

Andrew Johnson  
CS475

## Project 0 Write Up

UPTIME GAVE: 31 users, load average: 0.19, 0.24, 0.23

### RESULTS (1 Thread)

```
#define NUMT          1
#define ARRAYSIZE     10000
#define NUMTRIES      10000
```

Using 1 thread

Peak Performance =	549.36 MegaMults/Sec
Average Performance =	546.64 MegaMults/Sec

### RESULTS (4 Thread)

```
#define NUMT          4
#define ARRAYSIZE     10000
#define NUMTRIES      10000
```

Using 4 threads

Peak Performance =	898.62 MegaMults/Sec
Average Performance =	859.30 MegaMults/Sec

### Commentary

I ran this on the eCampus Server flip (Gave the current Uptime above). Using the computers in the Kelly Computer lab as well, but that won't matter much.

When going from 1 thread to 4 threads it seems to have done a bit less than double the original time. Which shows that we got a speedup of about 2 when going from 1 thread to 4 threads. This means it gives 50% speedup efficiency.

Since the computer is dividing such a large array into 4 equal smaller ones, the computer has less objects to use giving quite a bit of faster runtime. As seen from above. I believe if I were to use even larger numbers, within the bounds of the computer, the speedup would show more efficiency than what I got.