Parallel Computing
An MPI Case Study

Lena Kanellou, Manolis Ploumidis

ICS-FORTH

What is parallelism?

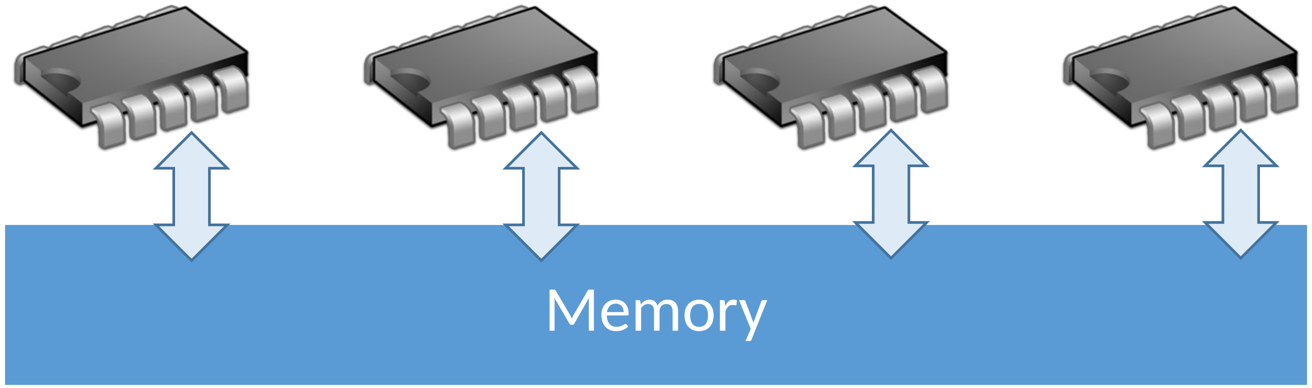
- Several processors collaborate to solve a problem, i.e. to execute a program

Why parallelism?

- Need for more and more performance and capacity
  - Scientific computing
  - Commercial computing
  - Computer graphics
- Exploit parallelism available in modern clusters - supercomputers

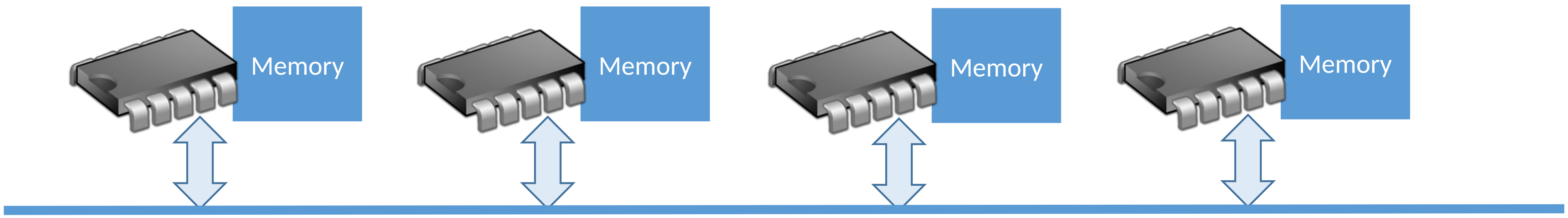
Shared memory multiprocessors

- May contain up to hundreds of processors
- Processes communicate over memory



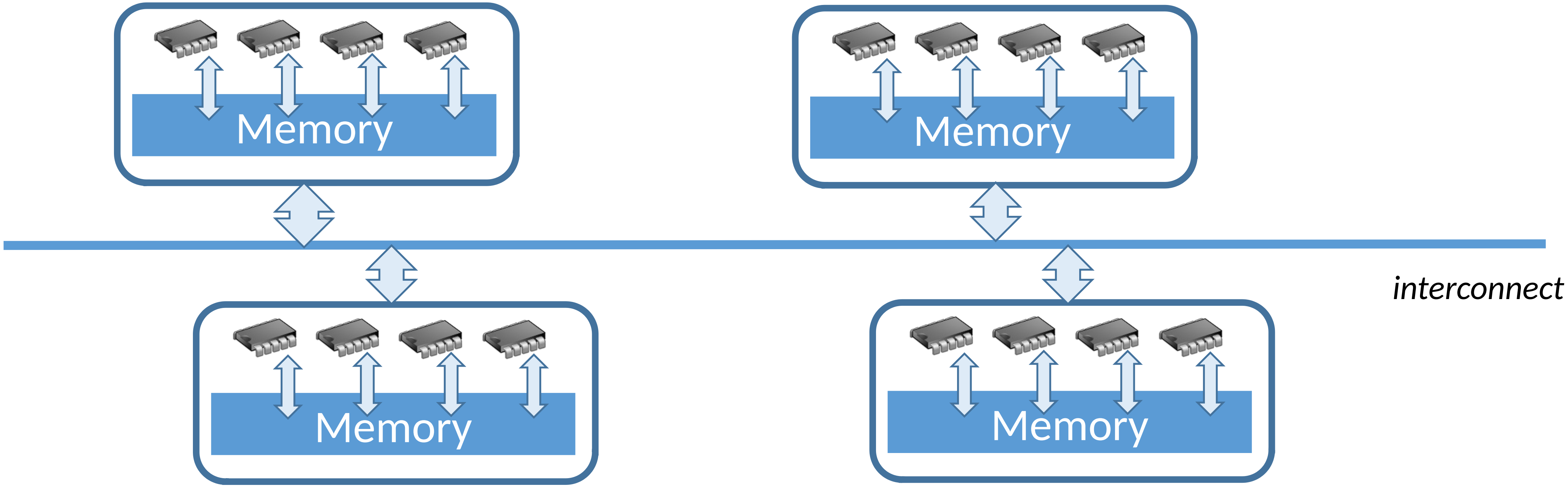
Message passing parallel machines

- Building block: full computing nodes
  - Processor
  - Memory
  - I/O controller
  - Network interface



Common SC/cluster paradigm

- Hybrid



How do we exploit all these resources?

- Hide machine/architecture details
- Programming models
  - Supported/tunned for each machine
- OpenMP
  - Shared memory multiprocessing architectures
- Message passing
  - Distributed memory architectures

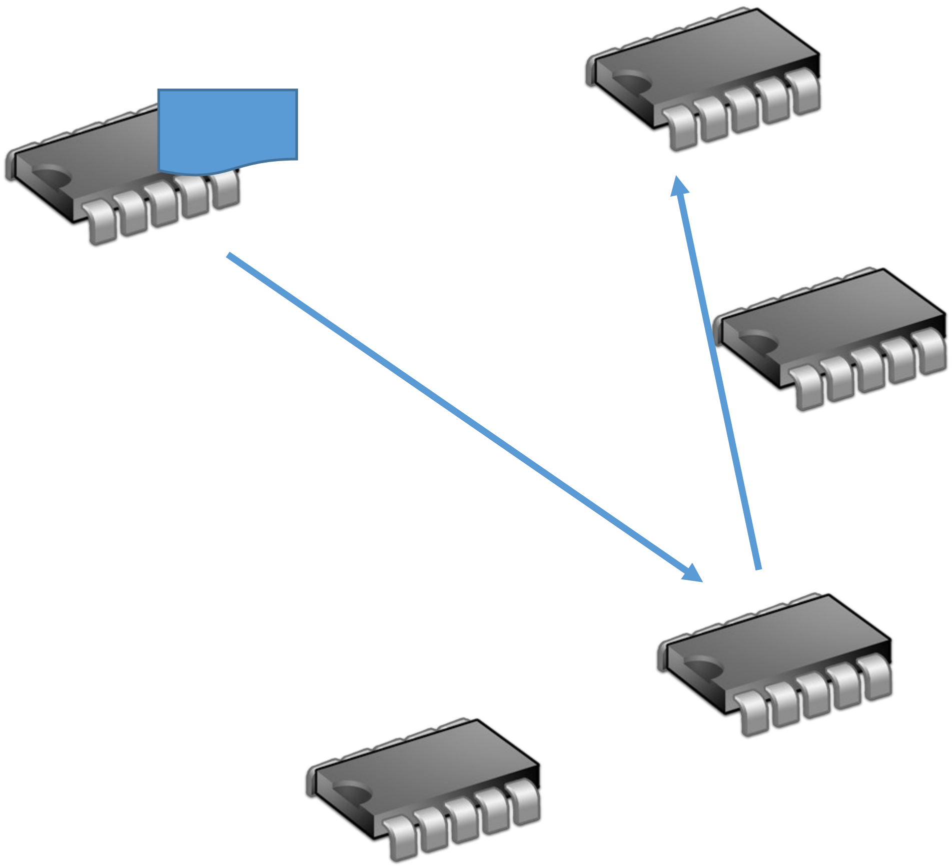
What is MPI

- Message Passing Interface
  - A specification for creating message passing libraries
- Originally designed for distributed memory architectures
- Currently adapted to handling various communication substrates
  - Shared memory
  - Distributed memory
  - Hybrid

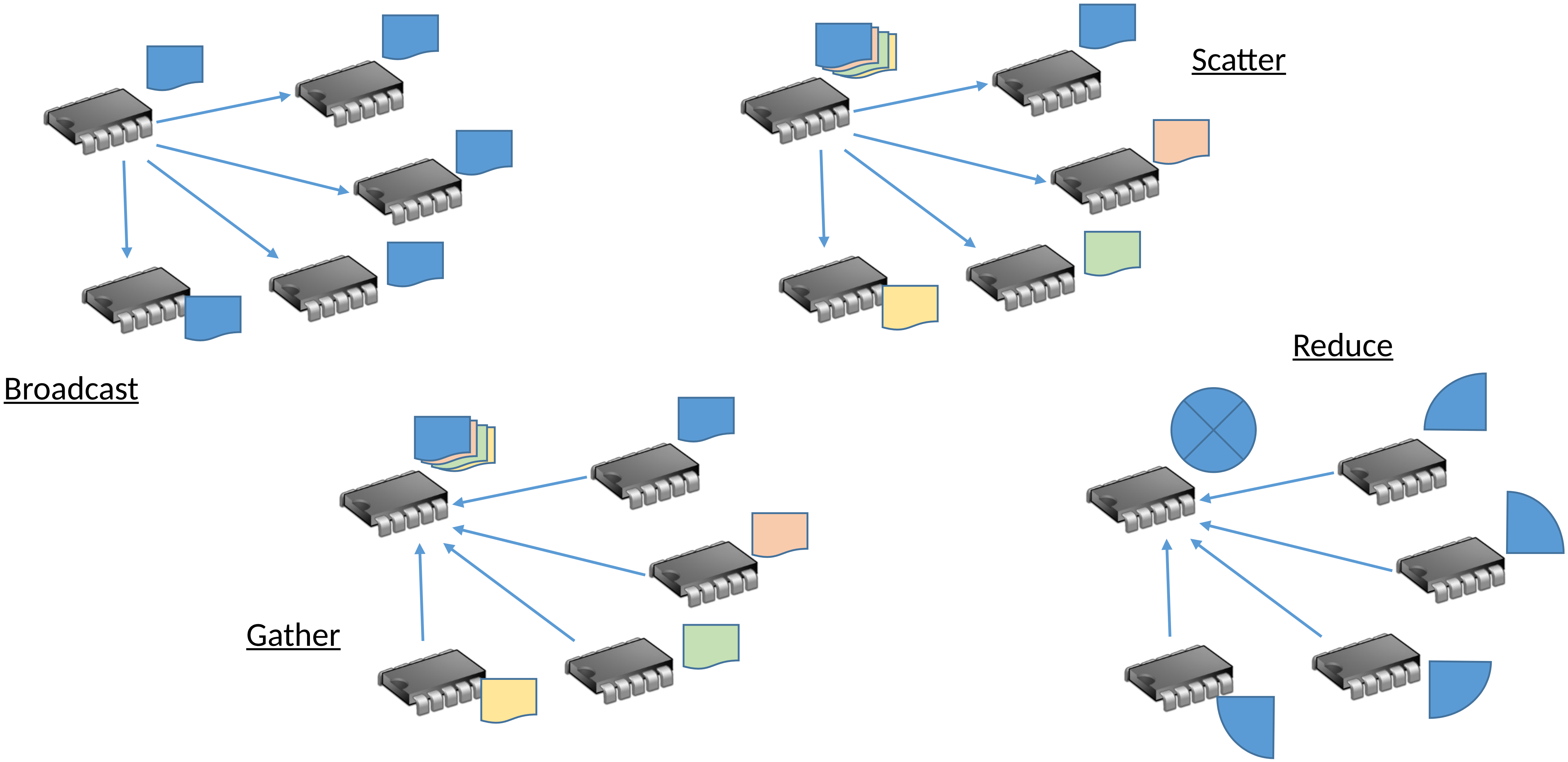
Communication primitives

- Point-to-point
  - Sender-receiver
- Collectives
  - One-to-all
  - All-to-one
  - All-to-all
- One sided primitives

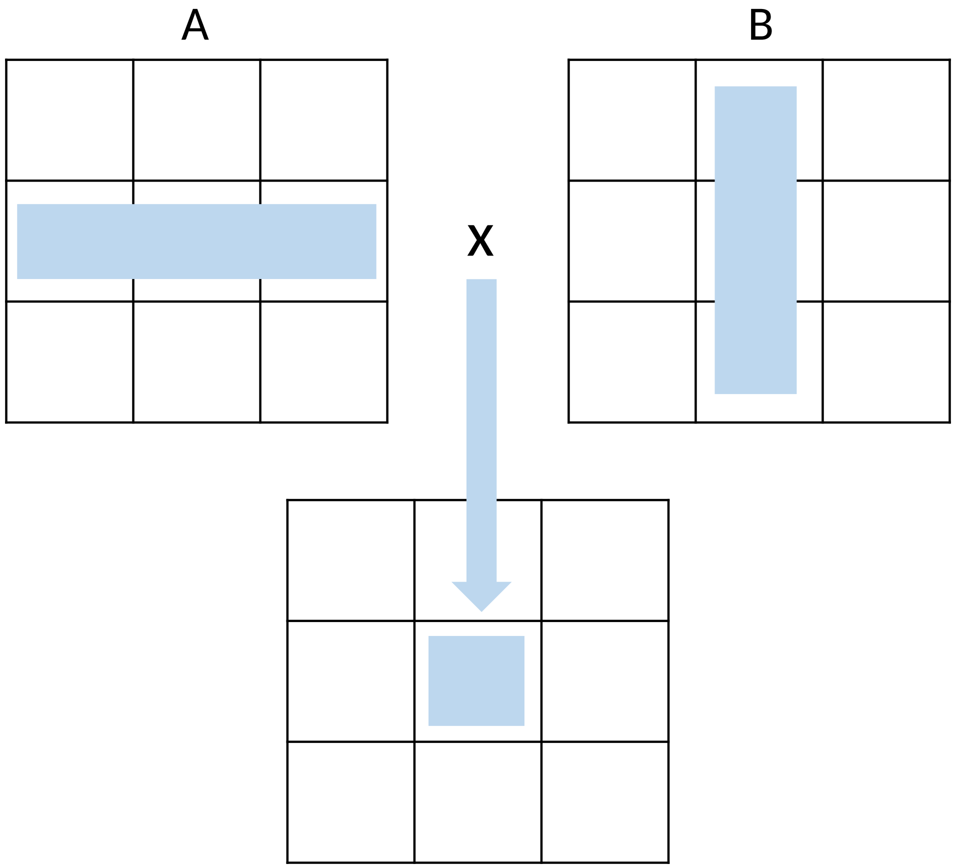
Point-to-point Communications



Collective Communications

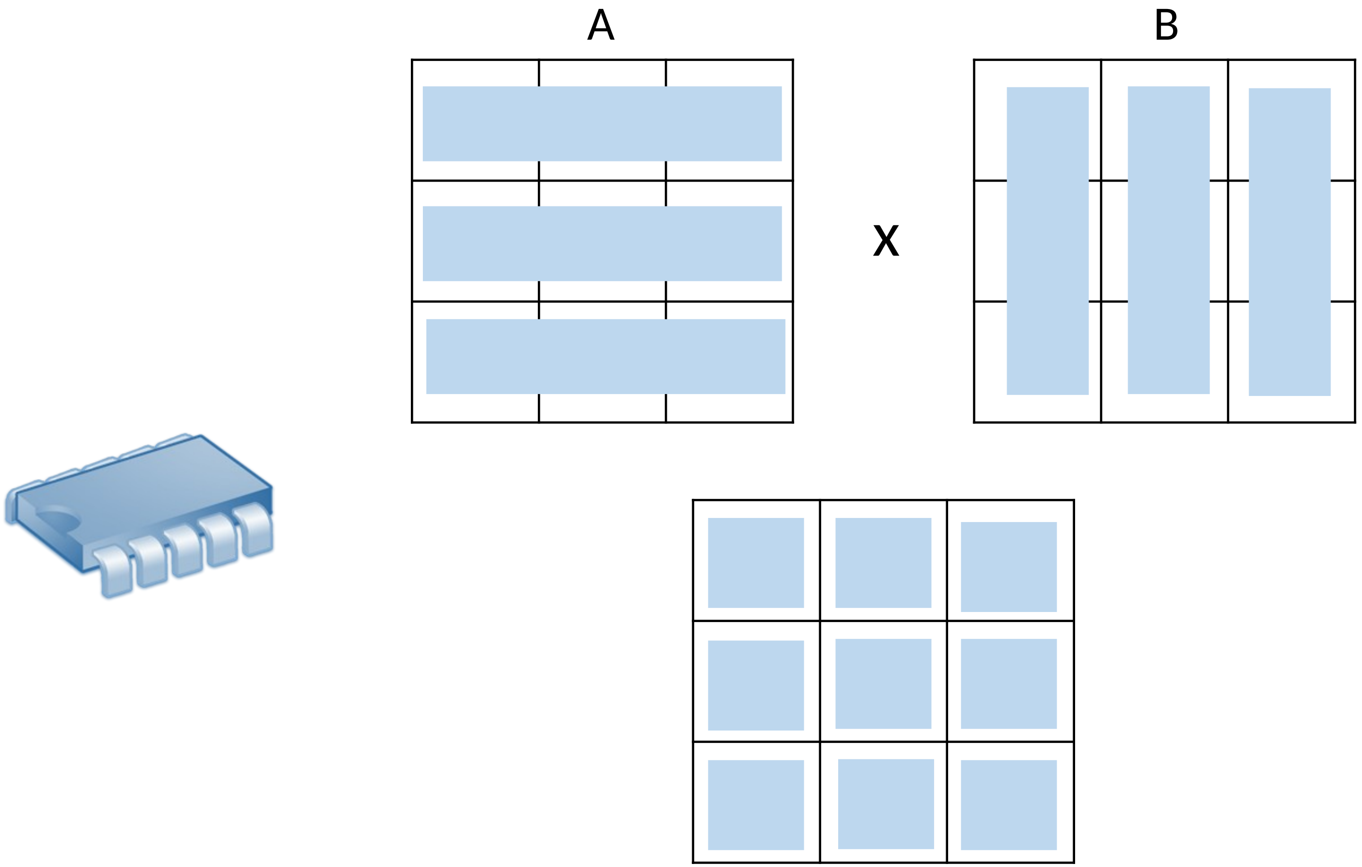


A parallelizable problem: Matrix multiplication



- Matrices occur when studying models with multiple variables
- In Biology, for example:
  - *Allele frequencies mutation*
  - *Conformational states of molecules*
  - *Growth of a structured population*
  - *Meta-population modeling*
  - *Age-structured population*

Matrix multiplication: the sequential case



Matrix multiplication: a possible parallelization