CPS: Medium: GOALI: Enabling Scalable Real-Time Certification for AI-Oriented Safety-Critical Systems

PI: Jim Anderson (UNC). Co-PIs: F. Don Smith (UNC), Ron Alterovitz (UNC), and Prakash Sarathy (Northrop Grumman)

Students: Joshua Bakita, Tanya Amert, Sergey Voronov, Syed Ali, Zelin Tong, Rohan Wagle, Sizhe Liu, and Angelos Angelopoulos

https://www.cs.unc.edu/~anderson/projects/rtai.html

UNC REAL-TIME SYSTEMS GROUP

NORTHROP GRUMMAN

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Motivation

Some of the most compelling use-cases for intelligent autonomy fall in safety-critical areas such as aviation and automotive. Certification is essential in these areas, but systems must be built compositionally; for example, a perception system must be able to be developed, certified, and upgraded independently of the planner, controller, or logging systems. Ensuing such independence is an unsolved problem for complex workloads, and for multicore platforms with accelerators.

On this poster, we use the GPU as a representative accelerator.

Project Focus

We decompose AI-oriented workloads into swappable **components** that are isolated from one another, allowing intra-component timing constraints to be verified independently. To isolate components from one another, we develop *time* and *space* partitioning techniques that support Multicore+GPU platforms, and update timing analysis.

Goals

1. Timing Analysis for Componentized Systems with GPUs
2. Time Partitioning for Componentized Systems with GPUs
3. Space Partitioning for Componentized Systems with GPUs
4. Methods for Validating Component Timing Constraints
5. Evaluation with "Real" AI-Oriented Avionics Workloads

Selected Publications (Out of 18 Total)

[1] J. Bakita & J. Anderson, "Extending HW Compute Partitioning on NVIDIA GPUs to Composable Systems," in sub. to EMSOFT'24.
[2] S. Ali, Z. Tong, and J. Anderson, "Predictable GPU Sharing in Component-Based Real-Time Systems," in sub. to ECRTS'24.
[3] R. Wagle, S. Liu, and J. Anderson, "Autonomy Today: Many Delay-Prone Black Boxes," in sub. to ECRTS'24.
[4] J. Bakita and J. Anderson, "Demystifying NVIDIA GPU Internals to Enable Reliable GPU Management," in RTAS'24.
[5] R. Wagle, Z. Tong, R. Sites, and J. Anderson, "Want Predictable GPU Execution? Beware SMIs!" in ICPADS'23.
[6] J. Bakita and J. Anderson, "Hardware Compute Partitioning on NVIDIA GPUs," in RTAS'23.
[7] T. Amert, Z. Tong, S. Voronov, J. Bakita, F.D. Smith, and J. Anderson, "TimeWall: Enabling Time Partitioning for Real-Time Multicore+Accelerator Platforms," in RTSS'21.
[8] S. Voronov, S. Tang, T. Amert, and J. Anderson, "AI Meets Real-Time: Addressing Real-World Complexities in Graph Response-Time Analysis," in RTSS'21.

Goal 1: Timing Analysis for Componentized Systems with GPUs

Components of tasks receive periodic time slices on some number of CPUs (ex: Fig 1). These CPUs are only partially available to a component, so we develop transformations to adapt conventional timing analysis [8].

To enforce time-partitioning, we uncover and address difficulties on modern systems, including SMI [5] and p-threads [3] bugs.

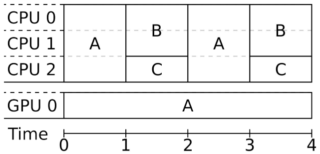

Fig. 1. Components A, B, and C on three CPUs and a GPU.

This allows each component to use an arbitrary scheduler, but does not address GPU use—past work only allows one component to use the GPU.

Goal 2: Enable GPUs in Componentized Systems: Time Partitioning

To allow multiple components to use the GPU, GPU tasks (kernels) must not overrun their time slice. By leveraging the predictable nature of GPU kernel execution times, we achieve this through preventing kernels that may overrun from launching [7]. This is shown at right, where we amend the system of Fig. 1 to allow Component B to also use the GPU.

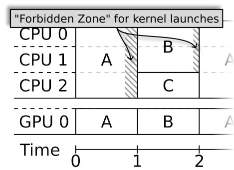

Fig. 2. Forbidden zones allow Components A and B to use the GPU at different times.

This allows multiple components to use the GPU, but not at the same time. We address this next.

Goal 3: Enable GPUs in Componentized Systems: Space Partitioning

To enable multiple components to use the GPU at the same time, we develop hardware-enforced partitioning of NVIDIA GPU engines [4] and compute cores [1, 6].

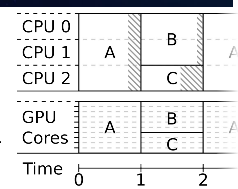

Fig. 3. Components B and C can concurrently use the GPU, once it is spatially partitioned.

This allows multiple components to concurrently use mutually-exclusive portions of the GPU; shown in Fig. 3.

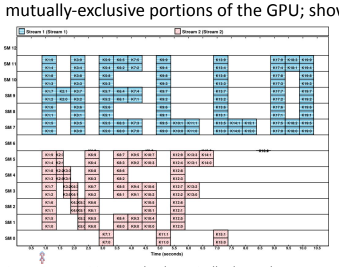

Fig. 4. GPU partitions can be dynamically changed, are hardware-enforced, and do not require task modification.

Our work [6] (an experimental example shown in Fig. 4) has led to collaboration with NVIDIA Research, and our other work [4] has uncovered flaws in three widely accepted GPU-management and -analysis techniques.

Goal 4: Methods for Validating Component Timing Constraints

The tasks inside a component may also need to share the GPU. We develop a per-SM locking protocol which allows for more than one task to simultaneously use the GPU at a time inside a component [2].

We have also completed three works which enable response-time bounding, budgeting, and support for new hardware for intra-component processing graphs (see online).

Goal 5: Evaluate with "Real" AI-Oriented Avionics Workloads

Our system which implements our work from Goal 1 and Goal 2, known as TimeWall, has been demonstrated to provide the necessary isolation between flight-critical and auxiliary tasks (each encapsulated in components) for a simulated quadcopter drone. We are presently porting this demonstrated to a physical drone with an embedded integrated NVIDIA GPU.

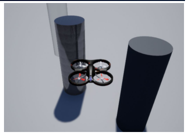

Fig. 5. Simulated quadcopter running on TimeWall

Please see our demo, which shows TimeWall at work.

Broader Impacts

The fruition of our work is a society filled with safer, longer-lasting, and more-efficient autonomous systems. Some already-realized broader impacts:

- Summer internships at Northrop Grumman, 2021, 2022, 2023.
- Bi-weekly meetings with DoD partners on evolving avionics certification.
- Outreach efforts to obtain industry perspectives from key players (e.g., Bosch, Apex.AI, NVIDIA).
- Two dissertations (Voronov & Amert).
- An undergraduate GPU research group.
- A reading group for undergraduate woman (more below).

Broadening Participation in Computing: TOPICS

Fig. 5: A recent TOPICS meeting.

- **TOPICS**: Talking Over Papers In Computer Science.
- A reading group for undergraduate woman in CS.
- We read papers out loud (you read that right) and discuss them.
- We have read papers on everything from quantum computing to AI to computer security to Turing Award lectures.
- We also talk about writing tips, applying to graduate school, and other things.
- We have only two rules:
  - No such thing as a dumb question.
  - No work outside of our one hour per week.
- A fun group with interesting discussions.

2024 NSF Cyber-Physical Systems Principal Investigators' Meeting
March 20-21, 2024

Award ID#: 2038855
Jan 2021 to Dec 2024 (est.)

NSF