UiT

THE ARCTIC UNIVERSITY OF NORWAY

OpenMP introduction

Edvard Pedersen
Edvard.Pedersen@uit.no

What is OpenMP?

- API for doing (semi-)automatic shared-memory parallel programming
- Parallelism acheived through compiler directives
    - OpenMP parallelizes the code automatically based on the directives

How to use OpenMP

- Compile with -fopenmp
- Add compiler directives to appropriate places in the code
  - Example: Mandelbrot code
- Use the OMP_NUM_THREADS environment variable to control the number of threads

Profiling and analysis

- OmpP is a nice profiling library
- For the sequential code, regular profiling is fine
- You may want to trace the sequential code to find good places to parallelize
- Valgrind DRD is useful for detecting data races

The tricky parts

- Shared memory
  - It's a nightmare!
- What can be shared?
  - For the mandelbrot code, everything is shared
  - For Markovian, not quite so simple
- #pragma omp default(none) shared(var1,var2) private(var3)
  - Gives you compile-time errors if there are data dependencies you haven't cleared up

The tricky parts cont.

- Restrict execution to one thread at a time
  - Omp_(un)set_lock(lock_t *lock)
  - #pragma omp atomic
  - #pragma omp critical
  - #pragma omp master
- Synchronization
  - #pragma omp barrier
  - #pragma omp taskwait

Tips and tricks

- Try to find the place where you have the most to gain by parallelizing
- Think about the data which is getting passed around, what can you share, and what do you need to duplicate