CSR: Small: Predictable Real-Time Computing in GPU-enabled Systems

PI: Cong Liu, Assistant Professor
Department of Computer Science, University of Texas at Dallas

Abstract

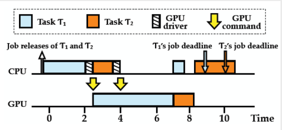
Given the need to achieve higher performance without driving up energy consumption, most chip manufacturers have shifted to multicore architectures, especially heterogeneous ones. Among heterogeneous processing elements, graphic processing units (GPUs) have seen wide-spread use. GPUs have the power to enable orders of magnitude faster execution of many applications. Thus, they are becoming increasingly applicable for general-purpose systems.

Unfortunately, it is not straightforward to reliably adopt GPUs in many safety-critical systems that require predictable real-time correctness, one of the most important tenets in certification required for such systems. An example is the advanced automotive system, in which timeliness of computations is an essential requirement of correctness due to the interaction with the physical world. It is challenging to ensure predictable real-time correctness in current GPU-enabled systems, preventing such systems from being legally certifiable and thus causing safety to be a major concern. The goal of the proposed research is to achieve predictable real-time computing in GPU-enabled systems by (i) establishing GPU-aware resource allocation methods that yield quantifiable guarantees on real-time correctness, and (ii) implementing an ecosystem of GPU resource management that enables GPUs to be predictably utilized in a real-time multi-tasking environment.

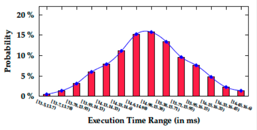
Challenges and Research Plan

Challenges

- Algorithmically, adding another type of computing resource brings new challenges on real-time resource allocation because the decisions on allocating CPU and GPU resources need to be judiciously coordinated



- Stochastic execution times on GPU



- Current system software support for GPUs is not well-designed to enable efficient utilization of GPUs in a real-time multi-tasking environment

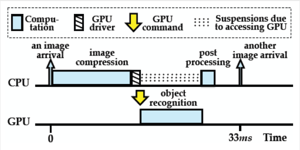
Research plans

- Develop GPU-aware scheduling algorithms and schedulability tests
- Enhance OS and GPU driver support for predictable real-time computing using GPUs
- Carry out overhead-aware schedulability studies and conduct case-studies

Methods

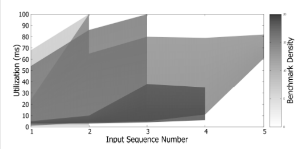
Suspension-based methods

- Treating the response time of any task's segment running on GPUs as suspension delays.
- Thus being able to reduce the hard CPU-GPU co-scheduling problem to a single resource (CPU) scheduling problem.
- Develop a set of suspension-aware real-time scheduling algorithms and schedulability tests



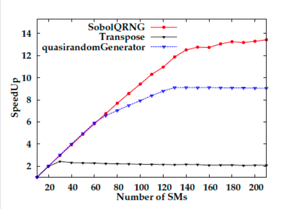
Server-based approaches to deal with variable execution times on GPUs

- Creating a server task with a fixed budget to execute each GPU-accelerated segment
- Task overruns on GPU can be handled by reclaiming the remaining capacity of other server tasks or creating a designated server task to specifically handle task overruns
- Stochastic resource allocation leveraging queueing theory

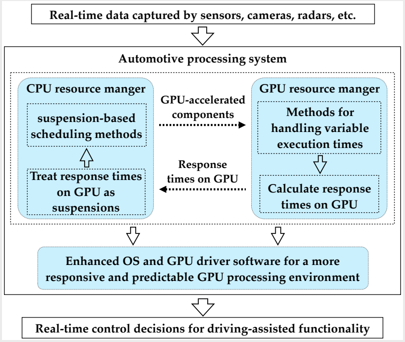


System software development for predictable GPGPU computing

- Fine-grained and predictable GPU computing core control for resolving the common GPU resource under-utilization issue
- Optimizing data transfer for improved latency



Approach overview



Real-time data captured by sensors, cameras, radars, etc.

Automotive processing system

CPU resource manger

suspension-based scheduling methods

Treat response times on GPU as suspensions

GPU-accelerated components

Response times on GPU

GPU resource manger

Methods for handling variable execution times

Calculate response times on GPU

Enhanced OS and GPU driver software for a more responsive and predictable GPU processing environment

Real-time control decisions for driving-assisted functionality

Implementation and Evaluation

- Implemented on top of LITMUSRT (the OS platform) plus GPES (our developed driver/runtime system for managing GPU resources)
- Overhead-aware schedulability studies
- Conduct case studies using real-world automotive workloads, e.g., advanced driving-assisted tools with DNN-based object recognition

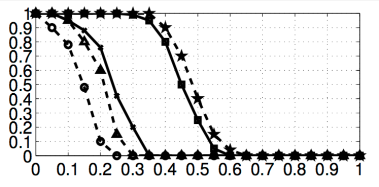
Broader Impact

Broader impact

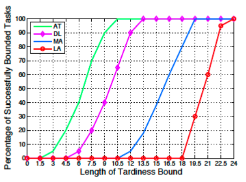
- The outcome of this project will pave the way to enabling safety-critical embedded systems equipped with GPUs to be certifiable
- At least two Ph.D. dissertations will be produced
- All software and systems created will be made publicly available
- Reporting our results in the leading scholarly venues
- Recruiting undergraduate and graduate students from underrepresented populations
- Participating in the Texas Alliance for Minorities in Engineering (TAME) program
- Giving tutorials on GPGPU programming to local high school teachers and students (particularly those participating in programming contests)
- Participating in numerous outreach programs organized by the UT-Dallas Computer Science department: in 2016, we had about 2100 participants, including 442 K-12 students and 653 college students (21% were women and more than 18% were minorities)

Results

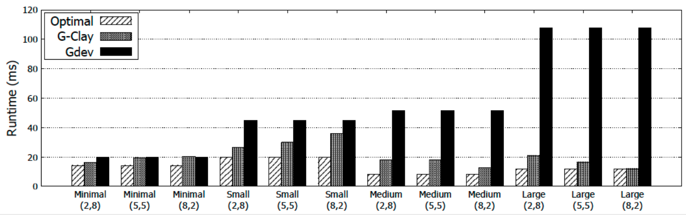
- A new set of suspension-aware real-time schedulers and tests



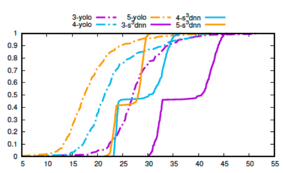
- GPU-oriented stochastic resource allocation strategies

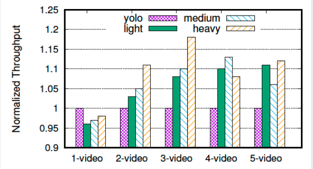


- OS-level GPU resource management ecosystems that improve the responsiveness and predictability of GPGPU computing



- Applying the overall approach to DNN-based object recognition workloads in an autonomous driving scenario





Publications

1. [RTSS'16] Zheng Dong and Cong Liu. Closing the loop for the selective conversion approach: a utilization-based test for hard real-time suspending task systems,
2. [RTSS'16] Zheng Dong, Yu Gu, Jiming Chen, Shaojie Tang, Tian He, and Cong Liu. Enabling predictable wireless data collection in severe energy harvesting environments,
3. [RTSS'16] Jianjia Chen, Wenhung Huang, and Cong Liu. k2Q: A quafratic-form response time and schedulability analysis framework for utilization-based analysis,
4. [INFOCOM'16] Guangmo Tong, Lei Cui, Weili Wu, Cong Liu, and Ding-Zhu Du.Terminal-Set-Enhanced Community Detection in Social Networks,
5. [HotCloud'15] Husheng Zhou, Yangchun Fu, and Cong Liu. Supporting Dynamic GPU Computing Result Reuse in the Cloud,
6. [RTSS'15] Jianjia Chen, Wenhung Huang, and Cong Liu. K2U: A General Framework from k-Point Effective Schedulabiliy Analysis to Utilization-based Tests,
7. [RTAS'15] Husheng Zhou, Guangmo Tong, and Cong Liu. GPES: A Preemptive Execution System for GPGPU Computing,