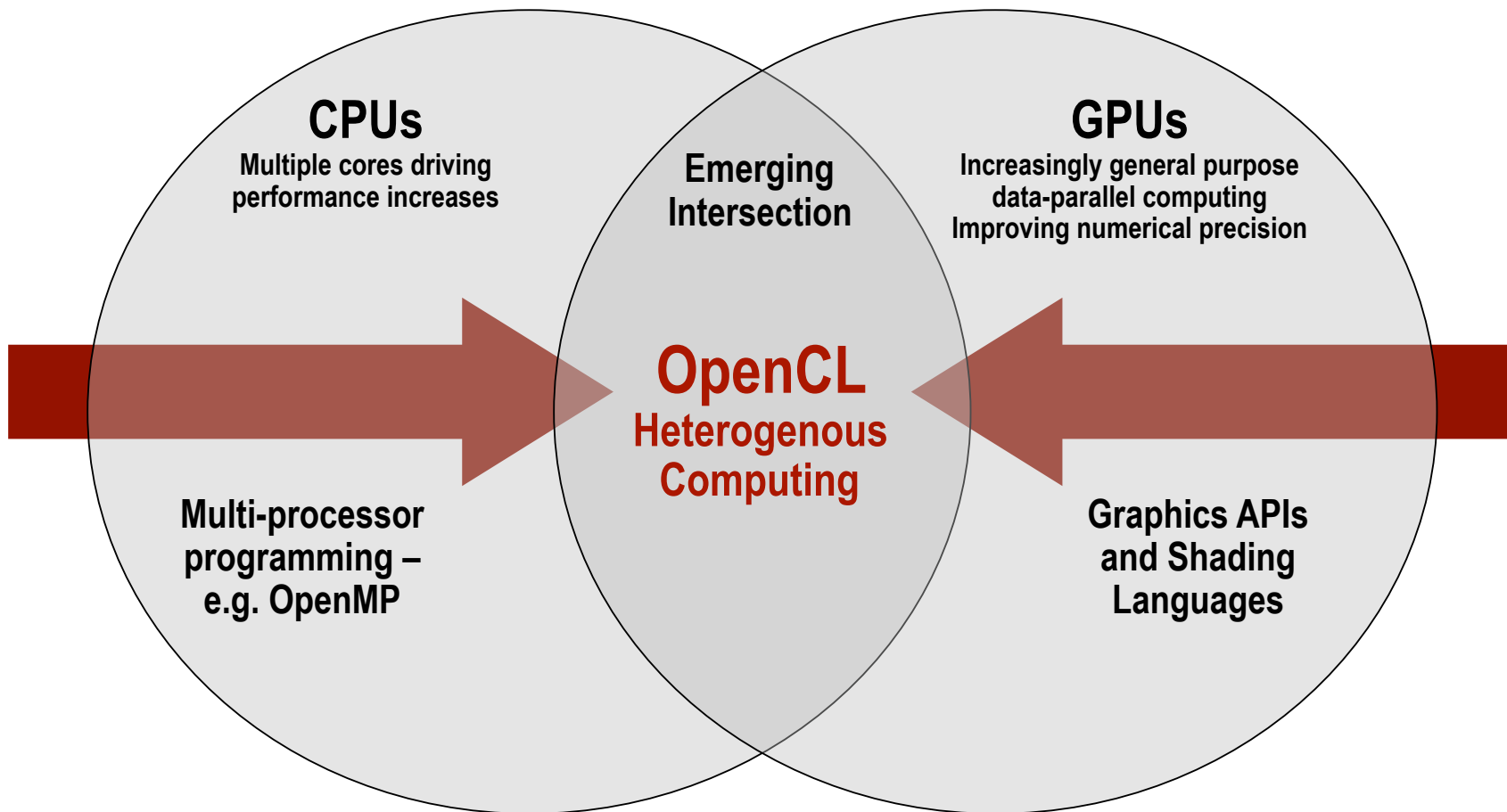




**OpenCL
Heterogeneous
Parallel Programming**

Neil Trevett, OpenCL Chair

Processor Parallelism



OpenCL – Open Computing Language
Open, royalty-free standard for programming heterogeneous parallel computing at the intersection of GPU and multi-core CPU capabilities

OpenCL Objectives

- **Use all computational resources in system**
 - Support both data- and task- parallel compute models
 - Program GPUs and CPUs as peers
- **High-performance, low-level abstraction at software/silicon boundary**
 - Device abstraction is low as possible while still providing cross-platform portability
 - Ecosystem foundation - will support domain-specific middleware and languages
- **Familiar to programmers**
 - Efficient C-based parallel programming model
 - Approachable – but primarily targeted at expert developers - no “convenience” functions
- **Implementable on a range of desktop, embedded and mobile devices**
 - Desktop and handheld profiles in one specification
- **Intimate interoperability with graphics APIs**
 - Including OpenGL and OpenGL ES
- **Drive future hardware requirements**
 - Applicable to both consumer and HPC applications

OpenCL Process

- **Draft specification under review and refinement today**
 - Standard Khronos working group process
 - Fast track schedule
- **Based on an initial proposal by Apple**
 - Performance-enhancing technology in Mac OS X Snow Leopard
 - Proposal developed in collaboration with industry leaders
- **Will be open, vendor-neutral royalty-free standard**
 - Developed under standard Khronos IP framework



What is OpenCL?

- **Language Specification**

- C-based cross-platform programming interface
- Subset of ISO C99 with language extensions - familiar to developers
- Well-defined numerical accuracy - IEEE 754 rounding behavior with maximum errors
- Online or offline compilation and build of compute kernel executables
- Includes a rich set of built-in functions

- **Platform Layer API**

- A hardware abstraction layer over diverse computational resources
- Query, select and initialize compute devices
- Create compute contexts and work-queues

- **Runtime API**

- Execute compute kernels
- Resource management

Using OpenCL with OpenGL

- **Both standards under one IP framework**
 - Enables very close collaborative design
- **Efficient, inter-API communication**
 - While still allowing both APIs to handle the types of workloads for which they were designed
- **OpenCL can efficiently share resources with OpenGL**
 - Textures, Buffer Objects and Renderbuffers
 - Data is shared, not copied
- **Applications can select compute device(s) to run OpenGL and OpenCL**
 - Efficient queuing of OpenCL and OpenGL commands into the hardware
 - Flexible scheduling and synchronization
- **Examples**
 - Vertex and image data generated with OpenCL and then rendered with OpenGL
 - Images rendered with OpenGL and post-processed with OpenCL kernels
 - etc. etc.

Summary

- **Open, royalty-free standard for heterogeneous parallel computing**
 - At the intersection of GPU and multi-core CPU capabilities
- **A commercially significant initiative**
 - Large industry impact and potential
- **Fast track**
 - Strong commercial motivation to get it done quickly
- **Will be a key influencer on the evolution of OpenGL**
 - Being developed under the same IP framework
- **If this is relevant to your company – please join Khronos and get involved!**
- **More technical details..**
 - Beyond Programmable Shading: Fundamentals
 - SIGGRAPH Core | Thursday, 14 August | 8:30 am - 12:15 pm | 403 AB
 - Affie Munshi, Apple – OpenCL Specification Editor