GPGPU for rapidly processing of Gravitational Wave Radiometry and Prospects for Virgo-cluster Hotspot

Kazuyuki Tanaka, Yuta Okada, Nobuyuki Kanda, Takahiro Yamamoto, Hirotaka Yuzurihara, KAGRA collaboration
Department of Physics, Osaka City University, Osaka, Japan
e-mail : ktanaka@yukimura.hep.osaka-cu.ac.jp

The calculation for Gravitational Wave analysis will take much cost, e.g. for huge number of templates in matched filter for compact star binary, sweeping many pixels and frequency band for the radiometry and so on.Then, to make those calculation rapidly, we try to employ GPGPU(General-Purpose computing on Graphics Processing Units) to the gravitational wave analysis. As the first instance, we are trying to apply GPGPU to Gravitational Wave Radiometry.

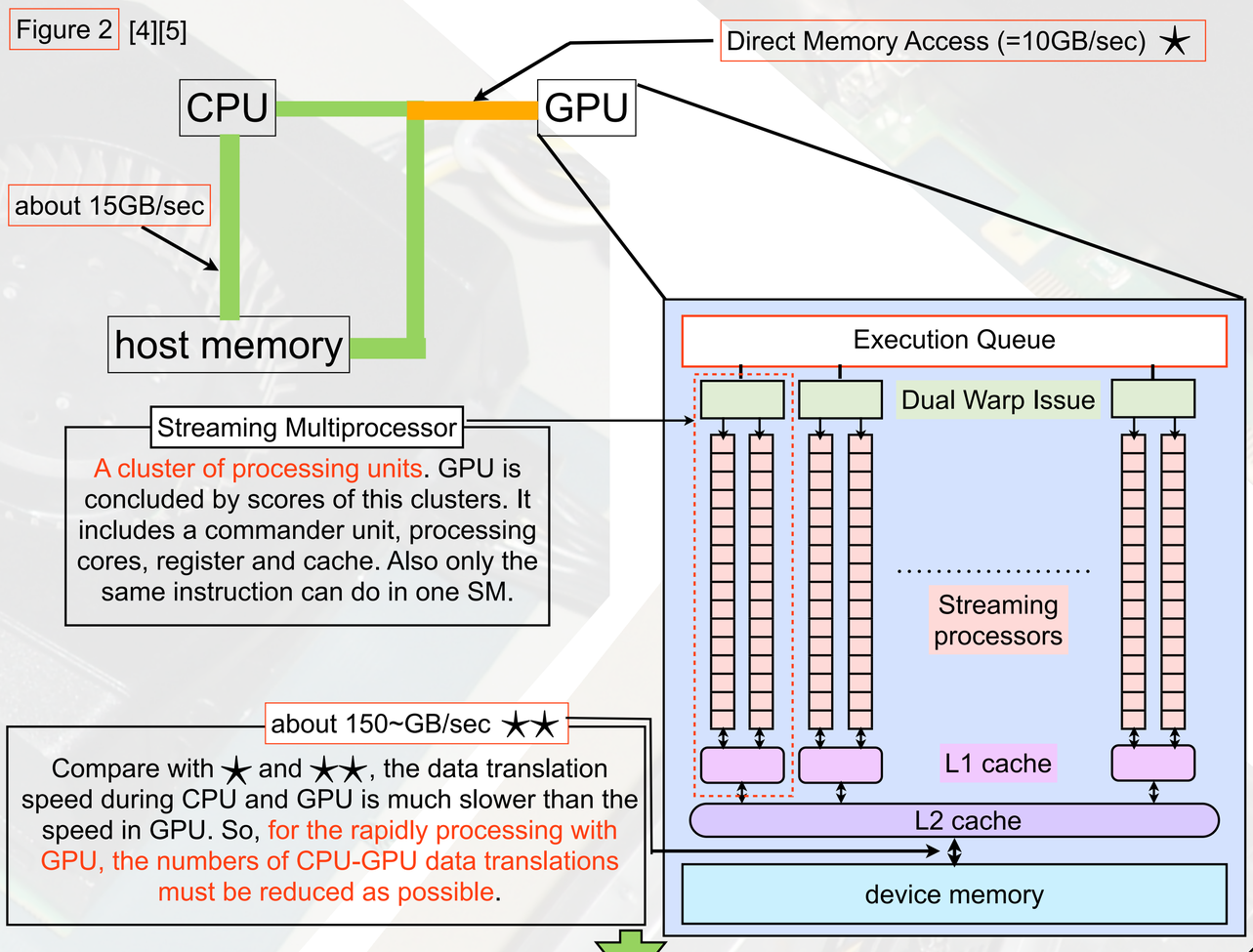
Gravitational Wave Radiometry is an analysis method, using two or more detectors and take that cross-correlation for a long time to see, for example stochastic gravitational wave background, continuous gravitational wave which comes regularly from particular direction.

Here, we show why GPGPU is useful for Gravitational Wave Radiometry, and what we can say is GPGPU has more advantageous if the frequency resolution become higher.

The construction of GPU

The most important feature of GPU here is the large number of cores of them. Although the latest CPU has yet less than 12 cores, GPU has some hundreds. (For example, Tesla2070 which used in our computer has 448 cores.) So in the case, calculation can be done in parallel, calculating on GPU will be much faster than on CPU.

Figure 2 [4][5]



Streaming Multiprocessor

A cluster of processing units. GPU is concluded by scores of this clusters. It includes a commander unit, processing cores, register and cache. Also only the same instruction can do in one SM.

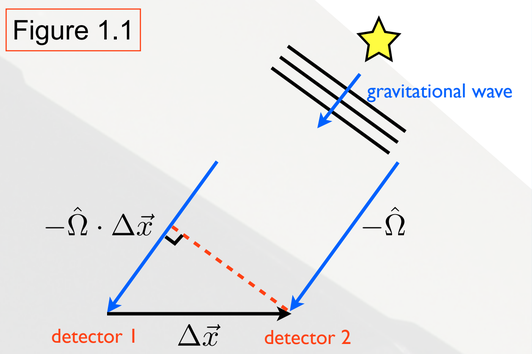
about 150~GB/sec ★★

Compare with ★ and ★★, the data translation speed during CPU and GPU is much slower than the speed in GPU. So, for the rapidly processing with GPU, the numbers of CPU-GPU data translations must be reduced as possible.

Gravitational Wave Radiometry [1][2][3]

Gravitational Wave Radiometry is an analysis method, using two or more detectors and take that cross-correlation for a long time to see, for example stochastic gravitational wave background, continuous gravitational wave which comes regularly from particular direction. As in the Figure 1.1, there will be time delay during the detectors. Then we can get the source directions by correct the time delay and survey all around the celestial sphere.

Figure 1.1



The statistic of Gravitational Wave Radiometry in the time t is

$$\Delta S(t,\hat{\Omega}) = \int_{-\infty}^{\infty} df\, \tilde{s}_1^*(f;t)\tilde{s}_2(f;t)\tilde{Q}(f,\hat{\Omega};t)$$

$\tilde{s_{1,2}}(f;t)$: the observation data of detector 1,2

$-\hat{\Omega}$:velocity vector

$\Delta\vec{x}$:the distance during the detectors

time delay : $\Delta t = \dfrac{\hat{\Omega}\cdot\Delta x}{c}$

$\tilde{Q}(f,\hat{\Omega};t)$ is called Radiometry Filter.

$$\tilde{Q}(f,\hat{\Omega};t) = \lambda\frac{\gamma^*(f,\hat{\Omega})H(f)}{P_1(f)P_2(f)}$$

$$\gamma(f,\hat{\Omega}) = \sum_{A=+,\times} F_1^A F_2^A e^{2\pi f i\hat{\Omega}\cdot\Delta\vec{x}/c}$$: correction term

$H(f)$: power spectrum of source

P_1, P_2 : noise spectrum of detectors

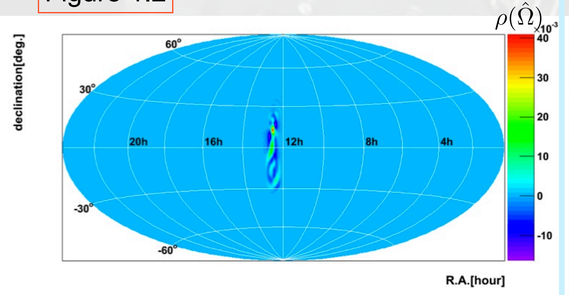
Then the Signal to Noise Ratio $\rho(\hat{\Omega})$ is

$$\rho(\hat{\Omega}) = \frac{\mu(\hat{\Omega})}{\sigma(\hat{\Omega})}$$

$$\mu = \langle \Delta S \rangle, \sigma = \left\langle [\Delta S - \mu]^2 \right\rangle$$

Figure 1.2 is a result of when we put a point source into the Radiometry Filter.

Figure 1.2



Put a point source on the place of Virgo-cluster, calculate $\rho(\hat{\Omega})$ and mapping it. Here, no considering about the noise.

Gravitational Wave Radiometry Flow with GPGPU

Figure 3.1 is a flow of Gravitational Wave Radiometry with GPGPU. (GPGPU = General Purpose computing on Graphics Processing Units)

In Gravitational Wave Radiometry, we separate the data into short terms(called chunk). Here, we have no use separate the data into longer than about 5 minutes, because of the relation with the resolution on the sky and earth rotation. After the separation, We translate the data from host memory to GPU's memory. The translation speed is not enough to the calculation speed of GPU, so for the rapidly processing, translate as long data as the GPU's memory can accept in one time is required. Then doing FFT to get the data in frequency domain. After that, through the Radiometry Filter and sum the results for all observing time. Then sweep around the sky to complete the mapping on the celestial sphere.

But in this phase, the point sources are spread out some scores of pixels. So to get clean map, which the spread bring into focus some pixels, we deconvolute it.

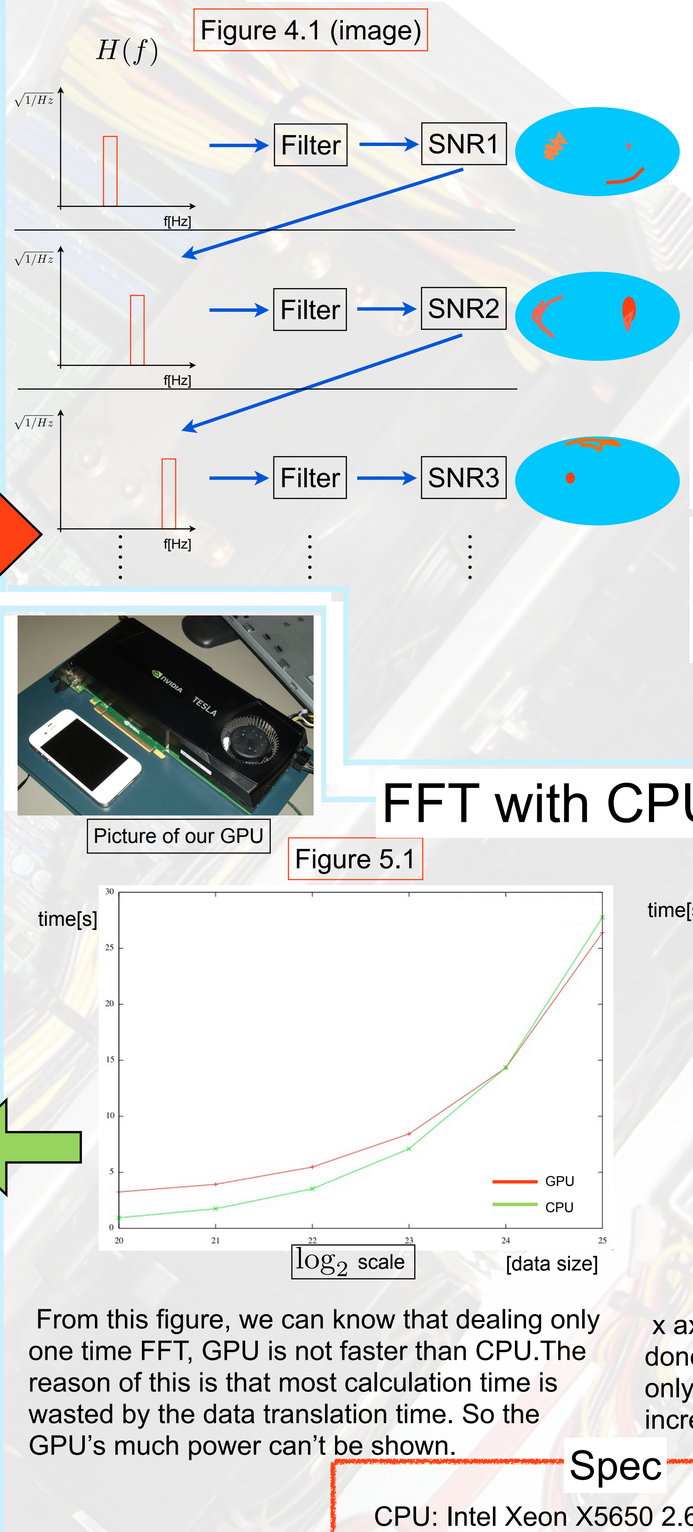
GPU is very good at the calculations which can do in parallel, on the other hand not good at troublesome calculations.(like programs which has many [if] process.) For these reasons, ☐ part will be calculated rapidly by GPU. Especially, the deconvolution part(Figure 3.2) will be able to make much faster than with CPU.

Figure 3.1

Figure 3.2 (deconvolution)

Sweep the frequency window for various band mapping

Figure 4.1 (image)

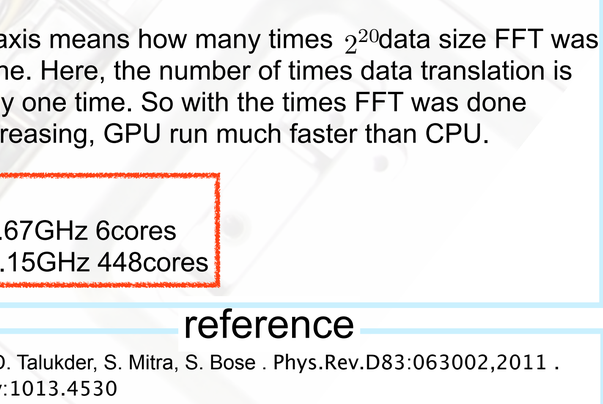


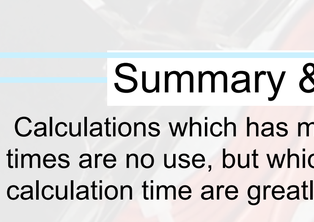
In Gravitational Wave Radiometry, we don't have to know the exactly spectrum of the source.

Though if we have much calculation power, we will be able to map the sky with particular frequency band, like in Figure4.1(image).

Also, we can decide more specific spectrum of sources in the same way.(Figure4.2)

Figure 4.2





Picture of our GPU

FFT with CPU and GPU

Figure 5.1

Figure 5.2

From this figure, we can know that dealing only one time FFT, GPU is not faster than CPU.The reason of this is that most calculation time is wasted by the data translation time. So the GPU's much power can't be shown.

x axis means how many times 2^{20}data size FFT was done. Here, the number of times data translation is only one time. So with the times FFT was done increasing, GPU run much faster than CPU.

Spec

CPU: Intel Xeon X5650 2.67GHz 6cores
GPU : Nvidia Tesla C2075 1.15GHz 448cores

Summary & Future

Calculations which has many data translation times are no use, but which dominated by calculation time are greatly good for GPU.

We'll make these analysis method can be dealt with GPU.

Applying GPGPU to CBC search, Burst, etc. (GPGPU may be good for CBC.)

reference

[1] D. Talukder, S. Mitra, S. Bose . Phys.Rev.D83:063002,2011 . arXiv:1013.4530
[2] S. Mitra, S. Dhurandhar, T. Souradeep, A. Lazzarini, V. Mandic, S. Bose, S. Ballmer . Phys.Rev.D77:042002,2008 . arXiv:0708.2728
[3] Y. Okada, N. Kanda, S. Dhurandhar, H. Tagoshi, H. Takahashi . arXiv:1112.3090
[4]http://www.softek.co.jp/SPG/Pgi/TIPS/public/accel/gpu-accel2.html
[5]はじめてのCUDAプログラミング : author; T. Aoki, A. Nukada : 工学社