

A scenic view of the University of Colorado Boulder campus. In the foreground, a large brick building with a central tower and an American flag on top is visible. The building is surrounded by lush green trees with some autumn-colored foliage. In the background, a large, rugged mountain with a rocky peak rises against a blue sky with light clouds.

Simple Application Timing

Be Boulder.



University of Colorado **Boulder**

What is application timing?

- Analysis of a program's behavior using information gathered as the program runs
- Why do it?
 - Good way to improve efficiency of scripts
 - Identify performance problems
 - Many times required for allocation requests

How would you do it?

- Measure time of execution of an entire program or simply a code snippet
 - Loops
- Timing functions within programs
 - Python, Fortran, C++, R have functions that allow you to measure the execution time of small code snippets
 - Can also do this with the Linux “time” command
- Can make changes to code to improve efficiency
 - Or...it's just informational

The Linux Time Utility

- The first place to start when profiling your program

```
time mpirun -np 4 ./prog.mpi
```

real	0m17.801s	Wall clock time
user	0m58.125s	Threads x Wall clock time
sys	0m0.081s	System overhead

Fine-grained Timing

- Often useful to time portions of a program
- Good idea when developing your own code
- Tough when its 3rd party software
- Useful functions:
 - Fortran: `system_clock`
 - C++ `clock()`
 - Python: `time.time()`
 - R `sys.time()`