**Programmable Dataflow Accelerators**

- Massive array of Processing Elements (PEs); each PE has ALU-like functional unit to perform operation every cycle (simple, programmable).
- PE's Private + shared memory sustain data reuse.
- Efficiently accelerate ML and media kernels.
- Architecture Variations:
  - Systolic arrays: TPU (Google), TensorCore (nVIDIA)
  - Spatially programmable architecture: Eyeriss (MIT), SCNN (nVIDIA), AI core (IBM), CSA (Intel)
  - Coarse-grained reconfig array: HyCUBE (NUS), DPU (Wave)

**Current Focus in System Stack**

- Front-End: Tensor Graph Optimizations
- Intermediate Representation (IR) with Scratchpad Memory
- Transformation for Dataflow Execution
- Architecture Specification

**DiRAC: Cycle-level μarch Simulation**

- Cycle-level Microarchitecture Simulation of DF Accelerator
- Need community infrastructure to learn and simulate DF accelerators.

**Loop Orchestration**

- Spatial loops: What subset of data each PE executes
- Temporal loops: Data Reuse in Memories.

**Adaptable Mappings = Better Results**

- Very high resource utilization
- Reuse of multiple operands
- Minimize DRAM accesses.
- Efficiently interleave compute with communication latency

**dMazeRunner Features**

- Non-expert programmers can explore space in seconds.
- Domain experts can perform directed search.
- Explore efficient designs for models/layers through DSE

**Acknowledgements**

This work was partially supported by funding from NSF grant CCF 1723476 – NSF/Intel joint research center for Computer Assisted Programming for Heterogeneous Architectures (CAPA) and from the grants NRF-2015M3C4A7065522 and 2014-3-00035 funded by MSIT.