Off-loading LET generator in PEACH2 : A switching hub for high performance GPU clusters

Chiharu Tsuruta, Yohei Miki**, Takuya Kuhara*, Takuji Mitsuishi*, Naru Sugimoto*, Hideharu Amano*.* **Keio Univ. , ** Univ. of Tsukuba*

Introduction

On-the-fly computation in the field-programmable gate array(FPGA) used for the switching hub is one potential way 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 hardware local essential tree (LET) generator used in the N-body simulation is implemented on the FPGA of PEACH2 (PCI Express Adaptive Communication Hub ver2), a low latency switching hub for high performance GPU clusters.

What is PEACH2 ?

PEACH2 (PCI Express Adaptive Communication Hub Ver.2) has been 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]. Hardwired logic of the FPGA in the PEACH2 chip is used for the packet buffer, switch, routing function, and DMA controller, just like any other switching hub. However, there is room to implement other functions specialized for applications executed in the system.

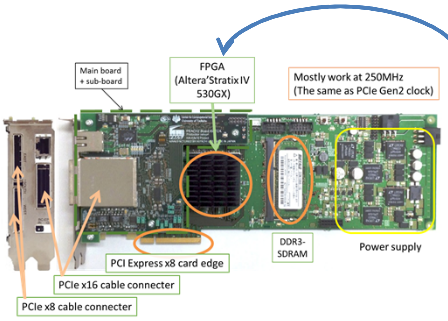


Fig 1 Photo of PEACH2

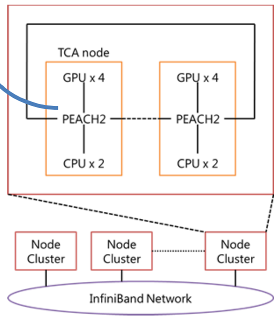


Fig 2 TCA system

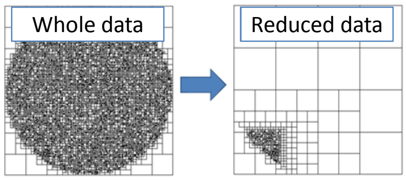
Target application : make LET

Making LET is the process of the part of N-body simulation.
N-body simulation is commonly used to investigate the structure formation and evolution history of galaxies by solving the following equation.

$$\boldsymbol{a}_i = \sum_{j=0 j\neq i}^{N-1} \frac{G m_j (\boldsymbol{x}_j - \boldsymbol{x}_i)}{(|\boldsymbol{x}_j - \boldsymbol{x}_i|^2 + \epsilon^2)^{\frac{3}{2}}}$$

The tree method is adopted by a lot of N-body code and has been proposed to reduce the computational cost to O[*Nlog*(*N*)] by using the multipole expansion technique [2].
When a tree method is executed in a GPU cluster such as HA-PACS, the tree is divided and distributed to each GPU for parallel processing.
So, we adopt locally essential tree (LET) [3].
The way of making LET proposed by Warren & Salmon is an algorithm to reduce the amount of communication among processes.
LET is a tree with a pruned data structure that contain the part of necessary data to calculate the gravitational force.

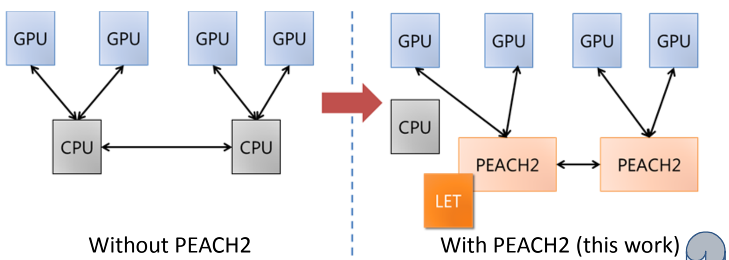
An example of data by using LET
Figs. taken from Ref[3]



LET generator

The role of LET generator

In the N-body simulation that are underway currently implemented, the data has to be transferred from the GPUs to the CPU and then the result has to be transferred back to the GPUs again. In contrast, in the PEACH2 implementation, the computation time of the LET generator is completely overlapped with the data transfer since it is on-the-fly computation.



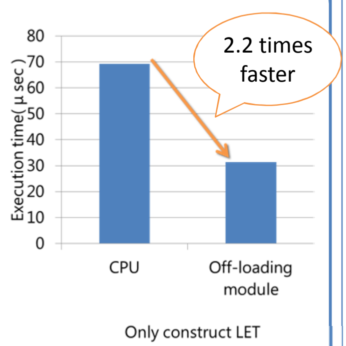
Advantages of LET generator on PEACH2

1. LET generation can be done on-the-fly during data transfer.
2. Data transfer between CPU and GPU for generating LET is removed.
3. CPU and GPU is free when LET is made in PEACH2

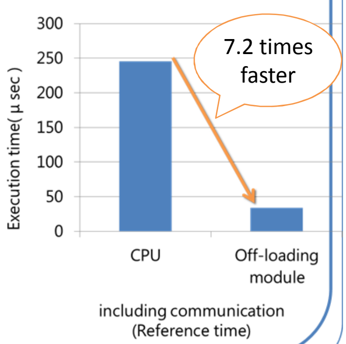
Evaluation of LET generator

The LET generator was designed in Verilog HDL and synthesized with Quartus II 13.1.

Right figure shows the execution time of the LET generator and CPU. The clock cycles in the LET generator are counted using RTL simulation. The number of particle is 4096. It appears that the LET generator works 2.2 times faster than the CPU.



This figure shows the execution time including communication between GPU and CPU or PEACH2.
The off-loading module time is the sum of LET generator and the time of the communication delay in the past.
We can estimate that the LET generator in PEACH2 achieves a 7.2 times faster execution than the CPU.



Reference

[1] Yuetsu Kodama et al. PEACH2: An FPGA-based PCIe network device for Tightly Coupled Accelerators. In *HEART2014*, June 2014.
[2] Barres and Hut. A hierarchical O(*NlogN*) force-calculation algorithm. In *Nature*, pages 446–449, December 1986.
[3] Warren and Salmon. Astrophysical N-body Simulations Using Hierarchical Tree Data Structures. In *Proceedings of the 1992ACM/IEEE Conference on Supercomputing*, Sep1992.