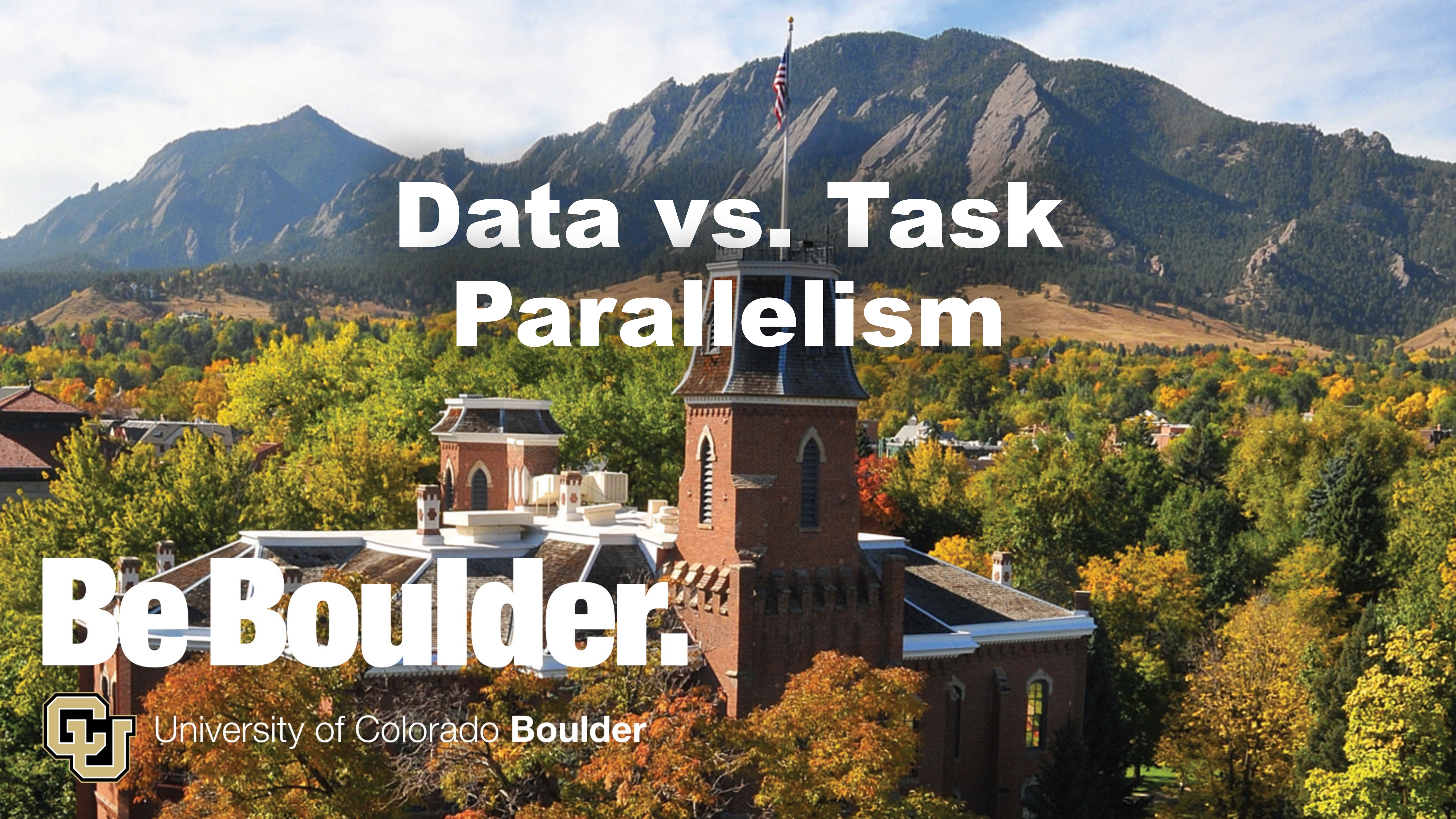
Data vs. Task Parallelism



Be Boulder.

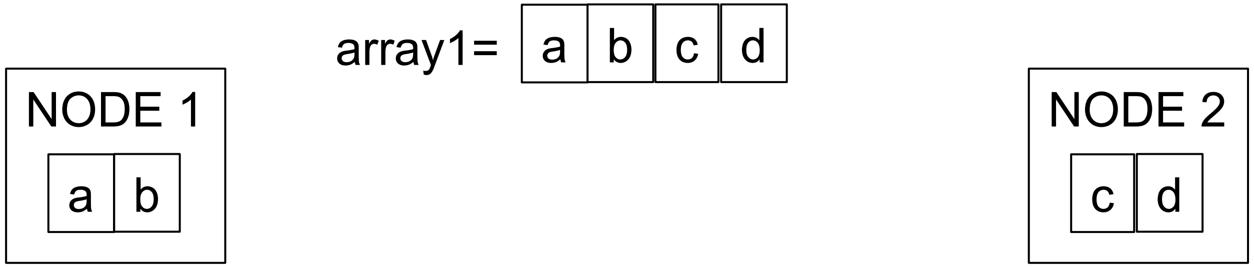
University of Colorado **Boulder**

Data and Task Parallelism

- Earlier discussed data parallel memory methods
- One of them was distributed memory, wherein different memory pools were accessed by different cores on a single node
- Data and task parallelism are a similar concept
- Data parallelism
  - Distribute the data across processors
- Task parallelism
  - Distribute the compute tasks across processors

Data Parallelism

- Different parts of a dataset are distributed across nodes



Task Parallelism

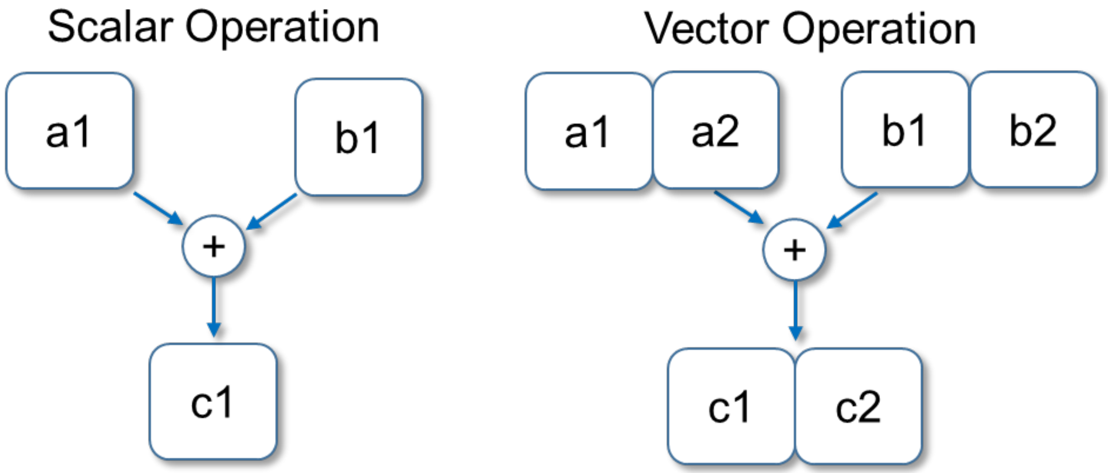
- Each processor executes a different task on the same dataset
    - Tasks (code, instructions) are spread out among the cores
    - Might be same instructions/code or different
- Distributed programming
- Example: Calculating wind speed from vector components across a geographic area. Divide vector calculation among processors

Data Parallelism - SIMD

- Two types of data parallelism we'll discuss here
  - SIMD – Single Instruction, Multiple Data
  - SPMD – Single Program, Multiple Data
- SIMD
  - Carry out the same instruction simultaneously multiple times across different elements of a dataset
  - Vector operation
    - Addition, subtraction, multiplication, division
  - Have to prepare your data to be vectorized

Vectorization

- Simply put, performing multiple math operations



```
Non-vectorized
a=rand(1,4)
b=rand(1,4)

for i=1:length(a)
   c(i)=a(i)+b(i)
end
```

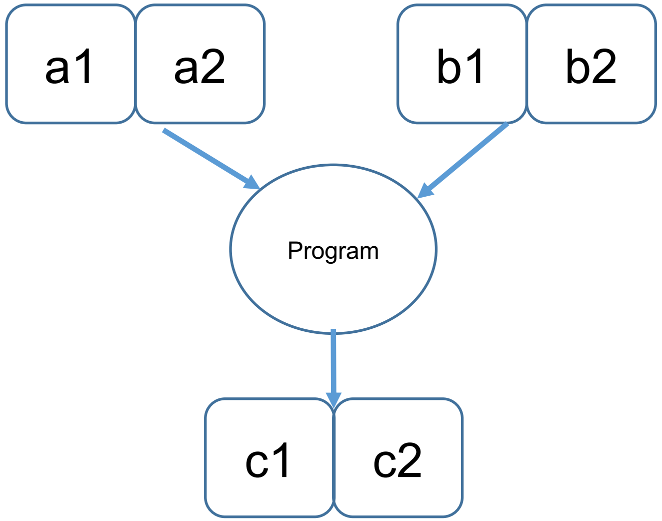
```
Vectorized
a=rand(1,4)
b=rand(1,4)

c=a+b
```

Python, R, etc.
Compiled languages – compiler can handle it

Data Parallelism - SPMD

- SPMD
  - Carry out the same program multiple times on different elements of a dataset
  - Calculate the wind direction from wind components



Why do this?

- Cleaner code
- Faster execution time
  - Eliminating loops!
- Usually not too challenging
  - Many languages have functions that make this easy to perform