OpenMP Thread Testing

Dr. Christopher Marcotte — *Durham University*

Hardware exploration

Under Linux, we can run `lscpu` to get information about the CPU. On macOS, a similar command is `sysctl -a | grep cpu`. On Windows, we begin to step into the arcane.

Exercise

Log in to Hamilton. How many physical cores are available on the compute nodes? How many on the login nodes? How would you set up your batch script to detect the former?

Solution

There should be 128 cores on the login nodes, and the same on the compute nodes.

Testing threads

Consider the following short program, `threadtest.c`:

threadtest.c

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

int main() {
  printf("Thread #%d reports: This program uses %d threads.\n",
         omp_get_thread_num(), omp_get_num_threads());

#pragma omp parallel
  printf("Hello world from thread #%d\n", omp_get_thread_num());

  return 0;
}
```

Compile it with `gcc -fopenmp -o threadtest threadtest.c` and then run it.

Exercise

How many threads are reported? Modify the code so that only the thread with `threadID 5` reports the correct number of threads?

Solution

Outside of a parallel region, the number of threads is 1.

threadtest_sol.c

```c
#include "omp.h"
                                        #include
```

```c
int main() {
#pragma omp parallel
  {
    if (omp_get_thread_num() == 5) {
      printf("Thread #%d reports: This program uses %d threads.\n",
             omp_get_thread_num(), omp_get_num_threads());
    }
    printf("Hello world from thread #%d\n", omp_get_thread_num());
  }
  return 0;
}
```

We need to check the thread number with `omp_get_thread_num()` and use that to only let thread 5 speak.

Aims

- Introduction to some hardware reports.
- Familiarity with simple usage of OpenMP library functions.
- Familiarity with OpenMP parallel regions and how they interact with the number of threads.