CSE305/Project - Fast Fourier Transform

Aya Hankir, Jiwon Park, Minh Khoa Tran

June, 2021

1 Overview

The goal of the proejct is to explore the Fast Fourier Transform (FFT).

Particularly, we manage to explore 2 objectives:

- Compression of weather data using FFT
- Parallelization of the FFT using thread pools

2 Cooley-Tukey Radix-2 Algorithm

The FFT version that we implement is the Cooley-Tukey Radix-2 [1].

It has the theoretical complexity is $O(n \log n)$, but it is limited to only the cases when n is a power of 2, or $n = 2^m$ with some m.

We implement 2 versions of the version:

- **Recursion**: This version uses the divide-and-conquer method. It is simpler to implement and understand the codeflow, but it has some overheads from the recursive calls.

  This corresponds to `fft_fw_rec` and `fft_bw_rec` functions in `lib.cc`.

- **In-place**: This version uses the Fast Bit-Reversal Algoritm to avoid recursive calls. It is faster (see benchmark), but it requires a special algorithm.

  This corresponds to `fft_fw_inp` and `fft_bw_inp` functions in `lib.cc`.

  The Fast Bit-Reversal Algorithm we use is from Cornell University [2].

3 Parallel FFT

The in-place FFT algorithm in theory can be improved using parallization. The scheme we use as the reference is from Tel Aviv University [3].

This corresponds to `fft_fw_par` and `fft_bw_par` functions in `lib.cc`.

In each stage of the main loop, we can launch multiple threads to perform the butterfly operations on each disjointed segments. We use a thread pool model in `pool.cc` to manage these threads.

4 Data Compression

An application of the FFT is data compression.

By applying FFT on an array, we can omit the higher-frequency coefficients to save storage space. The original data can be estimated using the reverse transformation.

Note that if the array size is not a power of 2 (constraint of the Radix-2 algorithm), we could pad it with some dummy data.

4.1 Dataset

We use historical weather data as the dataset for the compression.

The data used is the mean temperature measured in Orly, France from 2000 to 2019 [4]. It contains 7305 entries, which would be padded with 0 to the size of 8192, in order to meet the size constraint.

4.2 Efficiency

The algorithm shows poor result in practice.

Particularly, even we keep 8000 coefficients out of 8192, which is more than the number of terms in the original data, we still see an average error of 2.65 C degree, with the maximum error of 7.9 C degree.

An observation we notice in testing is that the coefficients after the FFT tend to be very large, with their average normed in the order of 8-10. This significantly impacts the quality of the approximation. Further testing also shows that the padding data does not correlate with the quality.

The Discrete Cosine Transform, a related algorithm, might be an interesting choice instead of the FFT.

5 Benchmark

We calculate and compare the running time of all 3 versions of the FFT. We also vary the number of workers in the worker pool.

The number is the average running time over 100 runs on an Intest i5-4300M CPU with 4 cores.

- Array Size: 8192

- 1-thread Recursive FFT: 4.3 ms
- 1-thread In-place FFT (inline): 2.7 ms
- 1-thread FFT: 4.5 ms
- 2-thread FFT: 4.2 ms
- 4-thread FFT: 3.9 ms
- 8-thread FFT: 6.1 ms

- 16-thread FFT: 10.3 ms
- 32-thread FFT: 19.3 ms

- Array Size: 65536

- 1-thread Recursive FFT: 45.3 ms
- 1-thread In-place FFT (inline): 26.6 ms
- 1-thread FFT: 35.6 ms
- 2-thread FFT: 30.5 ms
- 4-thread FFT: 26.8 ms
- 8-thread FFT: 27.4 ms
- 16-thread FFT: 33.8 ms
- 32-thread FFT: 47.2 ms

The result shows insignificant improvement for the parallelization. A possible reason for this is the size of the array is still small, and the speed up gained is lost from in the overhead of the thread pool. Further testing is required.

6 References

1. An algorithm for the machine calculation of complex Fourier Series, James W. Cooley and John W. Tukey

   https://doi.org/10.2307/2003354

2. Fast Bit-Reversal Algorithms, Anne Catherine Elster

   https://folk.idi.ntnu.no/elster/pubs/elster-bit-rev-1989.pdf

3. A parallel FFT on an MIMD machine, Amir Averbuch, Eran Gabber, Boaz Gordissky and Yoav Medan

   https://doi.org/10.1016/0167-8191(90)90031-4

4. European Climate Assessment & Dataset.

   https://www.ecad.eu