RECONF-HPC
Workshop on Reconfigurable High-Performance Computing

Organizers: Kazutomo Yoshii (ANL), Ryousei Takano (AIST), Taisuke Boku (University of Tsukuba)
Date: December 11 (Tue) afternoon, 2018
Location: Tenbusu-Naha Hall 3-2-10, Makishi, Naha-shi, Okinawa, Japan  http://www.fpt18.sakura.ne.jp/venue.html
Room: Tenbusu Hall, 4th floor

Abstract

Reconfigurable computers are expected to play an important role in the post-Moore era, offering a true co-design vehicle that could significantly improve both the performance and the energy efficiency of computation. While FPGAs-accelerated systems are becoming practical in the cloud and data centers, FPGAs or reconfigurable computers are still not common in large HPC systems. In this workshop, we will discuss future workloads, programming models, network/storage acceleration and clustering technologies that can enable reconfigurable high-performance computing. Audience participation is highly encouraged.

Agenda

<table>
<thead>
<tr>
<th>Time</th>
<th>Session I: chair: Ryousei Takano (AIST)</th>
</tr>
</thead>
<tbody>
<tr>
<td>13:00</td>
<td>Opening and welcome by Taisuke Boku (University of Tsukuba)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Time</th>
<th>Session I: Galapagos: A Stacked Approach to Enabling Multi-FPGA HPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>13:10</td>
<td>While FPGAs have already been shown to enable significant acceleration of many applications, it has always been a challenge to program them. To make FPGAs viable and easy to use as computing elements requires much more than just overcoming the challenges of programming with hardware description languages (HDL). Recently, the FPGA vendors have introduced OpenCL as a complete programming environment for FPGAs that truly makes them usable as computing devices. The OpenCL provided by the FPGA vendors is more than a high-level synthesis tool for a high-level language. The platform provides memory and I/O abstractions that completely abstracts the low-level issues faced by the HDL programmer using a barebone FPGA PCIe card. While this programming model is a significant advance towards making FPGAs into computing devices, it is still short of what is required to enable multi-FPGA applications that might be used in HPC. In this talk, I will describe our work on Galapagos, a stacked approach for enabling multi-FPGA computing environments. By providing multiple levels of abstraction in different layers, it is possible to build new programming models suitable for different types of applications and to experiment with lower abstraction layers, such as the communication layer, without impacting the application. We show that the abstractions can be provided with minimal impact on communication latency and no affect on link throughput.</td>
</tr>
</tbody>
</table>

Paul Chow (University of Toronto)

Paul Chow is a professor in the faculty of The Edward S. Rogers Sr. Department of Electrical and Computer Engineering at the University of Toronto where he holds the Dusan and Anne Miklas Chair in Engineering Design. He was a major contributor to the early RISC processor technology developed at Stanford University that helped spawn the rapid rise of computing performance in the past 30 years. Paul is now directing his research towards the next significant disruption in the way we do computing, which is computing at scale. His main interest is the integration of heterogeneous computing devices, with a specific focus on making Field-Programmable Gate Arrays (FPGAs) into platforms for computing at scale.

Paul has been the technical program and general chairs for FPGA, the premier conference on FPGAs, and for FCCM, the main conference for reconfigurable computing. He co-founded AcceLight Networks to build a high-capacity, carrier-grade, optical switching system, which did not survive the dot-com era but spent a lot of other people’s money. Paul is also a co-founder of ArchES Computing Systems, which still lives, and is developing reconfigurable computing technology for the data centre.
Opportunities of Accelerating HPC Kernels with FPGAs

Field-programmable gate arrays are becoming more promising for high performance computing. Unlike fixed hardware specialization through ASICs, reconfigurability allows FPGAs to be customized for a wide variety of HPC workloads. This talk first presents previous promising results of accelerating scientific kernels with FPGAs using an OpenCL compiler. Our results indicate that certain computation patterns are more likely to be amenable to efficient acceleration with FPGAs. We conclude this talk by discussing several specific examples that exhibit such patterns and how they could be mapped to efficient dataflow pipelines on FPGAs.

Naoya Maruyama (LLNL)

Naoya Maruyama is a researcher at Lawrence Livermore National Laboratory, where he studies cross-cutting domains of high performance computing and machine learning. Prior to joining LLNL, he was a Team Leader at RIKEN Advanced Institute for Computational Science, where he led research projects on high-level programming abstractions for heterogeneous architectures. He won several awards, including a Gordon Bell Prize in 2011 and Best Paper Award at SC16. He received Ph.D. in Computer Science from Tokyo Institute of Technology in 2008.

Establishing FPGA acceleration in HPC production systems and codes

HPC and data centers are looking into new approaches that allow for further performance scaling under given power and energy constraints. With progress in FPGA architectures and development tools, FPGAs have started to provide unique benefits in computing performance and energy-efficiency compared to other processor and accelerator technologies. While we have seen first large-scale deployments of FPGAs in public and private clouds and data centers, FPGAs still have to make inroads in general purpose HPC systems. At the Paderborn Center for Parallel Computing, we are at the forefront of this development and have recently put “Noctua” our first HPC cluster with FPGAs into production.

In this talk, I will share some of the experiences we made on our journey from the planning, to the procurement to the installation of the Noctua cluster and highlight critical aspects for FPGAs and how we addressed them. I will also present ongoing work on direct FPGA to FPGA connection infrastructure. Further, I will present first results from porting libraries and MPI-parallel HPC codes to the 32 Intel Stratix 10 FPGA boards in our cluster.

Tobias Kenter (Paderborn University)

Tobias Kenter is PostDoc researcher at the Paderborn Center for Parallel Computing (PC²). He has been involved in several research projects studying reconfigurable architectures, design flows and productivity, runtime systems and the application of FPGAs in HPC. His Ph.D. thesis was nominated by Paderborn University as candidate for the best German Ph.D. thesis in computer science in 2016. At PC², which is Paderborn University’s HPC center providing computing resources for computational sciences at Paderborn University and Germany-wide, Tobias Kenter is leading the activities on FPGA acceleration. PC² has recently deployed its first production HPC cluster with FPGAs.

C/C++ Front-end for Streaming Processing on FPGAs

Although Field-Programmable Gate Array (FPGA) is considered as one of the promising solutions to realize dedicated hardware for High-Performance Computing (HPC), it is difficult for non-experts to program FPGAs due to the gap between their applications and hardware-level programming models for FPGAs. In this talk, we propose a C/C++ based programming framework, C2SPD, to describe stream processing on FPGA. It uses SPGen, a data-flow High Level Synthesis (HSL) tool, as the FPGA backend. C2SPD provides directives to specify code regions to be offloaded onto FPGAs. Although the range of application is limited by its domain-specific approach, it can generate highly-pipelined hardware on FPGAs. A 2D-stencil computation kernel is written in C and C2SPD directives and the generated FPGA hardware achieves 175.41 GFLOPS by using 256 stream cores.

Jinpil Lee (RIKEN)

Coffee break

Session II: chair: Taisuke Boku (University of Tsukuba)

High Efficiency Dataflow Applications for Xilinx FPGAs

FPGAs have finally arrived in the datacenter. Now, applications are needed to make good use of the availability of FPGA resources from major cloud providers as well as the new Xilinx Alveo cards. In this talk, I will show an overview of applications that Maxeler has deployed on FPGAs in datacenters over the past decade, from Quantum Chromodynamics, NEMO, SPECFEM3D, QuantumEspresso, Seismic Imaging, Finance Risk, to AI (inference and training), video encoding, genomics and computational fluid dynamics.

Oskar Mencer (Maxeler)
### FPGA Virtualisation for Cloud Computing

While all major cloud service providers have FPGA instance offerings these days, this only includes single tenancy operation so far. This omits fundamental cloud principles, like for example that equipment can be better utilized across multiple applications with different requirements (e.g., needed resources for compute, memory capacity, I/O bandwidths etc.).

This talk will provide a quick journey through the FPGA virtualisation landscape, discuss upcoming requirements and reveal latest work. This includes aspects of physically implementing systems for multi-tenancy, system run-time management and security aspects.

For example, our group developed the concept of resource elasticity that uses a cooperative scheduling technique to keep FPGA resource utilisation high for delivering high performance even if workload and amount of resources changes at run-time. The talk will show how this is all managed by a run-time system and this service is provided entirely transparent to the tenants or users of the system. The talk will also shown how this scheme can be scaled up to large datacenter installations.

**Dirk Koch (The University of Manchester)**

Dirk Koch is a senior lecturer in the Advanced Processor Technologies Group at the University of Manchester. His main research interest is on run-time reconfigurable systems based on FPGAs, embedded systems, computer architecture and VLSI. Dirk developed techniques and tools for self-adaptive distributed embedded control systems based on FPGAs. Current research projects include database acceleration using FPGAs-based stream processing, HPC and exascale computing, as well as reconfigurable instruction set extensions for CPUs. In the new research project FORTE, Dirk is designing new kinds of FPGA fabrics using memristive materials.

Dirk Koch is author of the book "Partial Reconfiguration on FPGAs" and a co-editor of the book "FPGAs for Software Programmers" and his group is developing and maintaining the GoAhead framework that provides unique capabilities for building run-time reconfigurable systems.

### Application of FPGAs in High Performance and High Precision Computing

In this talk, I present our implementation of a high performance stencil application on FPGAs using an OpenCL based high-level synthesis tool. With slight modifications and additional optimizations to an OpenCL kernel designed for GPUs, our design shows comparable performance to our previous dedicated design entirely written in HDL. I also show the performance evaluation of high-precision floating-point arithmetic operations on recent FPGAs for applications in high energy physics.

**Naohito Nakasato (University of Aizu)**

Naohito Nakasato obtained PhD in Science (Astronomy) from Graduate School of Science, the University of Tokyo in 2000. Currently Senior Associate Professor and leader of CAIST ARC-HPC at University of Aizu. Member of IPSJ, JSIAM, IEEE, ACM, Astronomical Society of Japan.

Research interests include HPC, parallel computing architecture and reconfigurable computing.

### Automated Space/Time Scaling of Streaming Task Graphs on FPGAs

Software applications are often computationally intensive. To maximize the use of resources on a range of hardware platforms with a differing amount of parallel resources and minimize the runtime of software applications as much as possible, engineers need to manually modify and optimize each application for each platform with different resource restrictions or desired performance targets. This process is time consuming and can be prone to error. We propose an approach to automatically explore different ways of implementing an application for a performance target or a resource restriction defined by users. We explore automated space/time scaling of Streaming Task Graphs targeting a pipelined coarse-grained architecture (MPPA), a fine-grained architecture (FPGA fabric) and a Hybrid architecture. We provide an environment for pure software programmers to automatically explore space/time scaling of streaming task graph on reconfigurable platforms without any hardware engineering knowledge. Moreover, to achieve higher performance, we investigate using dynamic Partial Reconfiguration (PR) by time-sharing the FPGA resources.

**Hossein Omidian (Xilinx)**

Hossein started his career as a researcher (Hardware Developer) in ICTI research center in 2004. During his Masters, he worked for two different companies as a consultant helping to accelerate applications on FPGAs. After his Masters, he began working in oil and gas industry with a focus on accelerating seismic data processing applications on different platforms (e.g., FPGAs and GPUs) for four years. During that time, he co-founded a startup company (Gabian-pro) which was acquired in 2012. After selling the company, he moved to Vancouver, Canada to do his PhD at University of British Columbia. During his PhD, he did a 7-month internship at Xilinx in 2016. He was hired by Xilinx as a Senior Software Eng. after earning his PhD. Right now, he works in the FPGA architecture group at Xilinx exploring next generation architectures.

### Parallel Processing on FPGA Combining Computation and Communication in OpenCL Programming

We have been developed an environment named Channel over Ethernet (CoE) which enables the OpenCL programming on Intel FPGA where an FPGA can communicate with FPGAs on different computation node through the external communication link on the FPGA. The API for OpenCL programming is implemented as Channels on OpenCL supporting environment for easy use by application users on OpenCL. We will show the detailed implementation including how BSP should be modified for this work. We will also show the result on our prototype implementation with Himeno Benchmark which is a simple multi-dimensional stencil computation on Intel Arria10 FPGA with 40Gb Ethernet network. We will apply this system on Stratix10 base system in near future.

**Norihisa Fujita (University of Tsukuba)**

Norihisa Fujita is a postdoc researcher in Center for Computational Sciences, University of Tsukuba. He received PhD from University of Tsukuba in 2016. His research interest is parallel system using accelerators.
<table>
<thead>
<tr>
<th>1</th>
<th>short break</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1</th>
<th>Open discussion and wrap-up by Kazutomo Yoshii (ANL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td></td>
</tr>
</tbody>
</table>