Benchmarking SATI.c

Dean Yockey

My computer

My personal computer is a Lenovo ThinkCentre I bought for $75 on eBay. It's running the Linux Mint operating system. When running `lscpu`, the first few lines read:

```
Architecture:            x86_64
  CPU op-mode(s):        32-bit, 64-bit
  Address sizes:         39 bits physical, 48 bits virtual
  Byte Order:            Little Endian
CPU(s):                  4
  On-line CPU(s) list:   0-3
```

My computer has 4 CPUs.

SATI.c

I ran compiled and ran SATI.c after commenting out all printing statements.

Benchmarking

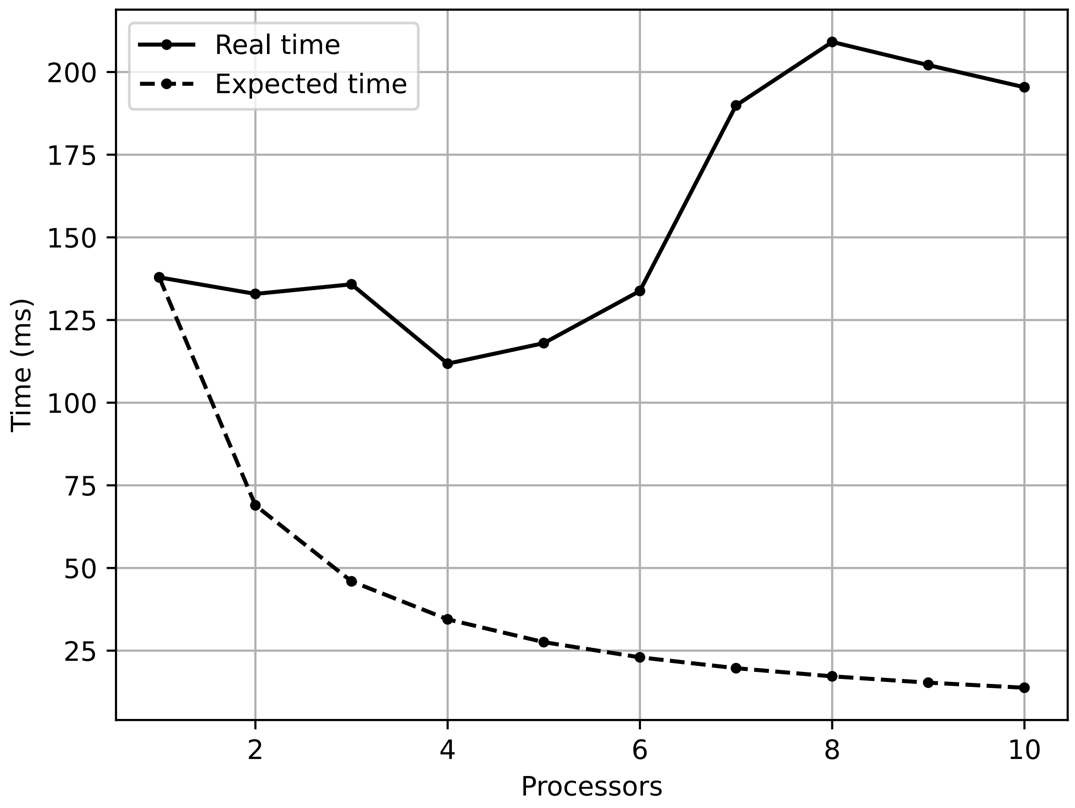
I used the program Hyperfine to automate the benchmarking. I created a bash file `test.sh` with the line:

```
hyperfine -w 3 -P p 1 10 "mpirun -n {p} --oversubscribe sati" > log.txt
```

The `-w 3` option made each benchmark run 3 warmup tests. The `-P p 1 10` option and `{p}` in the command tell Hyperfine to run this benchmark with a range of parameters from 1 to 10. I had to call `mpirun` with the `--oversubscribe` option, because most of the tests demand more processors than my computer has. I output the results to `log.txt`.

My results

To extract the results from the log file, and visualize them in a line graph, I wrote a python script. I used regex and matplotlib. This is my graph.



Takeaways

There's a very stark difference between my expected time and actual time. The addition of one or two processors had marginal impact on the time. Certainly nowhere near halving.

The best time came with 4 processors. This makes sense, since my computer has 4 processors.

At 7 processors, the time shoots up to significantly worse than the single-processor time. This is probably because simulating 7 processors adds communication time without any more real parallelization compared to 4.

I think my computer is poorly suited to this task.