GLOpenCL: OpenCL Support on Hardware- and Software-Managed Cache Multicores

Konstantis Daloukas
Christos D. Antonopoulos
Nikolaos Bellas

Department of Computer and Communications Engineering
University of Thessaly
Volos, Greece
Introduction

• OpenCL: a unified programming standard and framework
• Targets:
  – Homogeneous and heterogeneous multicores
  – Accelerator-based systems
• Aims at being platform-agnostic
• Enhances portability
Motivation

• Vendor specific OpenCL implementations target either only hardware- or software-controlled cache multicores
  – AMD Stream SDK – x86
  – IBM OpenCL SDK – Cell B.E.

• Lack of a unified framework for architectures with diverse characteristics
Contribution

• **GLOpenCL:** a *unified* OpenCL framework
  – Compilation infrastructure
  – Run-time support

• Enables native execution of OpenCL applications on:
  – Hardware-controlled cache multicores
  – Software-controlled cache multicores

• Achieves comparable performance to architecture-specific implementations
Outline

• Introduction
• The OpenCL Programming Model
• Compilation Infrastructure
• Run-Time Support
• Experimental Evaluation
• Conclusions
OpenCL Platform Model

From: http://www.viznet.ac.uk
OpenCL Execution Model

• An OpenCL application consists of two parts:
  – Main program that executes on the host
  – A number of kernels that execute on the compute devices

• Main constructs of the OpenCL execution model:
  – Kernels
  – Memory Buffers
  – Command Queues
OpenCL Kernel Execution “Geometry”

work-group $(wx, wy)$

- $sx = 0$
- $sy = 0$
- work-item $(wxSx + sx, wySy + sy)$

- $sx = Sx - 1$
- $sy = 0$
- work-item $(wxSx + sx, wySy + sy)$

- $sx = 0$
- $sy = Sy - 1$
- work-item $(wxSx + sx, wySy + sy)$

- $sx = Sx - 1$
- $sy = Sy - 1$
- work-item $(wxSx + sx, wySy + sy)$
Outline

• Introduction
• The OpenCL Programming Model
• Compilation Infrastructure
• Run-Time Support
• Experimental Evaluation
• Conclusions
Granularity Management

work-group

Integrated Memory Controller - 3 Ch DDR3

Core 0  Core 1  Core 2  Core 3

Shared L3 Cache

23/03/2011  HiPEAC 2011
__kernel void Vadd(...) {
    int index = get_global_id(0);
    c[index] = a[index] + b[index];
}

#define get_item_gid(N) \n    (__global_id[N] + (N == 0 ? __k : (N == 1 ? __j : __i)))

C code

__kernel void Vadd(...) {
    int index;
    for( i = 0; i < get_local_size(2); i++)
        for( j = 0; j < get_local_size(1); j++)
            for( k = 0; k < get_local_size(0); k++) {
                index = get_item_gid(0);
                c[index] = a[index] + b[index];
            }
}
Elimination of Synchronization Operations

OpenCL code

Statements_block_1
barrier();
Statements_block_2

triple_nested_loop {
    Statements_block_1
    barrier();
    Statements_block_2
    }

C code
Elimination of Synchronization Operations

C code

```
triple_nested_loop {
    Statements_block_1
    barrier();
    Statements_block_2
}
```

C code

```
triple_nested_loop {
    Statements_block_1
}
//barrier();
triple_nested_loop {
    Statements_block_2
}
```
Variable Privatization

```
C code

triple_nested_loop { 
    Statements_block_1
    x = ...;
}

triple_nested_loop { 
    Statements_block_2
    ... = ... x ...;
}

C code

triple_nested_loop { 
    Statements_block_1
    x[i][j][k] = ...;
}

triple_nested_loop { 
    Statements_block_2
    ... = ... x[i][j][k] ...;
}
```
Outline

• Introduction
• The OpenCL Programming Model
• Compilation Infrastructure
• Run-Time Support
• Experimental Evaluation
• Conclusions
Run-Time System Library

• Provide run-time support for:
  – Work management
  – Manipulation of memory buffers

• The run-time system provides a unified design for both architectures
Architecture for Hardware-Controlled Cache Multicores
Architecture for Hardware-Controlled Cache Multicores
Architecture for Hardware-Controlled Cache Multicores

Main Thread

Worker Thread

Worker Thread

Command Queue

C₁

C₂

C₃

Ready Command Queue

23/03/2011

HiPEAC 2011
Architecture for Hardware-Controlled Cache Multicores

Command Queue

Ready Command Queue

Work Queue 1

Work Queue N

23/03/2011

HiPEAC 2011

20
Architecture for Hardware-Controlled Cache Multicores

Main
Thread

Command Queue

Ready Command Queue

Work Queue 1

Work Queue N

23/03/2011

HiPEAC 2011

21
Architecture for Hardware-Controlled Cache Multicores

Command Queue

Worker Thread 1

Worker Thread N

Main Thread

C2

C3

23/03/2011

HiPEAC 2011

Work Queue 1

Work Queue N

22
Architecture for Hardware-Controlled Cache Multicores

Main Thread

Command Queue

Ready Command Queue

Work Queue 1

Work Queue N

23/03/2011

HiPEAC 2011
Architecture for Hardware-Controlled Cache Multicores

Main Thread

Command Queue

Ready Command Queue

Work Queue 1

Work Queue N

23/03/2011

HiPEAC 2011
Architecture for Hardware-Controlled Cache Multicores

Main Thread

Helper Thread

Async. Copy Queue

Worker Thread 1

Worker Thread N

Command Queue

Ready Command Queue

Work Queue 1

Work Queue N

23/03/2011

HiPEAC 2011
Architecture for Software-Controlled Cache Multicores

Host Side

Main Thread

Command Queue

Ready Command Queue

Helper Thread

Accelerator Side

Work Queue 1

Work Queue N

Worker Thread 1

Worker Thread N

Work related req./replies

SW cache req./replies

23/03/2011

HiPEAC 2011
Outline

• Introduction
• The OpenCL Programming Model
• Compilation Infrastructure
• Run-Time Support
• Experimental Evaluation
• Conclusions
Experimental Evaluation

- Evaluate the framework on two representative architectures:
  - Homogeneous, hardware-controlled cache memory processor – Intel’s E5520 i7 (Nehalem) processor
  - Heterogeneous, software-controlled cache memory processor – Cell B.E. processor

- Compare with vendor provided, platform-customized implementations
  - AMD Stream SDK for x86 architectures
  - IBM OpenCL SDK for the Cell B.E.
Vector Add – x86

Vector Add (1-D part.)

Execution time (sec)

- GLOpenCL
- AMD OpenCL

64 128 256 1K 2K 64 128 256 1K 2K 64 128 256 1K 2K 64 128 256 1K 2K
16M 32M 64M 96M
Vector Add – Cell

Vector Add (1-D part.)

Execution time (sec)

- **GLOpenCL**
- **IBM OpenCL**

<table>
<thead>
<tr>
<th>64</th>
<th>128</th>
<th>256</th>
<th>2K</th>
<th>64</th>
<th>128</th>
<th>256</th>
<th>2K</th>
<th>64</th>
<th>128</th>
<th>256</th>
<th>2K</th>
<th>64</th>
<th>128</th>
<th>256</th>
<th>2K</th>
</tr>
</thead>
<tbody>
<tr>
<td>2M</td>
<td>4M</td>
<td>8M</td>
<td>32M</td>
<td>2M</td>
<td>4M</td>
<td>8M</td>
<td>32M</td>
<td>2M</td>
<td>4M</td>
<td>8M</td>
<td>32M</td>
<td>2M</td>
<td>4M</td>
<td>8M</td>
<td>32M</td>
</tr>
<tr>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.01</td>
<td></td>
<td></td>
<td></td>
<td>0.01</td>
<td></td>
<td></td>
<td></td>
<td>0.01</td>
<td></td>
<td></td>
<td></td>
<td>0.01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.02</td>
<td></td>
<td></td>
<td></td>
<td>0.02</td>
<td></td>
<td></td>
<td></td>
<td>0.02</td>
<td></td>
<td></td>
<td></td>
<td>0.02</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.03</td>
<td></td>
<td></td>
<td></td>
<td>0.03</td>
<td></td>
<td></td>
<td></td>
<td>0.03</td>
<td></td>
<td></td>
<td></td>
<td>0.03</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.04</td>
<td></td>
<td></td>
<td></td>
<td>0.04</td>
<td></td>
<td></td>
<td></td>
<td>0.04</td>
<td></td>
<td></td>
<td></td>
<td>0.04</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.05</td>
<td></td>
<td></td>
<td></td>
<td>0.05</td>
<td></td>
<td></td>
<td></td>
<td>0.05</td>
<td></td>
<td></td>
<td></td>
<td>0.05</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.06</td>
<td></td>
<td></td>
<td></td>
<td>0.06</td>
<td></td>
<td></td>
<td></td>
<td>0.06</td>
<td></td>
<td></td>
<td></td>
<td>0.06</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.07</td>
<td></td>
<td></td>
<td></td>
<td>0.07</td>
<td></td>
<td></td>
<td></td>
<td>0.07</td>
<td></td>
<td></td>
<td></td>
<td>0.07</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

23/03/2011
AES Encryption – x86

AES Encryption (2-D part.)

Execution Time (sec)

- GLOpenCL
- AMD OpenCL

<table>
<thead>
<tr>
<th></th>
<th>GLOpenCL</th>
<th>AMD OpenCL</th>
</tr>
</thead>
<tbody>
<tr>
<td>8x8 HD</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>16x16 HD</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>8x8 UHD</td>
<td>2.5</td>
<td>5.0</td>
</tr>
<tr>
<td>16x16 UHD</td>
<td>2.5</td>
<td>5.0</td>
</tr>
</tbody>
</table>
AES Encryption – Cell

AES Encryption (2-D part.)

<table>
<thead>
<tr>
<th></th>
<th>GLOpenCL</th>
<th>IBM OpenCL</th>
</tr>
</thead>
<tbody>
<tr>
<td>8x8 HD</td>
<td>0.2</td>
<td>0.3</td>
</tr>
<tr>
<td>16x16 HD</td>
<td>0.3</td>
<td>0.4</td>
</tr>
<tr>
<td>8x8 UHD</td>
<td>3.8</td>
<td>3.9</td>
</tr>
<tr>
<td>16x16 UHD</td>
<td>3.9</td>
<td>3.8</td>
</tr>
</tbody>
</table>
BlackScholes – x86

Black Scholes (1-D part.)

- GLOpenCL
- GLOpenCL_async
- AMD OpenCL
- AMD OpenCL_async

Execution Time (sec)

64 128 256 1K 2K
16M
32M
64M
96M
BlackScholes – Cell

BlackScholes (1-D part.)

Execution Time (sec)

- GLOpenCL
- GLOpenCL_async
- IBM OpenCL
- IBM OpenCL_async

<table>
<thead>
<tr>
<th>64</th>
<th>128</th>
<th>256</th>
<th>1K</th>
<th>2K</th>
</tr>
</thead>
<tbody>
<tr>
<td>1M</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>128</td>
<td>256</td>
<td>1K</td>
<td>2K</td>
</tr>
<tr>
<td>2M</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>128</td>
<td>256</td>
<td>1K</td>
<td>2K</td>
</tr>
<tr>
<td>4M</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Outline

• Introduction
• The OpenCL Programming Model
• Compilation Infrastructure
• Run-Time Support
• Experimental Evaluation
• Conclusions
Conclusions

• GLOpenCL achieves comparable, or better performance than architecture specific implementations

• OpenCL standard leaves enough room for platform specific optimizations
  – May yield significant performance improvements

• Future Work: Reduce the effect of shared data structures
  – Memory access pattern analysis
Acknowledgements

This project is partially supported by the EC Marie Curie International Reintegration Grant (IRG) 223819