

# IRIS: A Portable Runtime System Exploiting Multiple Heterogeneous Programming Systems

**Jungwon Kim**, Seyong Lee, Beau Johnston,  
and Jeffrey S. Vetter

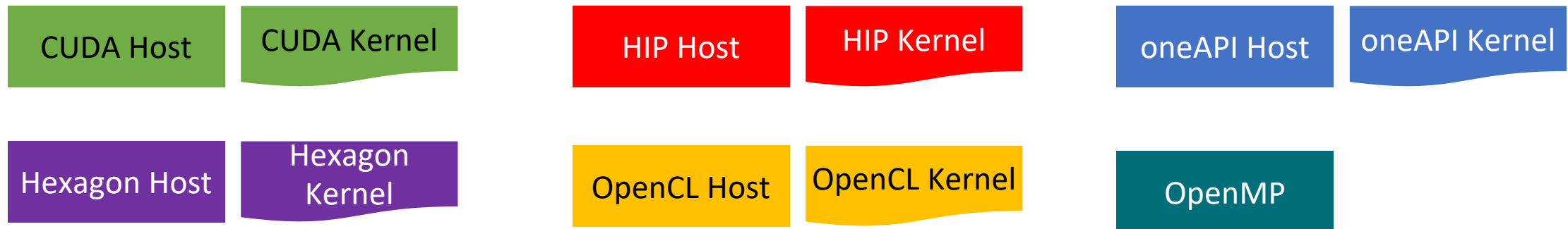
Oak Ridge National Laboratory

20 September 2021 @ IEEE HPEC '21

# No De Facto Standard for Heterogeneous Programming

- ORNL Experimental Computing Laboratory (ExCL) systems\*

Systems	Snapdragon	Jetson	Zynq	DGX			Oswald			Summit	Frontier
CPU	ARM	ARM	ARM	I	I	I	I	I	I	IBM	AMD
GPU	Qualcomm	NVIDIA		NVIDIA			NV	NV		NVIDIA	AMD AMD
FPGA			Xilinx				Intel	Intel			
DSP	Qualcomm										



\* ORNL ExCL: <https://excl.ornl.gov/>

# We Need Portability in Heterogeneous Programming

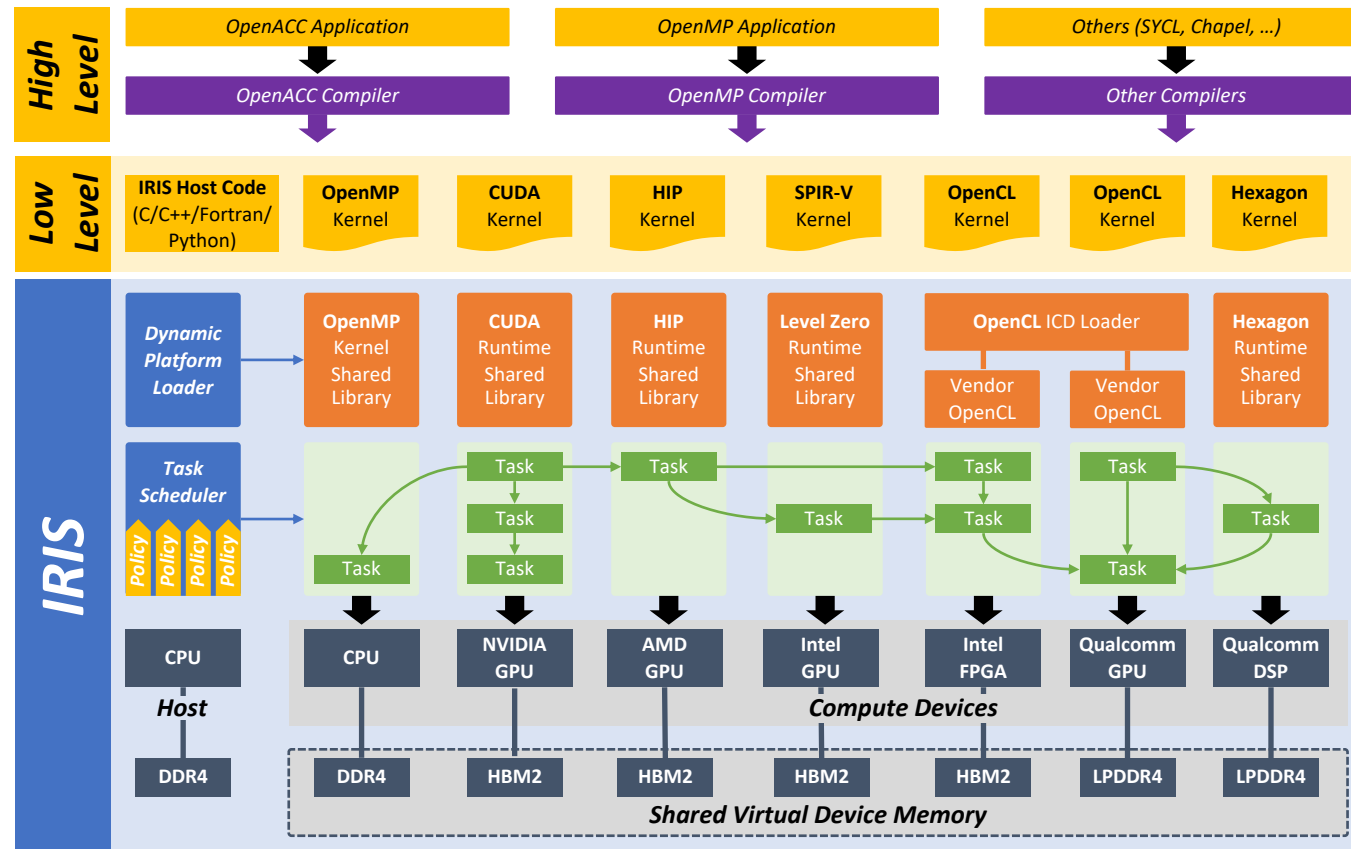
- Not portable program across different HW configurations

Systems	Snapdragon	Jetson	Zynq	DGX			Oswald			Summit	Frontier
CPU	ARM	ARM	ARM	I	I	I	I	I	I	IBM	AMD
GPU	Qualcomm	NVIDIA		NVIDIA			NV	NV	NV	NVIDIA	AMD AMD
FPGA			Xilinx				Intel	Intel			
DSP	Qualcomm										



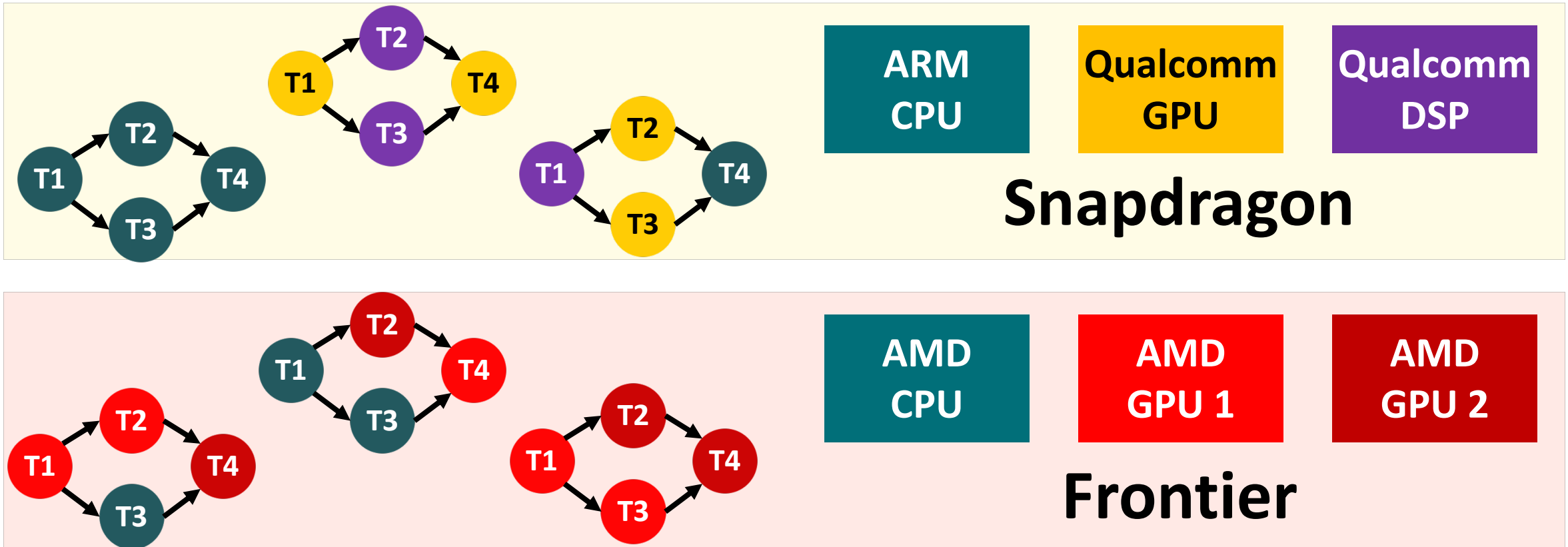
# Orchestrating Multiple Programming Systems

## • The IRIS Architecture



- Compilers
  - High level application → IRIS unified host code + native kernels
- Dynamic Platform Loader
  - Automatically discover all available accelerators and their programming systems
- Task Scheduler
  - Task: memory copy + kernel launch
  - DAG-style tasks graph across multiple devices
  - Device Selection Policies
- Shared Virtual Device Memory
  - An Illusion of single logical device memory across all physical device memories
  - Multiple local copies on multiple device memories (relaxed consistency model)

# Unified Host + Multiple Native Kernels + Shared Virtual Device Memory = **Portable Tasks & Flexible Scheduling**

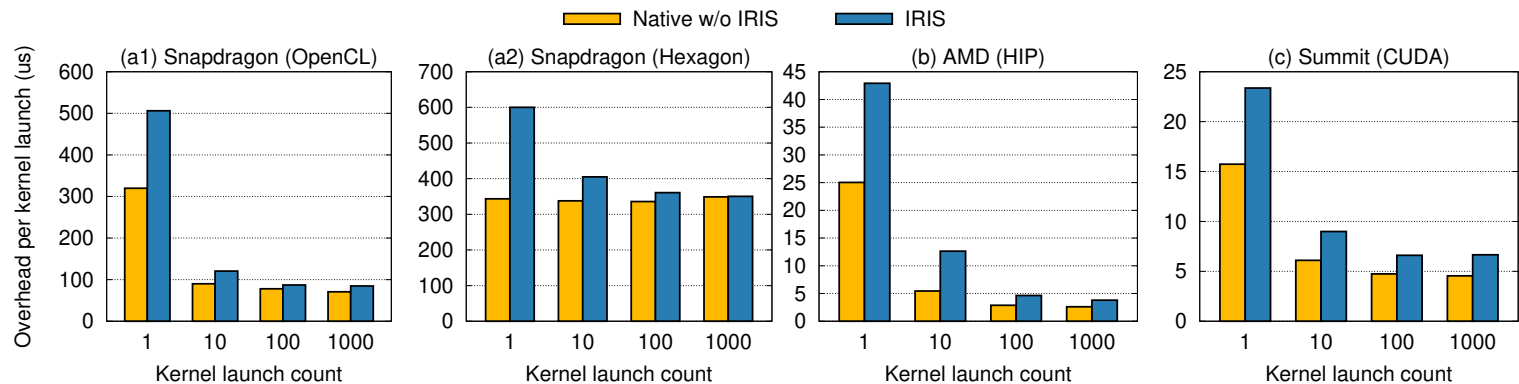


A task can be scheduled and run on any device.

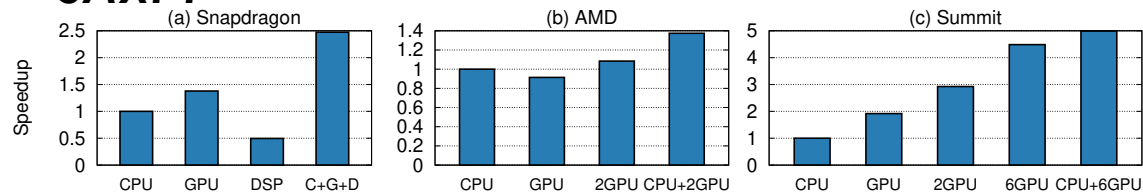
An IRIS application is portable across different heterogeneous systems.

# Evaluation

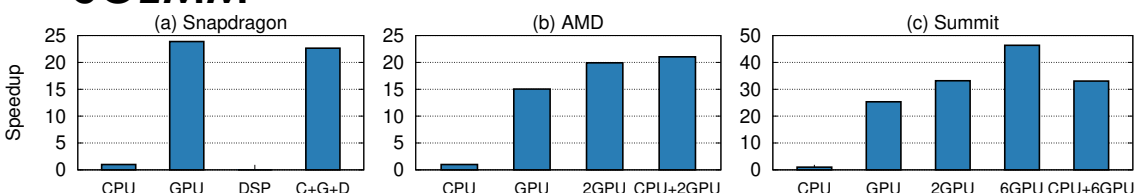
## Kernel Launch Overhead



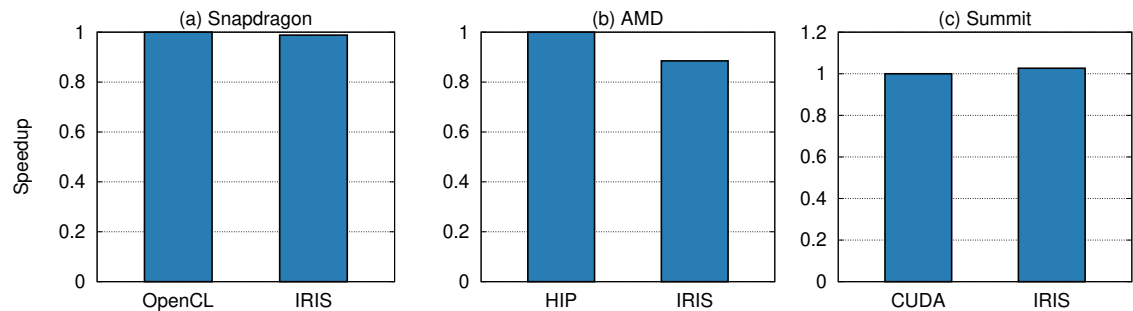
## SAXPY



## SGEMM



## LULESH



Systems	Snapdragon	AMD	Summit
CPU	Qualcomm OpenMP	AMD OpenMP	IBM OpenMP
GPU	Qualcomm OpenCL	AMD HIP	NVIDIA CUDA
DSP	Qualcomm Hexagon		

# Recap

<b>Situation</b>	No de facto standard for heterogeneous programming
<b>Task</b>	Achieving portability in heterogeneous programming
<b>Activity</b>	<p>We introduce a new portable runtime system, <b>IRIS</b></p> <ul style="list-style-type: none"><li>• Orchestrating multiple programming systems (CUDA, Hexagon, HIP, Level Zero, OpenCL, OpenMP)</li><li>• Unified Host + Multiple Native Kernels + Shared Virtual Device Memory → Portable Tasks &amp; Flexible Scheduling</li></ul>
<b>Result</b>	IRIS achieves portability, programmability, and performance

IRIS will be freely available soon at

<https://github.com/ORNL/iris>

# Acknowledgments

- This research used resources of the Experimental Computing Laboratory and the Oak Ridge Leadership Computing Facility at Oak Ridge National Laboratory, which are supported by the US Department of Energy's Office of Science of under contract no. DE-AC05-00OR22725.
- This research was supported by (1) the Defense Advanced Research Projects Agency's Microsystems Technology Office, Domain-Specific System-on-Chip Program and (2) the US Department of Defense, Brisbane: Productive Programming Systems in the Era of Extremely Heterogeneous and Ephemeral Computer Architectures.
- This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The US government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (<https://energy.gov/downloads/doe-public-access-plan>).