

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.
// Derived by Mike Heroux from a program of Tim Mattson.
```

```
#include <iostream>
#include <iomanip>
#include <cstdlib>
#include <omp.h>
```

```
int main () {
// Define the number of integration partitions to use
static long numSteps = 1000000000;

// Compute the length of each partition for the interval 0 to 1
double step = 1.0/((double) numSteps);

// Get the number of threads available on the system
int maxNumThreads = omp_get_max_threads();
std::cout << "Maximum number of threads available on
system " << maxNumThreads << std::endl;
```

```
for (int j = 1; j <= maxNumThreads;j++) {
double sum = 0.0;
omp_set_num_threads(j);
double startTime = omp_get_wtime();

#pragma omp parallel for reduction(+:sum)
for (int i = 1; i <= numSteps; i++) {
double di = (double) i;
double x = (di-0.5)*step;
sum = sum + 4.0/(1.0+x*x);
}

double pi = step * sum;
double runTime = omp_get_wtime() - startTime;
std::cout <<"pi estimate is " << std::setprecision(15) << pi
<< " computed in " << std::setprecision(5)
<< runTime << " seconds"
<< " on " << j << " threads."
<< std::endl;
}
}
```

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).