

Justin Davis

Golden, CO | jcdavis@mines.edu | 207-475-7880 | justindavis.github.io
linkedin.com/in/justinconnordavis | github.com/justindavis

Education

- Colorado School of Mines**, Doctorate of Philosophy in Computer Science Aug 2022 - Ongoing
- GPA: 3.9/4.0
 - Relevant Coursework: Theory of Computation, Advanced Computer Architecture, High Performance Computing, Distributed Systems, Heterogeneous Computing
- Colorado School of Mines**, Bachelor of Science in Computer Science Aug 2019 - May 2022
- GPA: 3.9/4.0
 - Completed a specialization in robotics and intelligent systems
- University of Maine Presque Isle**, Associate of Arts in Liberal Sciences Aug 2015 - May 2019
- GPA: 3.6/4.0
 - Completed concurrently while enrolled in high school at the Maine School of Science and Mathematics

Experience

- Graduate Research Assistant**, Colorado School of Mines – Golden, CO Aug 2022 - Ongoing
- Perform literature surveys for new and existing proposals relating to multi-accelerator workloads, high-performance computing, robot planning, DNN inference, and embedded systems.
 - Collaborate with professors and students on research papers and projects relating to joint compute and motion robot planning, efficient DNN execution, accelerator-aware DNN pruning for object detection, and concurrent DNN execution.
 - Maintain lab infrastructure, including server-grade systems, workstations, and embedded systems such as NVIDIA Jetson, Raspberry Pi, Xilinx Kria, Google Coral, and Luxonis OAK.
 - Train incoming undergraduate and graduate students on embedded systems and the associated hardware/software platforms.
 - Design and quote server-grade and workstation systems based on budget, user requirements, and long-term maintainability.
 - Design experiments focusing on system energy and power usage, latency, and throughput for applications such as computer vision, DNN inference, networking, and multi-accelerator usage.
- Heterogeneous Computing TA**, Colorado School of Mines – Golden, CO Aug 2023 - May 2024
- Guest lectured on non-von Neumann computer architectures and devices, such as VLIW, systolic arrays, GPUs, reconfigurable hardware (e.g., FPGAs, TPUs/DLAs), and ASICs.
 - Guest lectured on programming models for GPU and FPGA devices, covering CUDA, TensorRT, OpenCL, and SYCL.
 - Designed and graded assignments focused on the fundamentals of parallel computation using CUDA and GPUs.
 - Evaluated and assisted students with open-ended projects focusing on performance and efficiency comparisons between CPUs and GPUs and a processing unit of their choice.
- Operating Systems TA**, Colorado School of Mines – Golden, CO Jan 2023 - Dec 2023
- Guest lectured on memory hierarchy in modern systems and interrupts in operating systems.
 - Designed homework assignments covering virtual memory, Amdahl's Law, CPU scheduling, mutexes, and deadlock.
 - Tutored students and led review sessions on CPU scheduling, threading, Amdahl's Law, synchronization primitives, deadlock, memory management, and virtual memory.
 - Designed exam questions covering Amdahl's Law, threading, and CPU scheduling.

- Conducted lab sessions where students implemented techniques such as feature detection (SIFT, SURF, ORB), edge detection, morphology, camera coordinate frame conversions, and template matching.
- Evaluated and assisted students with open-ended, semester-long computer vision projects covering a wide range of topics, from optical character recognition to stereo visual odometry.

Publications

Context-aware Multi-Model Object Detection for Diversely Heterogeneous Compute Systems

DATE - March 2024

Justin Davis, Mehmet Belviranli

DOI: 10.23919/DATE58400.2024.10546645

Abstract: In recent years, deep neural networks (DNNs) have gained widespread adoption for continuous mobile object detection (OD) tasks, particularly in autonomous systems. However, a prevalent issue in their deployment is the one-size-fits-all approach, where a single DNN is used, resulting in inefficient utilization of computational resources. This inefficiency is particularly detrimental in energy-constrained systems, as it degrades overall system efficiency. We identify that, the contextual information embedded in the input data stream (e.g. the frames in the camera feed that the OD models are run on) could be exploited to allow a more efficient multi-model-based OD process. In this paper, we propose *SHIFT* which continuously selects from a variety of DNN-based OD models depending on the dynamically changing contextual information and computational constraints. During this selection, *SHIFT* uniquely considers multi-accelerator execution to better optimize the energy-efficiency while satisfying the latency constraints. Our proposed methodology results in improvements of up to 7.5x in energy usage and 2.8x in latency compared to state-of-the-art GPU-based single model OD approaches.

Awards: Outstanding Paper in the Autonomous System Design Initiative at DATE 2024, 3rd Overall in Computation and Theory in GRADS@MINES 2024

Ongoing Research

Priority-based Fast Multi-Object Tracking on Multi-Accelerator Systems

Ongoing

Abstract: Multi-object tracking (MOT) is a critical task in computer vision, essential for applications such as autonomous vehicles, surveillance, and robotics. It involves detecting and tracking multiple objects over time to understand dynamic environments and make informed decisions. State-of-the-art MOT systems rely on deep neural networks (DNN) for object detection, which can be computationally intensive and energy-consuming, especially in real-time scenarios on performance-limited and energy-efficient computing environments. This work aims to address the challenge of improving MOT execution efficiency on heterogeneous system-on-chips (SoCs) that integrate multiple accelerators, including CPUs, GPUs, and domain-specific accelerators like deep learning accelerators (DLA) and programmable vision accelerators (PVA). Our method leverages a novel multi-model, multi-accelerator execution strategy to enhance latency and energy consumption without compromising critical operational accuracy. By identifying the locations of high-priority (HP) and low-priority (LP) objects in the frame, our work runs the original, GPU-based detection model on reduced frames for HP objects and employs faster, lower-accuracy models for LP objects on the DLA. This approach significantly reduces the size of input data and optimizes the use of energy-efficient accelerators. Evaluations on the MOT17 dataset demonstrate up to 3.0x reduction in latency, 6.5x in energy, and 2.0x in power draw, showcasing the effectiveness of our method in real-world scenarios.

Planning for Shared Resources in Robot Motion and Heterogeneous Computational Scheduling

Ongoing

Abstract: Robots face a variety of resource limits, such as energy, power, time, and ability to dissipate heat. Physical motion requires such resources. Moreover, computation is a physical process that requires time and energy while producing heat. Effective operation in resource-constrained scenarios requires robots to consider the joint resource use of motion and computation. Our work integrates sampling-based motion planning and constraint-based computational scheduling to find robot plans for both motion and computation to satisfy resource constraints. Theoretically, we prove that our approach retains the probabilistic completeness of typical sampling-based methods. Empirically, we show that our method can find solutions with > 10 times frequency compared to the baseline in environments which are resource-constrained due to heat, power, and time.

Resource Management for Diversely Scaled Edge-Cloud Systems

Ongoing

Abstract: Computing systems are evolving to be more pervasive, heterogeneous, and dynamic. Many emerging domains, such as Internet of Things (IoT), federated learning, and smart buildings, rely on a diverse edge to cloud continuum where the execution of applications spans various tiers of systems with significantly different computational capabilities. Computing resources in each tier, such as processing units inside of in-the-field edge devices and high-performance servers in datacenters, are handled in isolation due to scalability and resource segregation. This practice results in task mappings limited to only a subset of all available processing units, preventing an efficient overall utilization of the system. In this work, we propose a *holistic* approach to capture diverse computational characteristics of *edge-cloud* systems with arbitrary topologies and to efficiently manage the computational resources with the whole continuum in the scope. Our approach is built upon a *multi-layer* graph-based hardware (HW) representation and a *modular* performance modeling interface that can capture interactions and *interference* between computational resources in the system. We introduce an *orchestrator* mechanism that leverages the graph-based HW representation to hierarchically locate processing units to which a given set of tasks can be mapped while respecting the isolation between the computational tiers of an edge-cloud system.

Awards

- Outstanding Paper in the Autonomous System Design Initiative at DATE 2024
- 3rd Overall in Computation and Theory in GRADS@MINES 2024
- Best Robot Under 1.5kg at the Colorado Spacegrant Challenge 2022

Projects

trtutils: General Purpose TensorRT Engines

github.com/justincdavis/trtutils

Tools Used: Python, TensorRT, CUDA, NVRTC, NumPy

Platforms: NVIDIA GTX/RTX and Jetson GPUs/DLAs

- Developed a general-purpose wrapper around TensorRT-compiled engines in Python, enabling faster development time when using TensorRT.
- Integrated NVIDIA Jetson APIs to measure power draw and overall energy usage during each inference of a compiled TensorRT engine.
- Integrated CUDA kernels compiled with NVRTC to achieve an order-of-magnitude speedup over previous CPU-based preprocessing methods in Python.

oakutils: Easy Pipelines and Custom Code on OAK RCV2

github.com/justincdavis/oakutils

Tools Used: Python, DepthAI, NumPy

Platforms: Luxonis OAK on Windows/Linux

- Developed easy-to-use high-level functions for creating pipelines onboard Luxonis OAK devices without requiring the use of the low-level API.
- Created abstractions that allow the device to be managed in an independent thread, with functionality occurring during callbacks.
- Integrated modules for handling calibration information, point clouds, and depth data, and for utilizing the onboard VPU embedded in the RCV2 SoC as a standalone DNN accelerator similar to NVIDIA DLA.
- Developed functionality for compiling PyTorch models to RCV2, enabling custom code execution in pipelines or as a standalone co-processor.

cv2ext: High-Level JIT Accelerated Computer Vision Functions

github.com/justincdavis/cv2ext

Tools Used: Python, OpenCV, NumPy, Numba

Platforms: OpenCV & Numba Compatible Devices

- Developed abstractions over OpenCV's I/O functionality, implementing them in separate threads to reduce latency and boilerplate code.
- Integrated the Numba JIT compiler for Python into common computer vision and bounding box operations, accelerating runtime without adding JIT complexity or management to the end-product code.

Technologies

Languages: Python, C++, C, CUDA, Lisp, Java

Tools: Docker, Git, Linux, Nsight Ecosystem, Scalene Profiler, Tracy Profiler, PDB, GCC, NVCC

Libraries: PyTorch, TensorRT, ONNX, OpenCV, VPI, Numba, NumPy, DepthAI