Brief Introduction to OpenMP

Luke Tierney

Department of Statistics & Actuarial Science
University of Iowa

October 11, 2007

Background

- OpenMP is a framework for shared memory parallel computing.
- OpenMP is a standard C/C++ and Fortran compilers.
- Compiler directives indicate where parallelism should be used.
  - C/C++ use `#pragma` directives
  - Fortran uses structured comments.
- A library provides support routines.
- Based on the fork/join model:
  - the program starts as a single thread
  - at designated *parallel regions* a pool of threads is formed
  - the threads execute in parallel across the region
  - at the end of the region the threads wait for all of the team to arrive
  - the *master thread* the continues until the next parallel region.

Brief Introduction to OpenMP

- Some advantages:
  - Usually can arrange so the same code can run sequentially.
  - Can add parallelism incrementally.
  - Compiler can optimize.
- The OpenMP standard specifies support for C/C++ and Fortran.
- Many compilers now support OpenMP, for example
  - newer versions of `gcc` and `gfortran`
  - Intel compilers `icc` and `ifort`
- The OpenMP runtime creates and manages separate threads.
- OpenMP is much easier to use than low level thread libraries.
- You still have to make sure what you are doing is thread-safe.