

Some Basic OpenMP Coding for Parallelism

The Main Question for Math 4610 at USU

- How many processors, have I. Create the “Hello World!” application and determine the number of processors available on your computer.
- What happens if you instantiate a virtual machine? To answer the question you should run the “Hello World!” your own computer. This is done in the first part of this module. Then, start a virtual computer and rerun the code on the computer and then on the virtual machine. Are the results different or the same.

Now, let’s see if we can apply OpenMP in a more important application.

- Write a code to compute the results of matrix multiplication. You have a code from an earlier module. It may be written in Python. You will need to rewrite the code in C. Use a version of the code that has a dot product as the inner most loop. Make sure the C code is working. This is called a **serial** version of the code.
- Now, for the “improvement”. The following code is on the web and gives an OpenMP version of matrix multiplication.

```

//
// Initialize the arrays
//
for(int i=0; i<100; i++)
{
    for(int j=0; j<100; j++)
    {
        A[i][j] = i;
        B[i][j] = j;
        C[i][j] = 0;
    }
}
//
// Starting the matrix multiplication
//
#pragma omp parallel num_threads(4)
{
    #pragma omp for collapse(3)
    for(int i=0; i<100; i++)
    {
        for(int j=0; j<100; j++)
        {
            for(int k=0; k<100; k++)
            {
                #pragma omp atomic
                C[i][j] = C[i][j] + ( A[i][k] * B[k][j] );
            }
        }
    }
}

```

Modify the code you have written to use OpenMP as done as above.

- Time the serial code against the parallel code you have just implemented. Which is the faster code for the computation as the size of the matrix increases?