CS 395 D  
SOFTWARE OPTIMIZATION  

ASSIGNMENT III  
Winter 2014

Assigned:  2/13/2014  
Due:  2/21/2014

This assignment can be submitted either in class or by email to jastratton@knox.edu

The following assignment will build on the analysis of the ImageSegmentation problem we examined in the last assignment. Your approach to optimizing this program should be based on your work from assignment #2. Because of the challenges with getting platforms up and running, if you were unable to submit assignment #2, you may resubmit assignment #2 with this assignment. The lab facilities in the SMC basement should now fully support the updated ImageSegmentation program, but keep in mind that if you are mounting a workspace over the network, then the output files are also being written over the network, and may be an unnecessary bottleneck for your program’s performance.

You will complete this assignment in teams. It is recommended that you complete this in the teams that will be working together for the final project, but team adjustment can be made as more details about each team’s project goals and approaches become more clear.

Part I: Parallelism analysis (9 points)  
This is somewhat of the follow-on to assignment #2, now that we have covered parallel programming topics. For each of the three main phases of the program (after image loading,) describe in what manner they could be parallelized with multiple threads. For each, write a paragraph or so describing the parallelism available, how you would divide up the work among multiple threads algorithmically, what kinds of synchronization or sharing would be required, and whether there are any “cleanup” steps to be performed to combine the results of multiple threads into one.

Part II: Optimization approach and lab report (30 points)  
Optimize the application to improve its performance to the best of your ability under the time constraints. The ultimate speedup improvements you are able to deliver is less important than a well thought out approach to designing and implementing your performance optimizations, and a good understanding and explanation of the results you achieved. Submit your insights and analysis in a “lab report” with the following components. You may organize the report as you choose, but each of the following components should be present for each of the major optimization attempts made.

A) Identification of what optimizations should be performed. Why did you decide to apply the optimization that you did, on the section of code that you chose? You should base your answer on the expected potential performance improvement, the expected amount of development effort required, and/or the particular learning goals
and personal interest in the approach you chose. Note that you may repeat this phase multiple times for different optimization steps after completing all steps for other optimizations – this is expected and fine.

B) Overview of the optimization’s design. What concrete changes to the software did you make to effect the optimization you identified? Describe any changes to the algorithm structure, including any major operation reordering.

C) Measured results and analysis of the optimization. How much did performance change, and can you explain why? If your results were good, maybe your initial assessment was correct. If your results were not as good, can you identify what was flawed with either your initial identification or implementation that did not provide the benefit you expected?

A good submission should provide enough detail that your classmates can understand what you tried, how well it worked, and how they might incorporate similar optimizations into their own versions of the application.