General Purpose GPU Programming

Advanced Operating Systems
Tutorial 7
Tutorial Outline

• Review of lectured material
• Key points
• Discussion
  • OpenCL
  • Future directions
Review of Lectured Material

- Heterogeneous instruction set systems
- Heterogeneous multi-kernel systems – Helios
- Main core with heterogeneous offload
  - Graphics offload hardware – GPGPU
  - Programming model
  - OpenCL
  - Integration with operating systems
- Heterogenous virtual machines – Hera JVM
- Hybrid models – Accelerator
  - Lazy encoding of SIMD-style operations and JIT compilation into type system
Key Points

- Increasing heterogeneity of hardware
- Programming models are complex
  - Too limited to run a full operating system
  - Too different to effectively run standard programming languages
- OpenCL-style offload model performs well, but is complex to program
- Attempts to hide complexity in VM have had mixed success
Discussion

- What is complexity versus performance trade-off in OpenCL – how does this limit usefulness?
- How can SIMD-style processing be more cleanly incorporated into modern languages?
- Is the embedded DSL approach of Accelerator a set in the right direction, or is the complexity of the VM excessive?
- How to use heterogenous processing resources?