Nov. 9, Assignment #3, due Nov. 18 @ 11:59 pm (10%)

Gardner Home | Course Home

Marking: For this assignment, the software and documentation are each worth 50%. Software that does not compile, link, and execute on lockhart will receive 0%. The report does not have to be beautifully bound, but should be presentable and not be hand written.

Where's Parallaldo? with OpenMP

Now that you're thoroughly familiar with this problem and have done a pthreads solution, let's go back to the drawing board. Get out your original pseudocode and, of course taking into account any flaws you discovered in the course of implementing the pthreads version, start by making a **serial program** that produces correct results.

Then, just like we've done in the labs, begin parallelizing it using OpenMP. That's all! The command line arguments are the same as for A2, and it produces the same output on stdout. Try to employ Parallel Inspector and Vtune Amplifier to good advantage. You'll be reporting on tool use in the documentation.

NOTE: It is not necessary to implement "interactive mode" in the OpenMP version, which should allow you to cut down the complexity.

Deliverables

- Source code via electronic submission using Subversion under tag **cis3090-A3** :

  1. OpenMP program called "ompaldo" in Visual Studio "solution." Each source file that you create must include a **file header** containing your name and student number, the filename, and a brief description of the purpose of the file. Submissions lacking proper file headers will be penalized.

  2. Accompanying Visual Studio files in the same directory, so that the solution can be readily opened and built. The odd-named files that are important are: .sln, .vcxproj, .vcsproj.user, and .vcxproj.filters. **Please avoid adding/committing the numerous, large files produced by Parallel Inspector and Vtune Amplifier.** We also don't need the subdirectories named Debug and Release.

- Documentation, printed and delivered to class.

  1. Write your impressions of working with Parallel Studio and OpenMP. Compare and contrast your pthreads experience.

  2. Report on your use of Parallel Inspector and Amplifier. Explain how they helped you, or why they didn't.

  3. Describe any changes you made to your application's architecture, data structures, and/or parallelization strategy in moving from a pthreads solution to OpenMP, and in trying to speed up the OpenMP version.

  4. Make a timing graph of total execution time (Y axis) vs. number of cores (1-16 with at least 8 data points) running the Release (optimized O2) version using the sample directories provided in Public Documents. Also make speedup and efficiency graphs based on the same data. Draw the linear speedup line on the speedup graph. Write some observations about the graphs. Did compiler optimization make a difference? **Given the results, how scalable was your parallel design's performance using OpenMP?**

  5. You don't need to rerun and graph every trial with the unoptimized Debug version, but do enough testing with O0 so you can report whether compiler optimization made a difference and to what extent.

Bonus marks

Same terms as for A2, including rematch.

Marks for Assignment #3

For the software, mark is 0-5 based chiefly on functionality:

5 = Parallel program that produces correct results, reports timings, and shows decent application of omp pragmas

1.5-4.5 = Same as 5, but with deductions according to deficiencies

1 = Parallel program that runs cleanly, and is coded with omp pragmas to address the problem, but results are all incorrect or missing

0 = Fails to compile and link, or crashes without producing any results

In addition, small 1/2 mark penalties were deducted for specific "sins" as in A1.

For the documentation, start with 5 marks, then deduct 1 mark for any item missing, or 1/2 mark for any lame or poorly-done item. If you didn't have any working software to graph, you'll lose at least 1 more mark to be fair to those who conducted timing and produced their graphs.

Gardner Home | Course Home