GPU-accelerated Searches for Gravitational Waves from Compact Object Binary Coalescence

Linqing Wen (UWA/Caltech)

on behalf of **Yuan Liu, Zhihui Du (Tshinghua U., China), Kipp Cannon (CITA), Shin Kee Chung, Shaun Hooper, David Blair, Amitava Datta (UWA)**

Objectives

To provide gravitational wave triggers within tens of seconds of the event to allow prompt electromagnetic follow up observations of Compact Binary Coalescence (CBC) event.

Astrophysical Motivation

- Transient electromagnetic emission from GW sources could happen within tens to hundreds of seconds of GW events
- Example: prompt Optical/X-ray emission of short-hard GRBs from NS-NS merger

Table 1. Time window and collimation of GW emission and EM afterglows of GRBs

Frequency Band	Typical or Predicted Observing Time window (relative to Gamma-ray trigger)	Collimation
GWs	- 1000 s (predicted for NS-NS coalescence for advanced detectors)	roughly isotropic
X-rays (prompt)	tens–hundreds seconds [6,13]	likely beamed
X-rays (afterglow)	hundreds of seconds–tens of hrs[13]	likely beamed
Optical (prompt)	tens–hundreds of seconds [14,15,13]	likely beamed
Optical (afterglow)	hundreds of seconds–days [16,17]	likely beamed
Radio (prompt)	hrs-days [18,19,20,21,22] (predicted)	beamed (?)
Radio (afterglow)	wks–yrs[23]	likely isotropic

- GWs will provide essential triggers to alert EM telescope to catch these event in the post-SWIFT era or when gamm-rays are beamed away

Graphics Processing Unit (GPU):

• Suitable for single instruction multiple thread model; Cost effective: (~Tflops/$400); easy to program, (C-like programming language); popular desktop solution; sizable existing computing library

GPU-accelerated time-domain search

- With proper choice of coefficients, simplest Infinite Impulse Response (IIR) filtering

$$y_k = a_1 y_{k-1} + b_0 x_k$$

can be equivalent to filtering data with damped sinusoid

- The sum of such IIR filters at various frequencies can retrieve 99% optimal matched filtering SNRs (Shaun Hooper's poster)

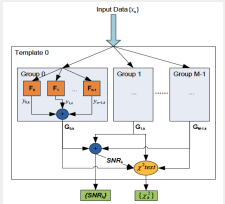


Figure 1: The SPIIR filtering mode integrated with χ^2 test

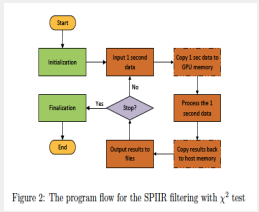


Figure 2: The program flow for the SPIIR filtering with χ^2 test

- Perfect for GPU processing. Each thread process one filter

- GPU vs CPU: x 50 speed –up

Table 3: The configurations for the experiments

Hardware	CPU	Intel Core i7 920 2.67GHz
	GPU	NVIDIA Geforce GTX 480
	Host Memory	6GB DDR3
Software	Operating System	Fedora 10 64-bit
	CUDA Version	3.0
	Host Compiler	gcc 4.3.2

Table 4: The speedup achieved by our methods over *CPU-Float*

Method	Overall Speedup Factor
Initial	5.7
Parallel Sum Reduction	14
Batched Calculation	21
Avoid Bank Conflicts	24
Texture Memory	38
Optimal Occupancy	47

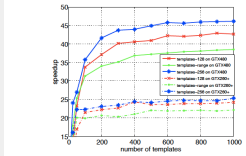


Figure 4: Speedup achieved by *GPU-Float* over single-threaded *CPU-FLOAT* on two different platforms. The platform with a GTX 260+ GPU (the last generation before Fermi) had a four-core Intel Core i5 750 CPU (2660MHz).

- **Single desktop real-time processing: for >5000 templates at 99% overlap for initial LIGO**

Table 5: The number of templates that could be processed in real time (means processing 1 second of data in 1 second)

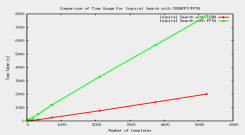
Templates	*CPU-Float*	*CPU-Double*	*GPU-Float*
templates-128	240	220	10670
templates-range	160	140	6280
templates-256	120	110	5650

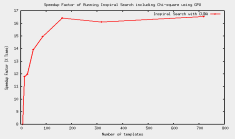
GPU-accelerated existing frequency domain pipeline

- The existing inspiral pipeline + chi-square test
- Implement CUDA FFT, checked into LAL
- In chi-square test :
  - Bundle multiple Fourier Transforms
  - data parallelism for sequential loop

Performance

- Geforce 8800 vs 2.5 GHz Intel quad 9300 CPU
- Inspiral search with realistic parameters: x 4
- Inspiral search+chi-square test: x 16





Summary

- Significant speed-up can be achieved with GPUs for inspiral search+chi-square test
- Implemented a new GPU-accelerated time-domain inspiral searches using IIR filterbank
- With GPUs, desktop real-time low-latency processing of inspiral search is possible even with advLIGO

(Chung, S. et al 2010 CQG)

Email: linqing.wen@uwa.edu.au