Assignment 3 Report

Firstly, to figure out what parts of the program take the most time, I have used combination of gprof and Hatchet.

(gprof result)

```
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls  Ts/call  Ts/call  name
 94.98      4.06     4.06                             smvp
  3.28      4.21     0.14                             phi2
  0.94      4.25     0.04                             element_matrices
  0.94      4.29     0.04                             mem_init
  0.00      4.29     0.00   151173     0.00     0.00  inv_J
  0.00      4.29     0.00     4162     0.00     0.00  area_triangle
  0.00      4.29     0.00        1     0.00     0.00  arch_parsecommandline
```

(Hatchet result)

```
                                 node_pid  core                                                module  ... time (inc)     instruction       time
name                                                                                                 ...
Loop@[quake]:0                 23967659982   336                                                     0  ...  29.819631               0  20.485325
[quake]:0                      25679635695   360                                                     0  ...  16.453013               0   9.141706
smvp                            1711975713    24                                                     0  ...   5.680787               0   5.674828
main                            1711975713    24                    /home/cahn12/CMSC416/assignment-3/quake  ...  10.237406        0x401150   3.264007
/usr/lib64/libm-2.28.so:0x7e602 3423951426    48  /usr/lib64/libm-2.28.so/usr/lib64/libm-2.28.so  ...   0.262454  0x7e6020x7e602   0.262454
```

```
0.000 main thread None
└─ 3.264 main [quake]
   ├─ 3.264 Loop@[quake]:0 [quake]
   │  ├─ 3.264 Loop@[quake]:0 [quake]
   │  │  ├─ 3.037 Loop@[quake]:0 [quake]
   │  │  │  ├─ 0.012 Loop@[quake]:0 [quake]
   │  │  │  │  └─ 0.012 Loop@[quake]:0 [quake]
   │  │  │  │     └─ 0.012 [quake]:0 [quake]
   │  │  │  └─ 3.025 [quake]:0 [quake]
```

```
└─ 0.006 [quake]:0 [quake]
└─ 5.675 smvp [quake]
   ├─ 5.675 Loop@[quake]:0 [quake]
   │  ├─ 5.132 Loop@[quake]:0 [quake]
   │  │  └─ 5.132 [quake]:0 [quake]
   │  └─ 0.543 [quake]:0 [quake]
   └─ 0.000 [quake]:0 [quake]
      └─ 0.000 <unknown procedure...[libgomp.so.1.0.0] [quake]
         └─ 0.000 [libgomp.so.1.0.0]:0 [libgomp.so.1.0.0]
            └─ 0.006 <unknown procedure...[libgomp.so.1.0.0] [libgomp.so.1.0.0]
               └─ 0.006 [libgomp.so.1.0.0]:0 [libgomp.so.1.0.0]
```

Both gprof and Hatchet show that smvp function consumes the most time. Loop inside the smvp were taking up most of the time so the loops inside smvp were parallelized.

To parallelize smvp function, a few lines were changed from serial code. The for loop inside the smvp function depends on my_cpu_id variable but in serial code was assigned in #pragma omp parallel private block not in the scope of the loop thus loop was not able to access my_cpu_id assigned in parallel. Such block served no purpose, so it was deleted, then the assigning was done inside parallel code. The rest of the code was untouched besides the addition of #pragma.

Loops inside main functions were the second most time consuming. Loops inside the main, except for those that were obviously not highly scalable due to low iteration count, were parallelized.

Results

On each of below results, doing diff quake-omp.out quake.out showed only difference to be the line displaying the time.

Screen shot shows the time stamp 3840 result on input file quake.in

1 thread

```
5903: -3.98e+00 -4.62e+00 -6.76e+00
16745: 2.45e-03 2.66e-02 -1.01e-01
TIME 9.85423 s
```

2 threads

```
5903: -3.98e+00 -4.62e+00 -6.76e+00
16745: 2.45e-03 2.66e-02 -1.01e-01
TIME 6.20592 s
```

4 threads

```
5903: -3.98e+00 -4.62e+00 -6.76e+00
16745: 2.45e-03 2.66e-02 -1.01e-01
TIME 5.15436 s
```

8 threads

```
5903: -3.98e+00 -4.62e+00 -6.76e+00
16745: 2.45e-03 2.66e-02 -1.01e-01
TIME 4.41603 s
```

16 threads

```
5903: -3.98e+00 -4.62e+00 -6.76e+00
16745: 2.45e-03 2.66e-02 -1.01e-01
TIME 3.21136 s
```

32 threads

```
5903: -3.98e+00 -4.62e+00 -6.76e+00
16745: 2.45e-03 2.66e-02 -1.01e-01
TIME 2.22884 s
```

64 threads

```
5903: -3.98e+00 -4.62e+00 -6.76e+00
16745: 2.45e-03 2.66e-02 -1.01e-01
TIME 3.20171 s
```