REDUCTION CALCULATOR IN AN FPGA BASED SWITCHING HUB FOR HIGH PERFORMANCE CLUSTERS

Takuya Kuhara†, Chiharu Tsuruta†, Toshihiro Hanawa††, Hideharu Amano†

†Dept. of ICS, Keio University, Yokohama Japan
††The University of Tokyo, Kashiwa Japan
email: †hlab_ac-crest@am.ics.keio.ac.jp, ††hanawa@cc.u-tokyo.ac.jp

ABSTRACT

Unused logic in the field-programmable gate array (FPGA) for the switching hub is one potential resource to accelerate the computation of data exchanged through the hub. However, for large scale scientific computation, it is difficult to implement such an accelerator on the FPGA used in high performance computers. Here, a reduction calculator for executing ARGOT (accelerated radiative transfer on grids using oct-tree) to solve the radiative transfer equation used for simulation of astronomical objects is implemented on the FPGA of PEACH2 (PCI Express Adaptive Communication Hub ver2), a low latency switching hub for high performance GPU (graphics processor unit) clusters. The implemented reduction calculator uses a pipelined tree of adders and works with a 150-MHz clock without affecting the switching hub functions. Use of the DMA (direct memory access) transfer with descriptors made it possible to improve the performance of CPU excution by a maximum of about 45 times in a real system.

1. INTRODUCTION

Acceleration of computation using both a field-programmable gate array (FPGA) and a graphics processor unit (GPU) has not been very successful, particularly in scientific computation. In this field, because the performance of GPUs is much better than that of FPGAs, it is difficult to appropriately divide and distribute a job to the GPUs and FPGAs. However, FPGA-based switching hubs are often used in various networking layers, including in interconnections in supercomputing clusters. Unused logic in such an FPGA for a switching hub can be used to accelerate the computation executed in a cluster. This approach is especially advantageous when the processed data need to be exchanged between GPUs attached to nodes and transferred through the switching hub. In other words, on-the-fly processing during data transfer is possible.

PEACH2 (PCI Express Adaptive Communication Hub Ver.2), is such a switching hub developed for low latency direct communication between accelerators through a PCIe standard I/O bus based on the concept of tightly coupled accelerators (TCA) architecture [1][2].

To utilize free logics, a reduction calculator is implemented with unused logic of PEACH2. Reduction calculation is a common computation step used in various matrix/vector computations. Data transfer is necessary because all data must be transferred to a single node. Executing a reduction calculation during the data transfer makes it possible to reduce the task of the CPUs and GPUs without any additional cost. In this paper, we focus on an astrophysics application and off-load the reduction calculation on PEACH2.

This study was supported by the JST/ CREST program entitled " Research and Development on Unified Environment of Accelerated Computing and Inter- connection for Post-Petascale Era " in the research area of " Development of System Software Technologies for post- Peta Scale High Performance Computing. "

2. TCA/PEACH2

2.1. HA-PACS/TCA

HA-PACS/TCA (Highly Accelerated Parallel Advanced System for Computational Sciences/TCA) is a testbed system developed for removing communication bottlenecks by using direct PCIe packet transfer between GPUs across nodes [2][1]. In usual, three steps are necessary when we communicate between GPUs across nodes in multi-GPU clusters. First, we copy the data from the GPU (node A) memory to the host CPU (node A) memory. Second, the data are transferred from the host CPU (node A) to another host CPU (node B). Finally, the data are copied from the host CPU (node B) memory to the GPU (node B) memory. Communication between the CPU and GPU, and between CPUs across multiple nodes, requires large latency. As shown in Figure 1, the TCA architecture provides an FPGA board called PEACH2 in each node. The TCA architecture enables direct communication between nodes through PCIe, which is commonly used as a bus in a node. That is, communication between GPUs can be done through PEACH2 without having to pass through any host CPUs.

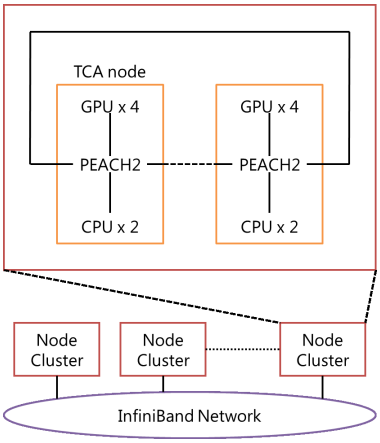

Fig. 1. Overview of HA-PACS/TCA design.

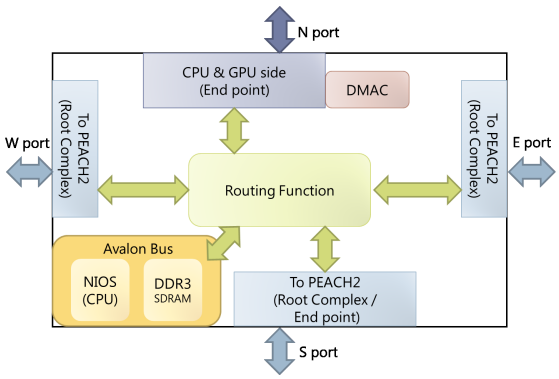

Fig. 2. Block diagram of PEACH.

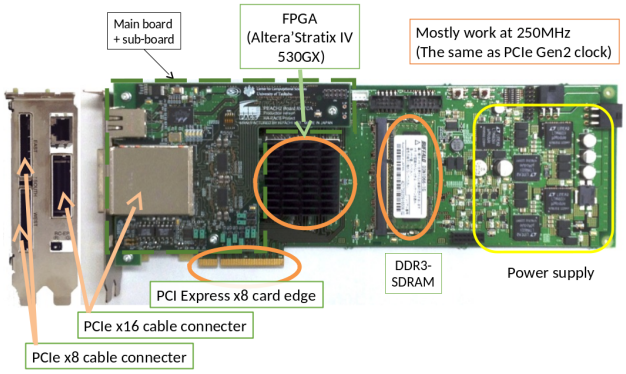

Fig. 3. Photograph of PEACH2.

2.2. PEACH2

Figure 2 illustrates a block diagram of a PEACH2 chip implemented on an Altera Stratix IV FPGA [3]. The main role of PEACH2 is to extend the PCIe, which is commonly used only as an I/O network, to connect multiple nodes of a cluster. PEACH2 has four ports: N, E, W, and S. Port N, an endpoint for PCIe Gen2 x8, is pushed into the PCIe connector on the host CPU board. Port E, an endpoint with Gen2 x8, and Port W, a root complex with Gen2 x8, are used for interconnection between nodes. Port S is a selectable PCIe Gen2 x16 port and is used to connect two ring networks formed by Ports W and E. The routing function embedded in the FPGA selects the destination port simply by checking the destination address of the PCIe packet on a single 512-GByte shared address space. The routing function provided in PEACH2 has control registers for the address mask and the lower and upper bounds. The destination port is statically decided by checking the address with the address mask. On the PEACH2, a memory access to a remote node is restricted to a memory write request. Memory read is difficult to implement efficiently, but the proxy write mechanism can achieve the same effect by using driver support. A DMAC (DMA controller) supports sophisticated block data transfer in the address space.

PEACH2 was implemented with Altera's Stratix-IV FPGA board and works with a 250-MHz system clock. A 512-MByte DDR3 SDRAM is provided on the board. The logic utilization is just 22%. A softcore NIOS CPU running at a 150-MHz clock speed is provided for the management of the PEACH chip. Figure 3 shows a photo of the PEACH2 board. Port N is implemented as a card edge, while cable connectors are used for other ports. A daughter board is used to extend Port S to x16 ports.

3. TARGET APPLICATION: RADIATIVE TRANSFER EQUATION

3.1. ARGOT

Radiation transfer (RT) is one of the most important physical processes of energy transfer in the form of electromagnetic wave, and of crucial importance in the formation of astronomical objects such as galaxies and stars. ARGOT[4][5] (accelerated radiative transfer on grids using oct-tree) was proposed as a high speed computation scheme for solving RT equations on 3-dimensional uniform cartesian mesh grids efficiently by using a tree data structure of the spatial distribution of radiating sources, and is parallelized by decomposing the simulation volume evenly into rectangular subvolumes with equal volumes along the cartesian axes. The parallel processing of ARGOT on multi-GPU systems is efficiently executed with the steps shown in Figure 4. Here, we assume a GPU cluster in which each node has a CPU and multiple GPUs.

Figure 4 includes the communication that occurs between the GPUs and the CPU in the reduction calculation. A bottleneck of the total parallel processing can occur since data transfer, synchronization, and serialized computation are required. If the cluster is connected to PEACH2, the reduc-

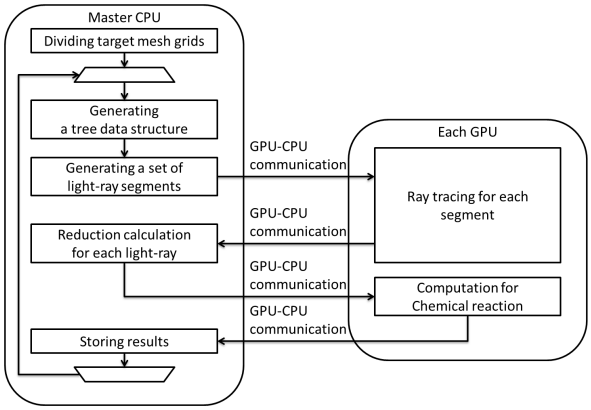

Fig. 4. Flow diagram of ARGOT

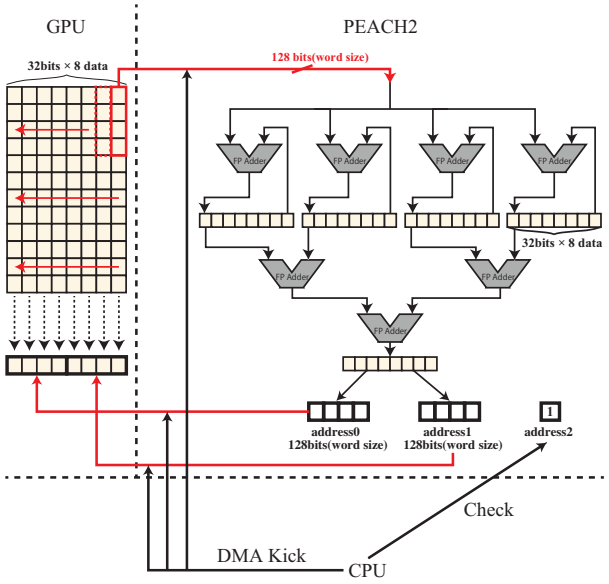

Fig. 5. Block diagram of the reduction calculator

tion calculation can be done in PEACH2, and the results can be transferred back to a GPU without sending them to the CPU. After the reduction calculation is carried out in each PEACH2, the final results can be computed through direct communication between PEACH2 boards, and sent back to a single CPU.

4. REDUCTION CALCULATOR IN PEACH2

4.1. Reduction Calculator Implementation

In PEACH2, the main switching hub functions include the PCIe interface, routing functions, multiplexers for packet switching, and the DMAC running at a 250-MHz clock speed. The reduction calculator must be implemented so as not to effect the operation of this part, so we implemented our logic as a functional module attached to the Avalon bus, as shown in Figure 2. The working memories inside the FPGA, the DDR3 SDRAM interface, and the Nios soft processor are connected to the bus, and this part works at a 150-MHz clock speed for easier implementation. The memory can be mapped to the same address space as the GPU and CPU memory modules connected to PEACH2, and writing to this memory area is done by sending data to the input registers of the reduction calculator. Since the bus width is 128 bits, the fundamental data size in the implementation is set to be 128 bits.

A diagram of the implemented reduction calculator is shown in Figure 5. It consists of a tree of seven pipelined 32-bit floating point adders that use Altera's Megafunction IP core. In ARGOT, the data for eight particles are independently summed up. Thus, each register has eight 32-bit fields for each particle. As shown in the figure, four sets of 32-bit data in the GPU memory are packed into 128 bits and transferred to a certain address mapped to PEACH2. They are added to the value in the registers in parallel and written back to the registers. Eight 32-bit results are mapped to two 128-bit registers and transferred in two cycles when the flag is ready. Although checking of the flag and triggering the DMA transfer of the results data are managed by the host CPU in the current implementation, the calculator can do it in future implementations. All data writing into a PEACH2 address is done by using DMA transfer controlled by PEACH2.

5. EVALUATION

5.1. Resource Utilization

Table 1 indicates the resource utilization before and after implementing the reduction calculator. It is clear that the designed reduction calculator requires only a small amount of hardware. Even with the combinational ALUTs (adaptive look-up tables), the total amount of hardware only increases by 1.3%. This shows that more complicated accelerator modules can be implemented in PEACH2 with a similar method.

5.2. Performance Improvement

The reduction calculator was implemented in an experimental cluster system that had the similar specifications in a system in the Center for Computational Science, University of Tsukuba[1]. Since HA-PACS itself has a lot of users, we conducted measurements using a minimal system with two GPUs and a host CPU. We used a high speed descriptor mode with working RAM, and we increased the data size to be computed from 1KB (32 particles) to 16MB (524288 particles). The time of 10 iterative executions was measured

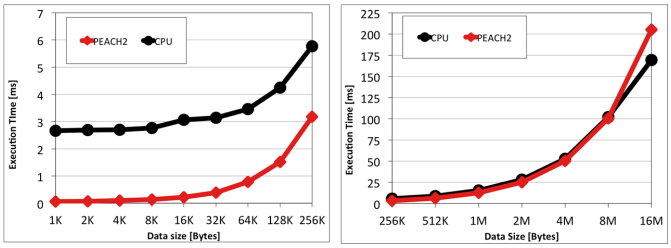

Fig. 6. Execution Time vs. Data Size (Left: 1K-256K/Right: 256K-16M)

Table 1. Resource Utilization of PEACH2 without/with reduction calculator

Tool	Quartus II 13.1		
Family	Stratix IV		
Device	EP4SGX290NF45C2		
ALUTs/ bits	Original PEACH2	PEACH2 with the reduc. calc.	Diff.(%)
Combinatorial	61,172	66,772	1.3
Mem. ALUTs	13,565	13,586	0.01
Registers	80,118	84,532	1.03
Block memory	2,717,248	2,718,272	0.005

with a timer and compared to the case when the reduction calculation was done in the host CPU as in the original ARGOT program.

The left figure in Figure 6 plots the execution time for small data sizes from 1K to 256K. Note that the horizontal axis uses a logarithmic scale. When the data size is small, the reduction calculator in PEACH2 achieves much better performance than the execution in the CPU. The performance was improved by about 45 times at maximum. However, the benefit decreases as the data size increases.

On the other hand, the right figure in Figure 6 plots the execution time for large data sizes (256KB to 16MB). For data sizes up to 4MB, the performance using the reduction calculator was almost the same as that of the host CPU, and for larger data sizes, the performance of the CPU was better. This is due to the difference in PCIe bandwidth. The direct data transfer between the CPU and GPUs can use PCIe Gen2 x16, whereas the data transfer between PEACH2 and the GPUs uses PCIe Gen2 x8.

6. RELATED WORK

Off-loading jobs to network interface has been researched recently, for big data processing. Some of the functions for database processing have been off-loaded to network interface FPGAs [6]. Most of them are off-loaded to a network interface FPGA, and the target application is networking or database processing. To the best of our knowledge, there has been no research on off-loading part of a scientific computation into a switching hub FPGA except Mellanox Fabric Collective Accelerator (FCA)[7]. It is a simple acceleration module for MPI collective communication, and not a reconfigurable module dedicated for the target application. It is difficult to compare our reduction calculator with FCA, since no report has been published yet. The reduction computation in a switching hub will become commn way in the near-future supercompting.

7. CONCLUSION

A reduction calculator for executing ARGOT to solve the radiative transfer equation used for simulation of astronomical objects was implemented on the FPGA of PEACH2 (PCI Express Adaptive Communication Hub ver2), a low latency switching hub for high performance GPU clusters. The implemented reduction calculator uses a pipelined tree of adders and works with a 150-MHz clock without affecting the switching hub functions. By using the DMA transfer with descriptors, we improved the performance of CPU execution by a maximum of 45 times in a real system. The implementation of the reduction calculator for multi-node systems is our future work.

8. REFERENCES

[1] Center for Computational Sciences, University of Tsukuba, http://www.ccs.tsukuba.ac.jp/.

[2] T. Hanawa, Y. Kodama, T. Boku, and M. Sato, "Interconnect for tightly coupled accelerators architecture," "IEEE 21st Annual Sympsium on High-Performance Interconnects (HOT Interconnects 21)", 2013.

[3] Yuetsu Kodama, Toshihiro Hanawa, Taisuke Boku, Mitsuhisa Sato, "PEACH2: An FPGA-based PCIe network device for Tightly Coupled Accelerators," in *HEART2014*, June 2014.

[4] T.Okamoto, K.Yoshikawa, M.Umemura, "ARGOT:Accelerated radiative transfer on grids using oct-tree," vol. 419. Blackwell Publishing Ltd, 2012, pp. 1365–1378.

[5] C.Scannapieco, and et.al, "The Aquila comparison project: the effects of feedback and numerical methods on simulations of galaxy formation," vol. 423. Blackwell Publishing Ltd, 2012, pp. 1726–1749.

[6] E.S.Fukuda, H.Inoue, T.Takenaka, D.Kim, T.Sadahira, T.Asai, and M.Motomura, "Caching Memcached at Reconfigurable Network Interface ," in *Proceedings of the International Conference on Field Programmable Logic and Application (FPL'14)*, Sept 2014.

[7] Mellanox, http://www.mellanox.com/products/fca/.