Automatic Offloading to Cluster Accelerators
Ciro Ceissler, Ramon Nepomuceno, Marcio Pereira and Guido Araujo
ciro.ceissler@students.ic.unicamp.br, {ramon.nepomuceno, mpereira, guido}@ic.unicamp.br
Institute of Computing - UNICAMP

For more information
www.hardcloud.org

Objective
HardCloud seeks to ease the task of integrating programs to FPGA-based accelerators. In HardCloud programmers could take their original code and add a few annotations and quickly evaluate if an FPGA accelerator is a suitable solution to a particular application.

Hardware Interface

```
module IP (
    input logic clk,
    input logic rst,
    input logic start,
    output logic finish,
    hc_buffer_if buffer
);
/* IP implementation */
endmodule : IP
```

Results

<table>
<thead>
<tr>
<th>ID</th>
<th>Benchmark</th>
<th>Freq.</th>
<th>Send</th>
<th>Recv.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AES-128</td>
<td>200 MHz</td>
<td>1.2 GB</td>
<td>1.2 GB</td>
</tr>
<tr>
<td>2</td>
<td>SHA-512</td>
<td>200 MHz</td>
<td>1.2 GB</td>
<td>64 B</td>
</tr>
<tr>
<td>3</td>
<td>Sobel Filter</td>
<td>400 MHz</td>
<td>0.8 MB</td>
<td>0.8 MB</td>
</tr>
<tr>
<td>4</td>
<td>Gaussian Filter</td>
<td>400 MHz</td>
<td>3.1 MB</td>
<td>3.1 MB</td>
</tr>
<tr>
<td>5</td>
<td>MD5</td>
<td>200 MHz</td>
<td>1.2 GB</td>
<td>0.6 GB</td>
</tr>
<tr>
<td>6</td>
<td>FFT</td>
<td>200 MHz</td>
<td>1.2 GB</td>
<td>1.2 GB</td>
</tr>
<tr>
<td>7</td>
<td>Reed-Solomon Decoder</td>
<td>300 MHz</td>
<td>1.6 GB</td>
<td>1.5 GB</td>
</tr>
<tr>
<td>8</td>
<td>FIR Filter 4th Order</td>
<td>200 MHi</td>
<td>1.2 GB</td>
<td>1.2 GB</td>
</tr>
<tr>
<td>9</td>
<td>Smith-Waterman</td>
<td>200 MHz</td>
<td>0.6 MB</td>
<td>0.6 MB</td>
</tr>
<tr>
<td>10</td>
<td>Gene Regulatory Network</td>
<td>200 MHz</td>
<td>0 GB</td>
<td>80 MB</td>
</tr>
</tbody>
</table>

Table 1: Benchmark runtime.

Acknowledgements
I sincerely express my gratitude to the IN-FIERI school, PETLAB, HUST, UNICAMP and CNPq for giving me the opportunity to participate this wonderful event.

Execution of a pre-compiled FPGA bit-stream

```
#pragma omp target device (HARP) map(to: X) map(from: Y)
#pragma omp parallel for
use(hrw) module(loopback)
// Software version of the loopback hardware module.
for(int i = 0; i < n; i++) {
    Y[i] = X[i];
}
```

The clause `use(hrw)` specifies that the annotated code block will use a pre-designed hardware, e.g., `module (loopback)`, to do the computation instead of the C code following the annotation.

The Clang/LLVM OpenMP 4.0 runtime was extended to enable the design of HardCloud.

Buffer Access Mode Functions

```
buffer [read request]
    read_stream(id, size)
    read_indexed(id, offset)

buffer [read response]
    valid()
    data()

buffer [write request]
    write_stream(id, data)
    write_indexed(id, offset, data)
```

- id: buffer identifier
- data: one cache-line
- size: data size in cache-lines
- offset: displacement within buffer (indexed mode only)

Access Modes

(a) Indexed Access Mode
(b) Stream Access Mode

The HardCloud Interface provides an effortless method to connect the IP core by exchanging information with the OpenMP runtime and reading/writing the shared memory in a seamless way, through the CCI-P interface. Two operation modes are available: stream and indexed.

HardCloud Architecture

Highlighted in gray are the blocks developed to enable HardCloud. HC_CSR manages communication with the host, while HC_Requestor controls the flow of data to/from shared memory.