Parallelization
Hardware architecture

Contents

- Introduction
- Memory organisation
- Topology
- Example

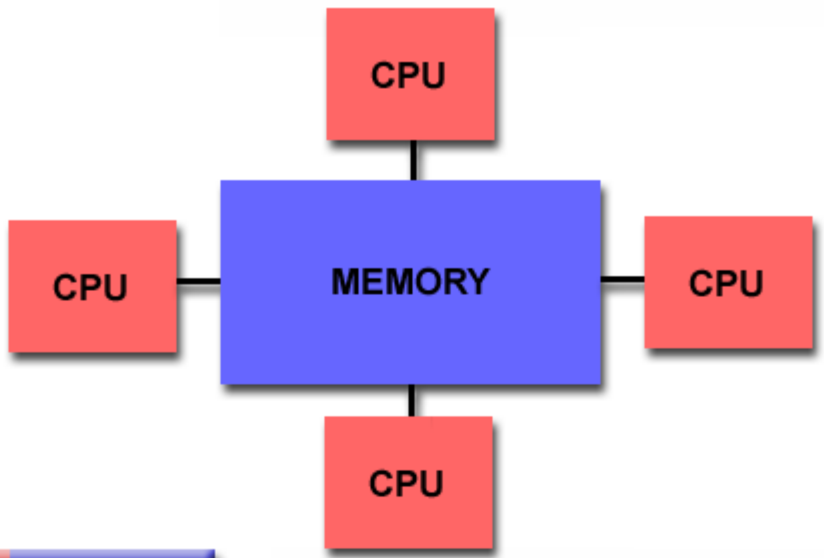
Why Parallel Computing

Primary reasons:

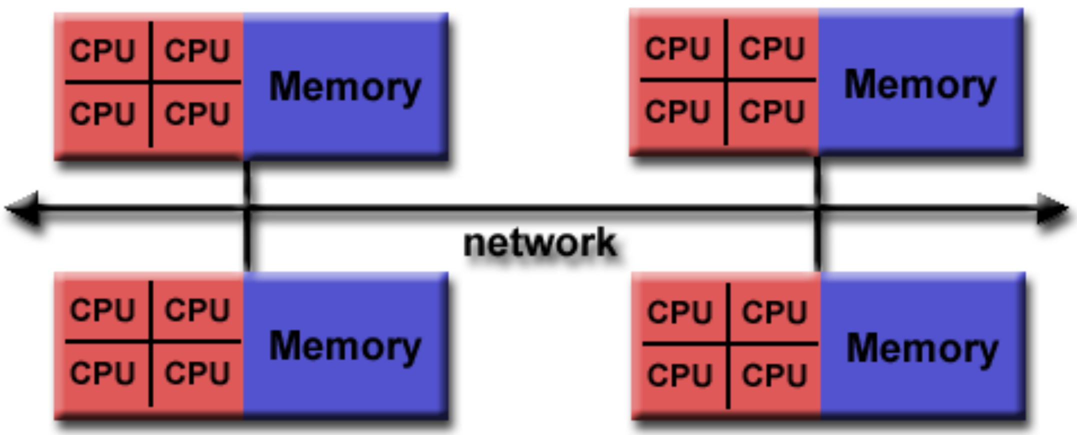
- Save time
- Solve larger problems
- Provide concurrency (do multiple things at the same time)

Memory organization

- Shared memory (SMP)
  - UMA
  - NUMA
    - CC-NUMA



- Distributed memory

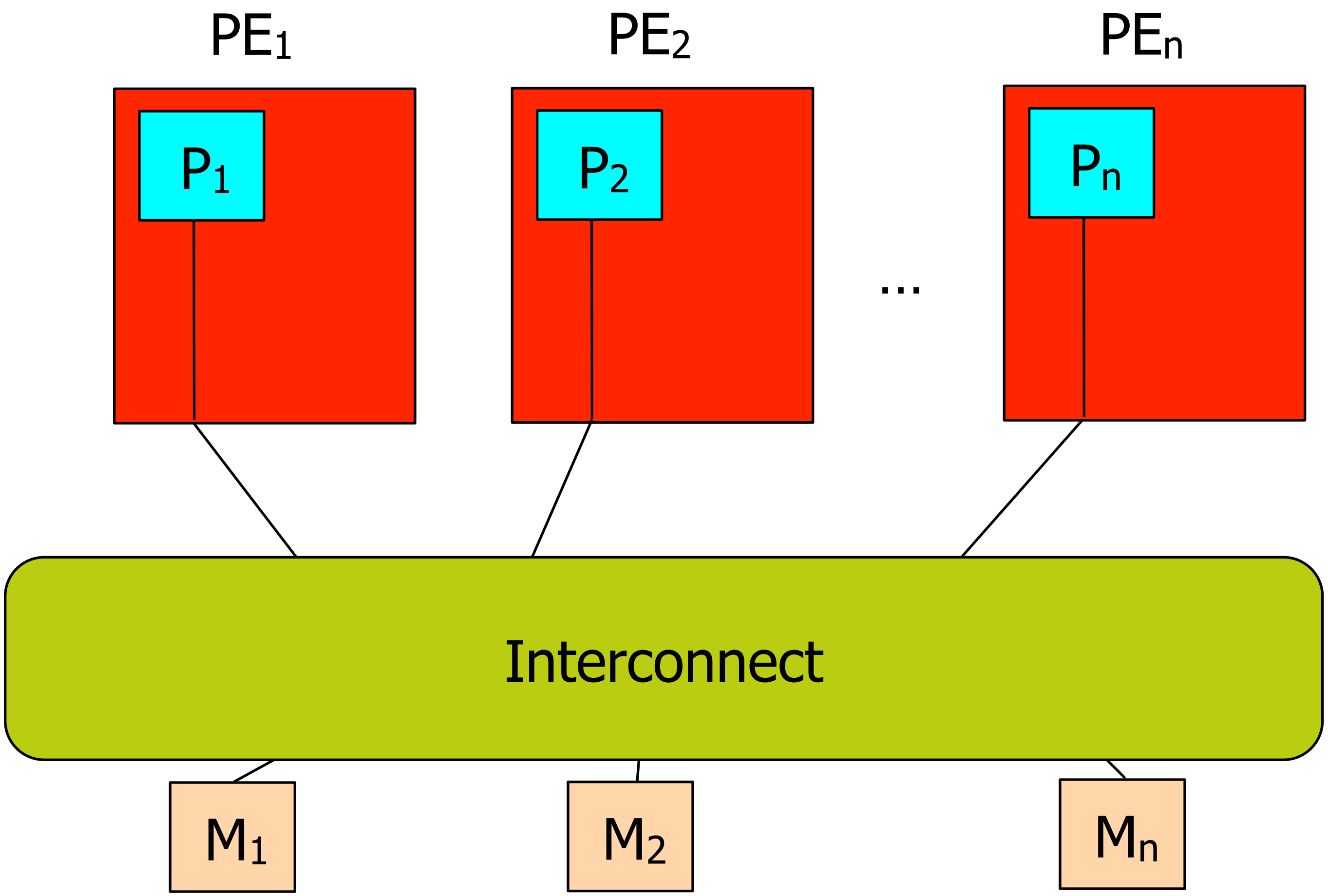


Memory Organization

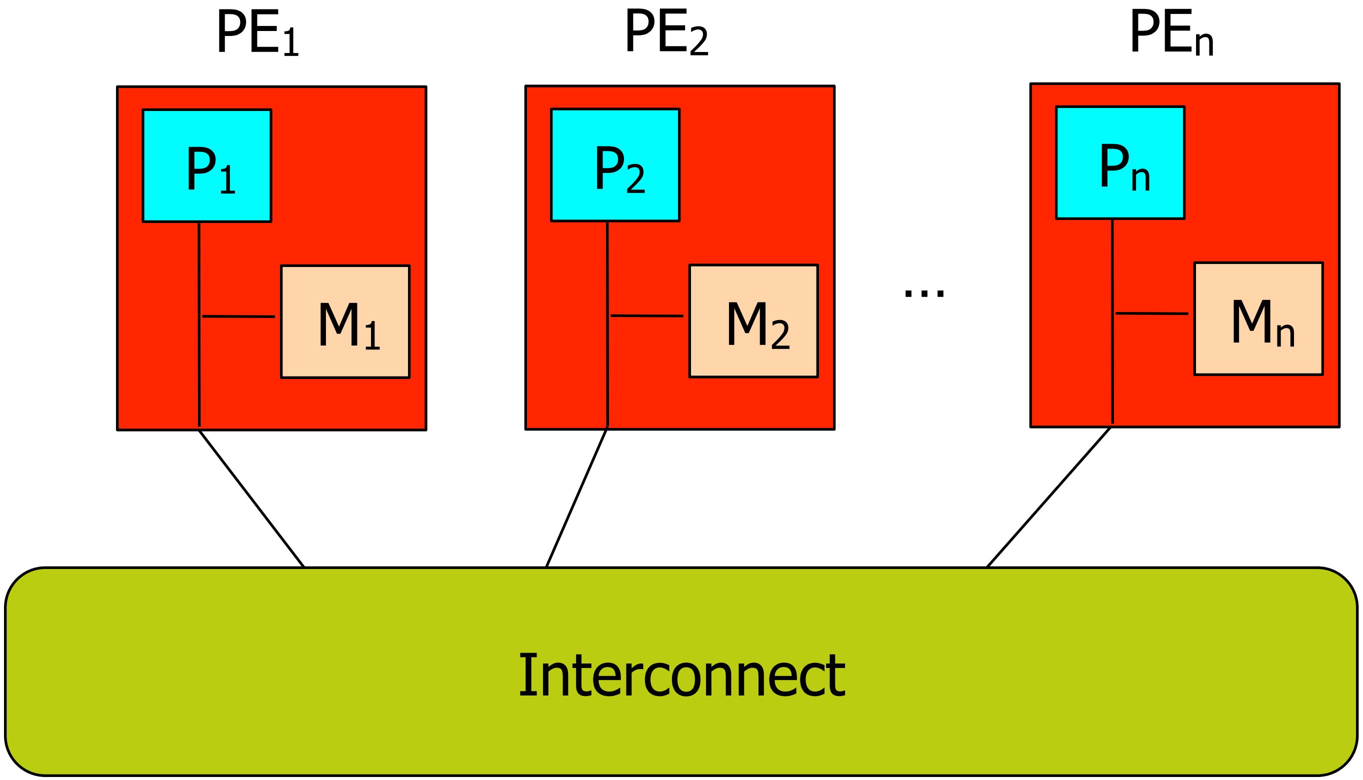
Symmetric shared-memory multiprocessor (SMP)

- Implementations:
  - Multiple processors connected to a single centralized memory
    – All processors see the same memory organization
    -> **uniform** memory access (UMA)
  - Shared-memory
    – All processors can access the entire memory address space through a tightly interconnect between compute/memory nodes
    -> **non-uniform** memory access (NUMA)

UMA (Uniform Memory Access)

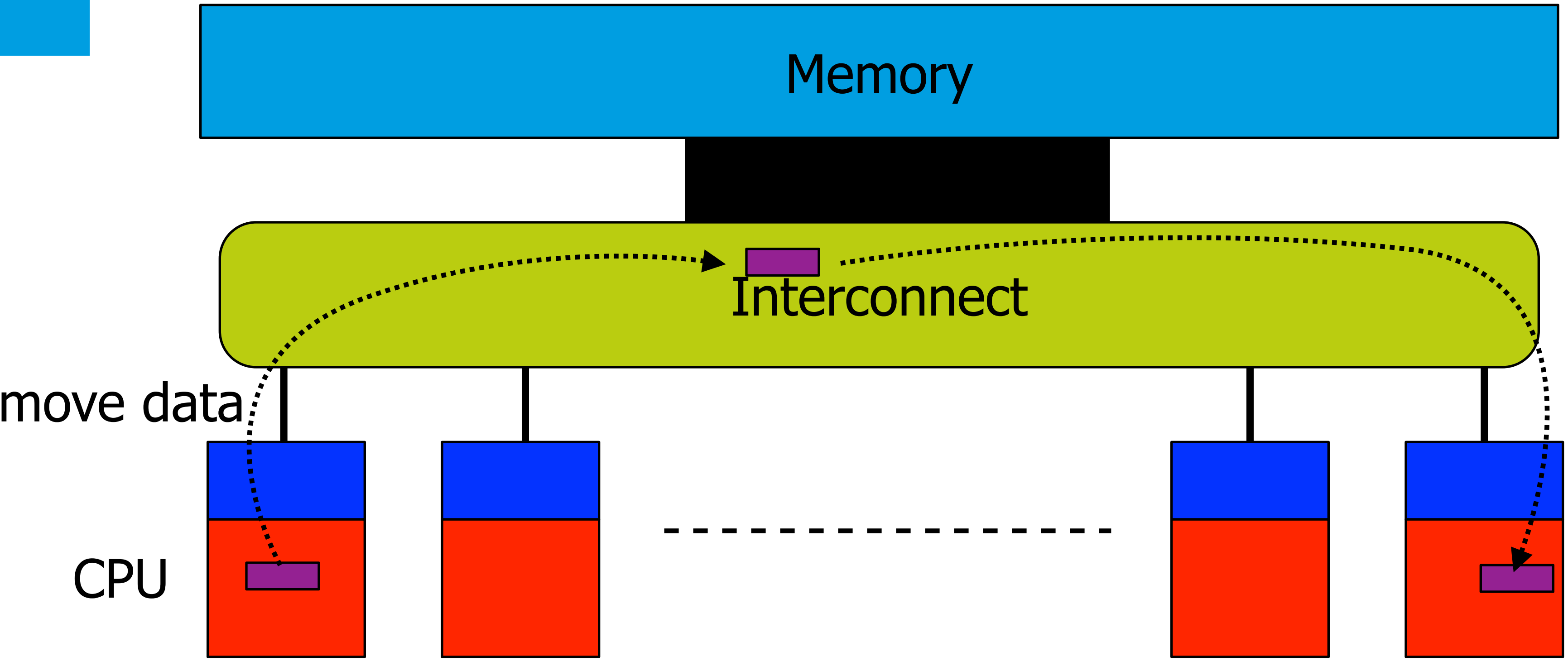


NUMA (Non Uniform Memory Access)



A Shared Memory Computer

Data movement is transparent to the programmer



SMP = Symmetric Multi-Processor

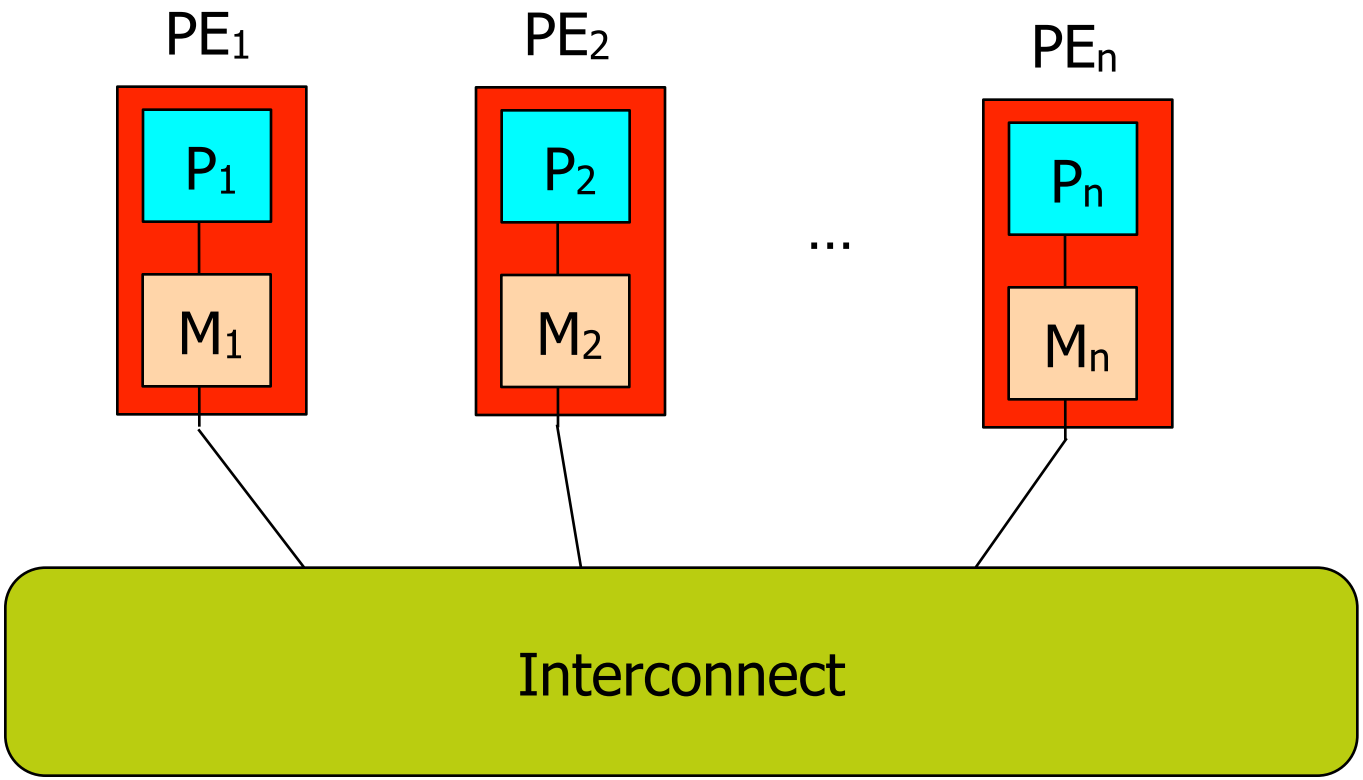
Note that the CPU can be a multi-core chip

Memory Organization

Distributed-memory

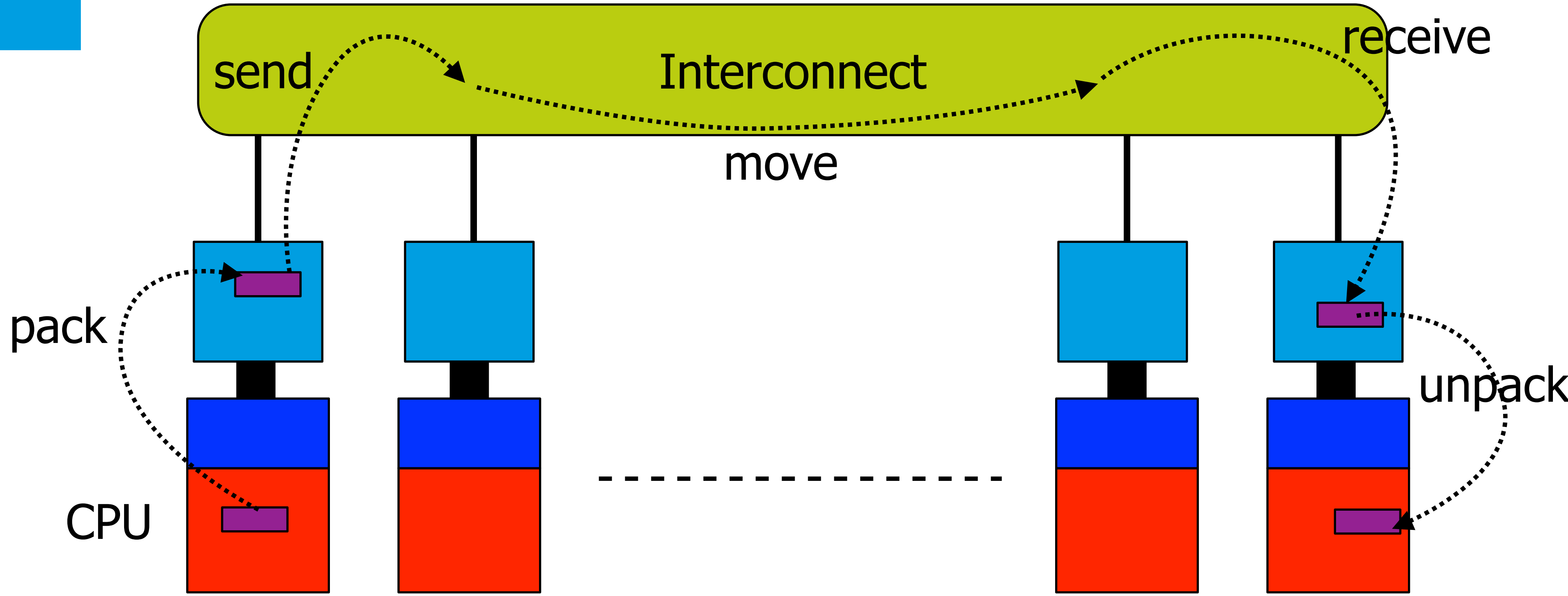
- Implementations:
  - Multiple processors connected by an interconnect
    – All processors have their own unique memory space
    – Can only access their own local memory address
    – Have to communicate though interconnect to move information/data to other processors

Distributed memory

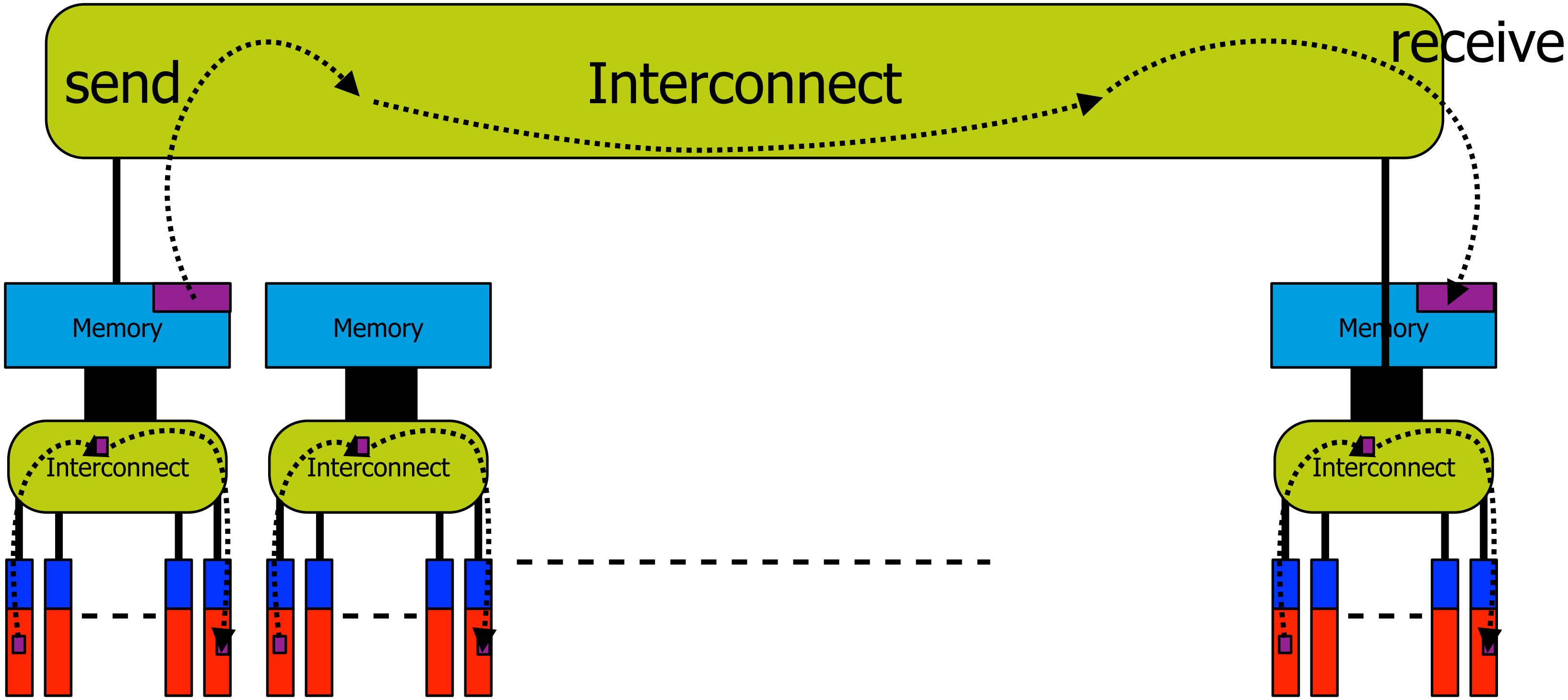


A Distributed Memory Computer

The system is programmed using message passing



Hybrid: MIMD with shared memory nodes => DelftBlue



Design Characteristics of a Network

Design Characteristics of a Network

- Topology (how things are connected):
  - Crossbar, ring, 2-D and 3-D meshes or torus, hypercube, tree, butterfly, ...
- Routing algorithm (path used):
  - Example in 2D torus: all east-west then all north-south
- Switching strategy:
  - Circuit switching: full path reserved for entire message, like the telephone.
  - Packet switching: message broken into separately-routed packets, like the post office.
- Flow control (what if there is congestion):
  - Stall, store data in buffers, re-route data to other nodes, tell source node to temporarily halt, discard, ...

Performance Properties of a Network: Latency

- Latency: delay between send and receive times
  - Latency tends to vary widely across architectures
  - Vendors often report hardware latencies (wire time)
  - Application programmers care about software latencies (user program to user program)

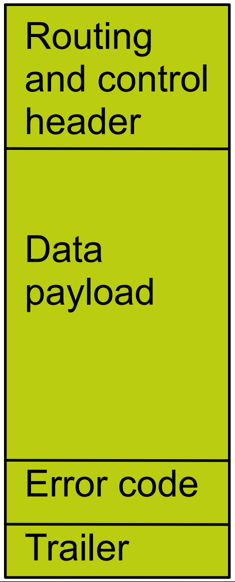
- Latency is important for programs with many small messages

Performance Properties of a Network: Bandwidth

- The bandwidth of a link = w * 1/t
  - w is the number of wires
  - t is the time per bit

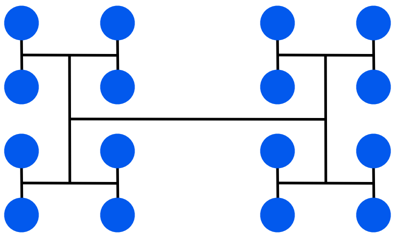
- Bandwidth typically in GigaBytes (GB), i.e., 8* 2^{20} bits
- Effective bandwidth is usually lower than physical link bandwidth due to packet overhead.

- Bandwidth is important for applications with mostly large messages

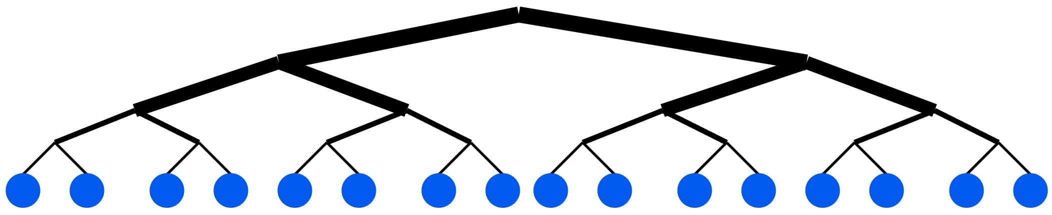


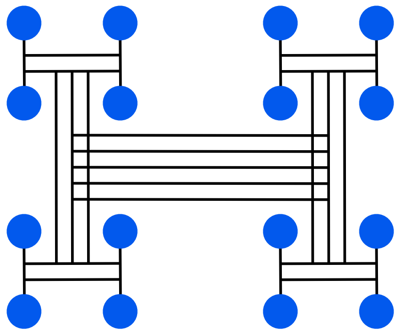
Common Network Topologies

Trees

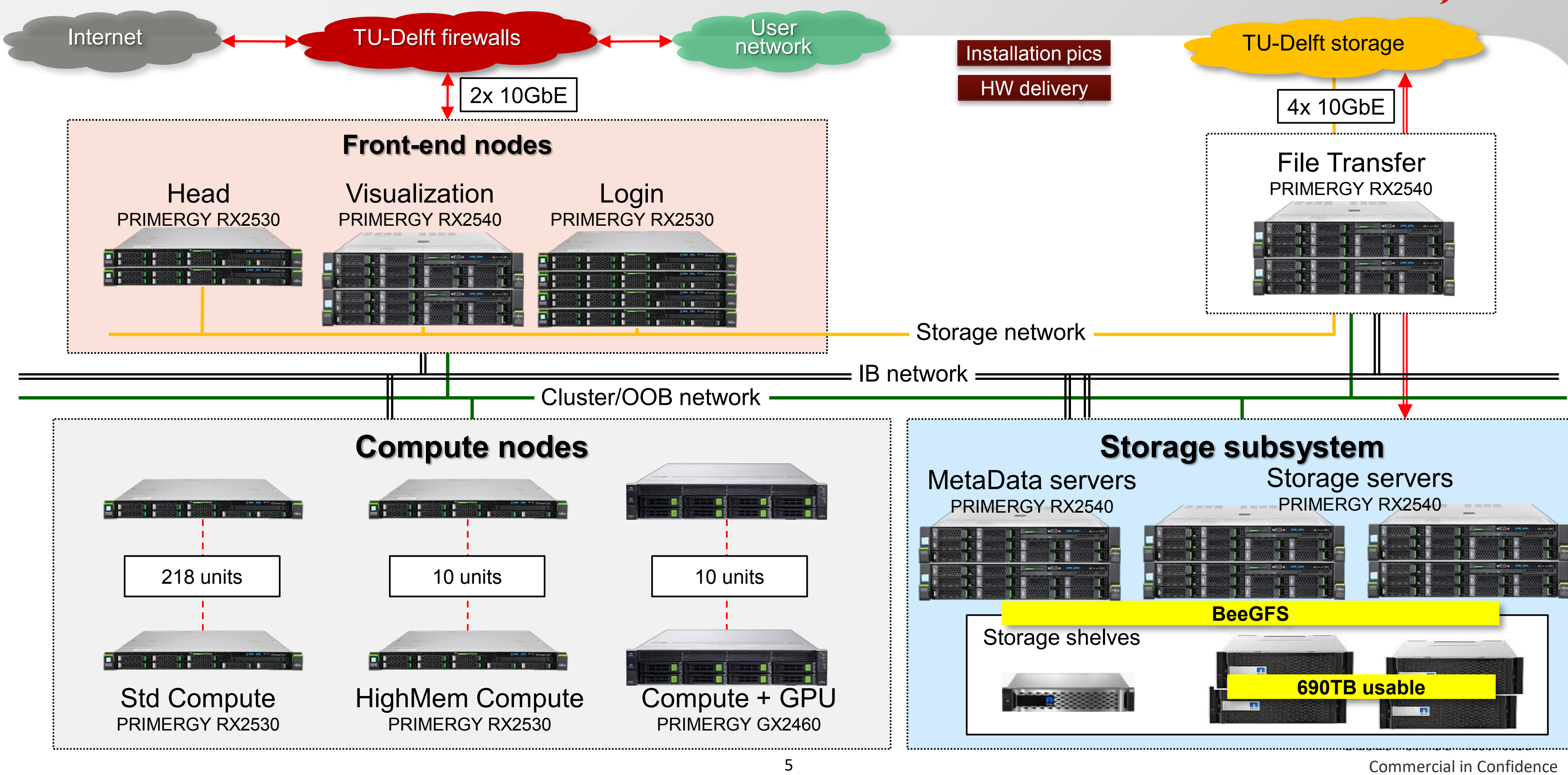


- Diameter = log n.
- Bisection bandwidth = 1.
- Easy layout as planar graph.
- Many tree algorithms (e.g., summation).
- Fat trees avoid bisection bandwidth problem:
    - More (or wider) links near top.





System components – servers / storage



Delft Blue

- Fat tree topology
- switch ports
  - Mellanox InfiniBand HDR100/HDR interconnect configured in a Full Bisectional Bandwidth (FBB) non-blocking network fabric.
-