First Interim Report

Yanqi Gu

Title: cWB Optimization for GW Signal Waveform Reconstruction Stage

Abstract: In these days, I am focusing on getting a better understanding of the background of cWB pipeline. I am getting familiar with Caltech computing grid and the cWB code, profiling to find the hotspots of the pipeline and get CUDA set up for further optimization.

1. Background

September 14, 2015, marked the historical gravitational wave signal detection, GW150914 [1], by the Advanced Laser Interferometric Gravitational Wave Observatories (aLIGO). Two different types of searches independently confirmed the strong signals from binary black hole mergers. Then for the second time in history, on June 15, signals are detected again by aLIGO, and the third gravitational wave is caught on June 1,2017. We can see that in the future, the detection of GW will become daily events, and with more and more data collected, a new era of astronomy is underway.

Our work focuses on profiling and optimizing the runtime performance of a key routine in our research, the Coherent Wave Burst (cWB) pipeline, which first detected GW150914 and it will continue to play an important role in O3 [2]. The aim of this project is to understand and address causes for the time-consuming part in the network section of cWB, and port to execution on GPU and Many Integrated Core (MIC) architecture for acceleration. As the waveforms are constructed as a superposition of Wilson-Daubechies-Meyer wavelets, the function under our optimizing consideration computes the weight of each wavelet in the reconstructed signal. We intend to design a parallel algorithm for the function, and try to get much better performance after porting the code to GPUs or many integrated cores (MIC).

2. Motivation

Optimization is essential! In the cWB pipeline, many optimizations have been implemented between O1 and O2, and the profiling results show that no single function takes more than roughly 10% to 15% of the execution time [3]. However, the cWB reconstruction stage still proves to be very time consuming, since in every single run more than 200000 sky locations for each data sample need to be searched [2].

GPU/MIC acceleration has shown great potential for helping cWB pipeline run faster [4]. A large range of GPUs is available for testing, including Nvidia Tesla P40, Nvidia Tesla P100, Nvidia GTX 1080Ti, Intel Xeon Phi Knights Landing (KNL). Optimization efforts in O2 and O3 LIGO run are scheduled as follows:

1.Redesign the algorithmic part of the cWB reconstruction to improve the detection and reconstruction performance, while cut down the runtime.

2.Use of CUDA to parallelize the computing part that is CPU intensive.

3.Use of SSE vectors to reorganize the structure of the cWB data containers

3. Progress

1. Get familiar with machines in computing platforms Caltech head nodes provide, including Knights Landing, Skylake and Broadwell,
2. Check out the latest version of cWB software, build and run the cWB code on pcdev [3,8,10,11,12]. These hosts give access to GPUs and CPUs of interest.
3. Guided by the cWB authors, we choose the background production setup dataset, 01_12Sep19Jan_C01_BKG_LF_rMRA_WP10_bench1, as our testing dataset, to see the time cost ratio and find the fat target for optimization.
4. CUDA is set up and simple call for CUDA function is added into the initial cWB code.
5. We conduct performance analysis with different platforms and compilers on Caltech computing nodes. The results are shown in Figure 1 and Figure 2.

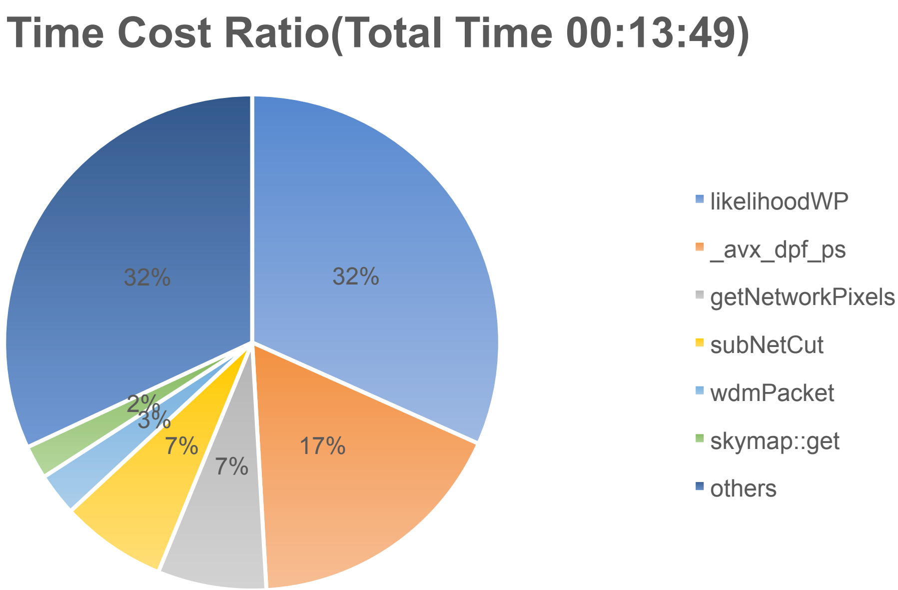

Figure 1. Built with gcc on Intel Xeon

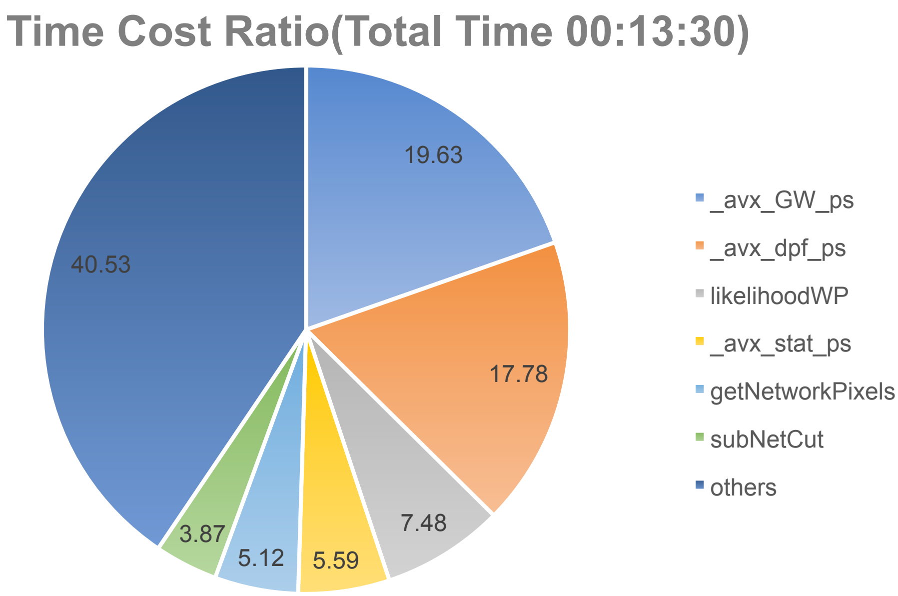

Figure 2. Built with icc on Intel Xeon Phi

Each figure lists top 6 time-consuming functions with specific compiler and platform. From Figure 1 we can see that likelihoodWP is the highest time-consuming routine for gcc, which costs 32% of total time, while from Figure 2, _avx_GW_ps is the top consumer when using Intel compiler, which costs 19.63% total time. This is an example of how profiling gives us areas to investigate in the coming weeks.

4. Opportunities, Problems and Challenges

In this project, we see several optimizing opportunities after profiling. We're going to design a new method for reconstruction stage of the pipeline, with parallelized algorithm and other CUDA optimization tools. Currently problems appear in run script editing, as the cWB pipeline is a big project and we need to be careful when setting the configurations. The challenging part will come in next few weeks, on algorithm designing and code deployment, which we'll see in the future.

Citation

[1] B.P.Abbott et al. Characterization of transient noise in Advanced LIGO relevant to gravitational wave signal GW150914.*arXiv:1602.03844(2016)*

[2] S.Klimenko et al. All-sky search: Performance and scaling.

[3] Peter Couvares. Progress on Data Analysis Computing Optimization

[4] Xiangyu Guo et al. Acceleration of low-latency gravitational wave searches using Maxwell-microarchitecture GPUs.*arXiv:1702.02256(2017)*