

For my final project I am going to attempt to find any un-parallelized sections of the open source software of Open Broadcasting Software (OBS) or switch any existing parallelized optimizations to the OpenMP format.

My motivation for doing this is that in the past I used OBS a great deal to live stream video game content to Twitch.tv and know that the software is used by many people. If i can optimize the software in such a way that will allow people with less powerful computers to livestream it would help the Twitch community greatly. I have never made a change submission to this project as it is worked on by quite a few people but perhaps i can find portions of the code that may be parallelized for slightly better performance.

OBS is currently implemented in C/C++ and uses Pthreads for their multithreading. If there are no portions that can be optimized I will attempt to convert some of the Pthread optimizations to OpenMP and see if there is any increase in performance.

Currently the audio and video compression algorithms are where the most parrallelism exists within this software and is where I focus most of my efforts. This is also a very delicate portion of the software and will require some research before I begin to edit the code and attempt to parallelize it. I don't know my way around this software well enough to provide a block diagram of what portions are / will be parallelized.

As for testing procedure I first plan on making sure that I can implement any of the above stated optimizations while keeping the software operational. Because of the complexity and delicacy of this software and the portions that can benefit from such optimizations this may be very difficult on its own. If i can get any of my optimizations working I will seek to test the software's performance using currently undetermined performance tools.

To start off I will seek to familiarize myself with the software and its core structure more during the first week. For the second week I will look into the source code more to find anywhere that optimizations may be made. For the remaining weeks i will be working bit by bit to implement and test these optimizations.