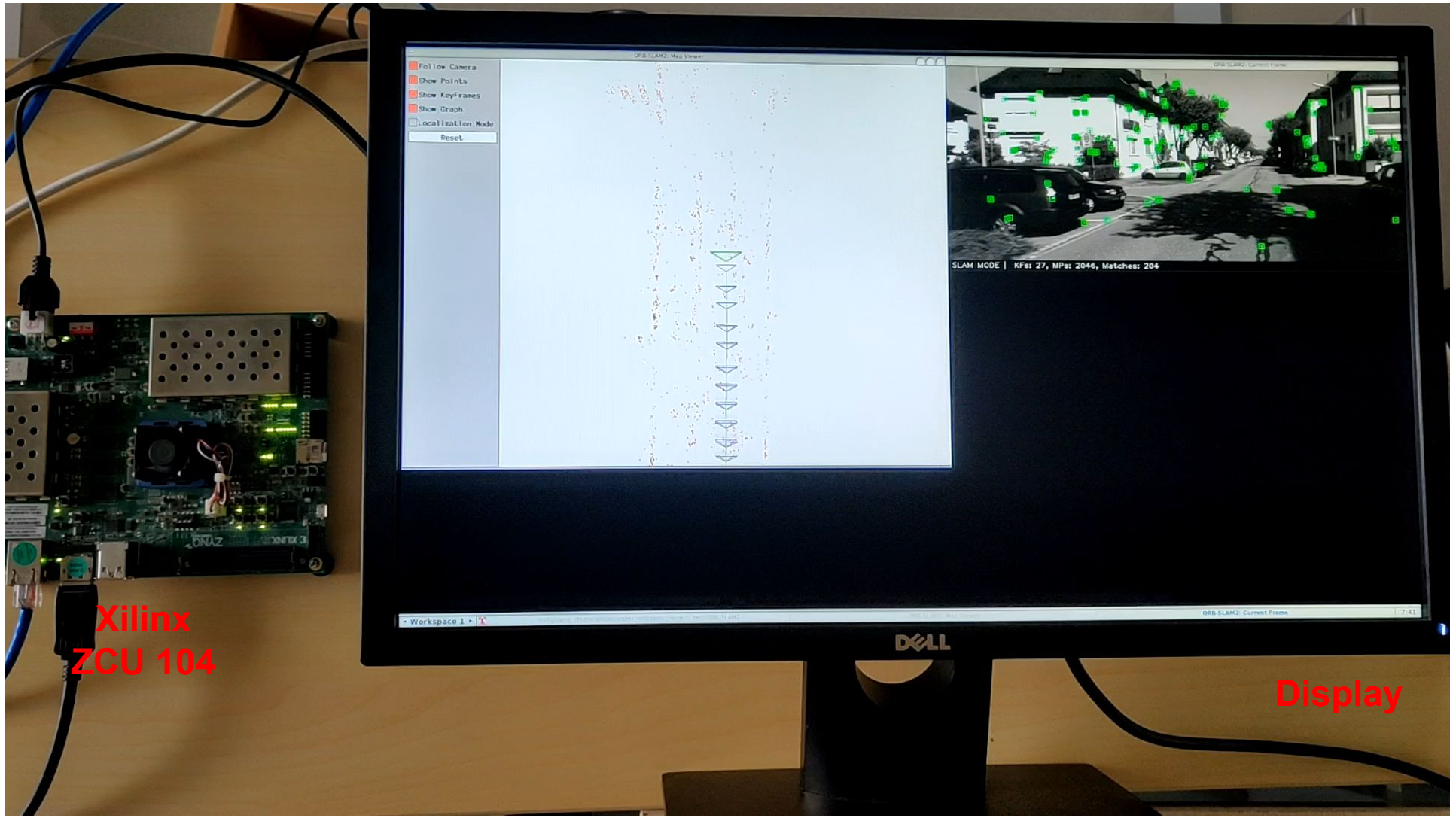
Insight: The multi-tiling processing demands compiler research!

ac2SLAM

FPGA Accelerated High-Accuracy SLAM with Heapsort and Parallel Keypoint Extractor

Cheng Wang, Yinkun Liu, Kedai Zuo, Jianming Tong, Yan Ding, and Pengju Ren.
Leader of the project
https://github.com/SLAM-Hardware/acSLAM
jianming.tong@gatech.edu

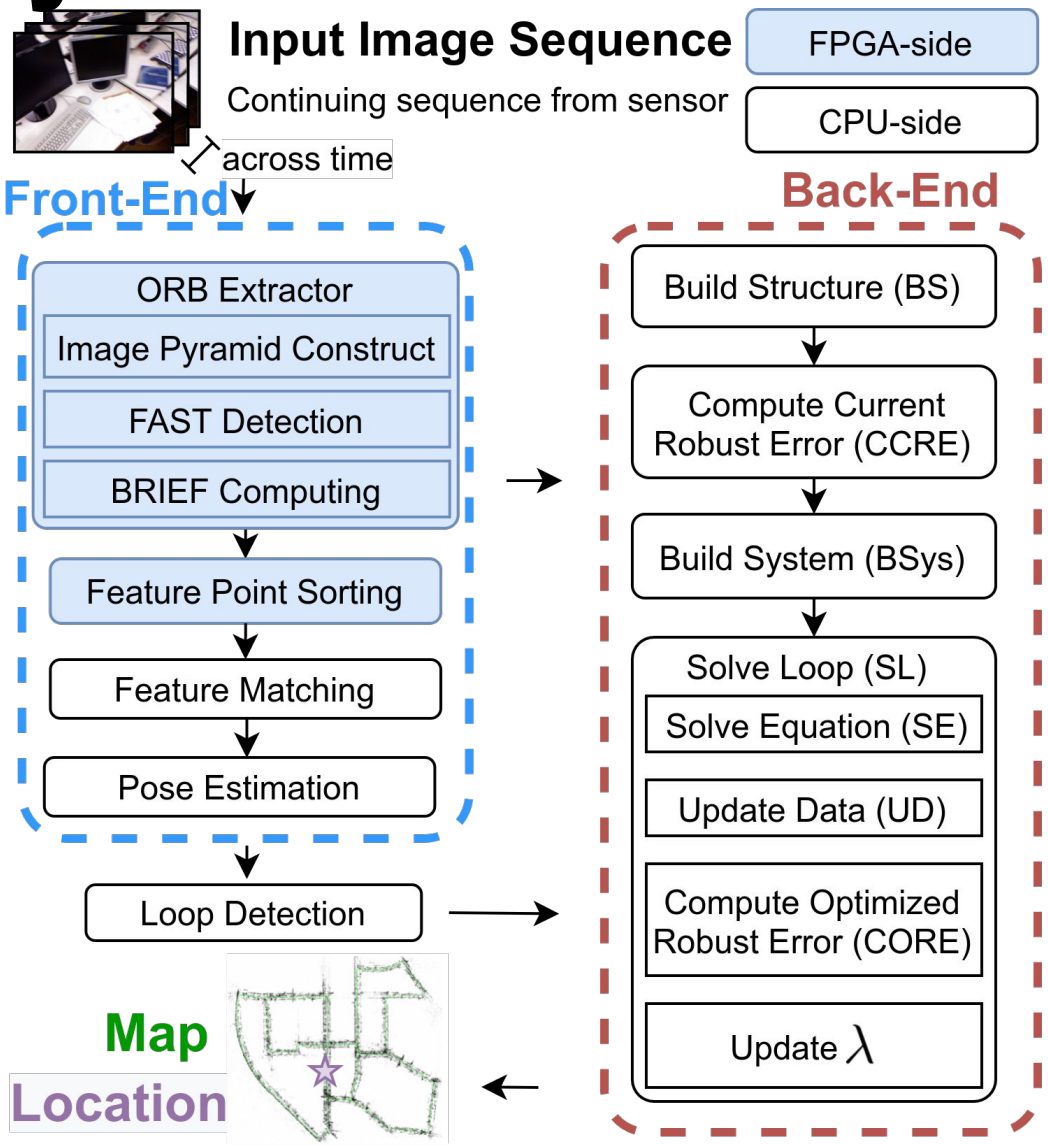
DEMO: SLAM on Embedded FPGA



SLAM = Simultaneous Location and Mapping

ac2SLAM: high-level system overview

- **Front-end (Feature Point Extraction)**
  - High Parallelism
  - Low Branches -> simple control
  - Accelerate on FPGA
- **Back-end (Optimization)**
  - High Parallelism
  - Multiple Branches -> complex control
  - Accelerate by multithreading on ARM CPU



Overview of FPGA Acceleration system

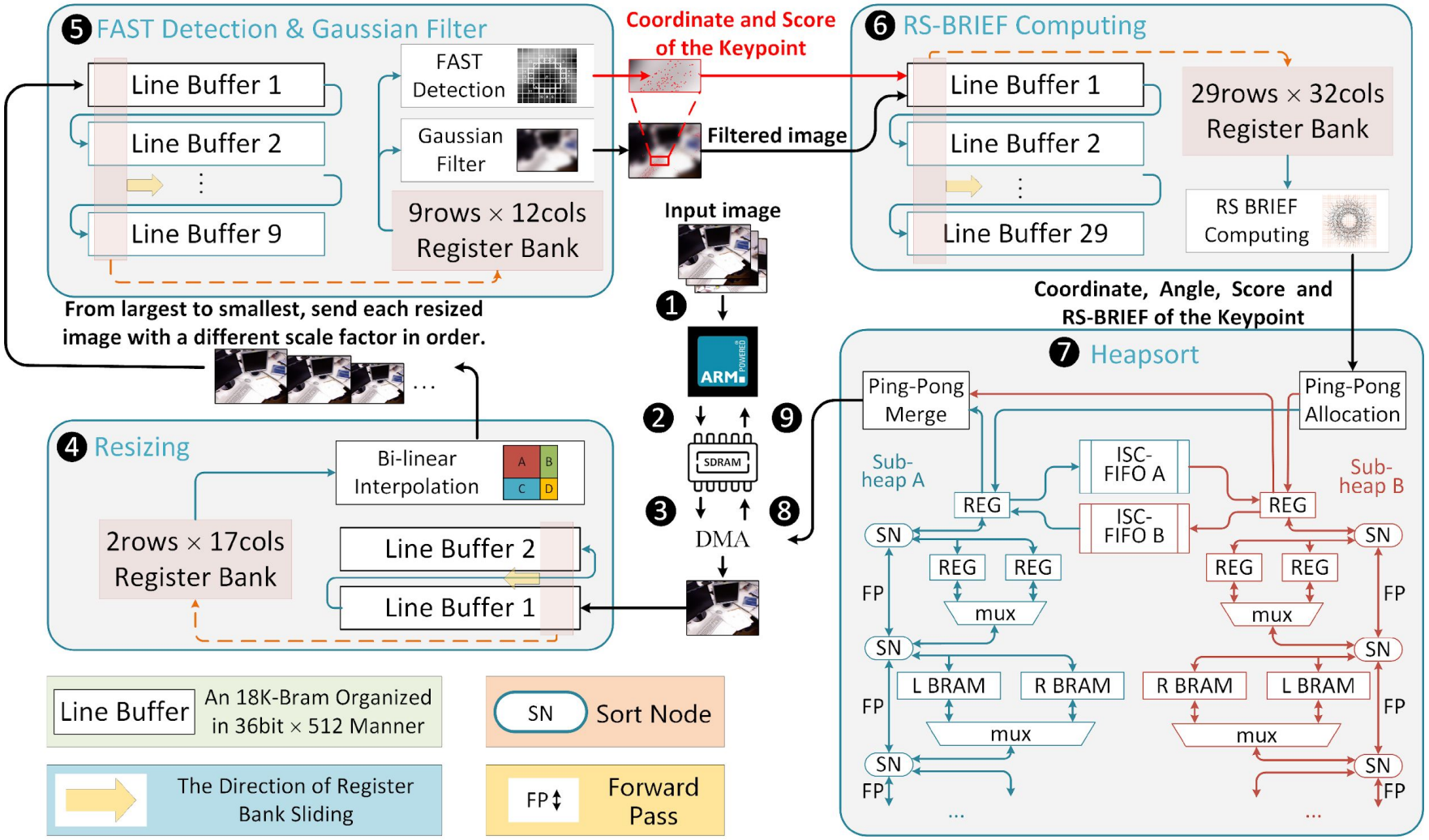
ac2SLAM: FPGA architecture overview

Streaming Architecture Challenges:

- Heterogeneous Engines
- Dynamic size of input

Solutions:

- Throughput matching
- Elastic workload support



ac2SLAM system FPGA architecture

ac2SLAM: Performance & Results

- **Performance**

Overall system Comparison

	TUM	KITTI
ac^{2}SLAM	15.5 fps	10.5 fps
ARM	7.5 fps	3.8 fps

Front-end (PL) Comparison

	TUM (640 × 480)	KITTI(1241 × 376)
ac^{2}SLAM	2.0 ms	5.1 ms
eSLAM	9.1 ms	not apply
ARM	80.2 ms	202.7 ms

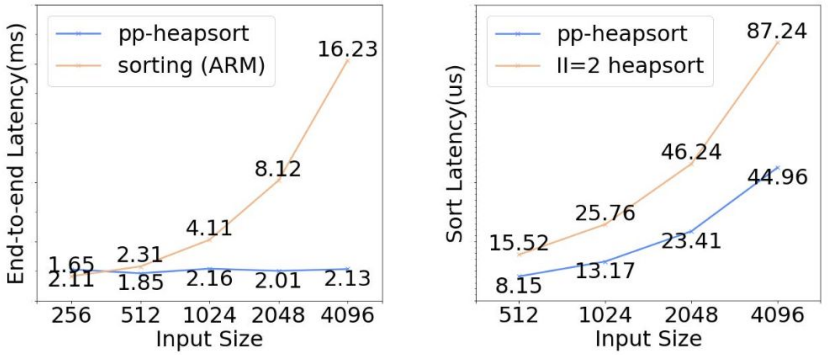
Overall 2X faster Compared with General CPU

40X & 4X faster compared with General CPU & Accel.

- **Overhead**: 1.55X Perf/Cost better than SOTA Accelerator

	LUT	FF	DSP	BRAM
ac^{2}SLAM	146572	74166	173	61
eSLAM	56954	67809	111	78

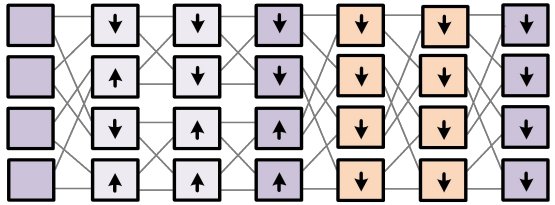
- **Scalability**: Linear Scalability to input size



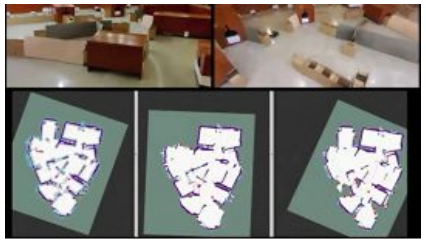
(a) end-to-end latency. (b) hardware latency.

Hardware Hierarchy Heapsort Scalability test with various input sizes（compared with CPU and DSA)

What can I do for u and myself?



SOM: Multi-stage multi-function network **(GLIVLSI)** U.S. patent

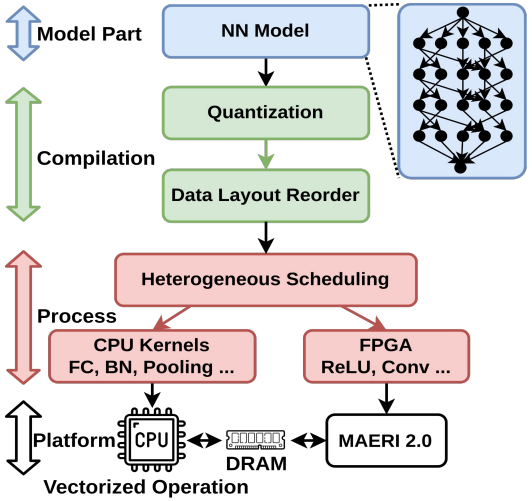


Multi-robot system
(ICRA) Open-sourced

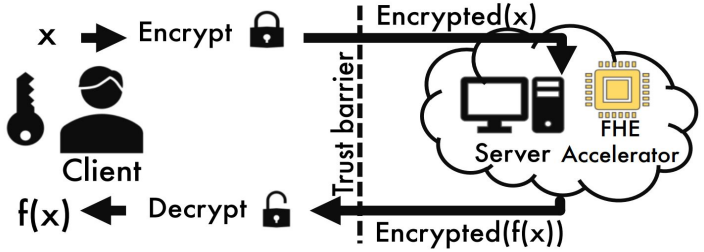
《On-chip Network》
Chinese Translator
On sale in China (thousands saling)

SLAM Accelerator
(FPT) open-sourced

MAERI 2.0
Main Developer



FHE Acceleration
Leader



My next steps

- **Speed gap for workload and architecture evolution**
- **Deployment gap for existing hardware and new appearing workload.**

Push boundary of compiler as the bridge between new workload an new/existing hardware!