

1. My Machine:

I ran this program on the flip server in linux from a windows 10 laptop. My main function is contained in a file called project2.cpp which I compiled by typing:

```
% g++ -l/usr/local/common/gcc-5.4.0/ project2.cpp -o proj2 -lm -fopenmp
```

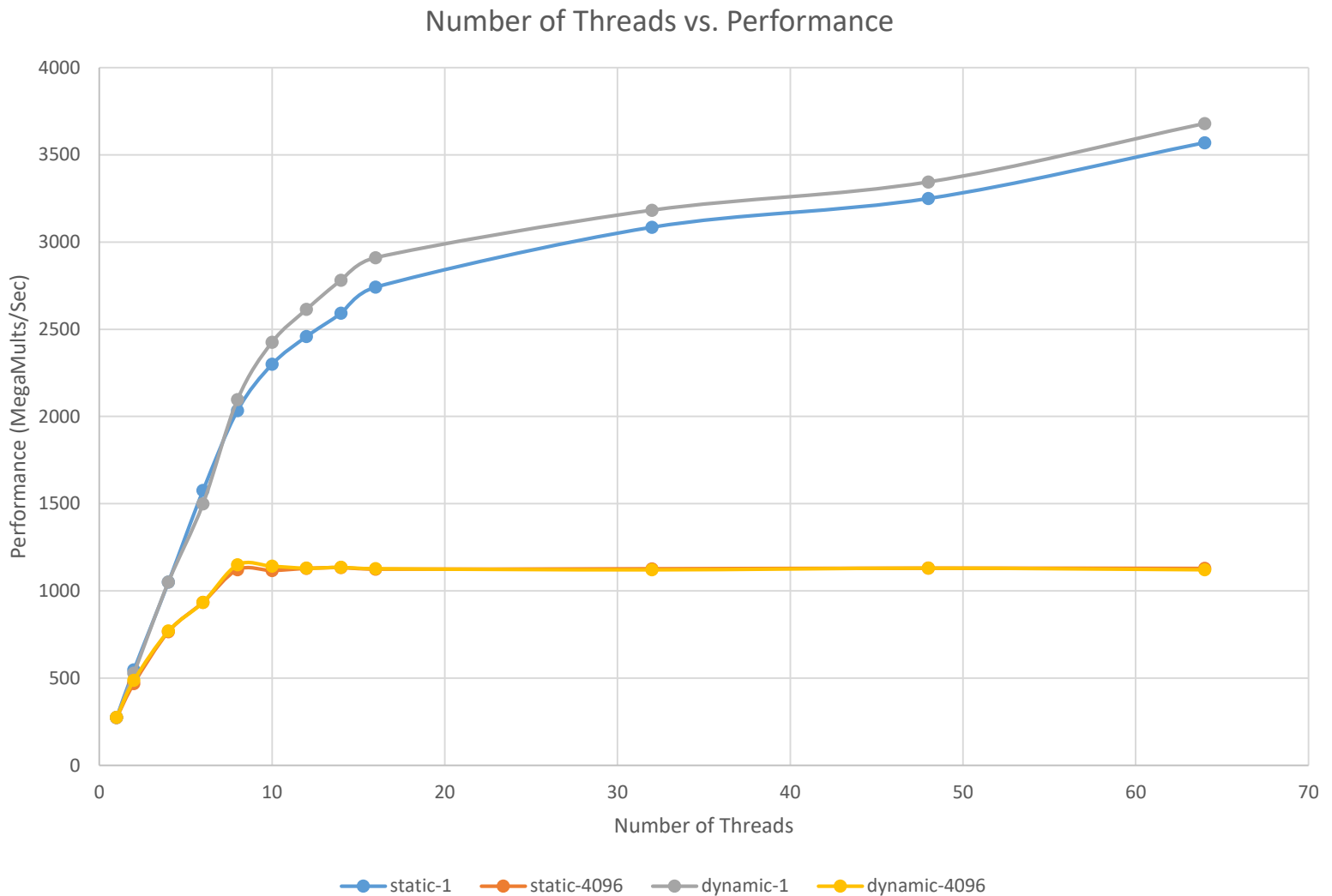
Then I executed the program with:

```
% ./proj2
```

2. My performance results:

	<b>static-1</b>	<b>static-4096</b>	<b>dynamic-1</b>	<b>dynamic-4096</b>
<b>1</b>	274.98	273.73	273.6	274.35
<b>2</b>	547.83	469.3	527.92	487.06
<b>4</b>	1051.36	765.9	1050.75	769.83
<b>6</b>	1575.55	934.35	1498.74	934.61
<b>8</b>	2034.43	1121.39	2097.21	1149.34
<b>10</b>	2299.36	1117	2425.55	1141.5
<b>12</b>	2458.81	1128.68	2613.92	1130.39
<b>14</b>	2591.51	1133.56	2780.81	1135.25
<b>16</b>	2741.84	1124.68	2910.51	1127.33
<b>32</b>	3085.03	1126.76	3182.39	1121.1
<b>48</b>	3249.64	1130.14	3344.01	1130.59
<b>64</b>	3569.62	1128.63	3679.3	1121.46

### 3. Graph of Performance:



### 4. Speed Patterns I Noticed:

As the number of threads increased, the performances of static-4096 and dynamic-4096 were very similar. This was also the case for the performances of static-1 and dynamic-1. The performance was much higher and increased much more rapidly for chunksize of 1 rather than chunksize of 4096. This held true for both dynamic and static scheduling.

### 5. Chunksize 1 vs. 4096:

In this program, I used an array of size 32678. Clearly, if we have a chunksize of 4096, we can have a maximum of 8 threads running at a time (because each will get a chunk of 4096 and  $32678/4096 = 8$ ). It can be seen clearly in the graph above in that both static-4096 (orange) and

dynamic-4096 (yellow) the performance stops increasing after 8 threads regardless of how many more threads are added.

For example, if we have an array of size 8, and chunksize of 2:

With 1 thread:

```
1:      ** ** ** **
```

With 2 threads:

```
1:      **      **
2:          **      **
```

With 3 threads:

```
1:      **          **
2:          **
3:              **
```

With 4 threads:

```
1:      **
2:          **
3:              **
4:                  **
```

With 5 threads:

```
1:      **
2:          **
3:              **
4:                  **
5:
```

With 6 threads:

```
1:      **
2:          **
3:              **
4:                  **
5:
6:
```

Clearly, increasing the number of threads > 4 has no effect in the above example and the case is the same in our original problem for number of threads > 8.

On the other hand, when we have a chunksize of 1, we can continue to drastically increase performance by adding more threads (for the parallelizable portion of the problem).

6. Static vs. Dynamic:

Because each iteration of the for-loop being parallelized takes the same amount of time to execute, there is little-to-no difference in the static vs dynamic performance. We can see in the graph above that static-4096 (orange) and dynamic-4096 (yellow) have almost exactly the same performance. While static-1 (blue) and dynamic-1 (grey) have similar performance as well.