Modernization Strategy

Test-Driven Development (TDD) grew out of the Extreme Programming movement of the 1990s. Although its basic concepts are as basic as the name suggests and its applications are not unique to the 1990s, TDD quickly reared its head toward software solutions by first writing tests that specify what the working software should do and then coding a software and test application code in order to pass the test. In the current context, TDD serves the purposes of ensuring that our parallelization exercises produce the expected results for representative test cases.

Table 1 shows 17 steps we employed in refactoring the PRM. The objectives of the different groups of tests were the following:

1. Experience the test suite without any parallelization
2. Experience the test suite with parallelization
3. Experience a suite of accuracy tests with parallelization
4. Experience a suite of accuracy tests

Case Study: PRM

The Particle Representation Model (PRM) describes bands on each octant of the hemisphere. Figure 2 shows the flattened bands. Each particle has properties such as band number and particle number within a band from the global particle number. The speedup obtained using the two different parallelization algorithms is shown in the Results section.

Conclusions and Future Work

We have demonstrated a strategy for parallelizing legacy Fortran 77 codes using Fortran 2008 coarrays. The strategy starts with building automated unit tests that check for regressions in accuracy or performance. In the PRM case study, our strategy involved two Fortran 77 codes by 10% and 40%, excluding the test and I/O infrastructure. The most significant code revision involved updating two nested loops that operate across images. The resulting parallel code achieves even load balancing. TAU identifies the chief bottleneck: a sequential sum. A simple parallel implementation of the coarray sum significantly improved the speedup of the problem, which may not be readily available to some extent in serial or parallel.

Acknowledgements

Funding for this work was provided by the Office of Naval Research under contract N00014-04-1-0159. Portions of the Sandia contribution to this work were supported by the Office of Basic Energy Sciences, U.S. Department of Energy, under Contract DE-AC04-94-AL85000. Portions of the Sandia contribution to this work were supported by the National Nuclear Security Administration under contract DE-AC04-94-AL85000. Portions of the Sandia contribution to this work were supported by the New Mexico Small Business Administration and the Office of Naval Research.

Results

More or less, we have considered the data from the TAU tool for the Fortran compiler and program and have plotted TAU plots of each image.

Considerations for future work include:

- Performance analysis of the Fortran 2008 coarray parallel implementation
- Evaluation of alternative summation algorithms
- Further refinement of the parallelization strategy

Figure 5 shows a TAU plot of each image’s runtime share for the dominant procedures which shows that the chief bottlenecks are collective sum procedures. The Fortran standards committee is working on a coarray intrinsic that will likely become part of the Fortran standard. We implemented a parallel version of the coarray procedure using a functional tree approach. A tool to increase the number of bands.

Turbulent flow theorists typically attempt to predict the statistics of a fluid velocity vector field after decomposing it into its mean and fluctuating components. As shown in Figure 1, the PRM tracks particles on a unit sphere in 17 steps. The modernization strategy involves 17 steps from automating the building and testing process to removing obsolete and deprecated features. Figure 3 shows a distribution of particles across four images using two different parallelization algorithms. The modernization strategy involves 17 steps from automating the building and testing process to removing obsolete and deprecated features.

Additional features in Fortran 2008 coarrays make it possible to perform collective operations on images without the overhead of passing data across image boundaries. This allows the PRM to perform collective operations on images without the overhead of passing data across image boundaries.

The Particle Representation Model (PRM) describes bands on each octant of the hemisphere. As shown in Figure 1, the PRM tracks particles on a unit sphere. The modernization strategy involves 17 steps from automating the building and testing process to removing obsolete and deprecated features.

The Particle Representation Model (PRM) describes bands on each octant of the hemisphere. As shown in Figure 1, the PRM tracks particles on a unit sphere. The modernization strategy involves 17 steps from automating the building and testing process to removing obsolete and deprecated features.

The Particle Representation Model (PRM) describes bands on each octant of the hemisphere. As shown in Figure 1, the PRM tracks particles on a unit sphere. The modernization strategy involves 17 steps from automating the building and testing process to removing obsolete and deprecated features.

Figure 2 shows the flattened bands. Each particle has properties such as band number and particle number within a band from the global particle number. The speedup obtained using the two different parallelization algorithms is shown in the Results section.