Fall 2017 ME759 Final Project Report
University of Wisconsin-Madison

# Stacking Low Frequency Earthquake Signal via GPU

Bin Guo

December 17th, 2017

## Abstract

In seismic study, low frequency earthquake (LFE) signals are processed to analysis non-regular earthquake events, which do not produce human sensible wave energy. Due to low energy release, LFE signals are usually buried in background noise. To improve signal to noise ratio (SNR), hundreds to thousands of repeated LFE signals are stacked to form a representative of a particular repeating LFE group. Current, two methods are widely used in stacking seismic signals, linear stacking and phase-weighted stacking (PWS). Linear stacking simply stacks LFE signals aligned in time axis. PWS analyzes phase information of seismic signal and stacks both amplitude and phase of LFE signals. In general, PWS will produce better SNR than traditional linear stacking. In this project, I implemented linear stacking and PWS with sequential and GPU parallel algorithms. The performance of two versions are compared. According to the tests, GPU improves the PWS efficiency while has minor effect on the linear stacking.

# 1. Introduction

Regular earthquakes are studied via processing their waveforms recorded on seismic stations. Two types of earthquake waves (P- and S-wave) are identified from recorded seismic waveform to locate earthquakes. If the seismic waveforms are with low signal to noise ratio (SNR) and P-wave arrival are not identifiable, S-wave will be picked to carry out earthquake relocation studies.

Low-frequency earthquakes (LFEs) were first reported in Japan benefiting from the construction of the Hi-net seismic array (Nishide et al., 2000; Katsumata and Kamaya, 2003). The Japan Meteorological Agency (JMA) discovered a type of seismic signal across many stations, which had no clear P-wave arrival but detectable S-wave arrivals. JMA located these events with only S-wave arrivals and found most of them distributed in the Tokai and Kii regions of Nankai Trough at depths ranging from 25 to 35 km. Signal with frequency lower than 10 Hz dominates in the LFE waveform. Due to low amplitude, the P-wave arrivals of LFE events are usually buried in background noise. Thus, only the S-wave arrival is used in LFE event locating.

## 1.1Linear stacking of LFE

In order to improve the SNR of LFE signals, previous studies (Shelly et al. 2006, 2010) used linear stacking method to suppress the background noise. Repeated LFEs are identified via cross-correlation through recorded signals. Then for each group of repeated LFEs, LFEs are aligned on the time axis and stacked. The final template LFE signal for each group has its amplitude be average over all repeated LFEs. Example of the linear stacked LFE signals are shown in Figure 1. With the improved SNR by linear stacking, P- and S-wave arrivals are able to be identified and used for later relocation studies.

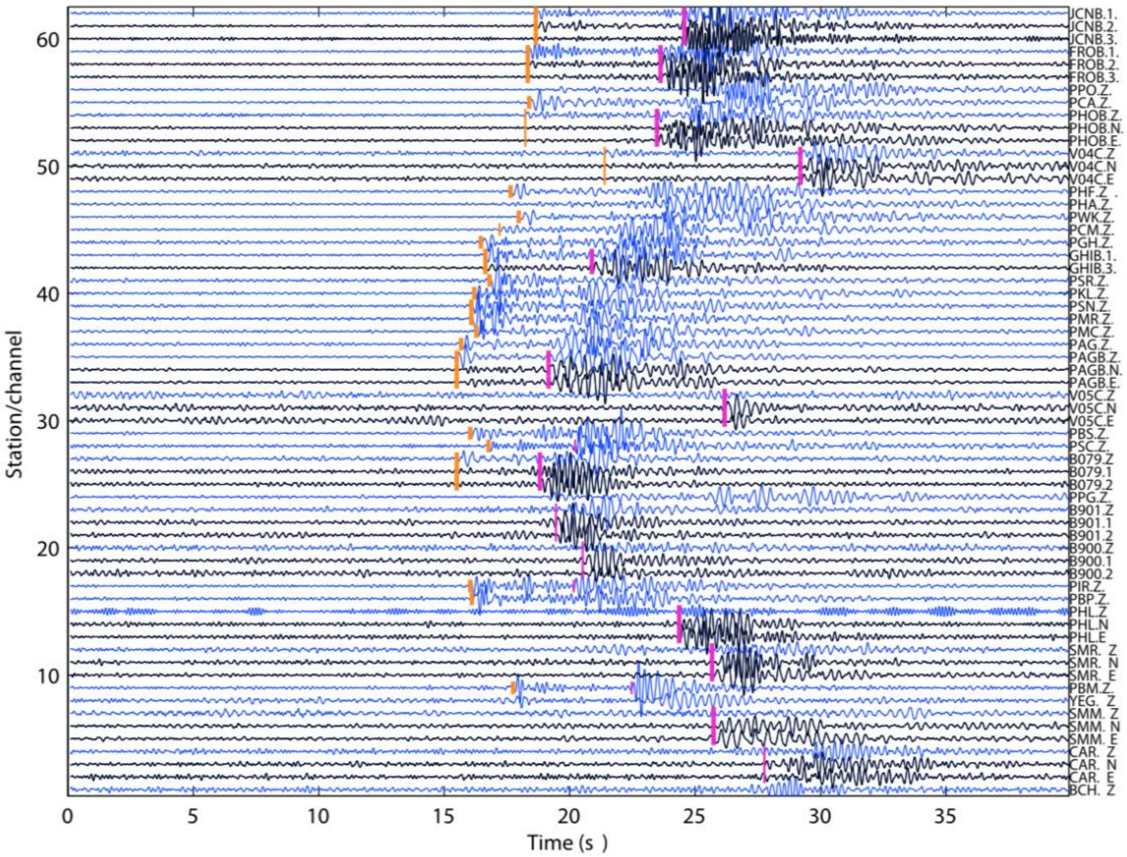



Figure 1. Sixty-two seismic records produced by stacking 400 similar events and their P and S arrival time picks. Orange lines are the identified P-wave arrivals. Purple lines are identified S-wave arrivals. Station names are labeled on the right Y-axis. Figure 2 from Shelly et al., 2010.

## 1.2 Phase-weighted stacking (PWS) for LFEs

Thurber et al. (2015) applied PWS method (Schimmel and Gallart, 2007) to LFE signals to improve the SNR further compared to linear stacking. In PWS, Hilbert transform is applied to each trace to get the instantaneous phase information of LFE signal. The analytic signals of each LFEs are combined by real part from amplitude of LFE signals and imagination part from Hilbert transformation of LFE signals. As shown in Figure 2, coherent instantaneous phase information shared by different signals will be enhanced after stacking while the random noise will be suppressed.

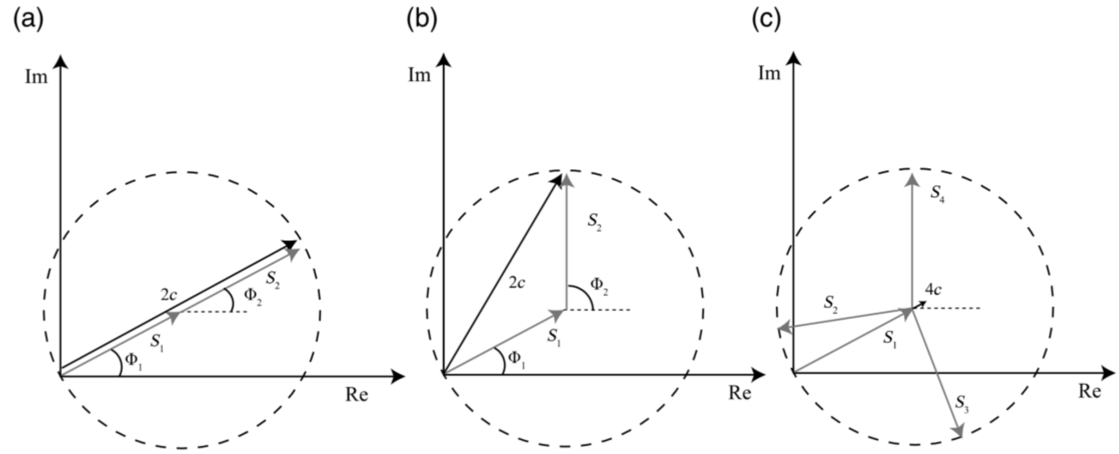



Figure 2. Example of adding different analytic LFE signals. The analytic LFEs will be stacked according to the real part with amplitudes and imagination part from Hilbert transformations. The stacking will enhance coherent instantaneous phase. Figure 1 from Thurber et al., 2015.

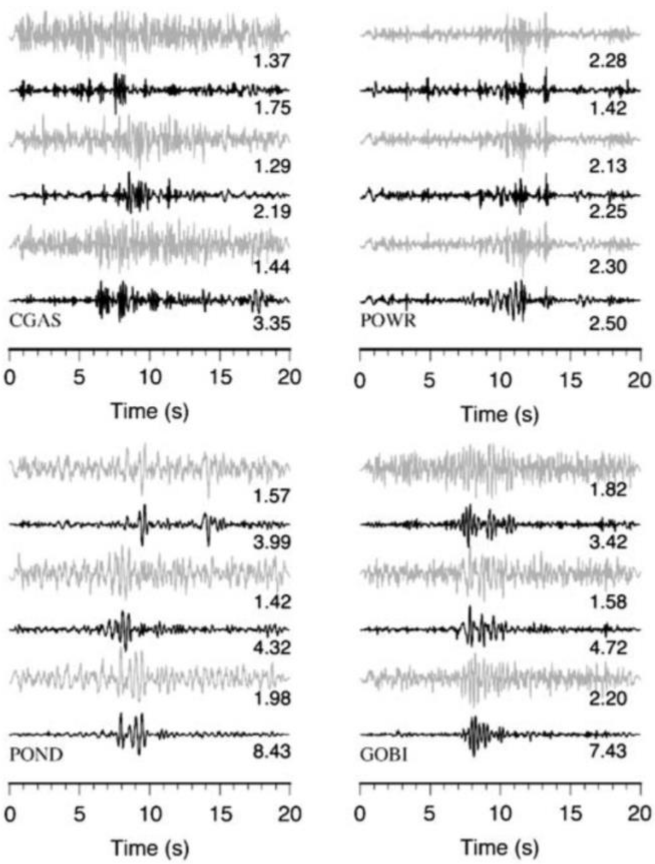



Figure 3. Comparison between linear stacked and PWS LFE signals. Black waveforms, LFE signals after PWS. Grey waveforms, LFE signals after linear stacking.

The SNR for each waveform is marked below each trace. From Figure 3a in Thurber et al., 2015.

# 2. Implementation

## 2.1 Linear stacking on GPU

Sequential version of linear stacking is trivial. For GPU implementation, I used CUDA to achieve the parallel stacking. Pinned memory is used on the host to hold the seismic traces read in. Two versions of GPU linear stacking are compared.

First version is to do a fine scale parallel stacking with each thread only calculating one float point for incoming traces. Assume number of traces is N, number of points in each trace is M. Set block dimension to be (1, 1, 1024). Thus, grid dimension is (1, N, N*M/1024 + 1). Then each thread is only responsible to stack one point onto the final result. In this implementation, atomic add of float points is implemented to avoid race condition.

Second version is to do a coarser scale parallel stacking with each thread responsible for one time point across all incoming traces. Set block dimension to be (1, 1, 1024). Grid dimension is (1, 1, N). In this case, I assume each incoming trace only has M not greater than 1024. With this implementation, there is no racing condition because each thread is accessing a different spot on the result trace.

## 2.2 PWS on GPU

There are three general steps to implement PWS on GPU. 1, use cuFFT to analyze the phase information of each trace; 2, stacking complex traces with Gaussian weighting matrix derived from phase information; 3, invert complex trace back to LFE amplitude information. In order to fully use the warps of each block, the thread number within each block is set up to be multiple of 32.

# 3. Experiment design

The LFE trace used in the test is shown in Figure 3. The LFE signal used is a 25.6 s trace recorded on a station named SMNB in California. Synthetic data are generated via adding random Gaussian noise with mean of 0 and standard deviation of 0.5 s. 1000 synthetic traces are generated for later stacking purposes. The P-wave arrival should be at 11 second of the LFE trace, SNR is calculated with the ratio between signal 5 seconds after and before P-wave arrival.

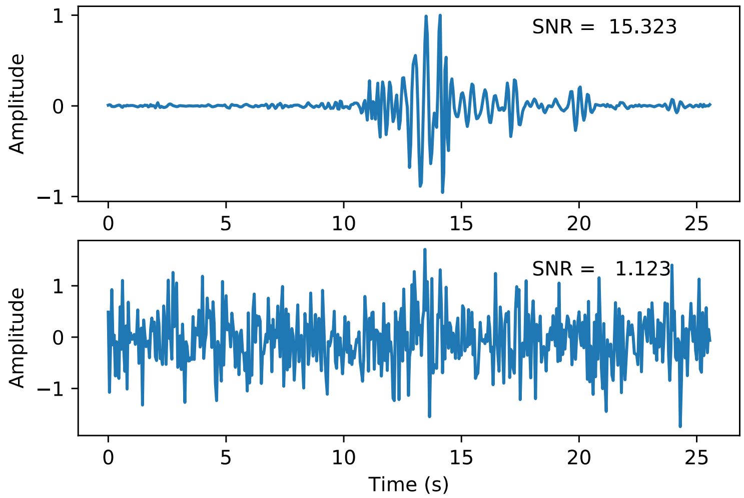



Figure 3. Original LFE waveform (upper panel) and synthetic LFE waveform (lower panel). SNR is marked.

We then conduct linear stacking and PWS on both CPU and GPU to test the performance with total number of traces span from [2, 5, 10, 20, 50, 100, 200, 300, 500, 700, 1000]. The sample rate of each trace is 0.025 and total number of points in each trace is 1024.

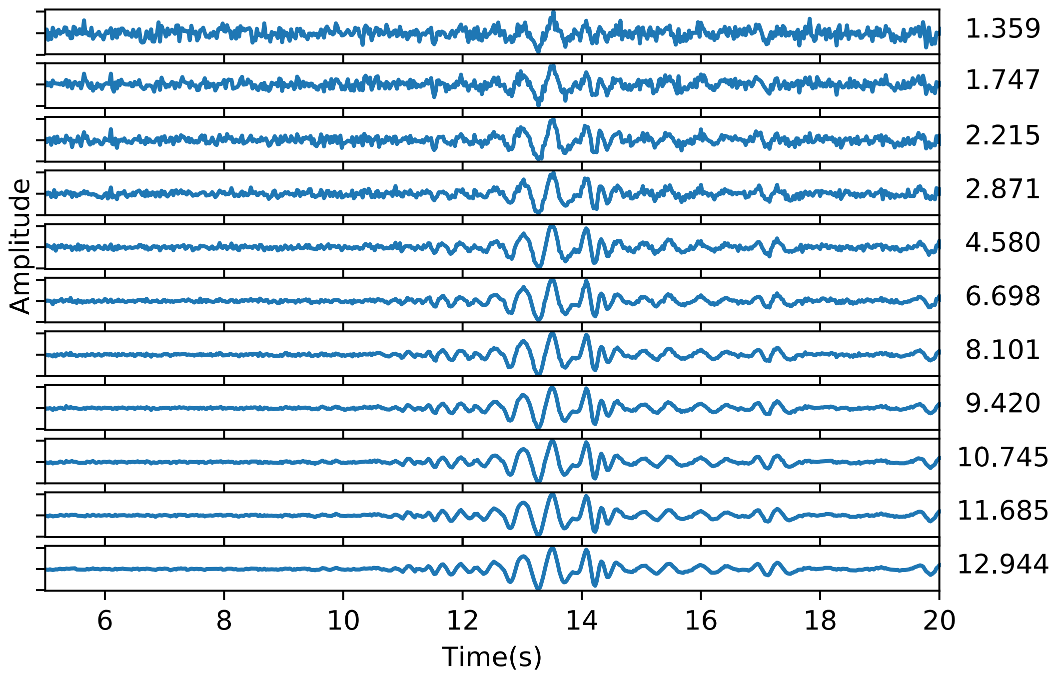
The GPU tested is GK110B and the CPU is Intel i7 3.5 G Hz in our seismological research lab.

# 4. Results and discussion

## 4.1Stacking results and SNR improvement.

As shown in Figure 4, the SNR improves as we stack more traces. The SNR increases from 1.395 to 12.944 for linear stacking while SNR increases from 1.963 to 197.933 for PWS. However, we used the signal with SNR 15.323 to generate synthetic data. Thus, PWS may over suppress some useful information in original seismic waveforms if we stack too many traces together. Because the seismic waveform may vary due to different site response and media in the wave path between earthquake source and seismometer. We can achieve similar SNR to original waveform (about 15) by PWS with 10 to 20 seismic traces. On the other hand, linear stacking needs more than 1000 traces to improve the SNR to the original data level.

(a)


Amplitude
1.359
1.747
2.215
2.871
4.580
6.698
8.101
9.420
10.745
11.685
12.944
6
8
10
12
14
16
18
20
Time(s)

(b)

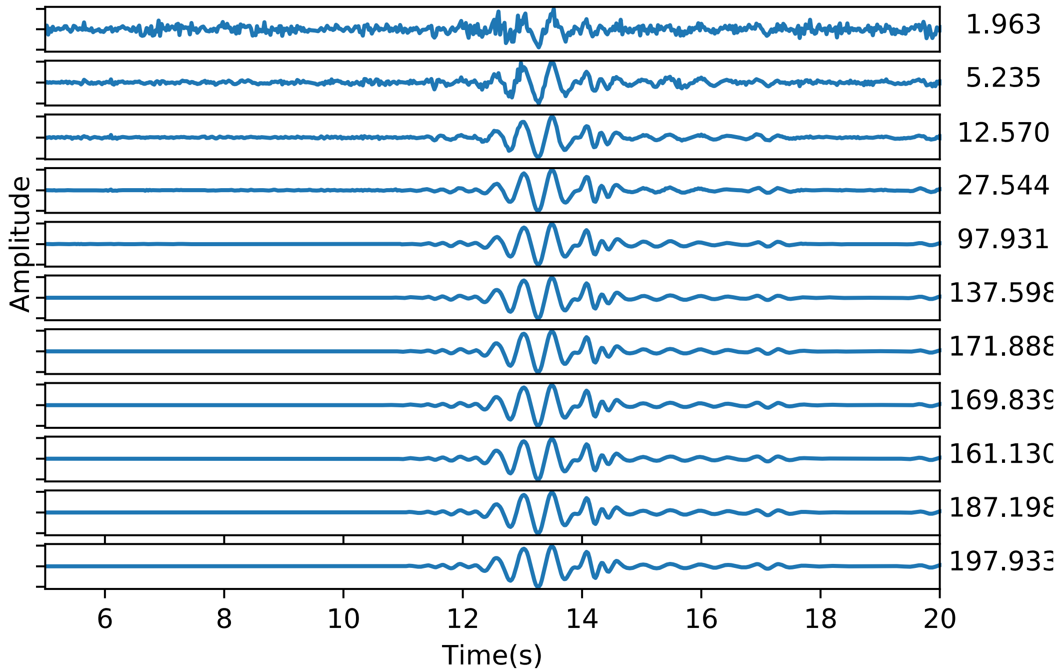



Figure 4. Stacking results. In both (a) and (b), the stacked wavefroms from top to bottom are with 2, 5, 10, 20, 50, 100, 200, 300, 500, 700, 1000 input traces. SNR for each waveform are marked on the right. All traces are normalized. (a) Linear stacking results. (b) PWS results.

## 4.2 Performance of CPU and GPU

For linear stacking, GPU does not really improve the efficiency (Figure 5). The run time is even longer than CPU in our tests. This may due to the communication time overhead introduced by GPU computing.

For PWS stacking, GPU improves the performance about 50 times (Figure 6). The significant performance improvement mainly comes from the FFT analysis. I used cuFFT package to perform FFT in my test.

As shown in Figure 6, in CPU version, PWS runtime increases drastically when number of traces increase from 2 to 100. In GPU version, the increase in runtime is approximately a linear function with respect to the number of traces. In previous discussion, we can see that 20 traces can lead to ideal results with PWS. Thus, GPU can lead to huge performance improvement in seismic waveform analysis with PWS.

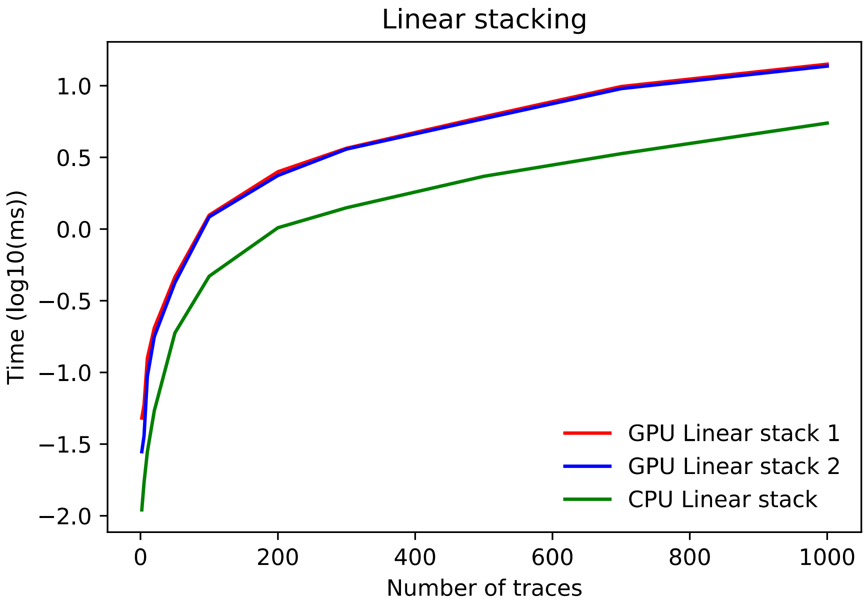



Figure 5. Linear stacking performance between GPU and CPU. We tested two versions of linear stacking with GPU, see details in implementation section.

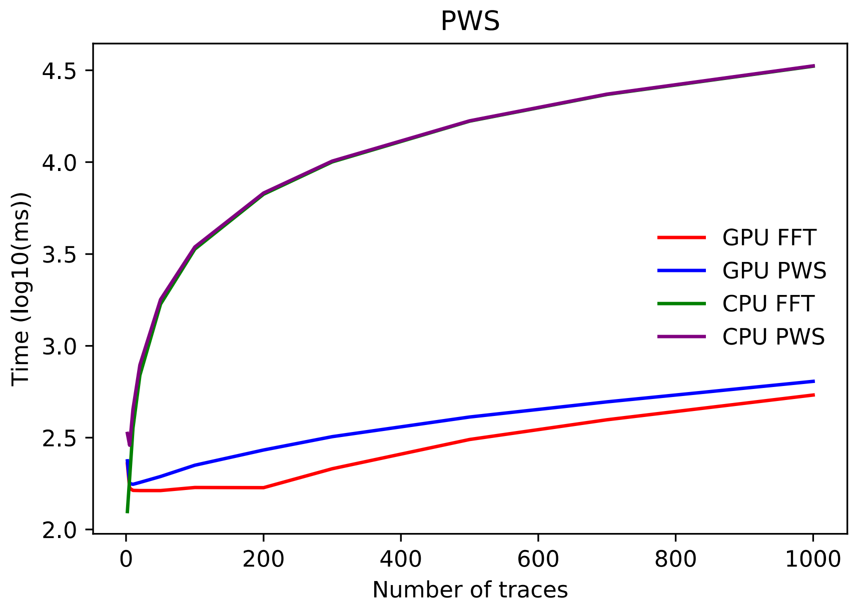



Figure 6. PWS performance between CPU and GPU.

# 5. References

Nishide, N., Hashimoto, T., Funasaki, J., Nakazawa, H., Oka, M., Ueno, H., ... and Takashima, T. (2000), Nationwide activity of low-frequency earthquakes in the lower crust in Japan. In Abstr. Jpn. Earth and Planet. Sci. Joint Meeting, sk-p002.

Katsumata, A., and Kamaya, N. (2003), Low-frequency continuous tremor around the Moho discontinuity away from volcanoes in the southwest Japan. Geophysical Research Letters, 30(1), 20-1.

Shelly, D. R., & Hardebeck, J. L. (2010), Precise tremor source locations and amplitude variations along the lower-crustal central San Andreas Fault. Geophysical Research Letters, 37(14).

Shelly, D. R., Beroza, G. C., Ide, S., & Nakamula, S. (2006), Low-frequency earthquakes in Shikoku, Japan, and their relationship to episodic tremor and slip. Nature, 442(7099), 188-191.

Thurber, C. and Zeng, X., 2014, Phase-weighted stacking applied to low-frequency earthquakes. Bull. Seismol. Soc. Am., 104(5), 2567-2572

Schimmel, M. and Gallart, J., 2007, Frequency-dependent phase coherence for noise suppression in seismic array data. J. Geophys. Res. 112, B04303