ME 5311: Week 14

Today

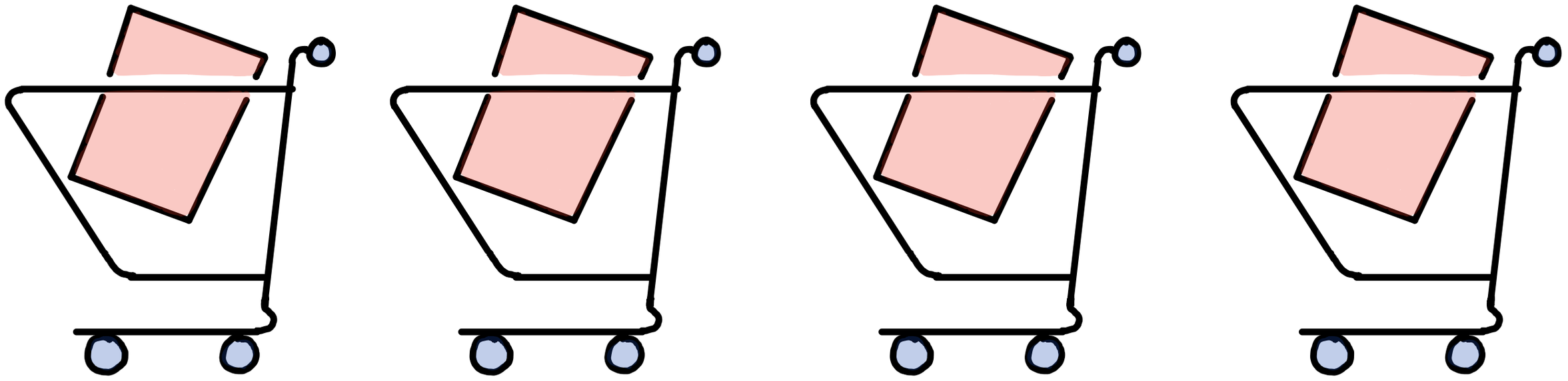
Introduction to parallel computing

- Goal: Understand basic principles of parallel computing for CFD
- Introduce and discuss the two most common approaches
- Get a flavor of “real” programming

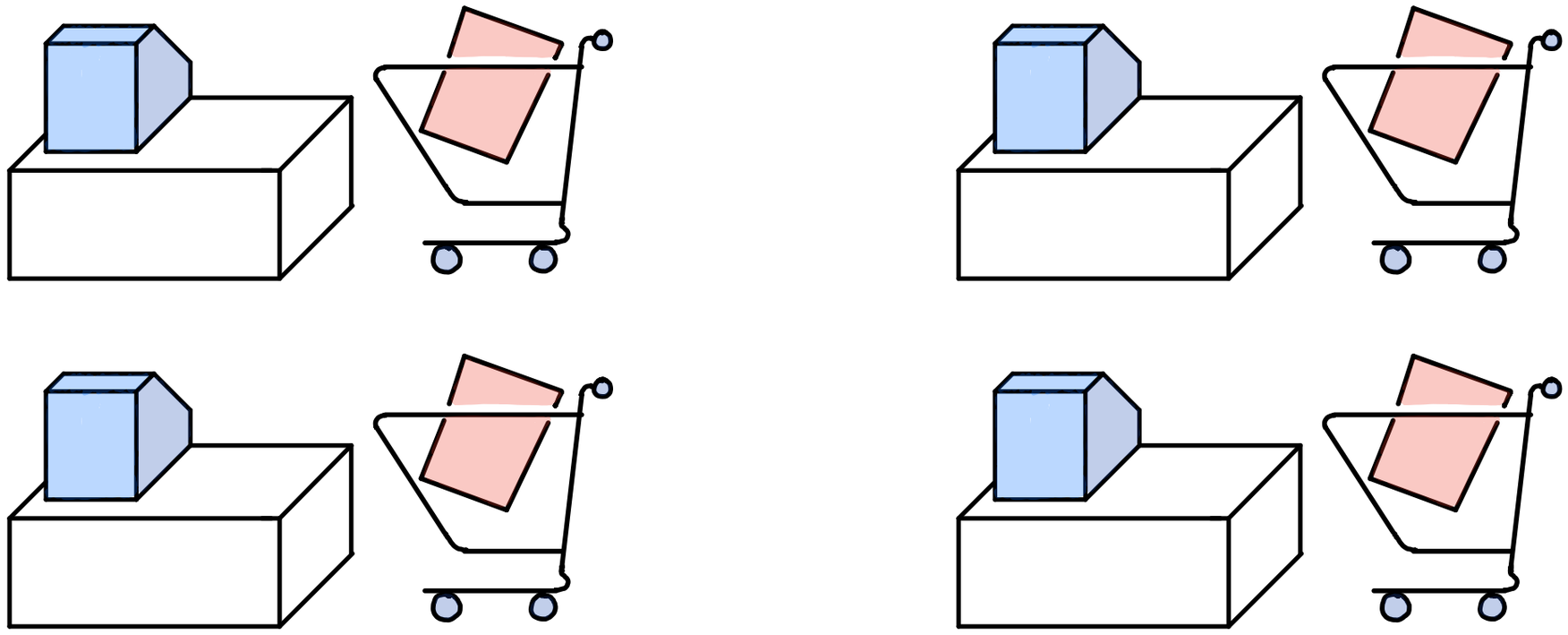
What is parallel computing?

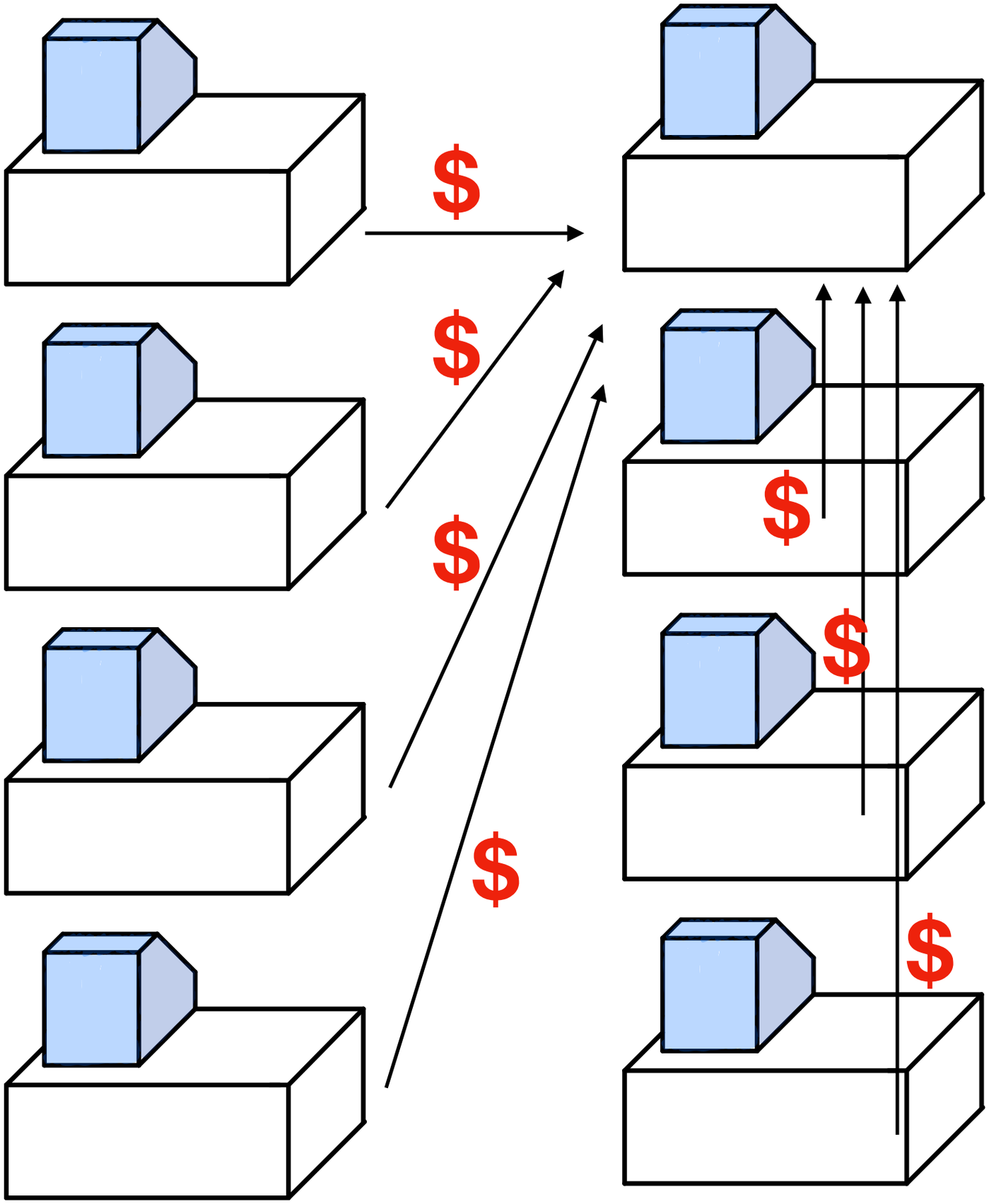
Need for parallel computing

- We would like to solve a “big” problem
- “Big” can mean:
  - Takes a long time to compute: split the work on multiple CPUs
  - Requires a lot of computer memory: use memory from different “computer nodes”



Splitting the work: granularity



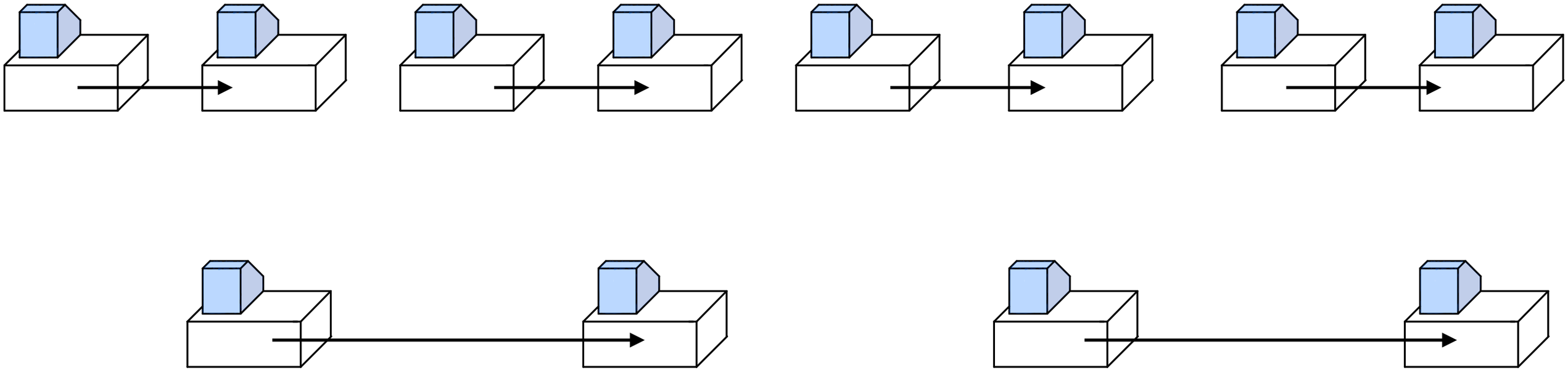


Processes must:

communicate

coordinate

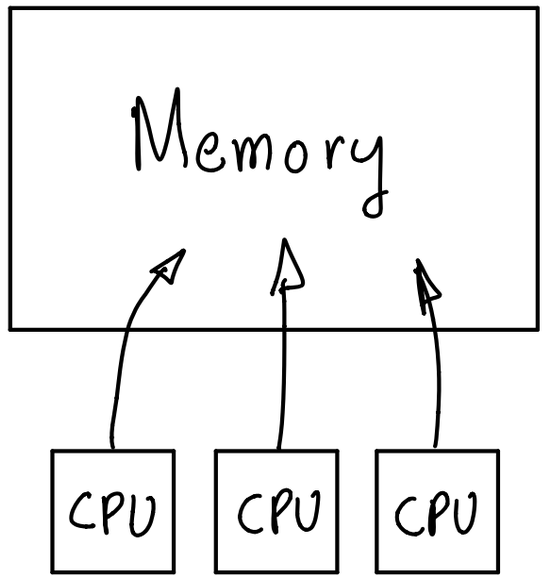
combine results



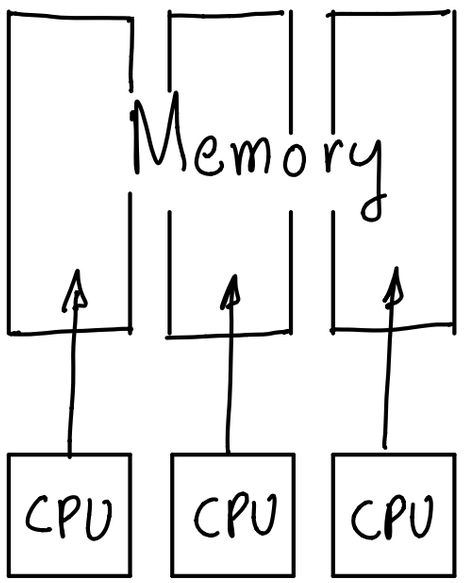
Basic principle: Program copies itself

Traditional parallel models

- **Shared memory**:
  OpenMP, Threads

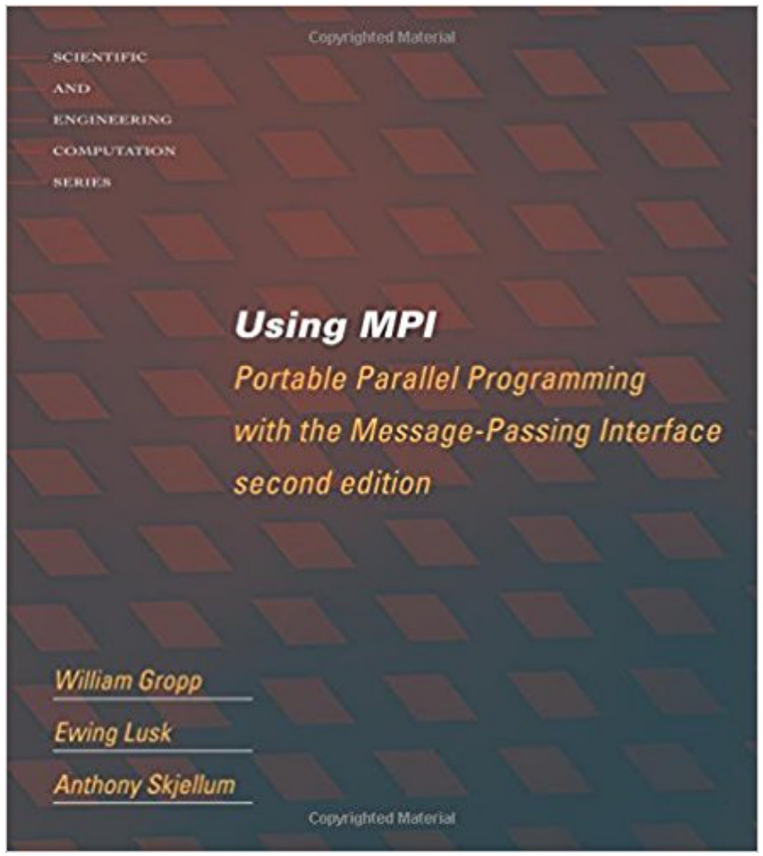


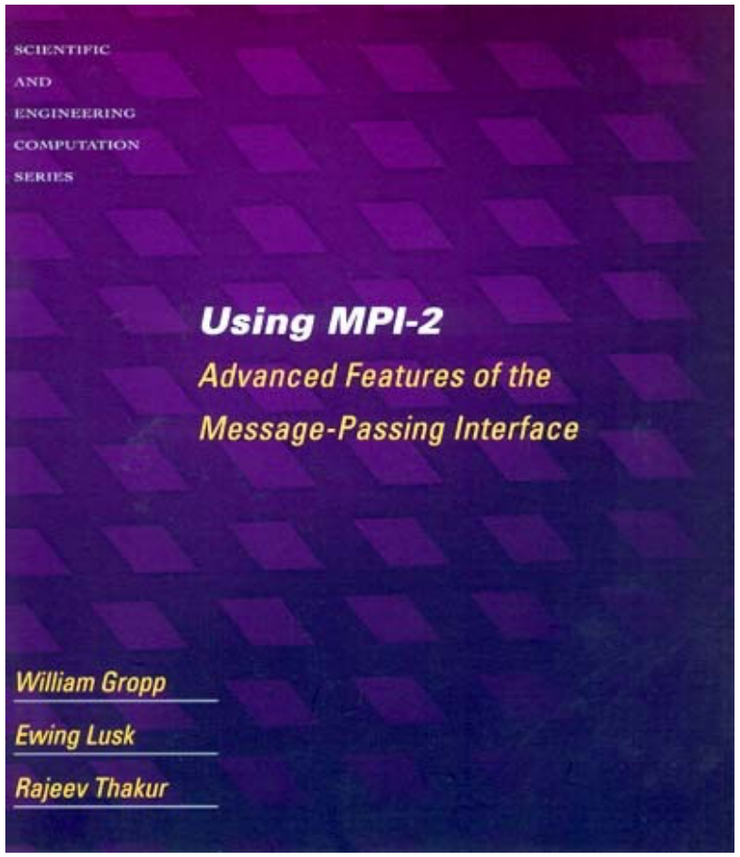
- **Described memory**:
  Message Passing Interface



Implementations

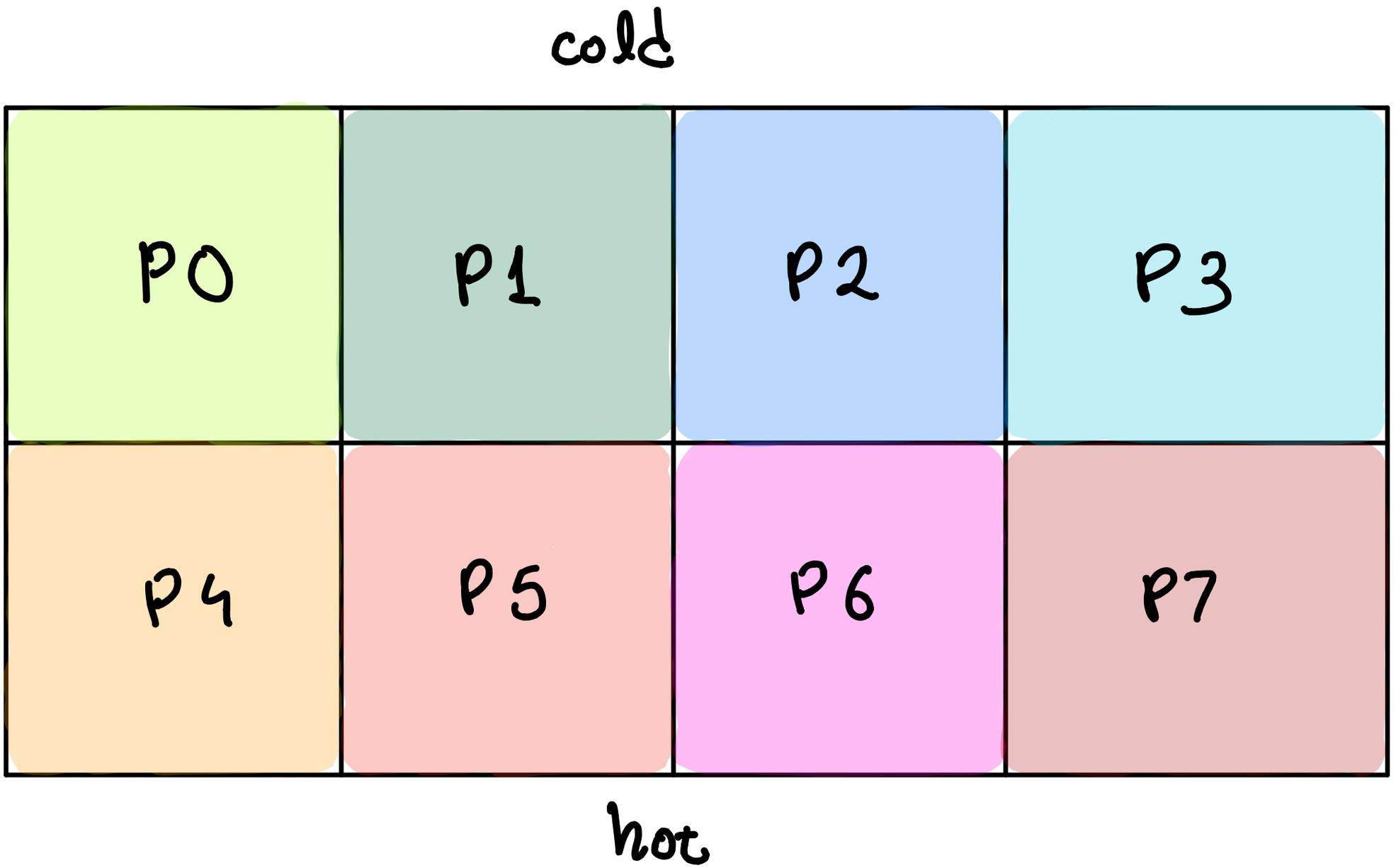
Goal is to “tell” the computer what you want to do…
e.g., send data from X to Z



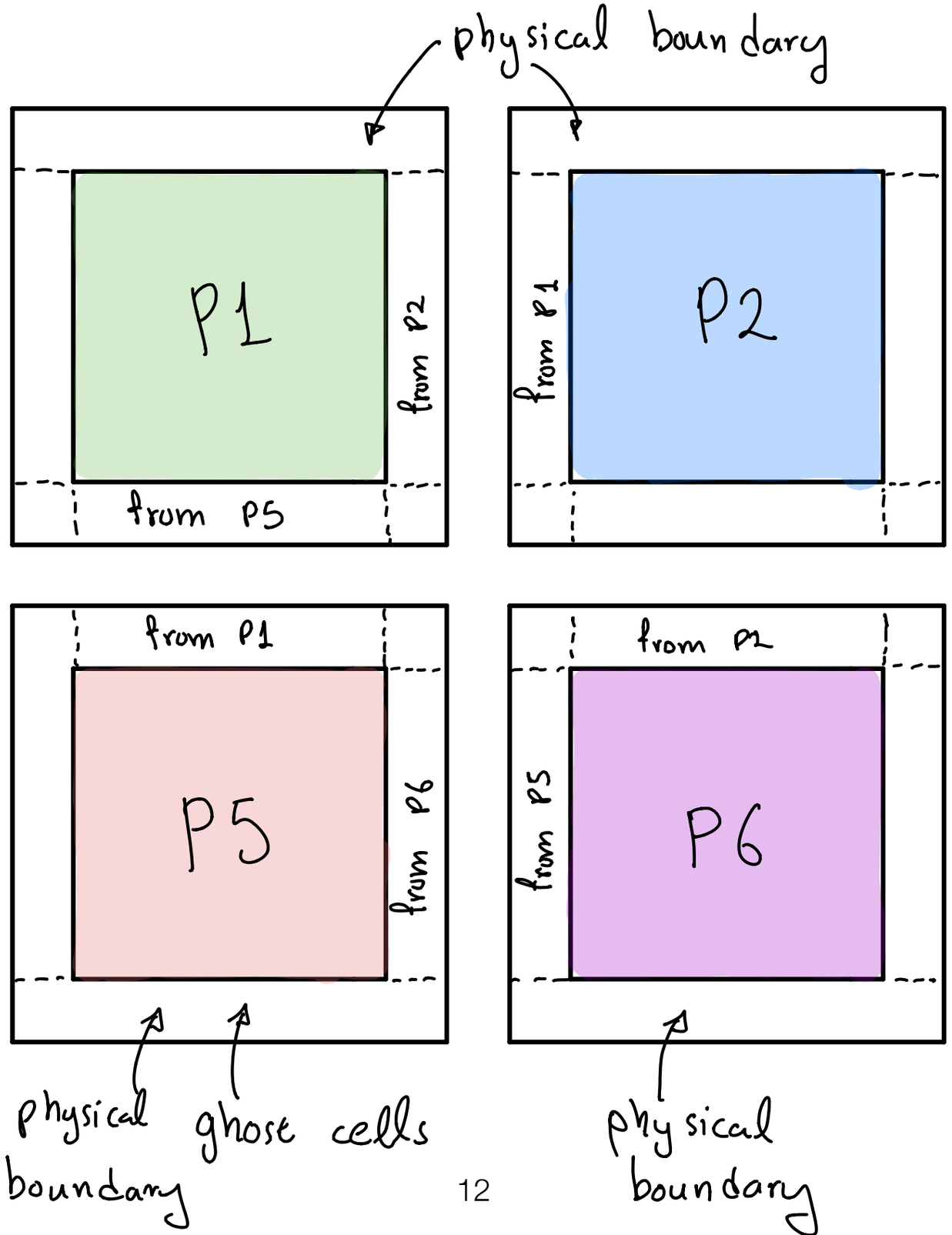


Implementation literature/books assume’s reader is experienced programmer

Computational domain decomposition



Process communication



Non-blocking communication

```
call mpi_isend(var(1,n2-2*nghost(1)+1,nghost(2)+1),1,xstride,
pxfwd, 130,  MPI_COMM_WORLD, req(1), ierror)

call mpi_irecv(var(1,1,nghost(2)+1), 1, xstride, pxback, 130,
MPI_COMM_WORLD, req(4), ierror)

call mpi_isend(var(1,nghost(1)+1,nghost(2)+1), 1, xstride, pxback,
140, MPI_COMM_WORLD, req(2), ierror)

call mpi_irecv(var(1,n2-nghost(1)+1,nghost(2)+1), 1, xstride,
pxfwd, 140, MPI_COMM_WORLD, req(3), ierror)

…

call mpi_waitall(16,req,stats,ierror)
```