Exploring the features of OpenCL 2.0

Saoni Mukherjee, Xiang Gong, Leiming Yu, Carter McC Cardwell, Yash Ukidave, Tuan Dao, Fanny Paravecino, David Kaeli

Northeastern University
Outline

• Introduction and evolution of OpenCL

• OpenCL 2.0 - new features

• Applications used to explore these features

• Result and analysis
OpenCL

- Programming and runtime framework
- Executes applications across heterogeneous platforms
- First version, OpenCL 1.0 was released in 2009

**OpenCL 1.0:** Basic programming model

**OpenCL 1.1/1.2:** Memory management & fine grain control

**OpenCL 2.0:** Support for emerging hardware capabilities & improved programmability
OpenCL 2.0 Features

- Shared Virtual Memory
- Dynamic Parallelism
- Generic Address Space
- Image Support
- Android Installable Client Driver Extension
• **Shared Virtual Memory**

• **Dynamic Parallelism**
  • Generic Address Space
  • Image Support
  • **Android** Installable Client Driver Extension
## Bigger picture

- **Goal:** A benchmark and micro benchmark suite with OpenCL 2.0 applications
- Features that are interesting in HSA and OpenCL 2.0

<table>
<thead>
<tr>
<th>HPC</th>
<th>Mobile/Embedded</th>
<th>Big Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Spectral Clustering</td>
<td>1. N-channel IIR Filtering</td>
<td>1. Rating System using MapReduce</td>
</tr>
<tr>
<td>5. Feature Selection and Outlier Detection</td>
<td>5. Convolution Neural Network</td>
<td>5. Gene Sequencing</td>
</tr>
<tr>
<td>6. 2-D Finite Difference Time-Domain</td>
<td>6. Shallow Water Simulation</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7. Color Histogramming</td>
<td></td>
</tr>
</tbody>
</table>
Bigger picture

- **Goal**: A benchmark and micro benchmark suite with OpenCL 2.0 applications
- Features that are interesting in HSA and OpenCL 2.0

<table>
<thead>
<tr>
<th>HPC</th>
<th>Mobile/Embedded</th>
<th>Big Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Spectral Clustering</td>
<td>N-channel IIR Filtering</td>
<td>1. Rating System using MapReduce</td>
</tr>
<tr>
<td>2. Connected Component Labeling</td>
<td>Multi-channel Noise Filter using FIR Filtering</td>
<td>K-means Clustering</td>
</tr>
<tr>
<td>4. Periodic Greens Function</td>
<td>AES encryption/decryption</td>
<td>4. Bayesian Estimation for Adaptive Spam Filtering/Learning</td>
</tr>
<tr>
<td>5. Feature Selection and Outlier Detection</td>
<td>Convolution Neural Network</td>
<td>5. Gene Sequencing</td>
</tr>
<tr>
<td>6. 2-D Finite Difference Time-Domain</td>
<td>Shallow Water Simulation</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Color Histogramming</td>
<td></td>
</tr>
</tbody>
</table>
Exploring the benefits of OpenCL 2.0

CyberSecurity: The Advanced Encryption Standard (AES)

- Adopted by US government for encryption
- Input as plain text with 256 bit key produces cipher text
- Blocks running concurrently
- Our results show that key expansion is faster on CPU than GPU
- 14 rounds of AES-256 are performed on GPU
Exploring the benefits of OpenCL 2.0

**Signal Processing**: Finite Impulse Response Filtering

- **Impulse Response** of finite duration
- Input: \(x[1\ldots n]\) and \(b[1\ldots N]\) → output: \(f[x]\)
- Number of taps: \(N = 1024\)
- Synthesized audio stream input
- Uses weighted reduction - very common parallel operation

\[
 f(x) = b_0 x[n] + b_1 x[n-1] + \ldots + b_N x[n-N] \\
 = \sum_{i=0}^{N} b_i x[n-i]
\]
**Signal Processing:** Infinite Impulse Response Filter

- Less processing power than FIR for same design
- Decomposed into multiple parallel 2nd-order (real and complex) IIR for performance
- $N_1$ – number of real poles
- $N_2$ – number of complex poles
- Number of channels = 64
- FIR coefficient: $c_0 = 3.0$
- Synthesized audio stream input

\[
H^z(z) = c_0 + \sum_{i=1}^{N_1} \frac{f_i}{1 + e_i z^{-1}} + \sum_{i=1}^{N_2} \frac{f_{N_1+2i-1} + f_{N_1+2i} z^{-1}}{1 + e_{N_1+2i-1} z^{-1} + e_{N_1+2i} z^{-2}}
\]
Exploring the benefits of OpenCL 2.0

Statistical Modeling: Hidden Markov Models

- Probabilistic meaning of hidden states without prior knowledge
- Targeting isolated word recognition
- Matrix form used for coalescing and computational efficiency
- Uses operations including
  - Matrix multiplication
  - Matrix vector
  - Parallel reduction
- Uses data & thread level parallelism
Ongoing OpenCL 2.0 Evaluation

- Baseline: OpenCL 1.2
- GPU model: AMD Radeon R9 290x (reported in paper)
  - Current use: AMD A10-7850K Radeon R7, Kaveri APU
- GPU Architecture:
  - Compute Cores: 12 (4 CPU & 8 GPU)
  - Global Memory: 512 MB
  - Max Clock frequency: 720 MHz
- GPU Driver: 1642.5 (VM)
AES Results

Optimizations explored:

- SVM
- X Dynamic Parallelism

- Input files contain excerpts of a book
- Input sizes are varied from 1MB to 1,000MB with constant 256 bit key
- Small benefits from SVM, which grow with input file size
- Child kernel is memory intensive, inhibiting dynamic parallelism
FIR Results

Optimizations explored:

✓ SVM

- FIR is a streaming application with different block sizes
- Results show that same kernel runs faster in OpenCL 2.0
- Consistent benefits from SVM, which grow with input block size
IIR Results

Optimizations explored:

✓ SVM
X Workgroup function

• Interesting feature - parallel reduction
• Workgroup function is useful for reduction, but did not work well
Exploring Workgroup Function further in IIR

- Workgroup function is useful for reduction, but did not work well in OpenCL 2.0
- It works better in HSAIL on HSA, but not as good as reduction
Hidden Markov Model Results

Optimizations explored:

✓ SVM
✓ Dynamic Parallelism

• Updating expected values for each hidden state is an independent operation - perfect for Dynamic Parallelism!
Data Mining: K-means algorithm

- Well known clustering algorithm.
- K-means with different number of objects, 34 features, 5 clusters
- Input file contains features and attributes
- Consistent benefits from SVM

K-means Results

Optimizations explored:

 ✓ SVM

![Bar chart showing execution time comparison between OpenCL 1.2 and OpenCL 2.0 for different number of objects.](chart.png)
Shallow Water Simulation Results

Physics simulation: Shallow Water Engine

Optimizations explored:

✅ SVM

- Depicts complex behavior of fluids, wave modeling for interactive systems
- Predicts matters of practical interest, e.g. internal tides in strait of Gibraltar
- Mathematically and computationally intense, so expensive to do real-time
Summary

- OpenCL 2.0 introduced new features
- We have explored the benefits of using them with some benchmarks from a variety of domains
- SVM provides consistent benefits
- Exploring issues with utilizing the work-group function
- The benchmark suite will be released Summer 2015
THANK YOU TO OUR SPONSORS