Modular Power Orchestration at the Meso-scale

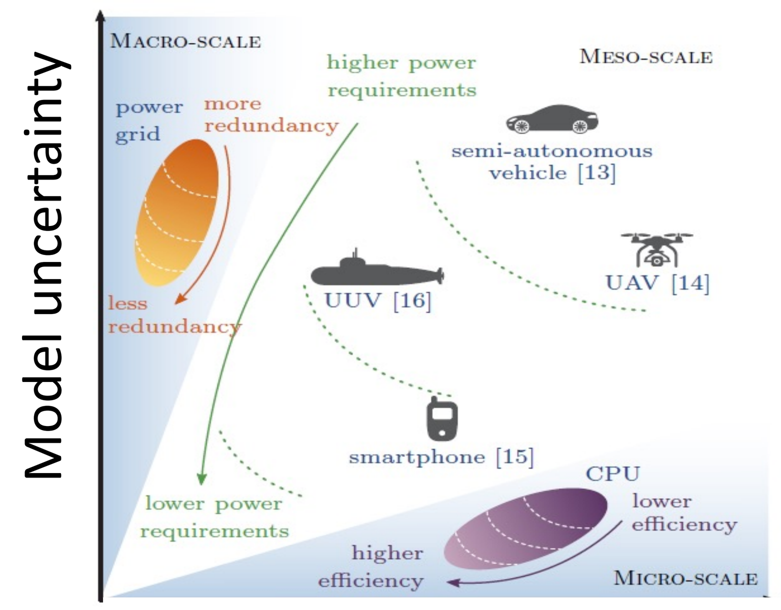
Washington University in St. Louis

PI: Xuan Zhang; Co-PI: Christpoher Gill {xuan.zhang, cdgill} @wustl.edu

https://xzgroup.wustl.edu/; https://github.com/xz-group

Motivation for Meso-scale Orchestration

- The generation, storage, allocation, and distribution of power, as well as computational resources among the modules demands flexibility
- The semantics of precise resource management and the complexity of high-level tasks are mismatched
- High model uncertainty exists due to diverse sources of variability, especially at run-time



Semantic mismatch

Research Approach

A bottom-up approach to developing resource (power/compute) resource management techniques for meso-scale systems across the system layers:

- T1: Efficient circuit-and-architecture-level power delivery
- T2: Architecture-and-operating-system-level real-time resource scheduling
- T3: Principled investigation and evaluation of mission-level CPS performance

(T1 tasks have been presented in prior years; we focus on T2 and T3 tasks here)

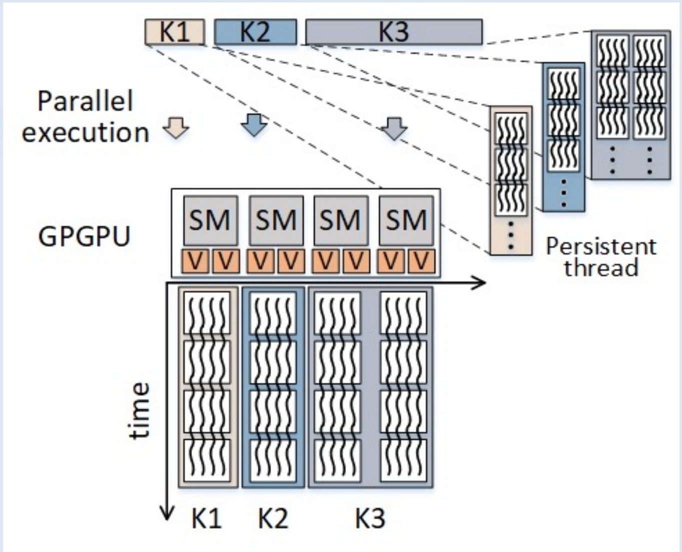
Cross-layer System Resource Management

Real-time GPU scheduling

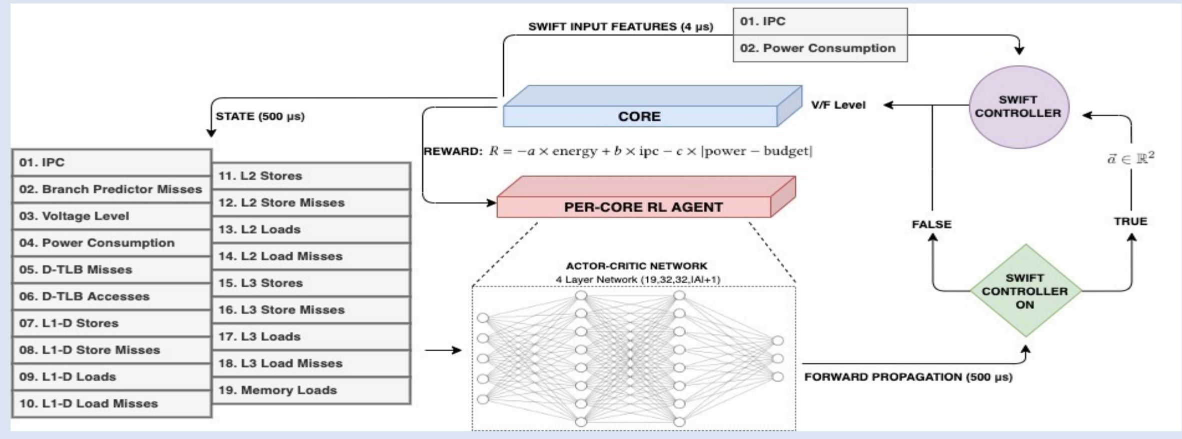
- System-level resource partitioning improves utilization
- OS-level real-time scheduling guarantees hard deadlines

Learning-based energy management

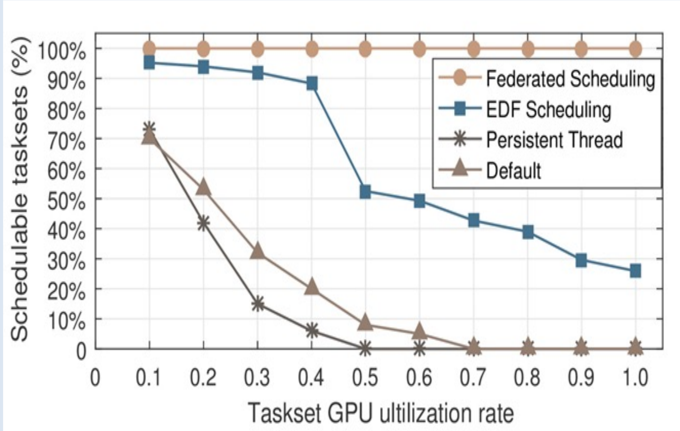
- Fast microsecond timescale power management (DVFS)
- Learning-based hierarchical controller structure



Resource partitioning



Reinforcement learning model as local controller

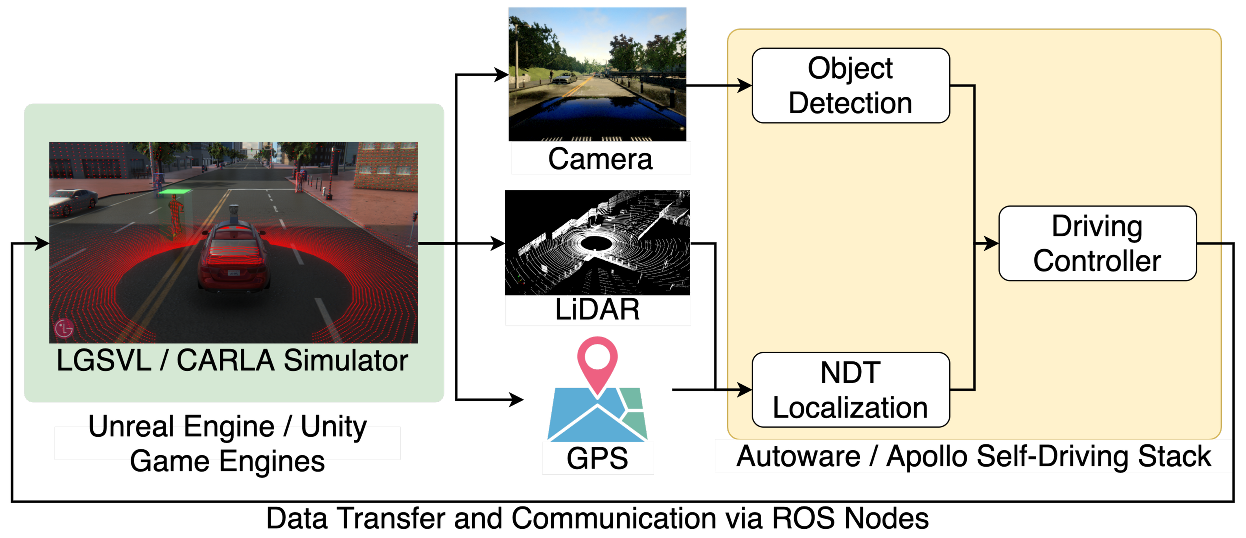


GPU schedulability

Full-stack Self-driving Evaluation Framework

- Autonomous vehicles (AVs) infrastructure leverage high-fidelity simulators like CARLA / LGSVL running on Unreal Engine / Unity.
- Full self-driving stacks such as Autoware / Apollo use the sensor data from the simulator to drive the vehicle
- ROS nodes are used for relaying sensor data between the simulators and the controllers

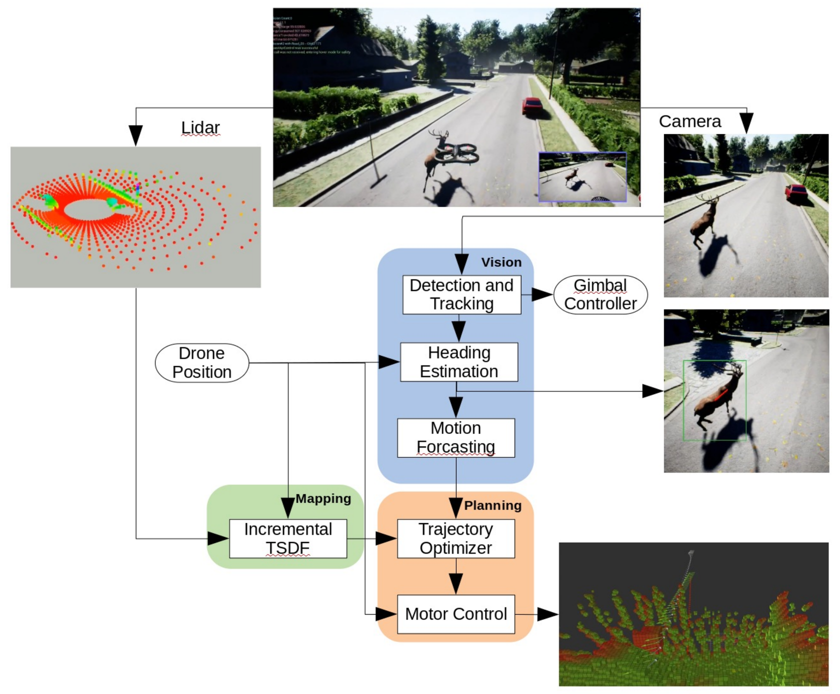
Robust self-driving against adversarial settings



Self-Driving Infrastructure

Mission-level Real-time Benchmarking

- AirSim APIs interface with simulate hardware and provide ground-truth data about environment such as collisions, actor position, etc.
- GPU intensive workloads (e.g. machine perception)
- ROS2 serves as real-time platform
- Cinematography mission is used as an application example to evaluate system end-to-end performance



Related Publications

- Boloor, A., et. al. (2020). “Attacking vision-based perception in end-to-end autonomous driving models”. *Journal of Systems Architecture*, *110*, 101766.
- Boloor, A., et. al. (2021). “Optical Trojans: Assisting Adversarial Perturbations with Coded Defocus”. (Pending review).
- Yang, J., et. al. (2020). “Finding Physical Adversarial Examples for Autonomous Driving with Fast and Differentiable Image Compositing”. *arXiv preprint arXiv:2010.08844*.
- Zou, A., et. al. (2021). “RTGPU: Real-Time GPU Scheduling of Hard Deadline Parallel Tasks with Fine-Grain Utilization”. *arXiv preprint arXiv:2101.10463*.
- Zou, A., et. al. (2020,). “F-LEMMA: Fast Learning-based Energy Management for Multi-/Many-core Processors”. In *Proceedings of the 2020 ACM/IEEE Workshop on Machine Learning for CAD* (pp. 43-48). (Best Paper Nomination)
- Zou, A., et. al. (2021). “System-Level Early-Stage Modeling and Evaluation of IVR-assisted Processor Power Delivery System”. (Under review).

2021 NSF Cyber-Physical Systems Principal Investigators' Meeting
June 2-4, 2021

Award ID#: NSF CNS #1739643