Meeting Notes and Assignment 6/28/19

Paul Scanlon and Matt Murdoch

Colorado State University, Research for Dr. Sanjay Rajopadhye, Summer 2019

Key Meeting Points

Last week we augmented the execution time and makefile of Shadows. Sanjay explained how the $NormalizeReduction(prog)$ followed by $AShow$ suggests Alphaz optimization based on the current implementation. We achieved the optimization to linear run time.

Matt explained the python script for optimization with respect to the ”caster”. Then Paul explained how this relates to the described concept of a shifting triangle.

When using the check command, we are getting input from *standard in*, and we are initializing the input variables. Then we call timing. The missing implementations in our makefile are timing, random, and random timing alone. The reason for having these implemented is for faster testing and implementation options.

The point of the Scan assignment is to parallelize Scan. Take the sequential, order N work, and parallelize it using the prefix *sum* operation. The next step would be to time that. This may be work in the next few weeks.

When you get timing data, how can you test that the same input was given. If this is important, we need to seed the random number in the wrapper, provide it first in the wrapper. Line 106 seeds with the time of day.

What causes the huge time difference at a problem size of 500,000 in Shadows? This merits further investigation. Dig into this as Sanjay does not have an immediate answer.

Logically, there should be bounds on T. We have to impose this in the verify and wrapper script. The T constraint in alpha is about the dimension of T, you cannot specify positive or negative values of T in AlphaZ.

The compiler reasons about distributivity to change from *and* to *max*.

Assignment

1.) Implement the rest of the makefile with respect to the suggested revisions above.

2.) Why does a factor of 4 variance in execution time occur in the optimized Shadows AlphaZ algorithm at 500,000 problem size. This will be a difficult question.

3.) What about the flag nesting in the wrapper makes sense?

4.) Test the zero calibration time of the timer function.

5.) MCP - Implement string parenthesization in AlphaZ and run a verification script similar to verify in Shadows.

6.) Consider what is a legal way to fill the entries of the table in the parenthesization algorithm for optimization. This will be a difficult question.