INSTITUTE OF COMPUTER SCIENCE

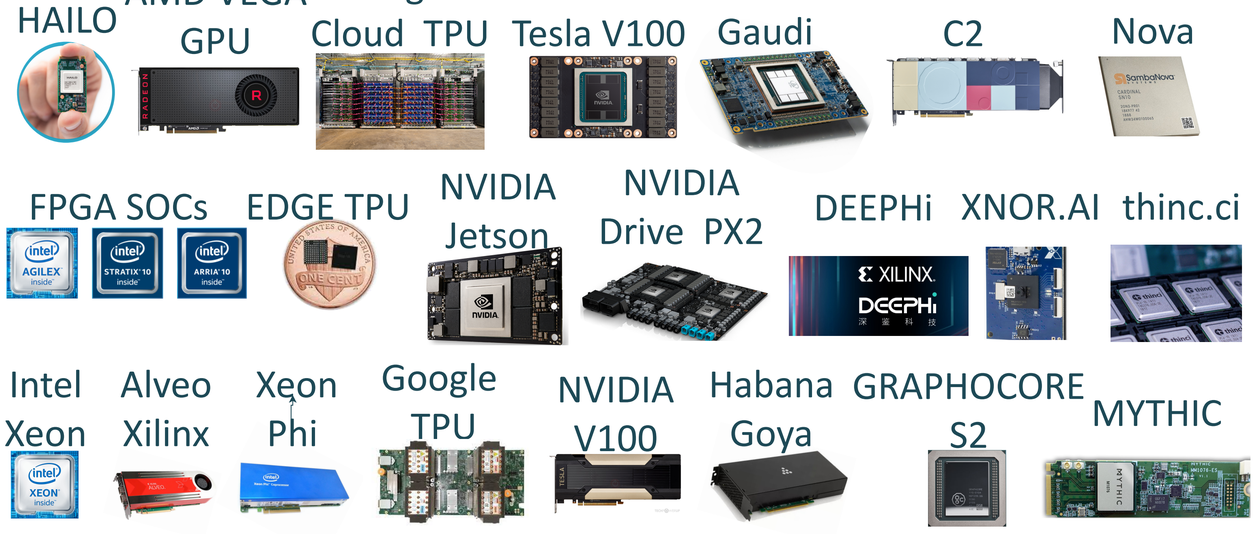
ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
UNIVERSITY OF CRETE

Arax: A Runtime Framework for Decoupling Applications from Heterogeneous Accelerators

M. Pavlidakis, S. Mavridis, A. Chazapis, G.Vasiliadis, and A. Bilas

Towards Extreme Heterogeneity

HAILO, AMD VEGA GPU, Google Cloud TPU, NVIDIA Tesla V100, Habana Gaudi, GRAPHOCORE C2, Samba Nova

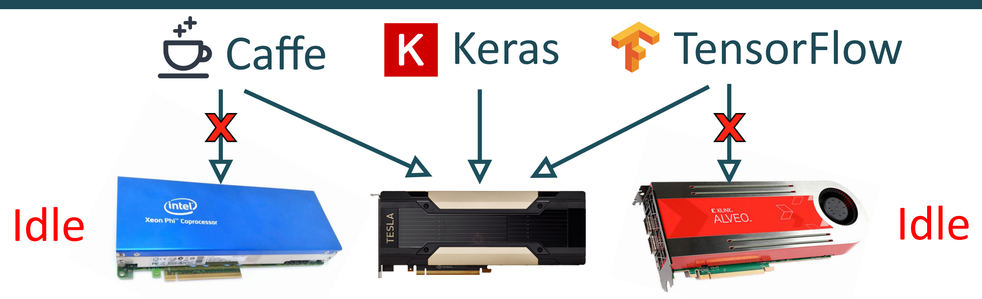


FPGA SOCs, EDGE TPU, NVIDIA Jetson, NVIDIA Drive PX2, DEEPHi, XNOR.AI, thinc.ci

Intel Xeon, Alveo Xilinx, Xeon Phi, Google TPU, NVIDIA V100, Habana Goya, GRAPHOCORE S2, MYTHIC

Existing Programming Models

Caffe, Keras, TensorFlow

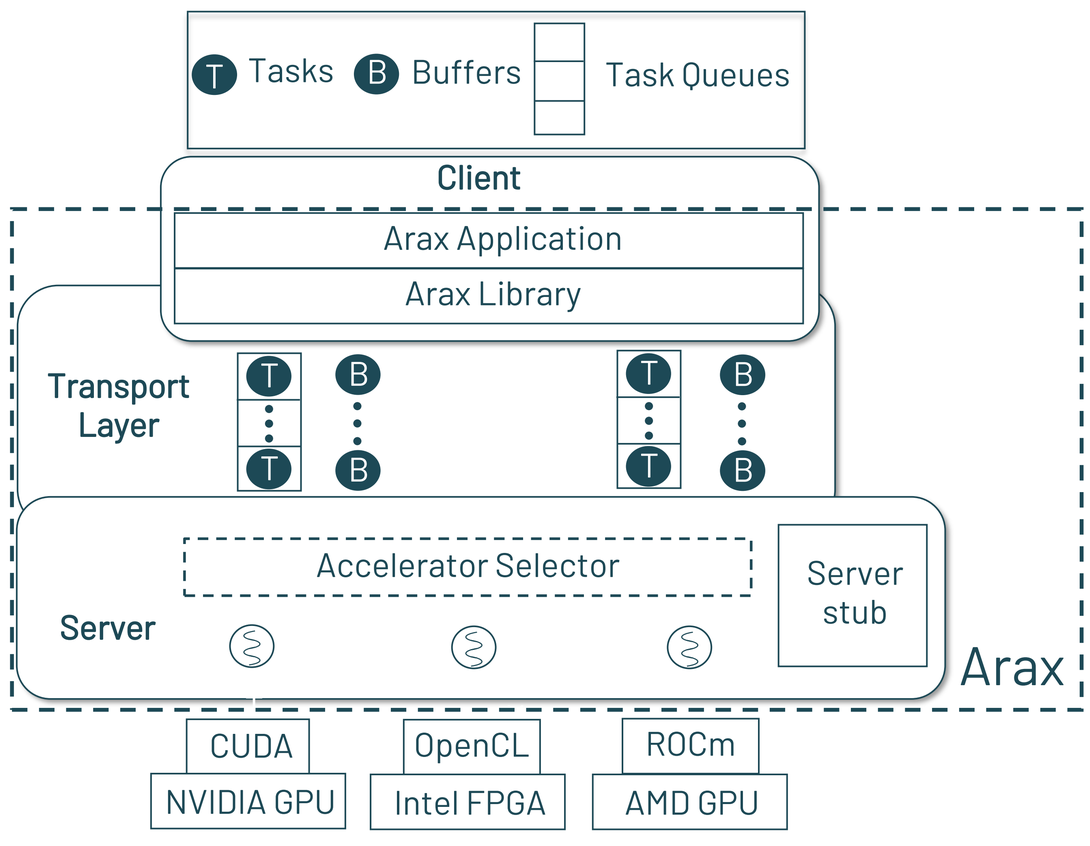


✗ Applications are statically bound to accelerators
✗ Leading to load imbalances

Arax: A Runtime System for Accelerators

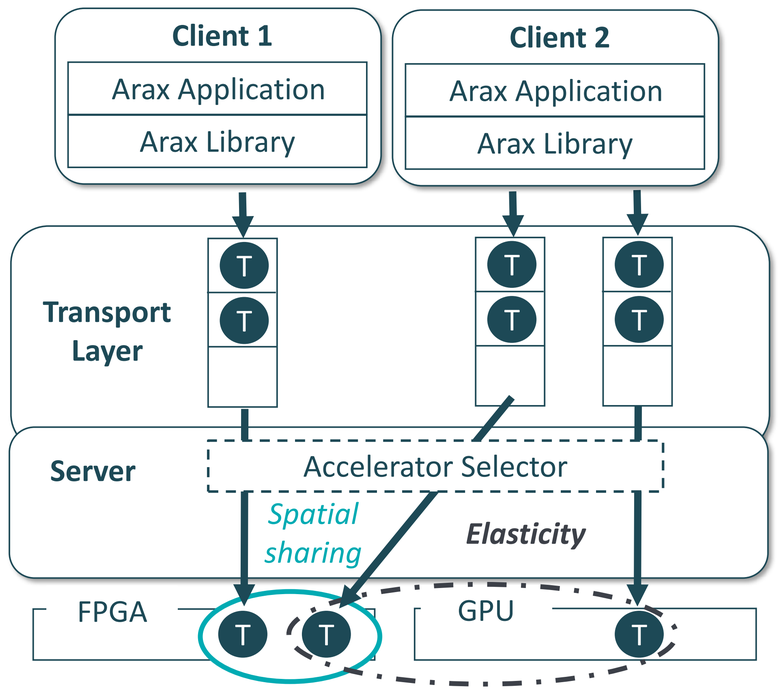
✓*Arax* abstracts applications from accelerators using RPC
✓*Arax* performs dynamic task assignment & memory management

RPC: Clients, Transport, Server



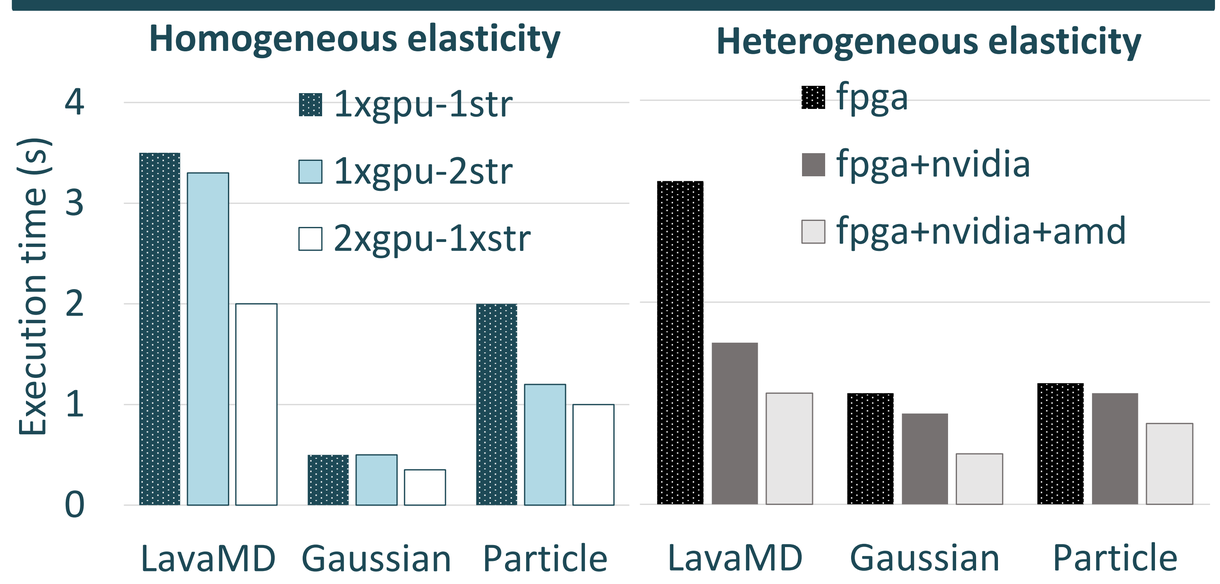
- Clients use Arax accelerator agnostic library
- The transport layer is shared across the server and the clients
- The server:
  - ➤ Receives tasks
  - ➤ Maps tasks to kernels
  - ➤ Executes kernels

Dynamic Accelerator Assignment



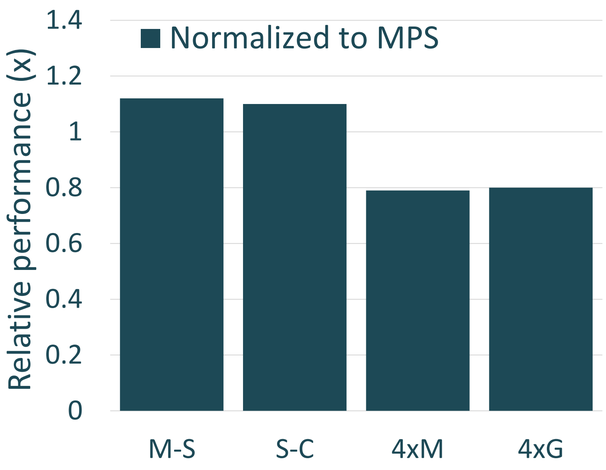
- Multiple tasks can execute to the *same* accelerator
  - ➤ Accelerator spatial sharing → improve resource utilization
- Tasks from the *same app.* can execute to different accelerators
  - ➤ Application elasticity → improve application performance

Application Elasticity



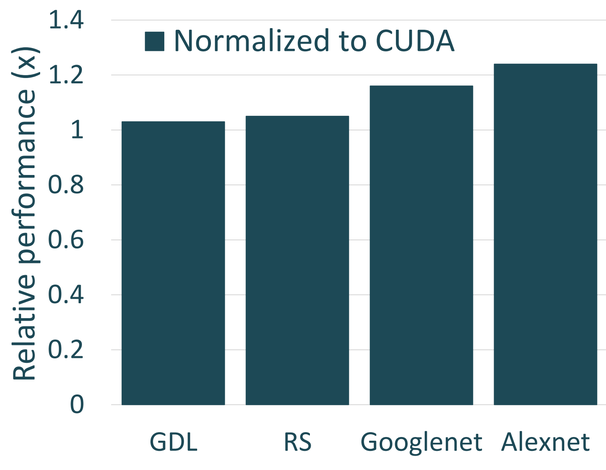
✓ *Arax* increases application resources → 3x less execution time

Accelerator Sharing



✓ Comparable to MPS →NVIDIA

Arax Overheads



✓ 12% geomean degradation

Arax highlights

✓ Runs upon CPUS, NVIDIA-AMD GPUs, and FPGAs
✓ Supports Caffe, TensorFlow, and Rodinia applications
✓ Achieves dynamic resource adaptation (up to 3x with elasticity)
✓ Offers equal performance to MPS (up to 20% improvement)
✓ Has near-native performance (up to 12% degradation)

Acknowledgments

HiPEAC, DEEP-SEA, THE EUPILOT

GitHub link