CREST
Core Research for Evolutionary Science and Technology

Development of System Software Technologies for post-Peta Scale High Performance Computing

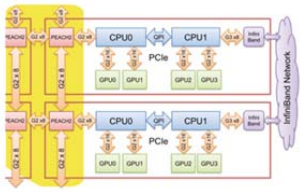
Research and Development on Unified Environment of Accelerated Computing and Interconnection for Post-Petascale Era

Low Latency Communication for Strong Scaling with Next Generation Parallel Accelerated Computing

AiS (Accelerator in Switch) and FPGA interconnection

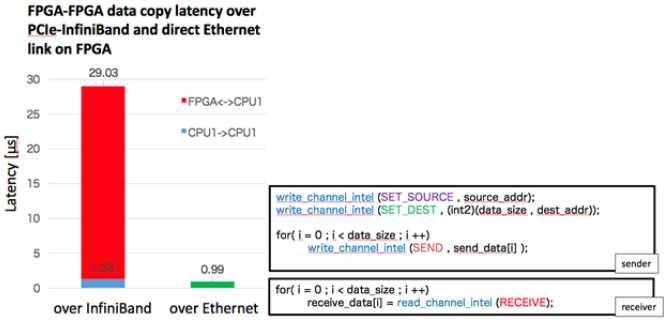
As an original research for short latency communication among GPUs over nodes to achieve strong scalability on next generation accelerated computing, we have been developing a prototype system named PEACH2 under concept of TCA (Tightly Coupled Accelerators). PEACH2 can realizes 2.0μs of latency for GPU-GPU remote memory copy enabling single address space for PCIe even over multiple nodes. PEACH2 is implemented by FPGA for flexible design and suitability for PCIe interface and protocol.

PEACH2 board by FPGA
(Altera Startix-IV)



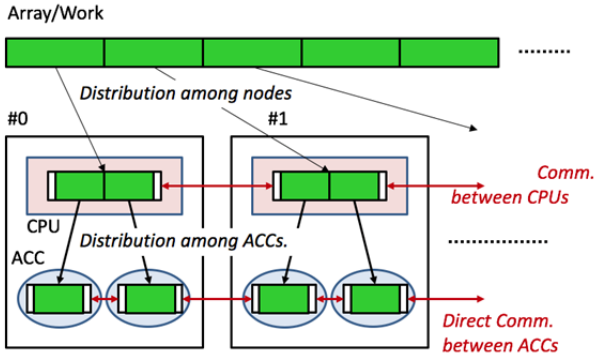
concept of TCA

After we found the limitation on scalability and bandwidth of conventional PCIe solution, we stepped into more aggressive solution named AiS (Accelerator in Switch) to exploit high potential of recent FPGA. Here, we implement both computation and communication with high level FPGA programming by OpenCL for partial off-loading of user applications. We have developed Verilog HDL module which enables to drive the high speed optical interconnect from OpenCL code. It reduces latency for FPGA-FPGA remote data copy in only 1μs while traditional way through PCIe switch on CPU and external InfiniBand HCA takes approximately 30x longer latency.



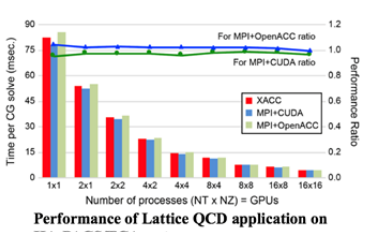
XcalableACC a directive-based language extension for accelerated parallel computing

To solve the low productivity on programming by MPI+CUDA style, we are developing new language framework XcalableACC (XACC) where OpenACC description is involved to XcalableMP PGAS language. Either PEACH2/TCA or ordinary MPI interface can be applied for compiler generated inter-GPU communication to provide highly productive and effective GPU coding. The base language XcalableMP is for distributed-memory parallelism, and computation off-loading by OpenACC description is applied to distributed arrays automatically.



Parallelization and off-load model of XACC

For a new target of accelerator, we are implementing an OpenACC compiler for PEZY-SC processor which provides one of the best performance/power ratio in the world. As well as basic OpenACC feature implementation, we will apply it to a new XACC compiler.



Performance of Lattice QCD application on HA-PACS/TCA system

```
Quark_t v[NT][NZ][NY][NX];
// distributes p with XMP directives
#pragma xmp shadow v[1][1][0][0]
#pragma acc enter data copyin(v)

#pragma xmp reflect (v) orthogonal acc …
..  …
#pragma xmp loop (iz,it) on t[it][iz]
#pragma acc parallel loop collapse(4) …
for(it = 0; it < NT; it++){
  for(iz = 0; iz < NZ; iz++){
    for(iy = 0; iy < NY; iy++){
      for(ix = 0; ix < NX; ix++){
        …
```

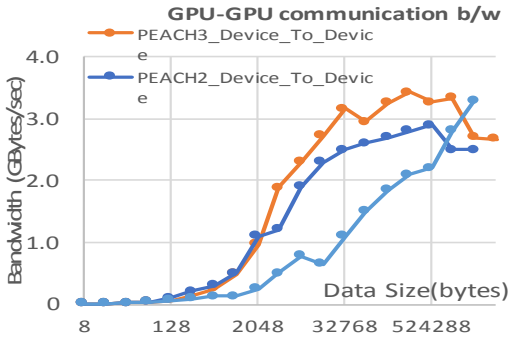
XACC coding for Lattice QCD

Japan Science and Technology Agency

- XcalableACC reference compiler is available on https://github.com/XcalableMP/XACC-Specification
- GPU accelerated Hartree-Fock matrix routine will be published as OpenFMO contributed code on March 2018.
- OpenCL enabled optical interconnection BSP and Verilog HDL library for Intel Arria10 FPGA will be published on March 2018.

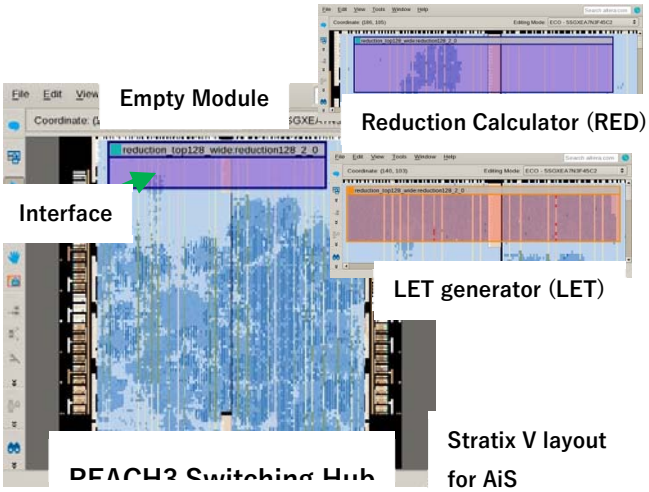
PEACH3: PEACH with PCIe gen.3

The bandwidth of GPU-GPU communication is about 3 times of MPI/Infiniband and 25% larger than that of PEACH2 for 32KB data. The performance of the BFS (Breadth First Search) of Graph500 was 1.16 times better than that of PEACH2 and 1.3 times better than that with MIP/Infiniband.



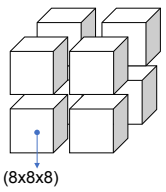
AiS on PEACH3

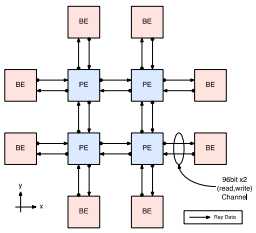
AiS framework for the reduction calculator (RED) and Locally Essential Tree (LET) generator for an astrophysics application is implemented on PEACH3. The former achieved 45x and the latter achieved 7.2x performance improvement including the time for data exchange. Partial reconfiguration is introduced to separate the switching hub and accelerators as shown in the below layout.



ART: Space Radiative Transfer on FPGA

ART (Authentic Radiation Transfer) is one of algorithms used in ARGOT (Accelerated Radiative transfer on Grids using Oct-Tree) program. ARGOT solves space radiative transfer problem and has been developed in Center for Computational Sciences in University of Tsukuba.
ART is a dominant part of ARGOT program and dominates more than 90% of its computation time. We selected ART method to demonstrate the practical use of AiS concept. Before implementing network functionality in an FPGA, we have been optimizing ART algorithm on FPGA using OpenCL HLS (High Level Synthesis). OpenCL enables us to program FPGA without HDL (Hardware Description Language) for high productivity and easiness of application coding. Since our implementation is work in progress, problem size which FPGA can solve is limited by the size of BRAM in an FPGA chip. We apply 16^3 problem size for performance evaluation because it is the maximum size where FPGA can solve. It will be expanded to use DDR memory very soon. We will combine this computation part by OpenCL with the networking module to realize AiS system.





CPU	Intel Xeon E5-2660 v4 × 2
GPU	NVIDIA Tesla P100
FPGA	BittWare A10PL4 (Intel Arria10 GX 1150)
CPU Compiler	gcc 4.8.5
GPU Compiler	CUDA 8.0
FPGA Compiler	Quartus Prime Pro 16.1.2, Intel OpenCL SDK for FPGA

Implementation	Performance [M mesh/sec]	Speedup v.s. CPU
CPU (14 cores)	117.49	1.00
GPU (P100 x1)	105.28	0.90
FPGA (at 236.11MHz)	1714.97	14.60

*mesh/sec means how many meshes are traced by rays in a second

Project information:
Project leader: Taisuke Boku (University of Tsukuba)
Project members: Hideharu Amano (Keio University), Hitoshi Murai (AICS, RIKEN) and Masayuki Umemura (University of Tsukuba)
Contact: **taisuke@cs.tsukuba.ac.jp**