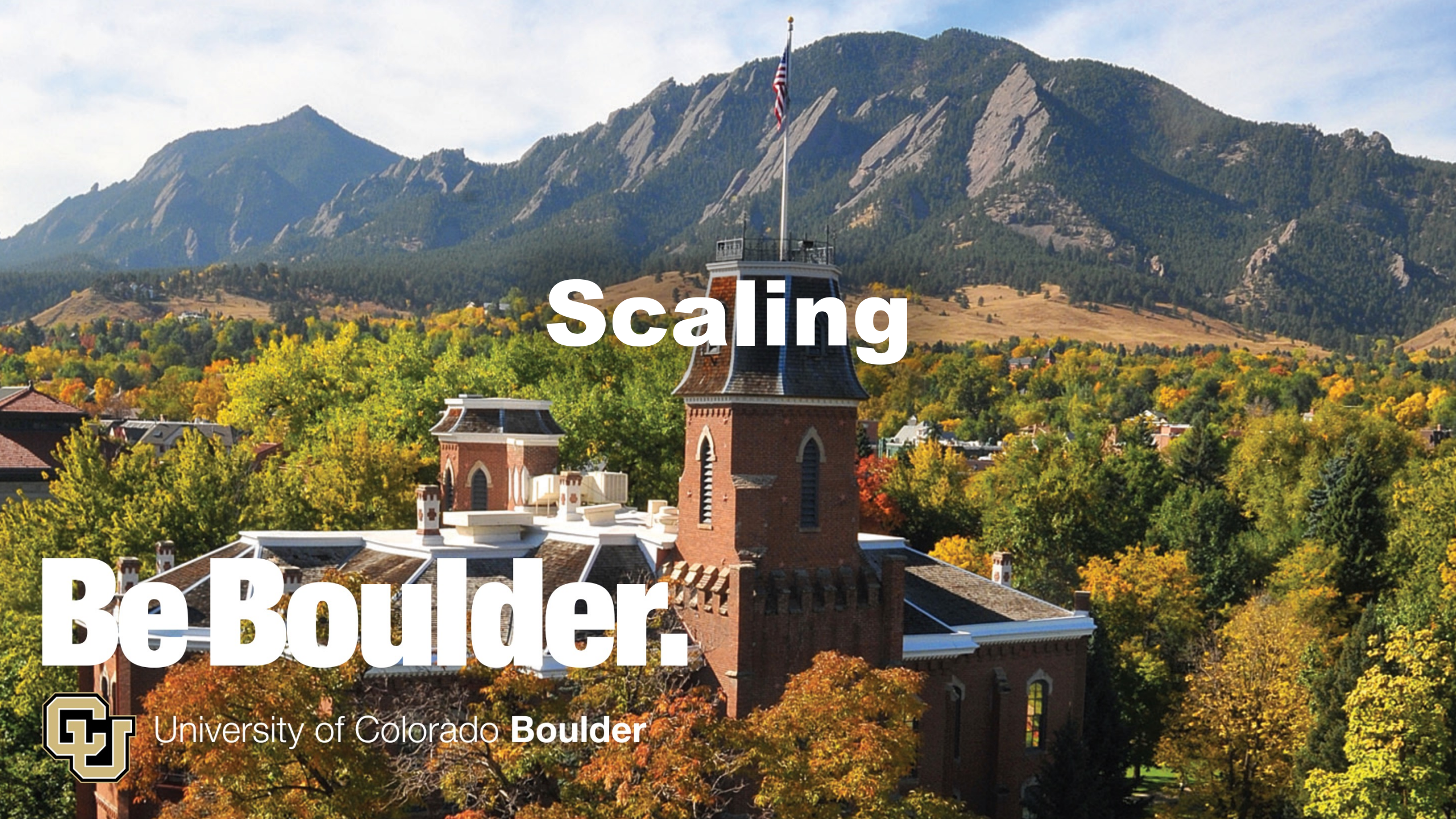

Scaling
Be Boulder.
University of Colorado Boulder

Scaling

- Scalability: the measure of how well a program utilizes multiple processors to (i) decrease time to solution or (ii) run a larger problem.

- How do we measure scalability?
  - Run test problems at multiple core counts and measure elapsed time
  - As you increase your core count, how does your application perform relative to ideal behavior?

Types of Scaling

- Weak scaling:
  - Increase problem size alongside the number of cores used.
    - *“Can I run a larger problem?”*
    - *“How large of a problem can I run in a given time frame and with given amount of compute resources?”*

- Strong scaling:
  - Fix the problem size, and increase the core count.
    - *“Can I decrease the time to solution?”*

Ideal Scaling

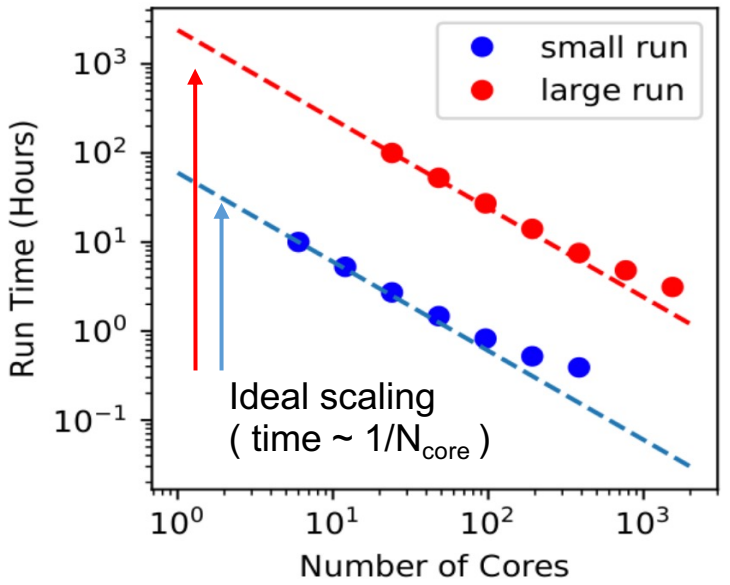
- The expected performance in the absence of any communication overhead
  - Never achieved in practice

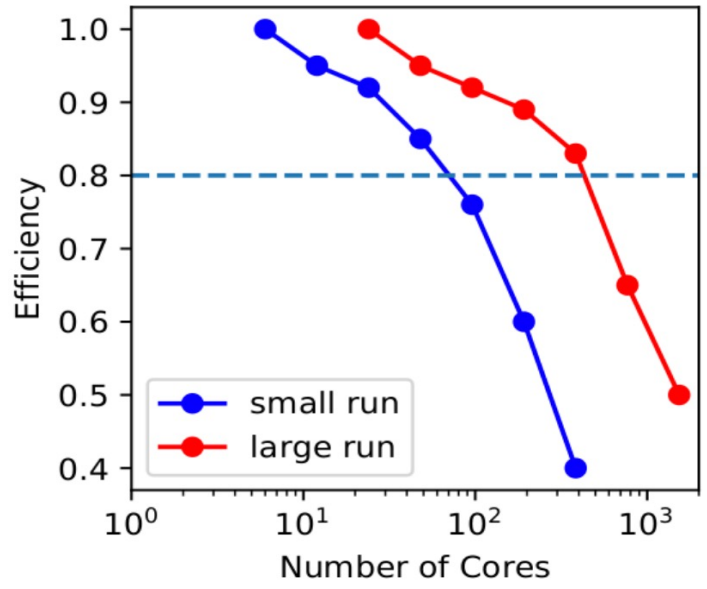
- Ideal weak scaling:
  - Time to solution is independent of core count

- Ideal strong scaling:
  - Time to solution decreases as 1/number of cores

Strong Scaling

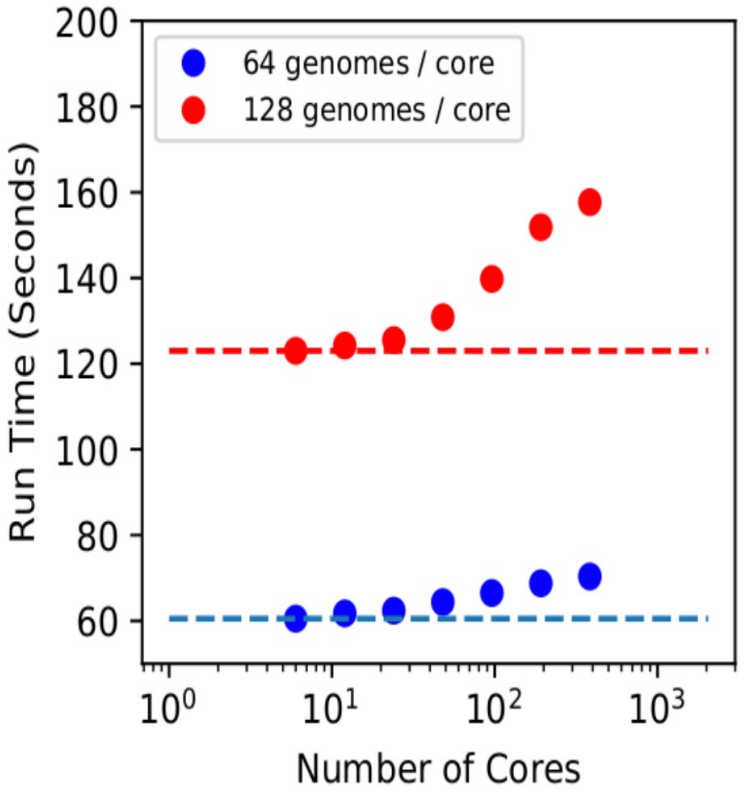
- Fix problem size, increase core count
- Shows how execution time decreases as number of processors increases
- Deviation from ideal scaling → Diminishing returns at large core counts (communication overhead)
- Characterized via efficiency:
  expected time / measured time
- Rule of thumb: run at > 80% efficiency

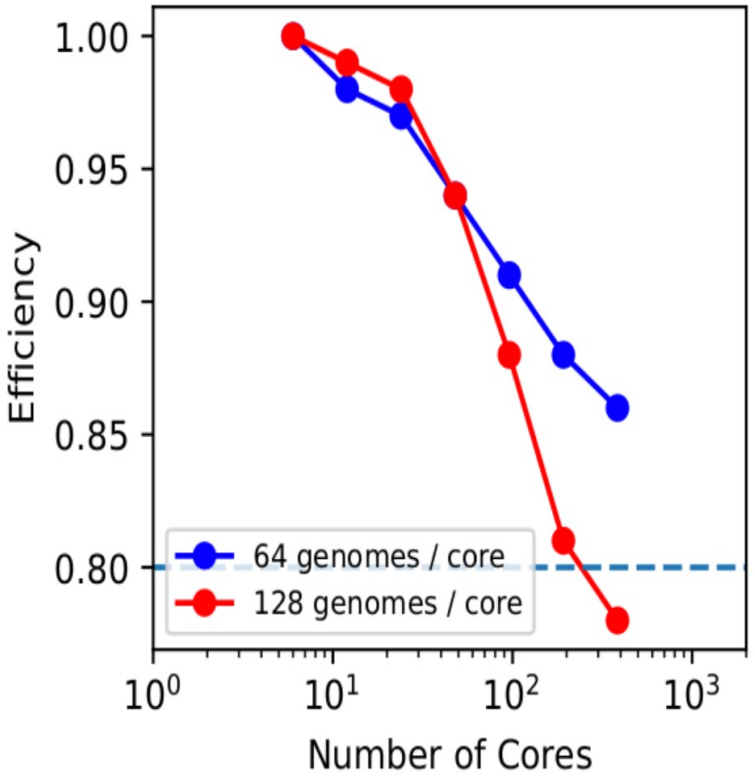




Weak Scaling

- Increase problem size in proportion to process count
- Work per core remains constant
- Ideal performance = constant run time





Scaling Study Procedure: Summary

- Pick a few representative problem sizes (e.g., number of images to process, size of grid domain, number of genomes to examine)
- Decide on a sensible duration for the test. If code iterates in some fashion, run for just a few iterations
- Run code for a short time at each problem size and with multiple core counts
- For each test, record or calculate elapsed time, expected time, efficiency
- Plot results and ask for CPU time!