GPU Computing Development with GPU Ocelot

Full-day tutorial previously held at PACT’11, HPCA’12, and Microsoft Research. The proposed tutorial is derived from previous half-day versions augmented with recent research results, developments, and live demonstrations of presented concepts. Expect 20-50 attendees.

- Andrew Kerr <arkerr@gatech.edu>, Georgia Institute of Technology
- Gregory Diamos <gregory.diamos@gatech.edu>, Georgia Institute of Technology
- Rodrigo Dominguez <rdomingu@ece.neu.edu>, Northeastern University
- Sudhakar Yalamanchili <suhda@gatech.edu>, Georgia Institute of Technology

Abstract

GPU Ocelot is an open-source dynamic JIT compilation framework for GPU compute applications targeting a range of GPU and non-GPU execution targets. Ocelot supports CUDA applications and provides an implementation of the CUDA Runtime API enabling seamless integration with existing CUDA applications. Its JIT compiler supports four backend execution targets - (1) an emulator that implements NVIDIA’s Parallel Thread Execution (PTX) instruction set architecture, (2) NVIDIA, (3) AMD GPUs, and (4) a translator to LLVM for efficient execution of GPU kernels on multicore CPUs. Existing CUDA applications are seamlessly supported.

Ocelot facilitates research and development on several fronts. First, Ocelot improves developer productivity of GPU compute applications by providing an infrastructure for building event trace analyzers using the emulator. This tutorial will describe how analyzers can be built and cover several existing analyzers for (1) correctness checks, (2) debugging and (3) performance tuning. Second, as a JIT compiler infrastructure, Ocelot provides facilities for compiler research including interfaces to an internal representation of PTX programs in support of optimization passes for massively data parallel computer kernels. Third, with an open source re-implementation of the CUDA runtime, Ocelot enables research into scheduling, resource allocation, and operating systems. Finally, Ocelot enables research in heterogeneous architectures via trace generation interfaces, PTX emulation and support for detailed workload characterization on GPU and CPU devices.

Target Audience

GPU Ocelot is intended to facilitate research into three primary areas: GPU computing, compiler research, heterogeneous architectures. Each of these communities are among the primary audience of the PACT conference, and we hope to provide them with an understanding of each of the capabilities of Ocelot as well as how this infrastructure may enable their own research.

Outline

1. Introduction
   (a) Project Goals
   (b) Ocelot Structural Overview
   (c) Enabled Research

2. Ocelot: Architecture
   (a) CUDA Runtime API implementation
   (b) Ocelot Device Abstraction
      i. Virtual method interface
      ii. Memory allocations
   (c) Parallel Thread Execution (PTX) IR
      i. Introduction to ISA and internal representation
      ii. Writing PTX transformations and Pass Manager
      iii. Details of the parser and emitter

3. Ocelot: Supported Devices
   (a) PTX Emulator
      i. Implementation overview
      ii. Trace generation interface
      iii. Events, machine state, and emulation
   (b) Multicore Host Backend
i. Translating kernels for Execution on the Host
ii. Translator interface
iii. PTX to LLVM Translation
  (c) NVIDIA GPU Backend
  (d) AMD Radeon Backend
    i. Introduction to Caracal
    ii. Language features supported
    iii. Structural Analysis (Control Tree)
    iv. Unstructured Control Flow
    v. Performance optimization research

4. GPU Computing Productivity Tools
   (a) Correctness Checks
   (b) Performance debugging
   (c) PTX Instrumentation
     i. Defining and compiling PTX-to-PTX procedural transformations
     ii. Registering instrumentation tools with Ocelot
     iii. Obtaining and analyzing instrumentation results
   (d) Device switching

5. Compiler Research and Execution Model Translation
   (a) Unstructured to Structured control flow
   (b) Subkernel partitioning, lazy compilation, and scheduling
   (c) Vectorization, VLIW instruction scheduling
   (d) Glimpses, Datalog to PTX, Heterogeneous Virtual Machines

6. Architecture Research
   (a) Reconvergence mechanisms
   (b) Interfaces to MACSIM (cycle-accurate GPU simulator)
   (c) Fast timing models
   (d) Performance prediction, workload characterization

Presenter Biographies

Andrew Kerr is a PhD student with the Computer Architecture and Systems Lab at Georgia Tech. His research interests focus on the interaction between architectures, compilation tools, programming languages, and software design. Andrew is currently pursuing a PhD focusing on the challenges of porting applications to heterogeneous collections of processors with novel architectures and execution models.

Gregory Diamos is a research scientist at NVIDIA Research and recent PhD graduate of Georgia Tech. His current research interests follow the industry shift from ILP to many core architectures, where the ability to tightly integrate heterogeneous architectures offers the potential for dramatic improvements in efficiency at the cost of increased design complexity; his research is directed toward maintaining this efficiency while reducing design complexity.

Rodrigo Dominguez is a PhD student at Northeastern University working under the supervision of Dr. David Kaeli. His research interests are in the area of many-core heterogeneous processors and compilers, especially binary translation systems and GPUs. His research addresses the problem of code portability among different heterogeneous architectures. He received his MS degree from the Georgia Institute of Technology. Before joining Northeastern, he worked as a Software Engineer.

Sudhakar Yalamanchili earned his Ph.D degree in Electrical and Computer Engineering in 1984 from the University of Texas at Austin. He was at Honeywells Systems and Research Center in Minneapolis before joining the ECE faculty at Georgia Tech in 1989 where he is now a Joseph M. Pettit Professor of Computer Engineering. His current research focuses on addressing the software challenges of heterogeneous architectures and solutions to power and thermal issues in many core architectures and data centers. Since 2003 he has been a Co-Director of the NSF Industry University Cooperative Research Center on Experimental Computer Systems at Georgia Tech.