Wavelet Transforms in C++

Liping and Albert

October 2021

1 Proposal

We would like to implement wavelet transforms in C++ using Fourier Transforms. We would like to write the DFT routine in parallel, generate the filters used in parallel, and do the matrix multiplication routine in parallel with my own code.

Here is the process we will use:

- Make the filters (there are usually multiple) in frequency in parallel.
- Calculate FFT of original image using a parallel routine.
- Do matrix multiplication for FFT in parallel and calculate IFFT in parallel.
- This will output a convolution of the original image with one of the filters.

We would like to use openMP for these routines first so that we can use it on CPUs. Then we would like to use MPI to split it up over multiple processors and combine it with openmp in each procesor. The idea will be to do each convolution with an image on one processor, so the number of ranks will be fixed for this problem.

We will try to test this versus an implementation Albert previously made given at this link. A success is if the images match and the result is faster than the naive python implementation.

Goals:

- Implement filters in parallel by first week of November.
- Implement 2D FFT and 2D IFFT in parallel by middle of November.
- Implement matrix multiplication by third week of November or use LLPACK library if we run out of time.
- Write paper over the last two weeks.

Another benchmark for success we will use is seeing if the norm between the two wavelet transforms is very small. We will try to test the run time versus the number of threads and see if our results are faster.