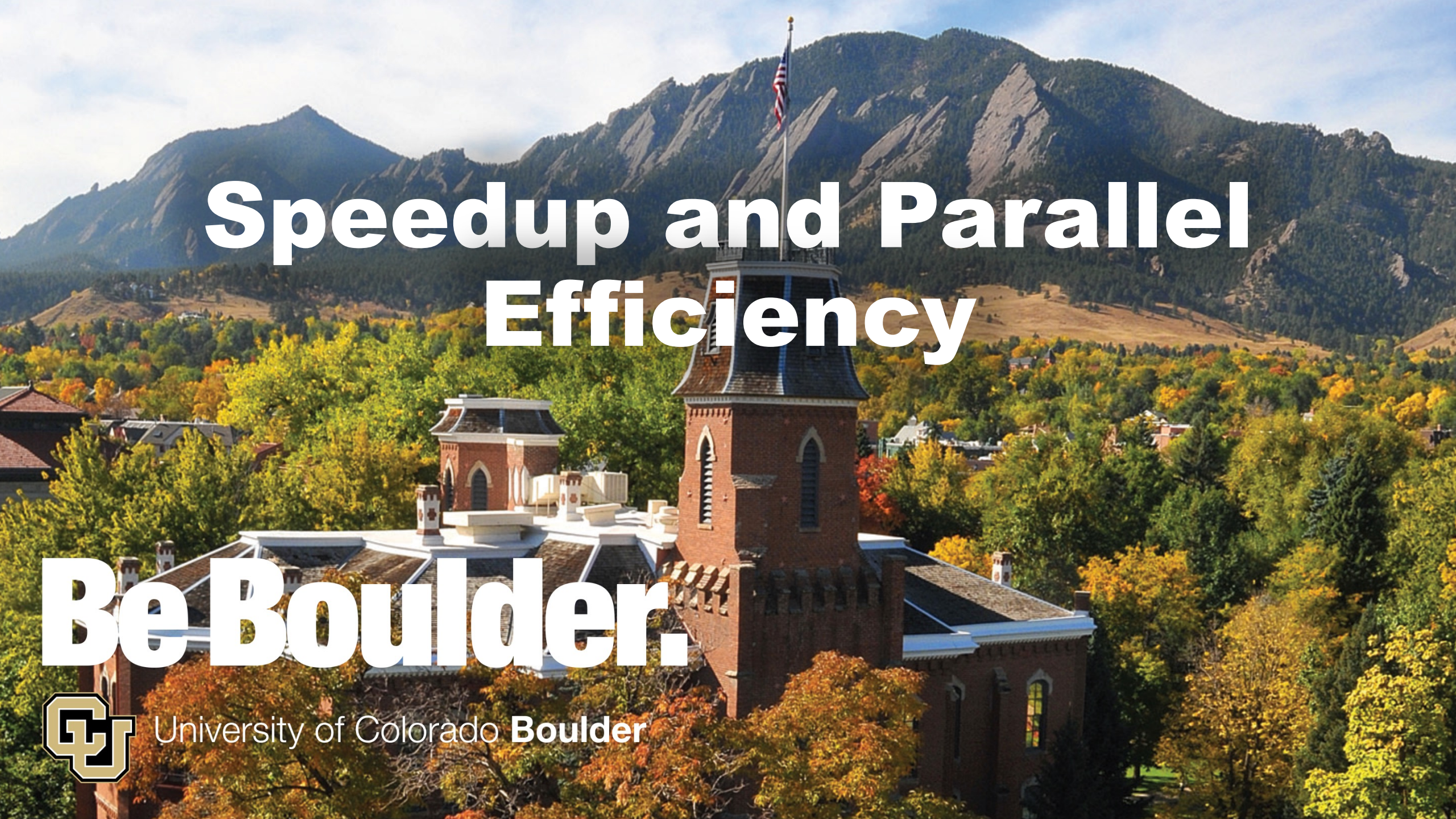

Speedup and Parallel Efficiency
Be Boulder.
University of Colorado Boulder

What is this?

- We've touched on the idea of HPC architecture and how HPC infrastructure is best utilized for parallelized codes
- We've also talked a little about parallelized codes and why you might want to rewrite your serial code
- Let's talk about ways that you want to determine if that's the best plan
  - After all, it might take a lot of time an effort to convert your serial code to parallel
    - Is it worth it?

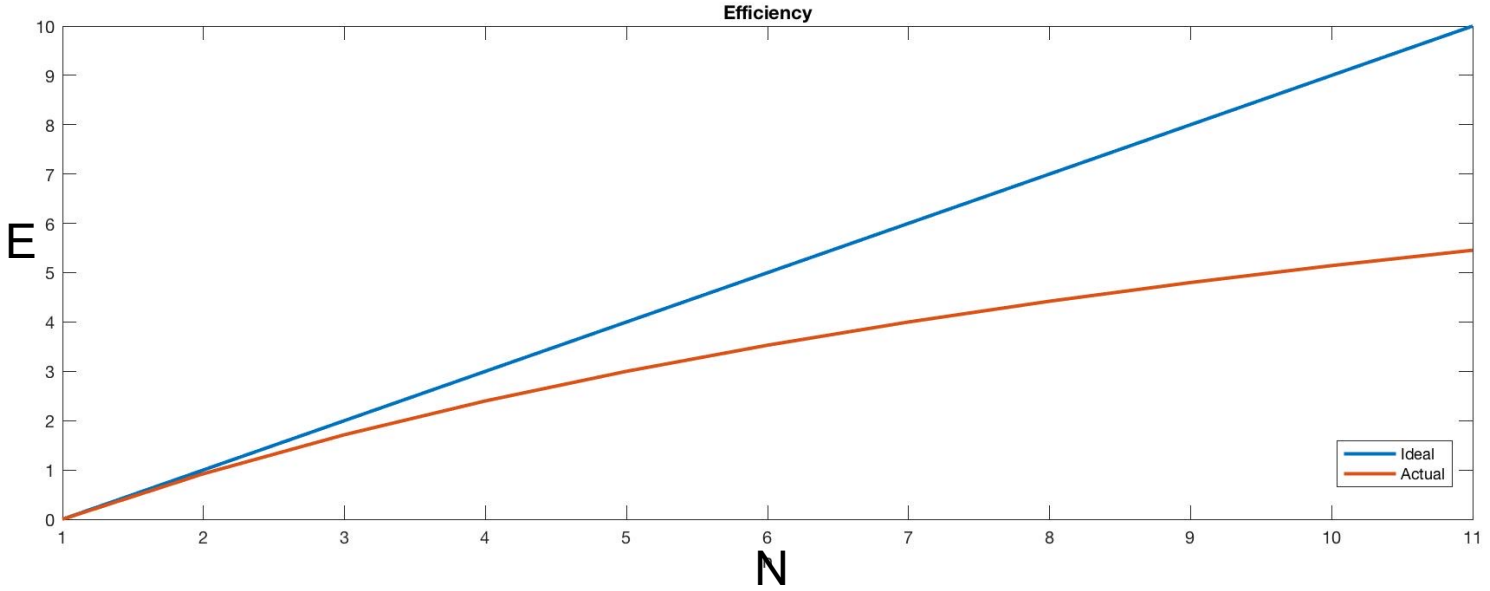
Speedup

- Speedup is defined as the ratio of the serial runtime (T_s) of a problem to the parallel runtime (T_p) of a problem
  - Parallel run time will depend on how many processors (p) are being used
  - Run time is the time from the start to end time of execution

$$S = \frac{T_s}{T_p}$$

- The linear speed up S is the ideal case

Efficiency



- Measures the amount of time that a processor is usefully utilized

$$E = \frac{expected\ time}{actual\ time}$$

- Expected time: T_s/N (problem size constant)
- Expected time: T_s (problem size increases with cores)
- Actual time: T_p
- Program that scales linearly has a parallel efficiency of 1

Uses

- Speedup and efficiency may be used to:
  - What would the improved time (speedup) of your code be if it was parallelized?
    - Depends in part on how much of the code can be parallelized
      - If only half, can at best see a factor of 2 speedup
    - Is this worth your time to parallelize your serial code?
  - To point out where you might have a diminishing return
    - At what point are you adding too many cores to solve your problem
    - Remember the lawn example? Overhead, communication
    - Generally want your efficiency to be around 80%