Programming and optimizing for heterogeneous CPU-GPU architectures

Course description

The imminent future of parallel architectures is a tighter integration of different types of processing cores, namely CPUs and GPUs. HSA architecture or NVIDIA Pascal are initial steps towards this direction. They show that the trend in the upcoming years will be CPU and GPU threads working concurrently on the same data.

Brand new features such as unified virtual address spaces and memory coherence make it possible, enabling seamless sharing of data structures. System-wide atomic operations are also available to perform fine-grained synchronization.

CPU and GPU will be able to work readily and flexibly on those parts of the applications that fit best their processing power. Higher performance and much higher energy efficiency will be achieved, ensuring longer life of the batteries of mobile devices and a more constrained power budged for HPC systems.

In terms of programmability, these features soften the learning curve for heterogeneous programming. Programmer productivity increases by eliminating the need for explicit memory management.

In this course, you will be introduced to the current heterogeneous architectures, and how to program them using mainstream languages, such as CUDA and OpenCL, and higher level languages as C++AMP.

Contents

- Heterogeneous architectures
  - Traditional heterogeneous systems
  - Integrated platforms
  - CPU-GPU integration: memory coherence, atomic operations
  - State of the art: HSA architecture, Intel processors, NVIDIA Pascal
- Programming models
  - CUDA 8.0
  - OpenCL 2.0
  - C++AMP
- Patterns of heterogeneous programs
  - Workload partitioning
  - Producer-consumer
  - Switching
- Case studies
  - Data manipulation, graph algorithms, image and video processing

Duration

10 hours, including hands-on labs.

Target audience

Graduate students with a background on C programming (basic knowledge of CUDA or OpenCL is desired). Students are required to bring their own laptops.

Agenda

Day 1 (9:30-11:30)
- Introduction to heterogeneous computing
- Review of traditional heterogeneous systems: the accelerator model
- Hands-on lab 1: Traditional host-device interaction

Day 2 (9:30-11:30)
- Integrated heterogeneous architectures: NVIDIA Jetson, NVIDIA Pascal, HSA, Intel IGP, FPGAs
- Latest features of heterogeneous architectures: unified memory, memory coherence, system-wide atomic operations
- CUDA 8.0, OpenCL 2.0, C++AMP
- Hands-on lab 2: Unified memory and system-wide atomics

Day 3 (9:30-11:30)
- Heterogeneous patterns: data partitioning
- Hands-on lab 3: Compute-bound (Bezier surface) and memory-bound (Data manipulation) computation

Day 4 (9:30-11:30)
- Heterogeneous patterns: task partitioning
- Hands-on lab 4: Coarse-grained (Breadth-First Search) and fine-grained (RANSAC) partitioning

Day 5 (9:30-11:30)
- Concluding remarks
- A glimpse into the future

About the lecturer

Juan Gómez-Luna received the B.S. and M.S. degrees in Telecommunication Engineering from the University of Sevilla, Spain, in 2001, and the Ph.D. degree in Computer Science from the University of Córdoba, Spain, in 2012. Since 2005, he has been a lecturer at the University of Córdoba. His research interests focus on the parallelization and optimization of applications on GPUs and heterogeneous systems. He collaborates with research groups in the University of Illinois at Urbana-Champaign, the Barcelona Supercomputing Center, and the Norwegian University of Science and Technology. He is a coauthor of the book "Heterogeneous System Architecture: A new compute platform infrastructure".

Contact

If you are interested, please, send an email to el1goluj@uco.es.