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 add a few annotations and quickly evaluate if an FPGA accelerator is a suitable solution to a particular application.
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 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-
e.g., module (loopback), to do the computation instead of the C code following
The Clang/LLVM OpenMP 4.0 runtime was extended to enable the design of HardCl
Hardware Interface
module IP (
input logic clk,
input logic rst,
input logic start,
output logic finish,
hc_buffer_if buffer
);
/* IP implementation */
endmodule : IP
Buffer Access Mode Functions
buffer
[read request]
[read response]
[write request]
read_stream(id, size)
read_indexed(id, offset)
valid()
data()
write_stream(id, data)
write_indexed(id, offset, data)
• id: buffer identifier
• size: data size in cache-lines
• data: one cache-line
• offset: displacement within buffer (indexed mode only)
Results
Figure 1: Logarithmic FPGA speedup over CPU
Table 1: Benchmark runtime.
Access Modes
(a) Indexed Access Mode
(b) Stream Access Mode
Figure 2: Operation Modes Available
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
Figure 3: 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.
Acknowledgements
INFIERI2019@ HUST
A new in-beam proton therapy monitoring system based on digital MVT readout
Introduction
Design and Implementation