Brief OpenMP

CSCI 317

Mike Heroux

Simple OpenMP Program

```
/* This program will numerically compute the integral of
                4/(1+x*x)
from 0 to 1.  The value of this integral is pi -- which
is great since it gives us an easy way to check the answer.
History: Written by Tim Mattson, 11/99.        */

#include <stdio.h>
#include <omp.h>

static long num_steps = 100000000;

double step;
int main () {
  int i;
  double x, pi, sum = 0.0;
  double start_time, run_time;

step = 1.0/(double) num_steps;
for (i=1;i<=4;i++) {
  sum = 0.0;
  omp_set_num_threads(i);
        start_time = omp_get_wtime();
  #pragma omp parallel
  {
    #pragma omp single
    printf(" num_threads = %d",omp_get_num_threads());

    #pragma omp for reduction(+:sum)
    for (i=1;i<= num_steps; i++) {
      x = (i-0.5)*step;
      sum = sum + 4.0/(1.0+x*x);
    }
  }
        pi = step * sum;
        run_time = omp_get_wtime() - start_time;
        printf("\n pi is %f in %f seconds and %d threads
          \n",pi,run_time,i);
}
}
```

Compiling/Using OpenMP code

- GCC:
  - Add –fopenmp to compile line:

    g++ -fopenmp …
- Intel:
  - Add –openmp

    Add #include <omp.h> to source code.
- Set number of threads:
  - Use omp_set_num_threads(numthreads) in your code.
  - Use:

    setenv OMP_NUM_THREADS numthreads (for tcsh or csh shells).

    export OMP_NUM_THREADS=numthreads (for bash shell).