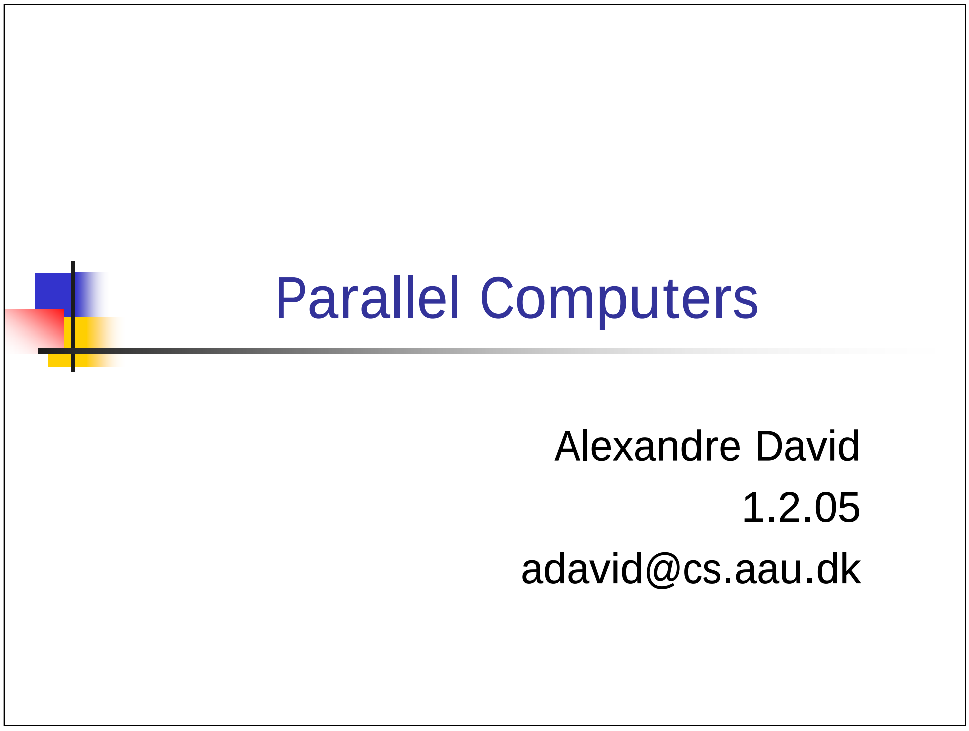
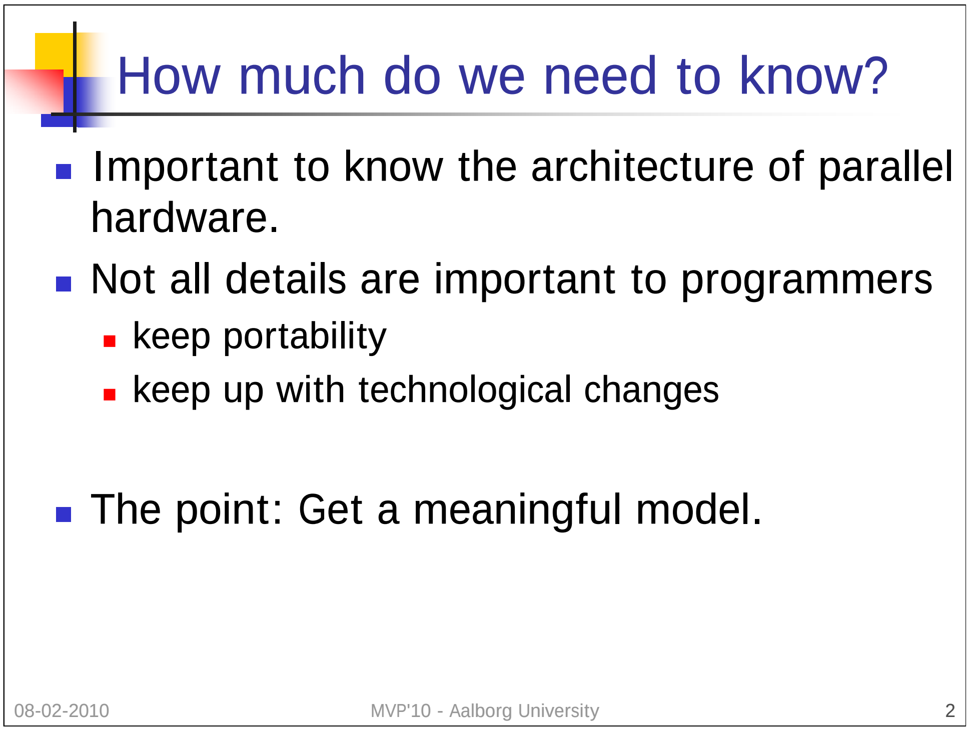


Parallel Computers

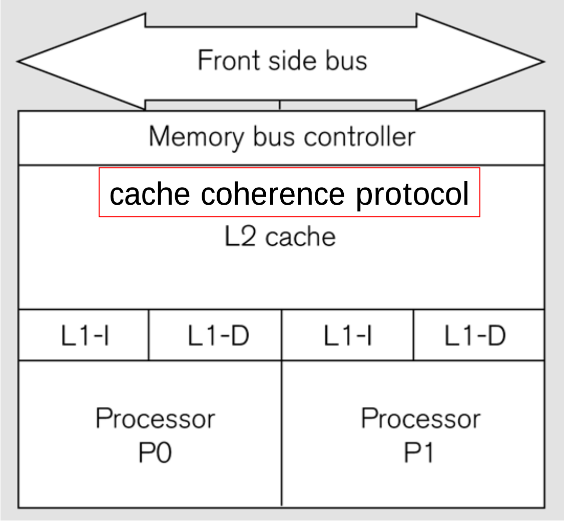
Alexandre David
1.2.05
adavid@cs.aau.dk



How much do we need to know?

- Important to know the architecture of parallel hardware.
- Not all details are important to programmers
  - keep portability
  - keep up with technological changes
- The point: Get a meaningful model.

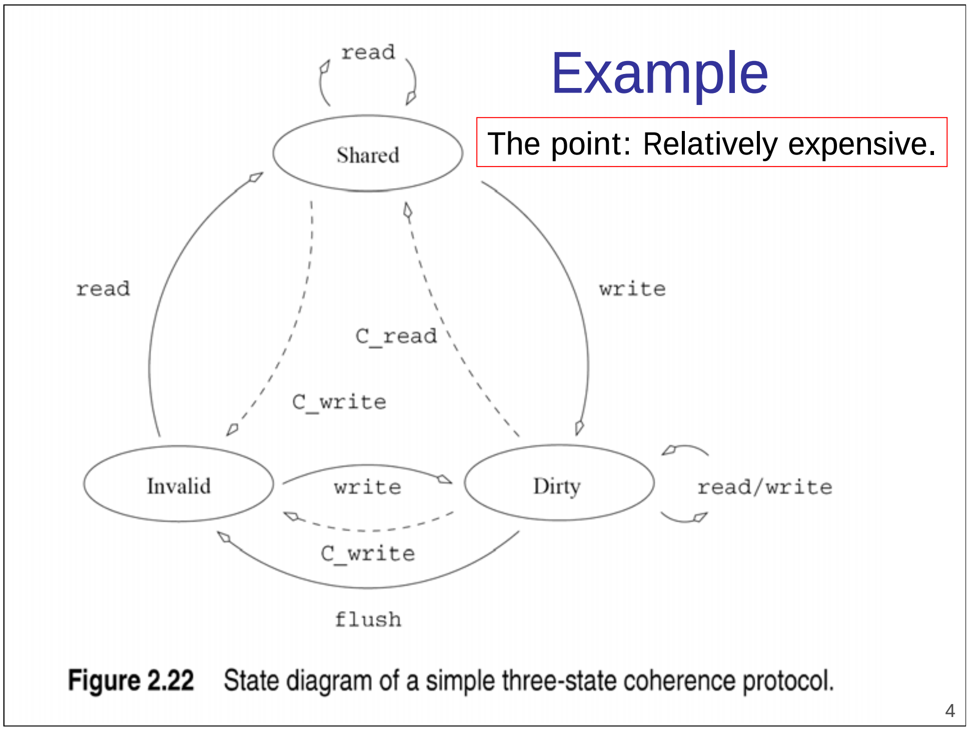
Intel Core Duo



cache coherence protocol

Modified
Exclusive
Shared
Invalid

more shared L2
low latency

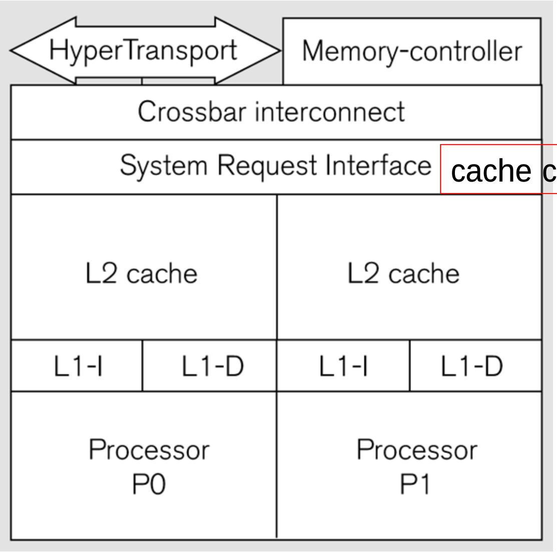


Example

The point: Relatively expensive.

Figure 2.22 State diagram of a simple three-state coherence protocol.

AMD Dual Core Opteron

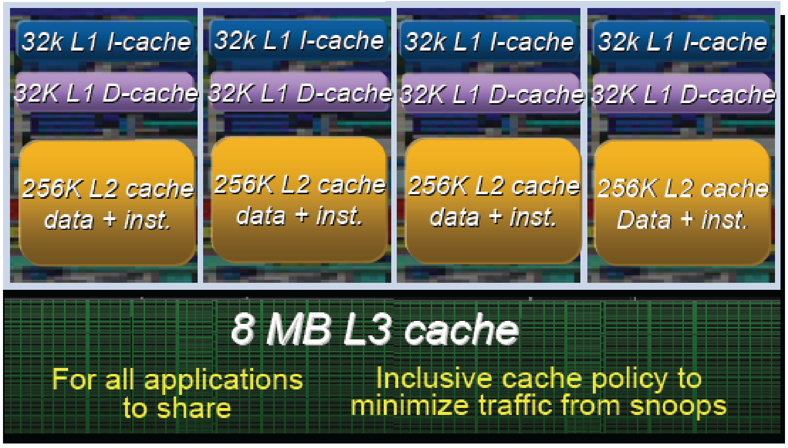


easier for SMP

cache coherence protocol

Modified
Owned
Exclusive
Shared
Invalid

Core i7



32k L1 I-cache | 32k L1 I-cache | 32k L1 I-cache | 32k L1 I-cache

32K L1 D-cache | 32K L1 D-cache | 32K L1 D-cache | 32K L1 D-cache

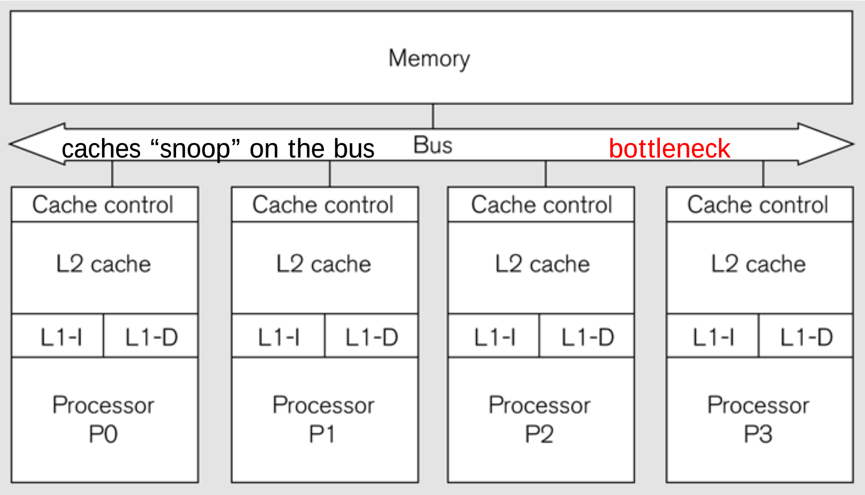
256K L2 cache data + inst. | 256K L2 cache data + inst. | 256K L2 cache data + inst. | 256K L2 cache Data + inst.

8 MB L3 cache

For all applications to share

Inclusive cache policy to minimize traffic from snoops

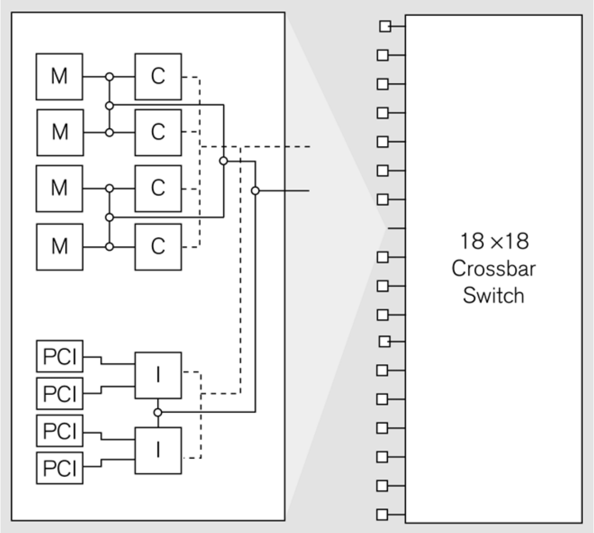
SMP



caches “snoop” on the bus

bottleneck

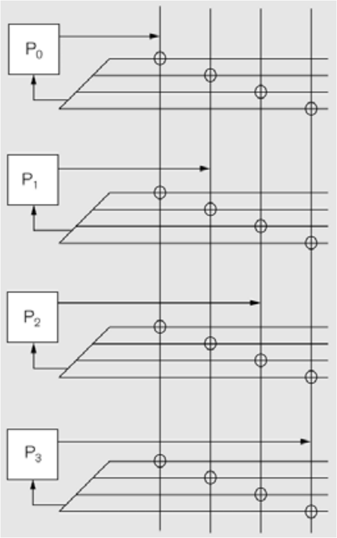
Larger SMP – Sun Fire



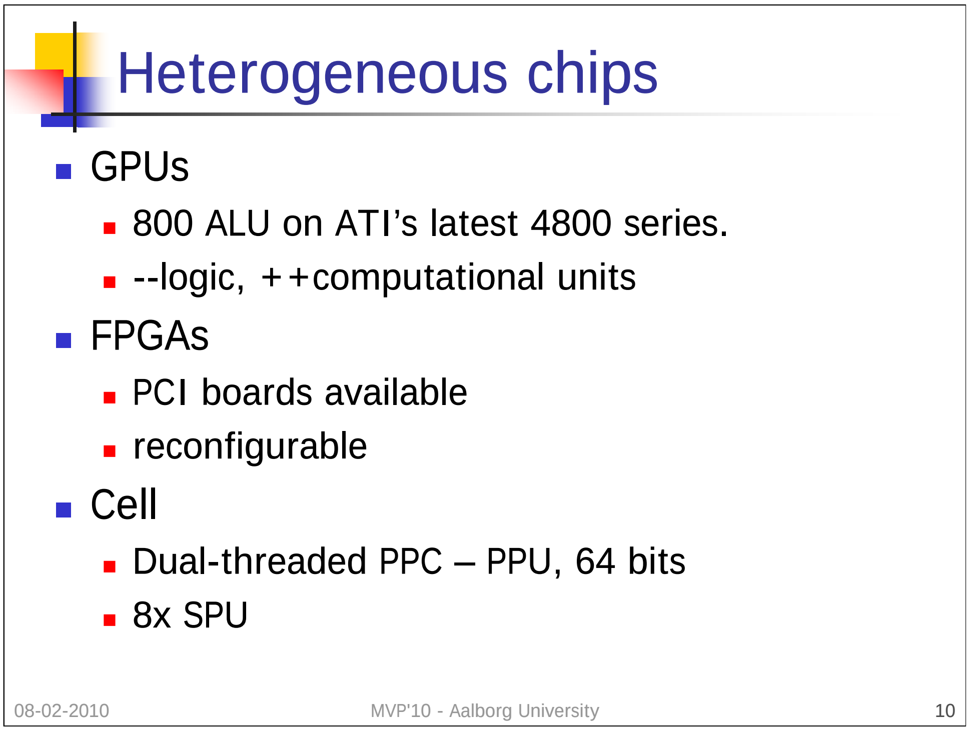
18 boards connected by a crossbar switch.
Snooping buses.
Directory based cache coherence protocol.
Scalable/higher latency.

Note: Expensive hardware.

Crossbar



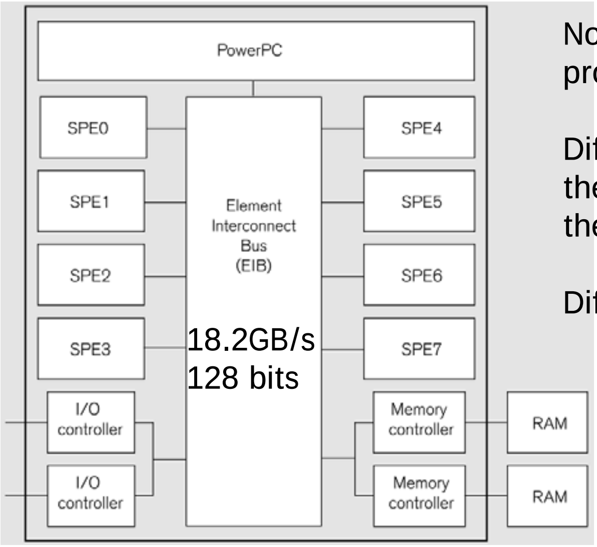
- N x N connections.
- Expensive, limited.



Heterogeneous chips

- GPUs
  - 800 ALU on ATI’s latest 4800 series.
  - --logic, ++computational units
- FPGAs
  - PCI boards available
  - reconfigurable
- Cell
  - Dual-threaded PPC – PPU, 64 bits
  - 8x SPU

Cell architecture

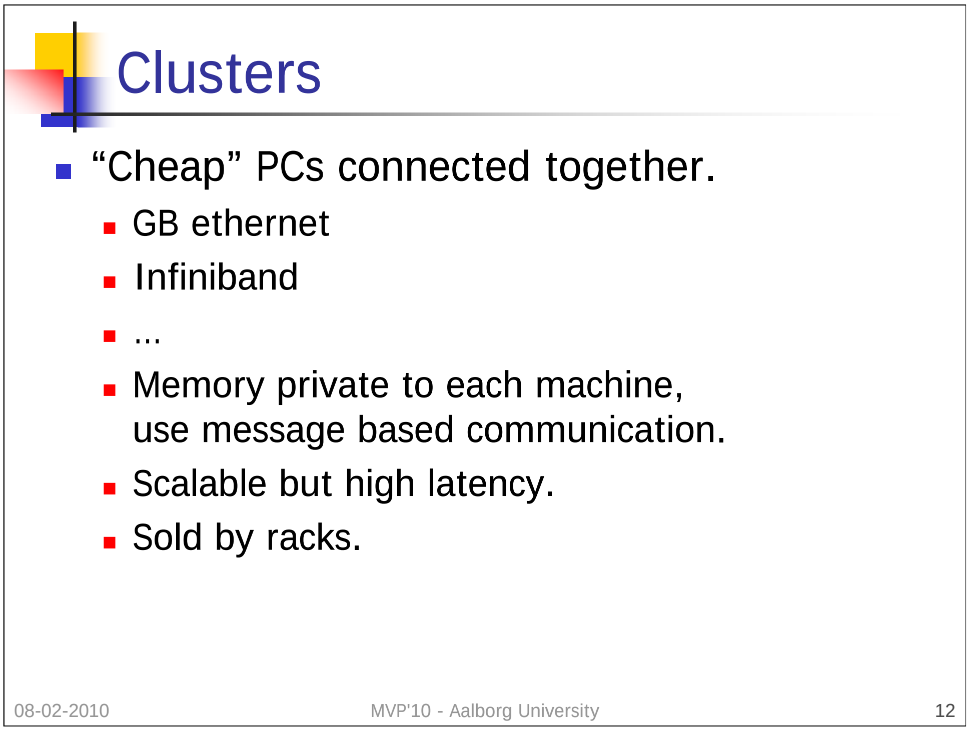


18.2GB/s
128 bits

No cache coherence protocol.

Different philosophy:
the PPU is a coordinator,
the SPUs do the job.

Difficult to program.



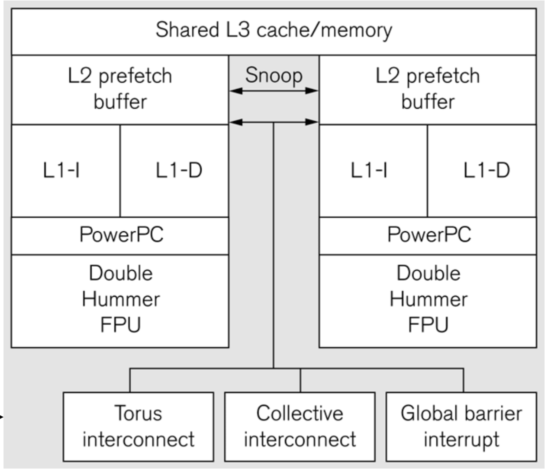
Clusters

- “Cheap” PCs connected together.
  - GB ethernet
  - Infiniband
  - ...
  - Memory private to each machine,
    use message based communication.
  - Scalable but high latency.
  - Sold by racks.

BlueGene

65536 x
@ 700MHz

interesting part →



Shared L3 cache/memory

L2 prefetch buffer

Snoop

L2 prefetch buffer

L1-I

L1-D

L1-I

L1-D

PowerPC

PowerPC

Double Hummer FPU

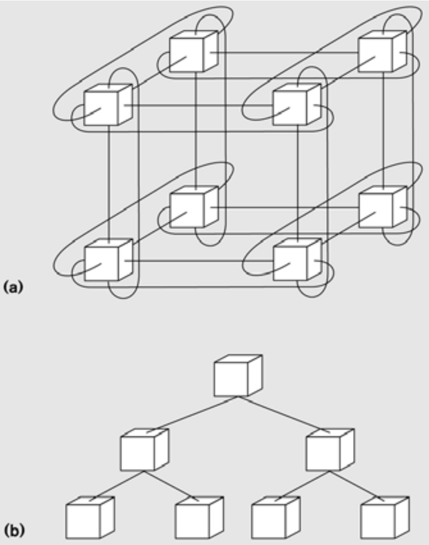
Double Hummer FPU

Torus interconnect

Collective interconnect

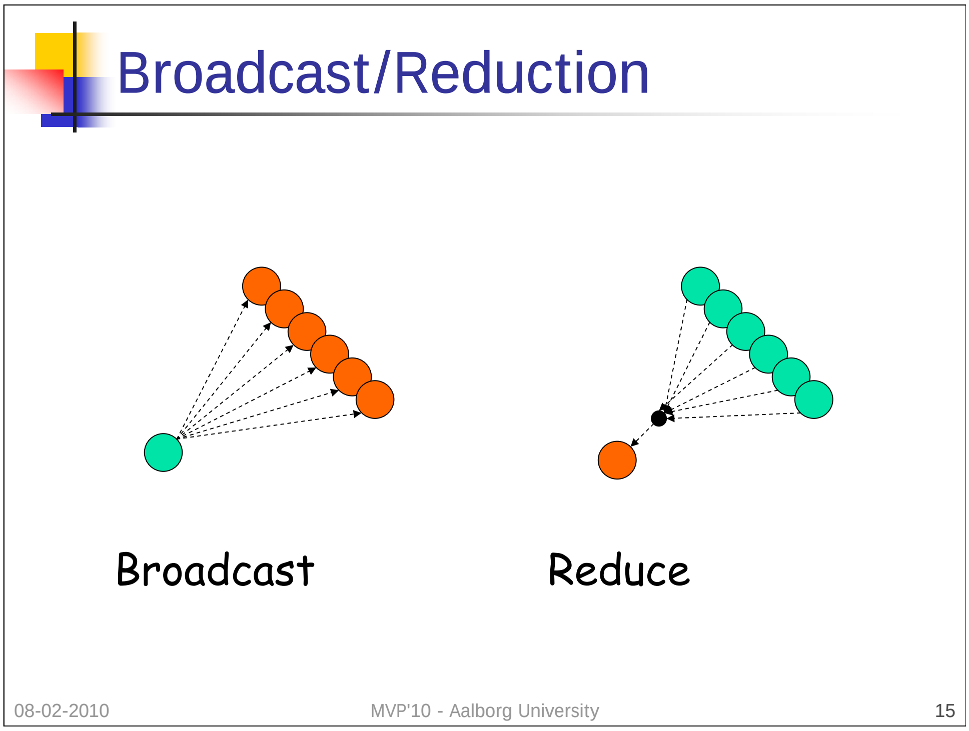
Global barrier interrupt

Interconnect



3-D torus for standard data transfers.

Collective network for fast reductions. Very powerful.



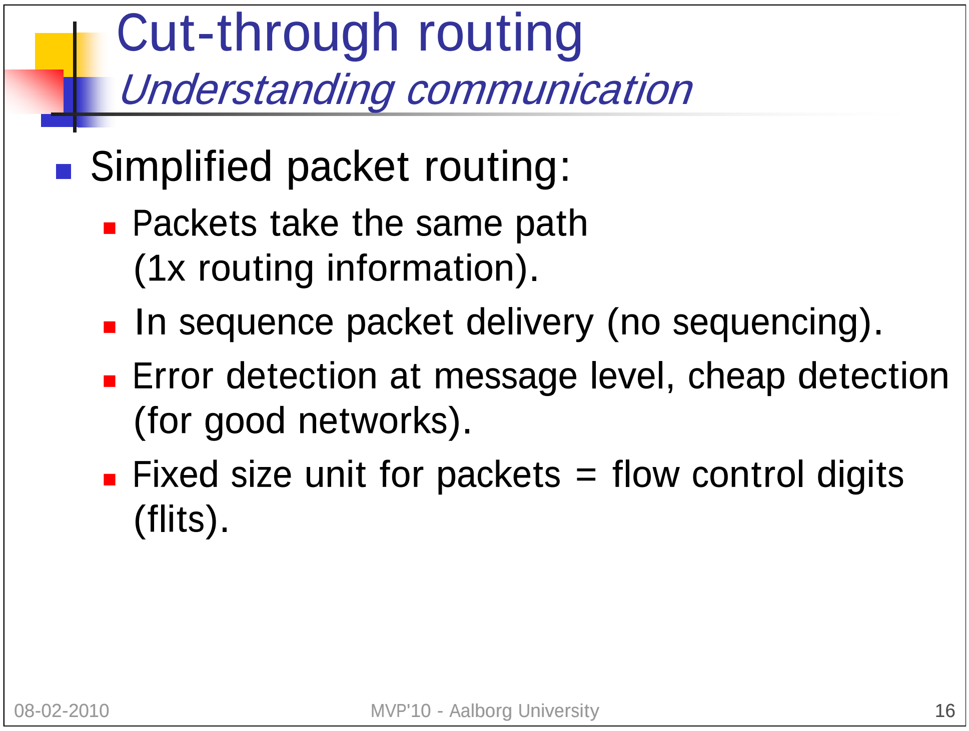
Broadcast/Reduction

Broadcast

Reduce

08-02-2010 MVP'10 - Aalborg University 15

This is the logical view, what happens from the programmer’s perspective.



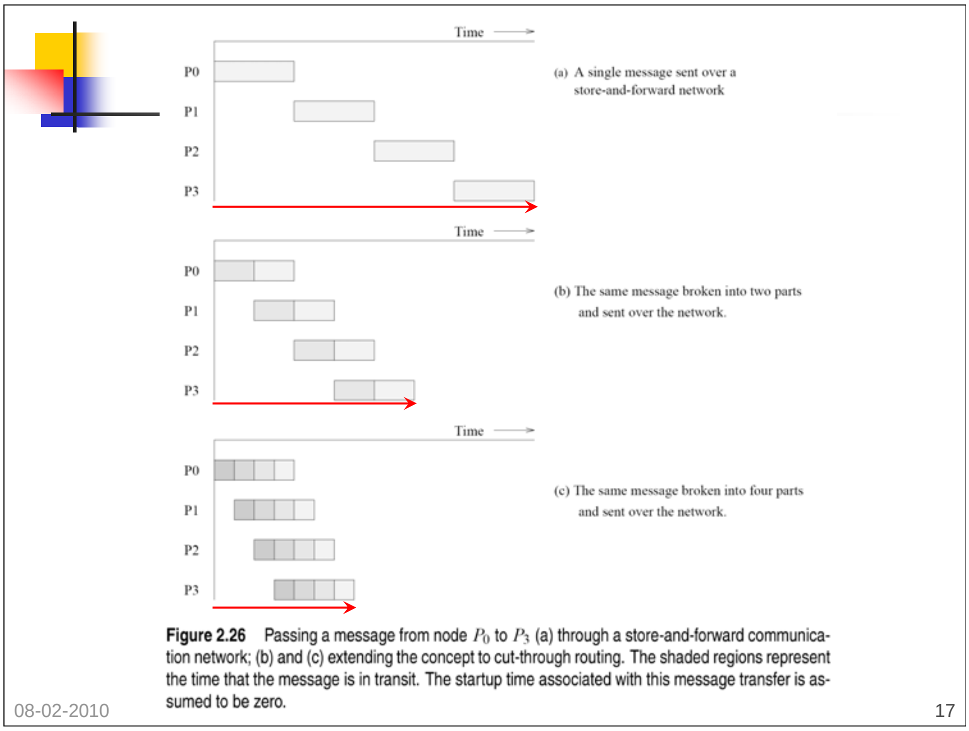
Cut-through routing

Understanding communication

- Simplified packet routing:
  - Packets take the same path (1x routing information).
  - In sequence packet delivery (no sequencing).
  - Error detection at message level, cheap detection (for good networks).
  - Fixed size unit for packets = flow control digits (flits).

08-02-2010 MVP'10 - Aalborg University 16

It is an optimization for interconnection networks of parallel machines since error rates are very low (dedicated network).

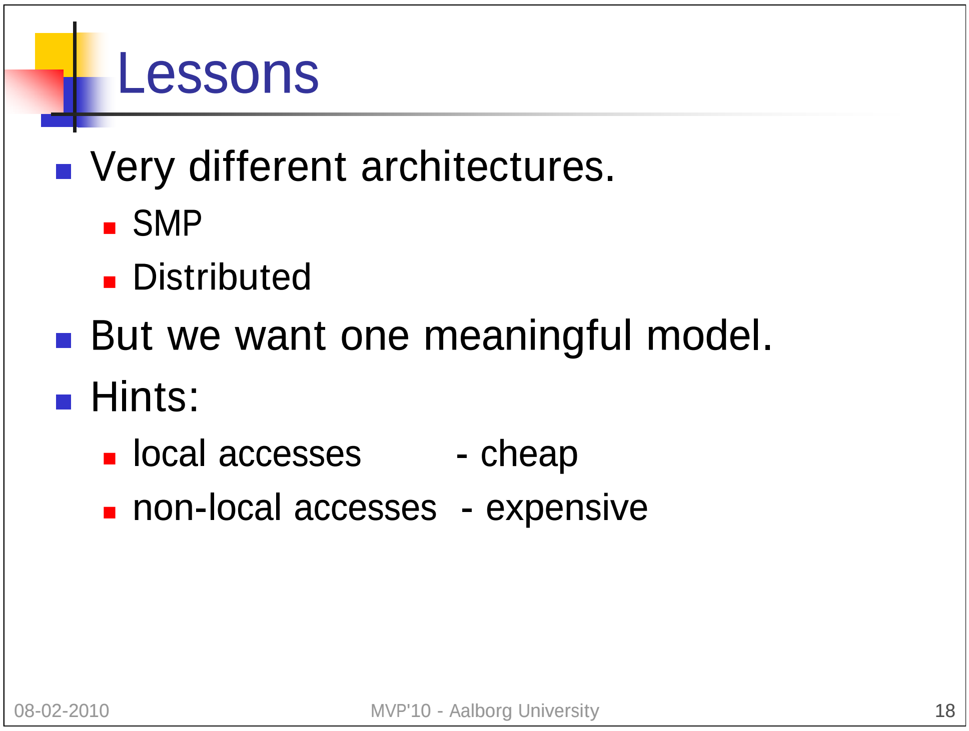


(a) A single message sent over a store-and-forward network

(b) The same message broken into two parts and sent over the network.

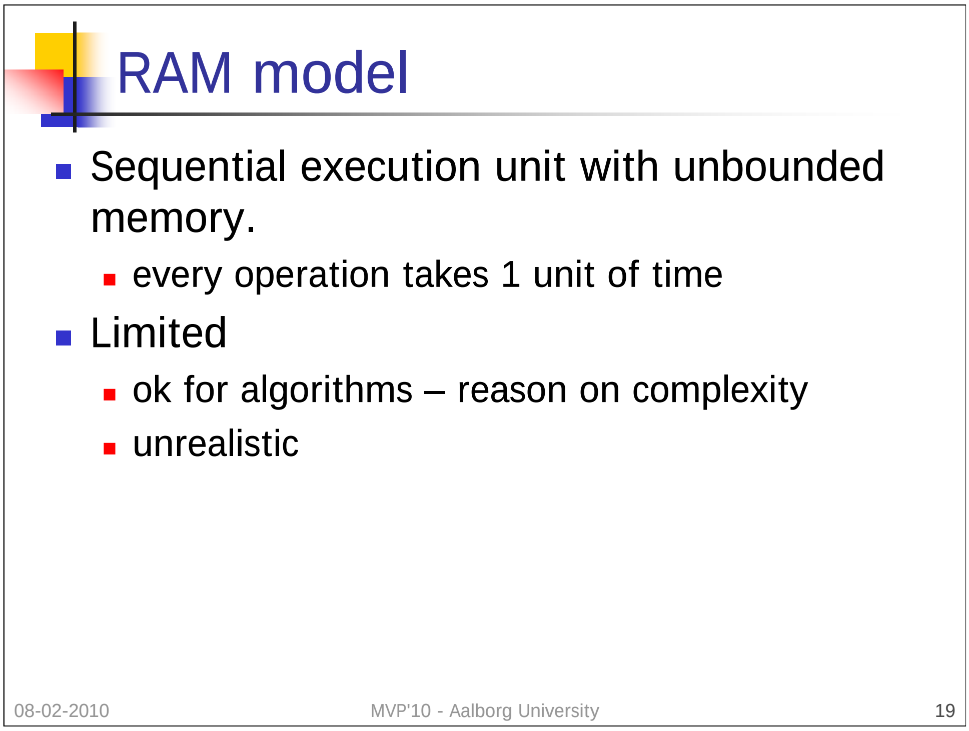
(c) The same message broken into four parts and sent over the network.

Figure 2.26 Passing a message from node P_0 to P_3 (a) through a store-and-forward communication network; (b) and (c) extending the concept to cut-through routing. The shaded regions represent the time that the message is in transit. The startup time associated with this message transfer is assumed to be zero.



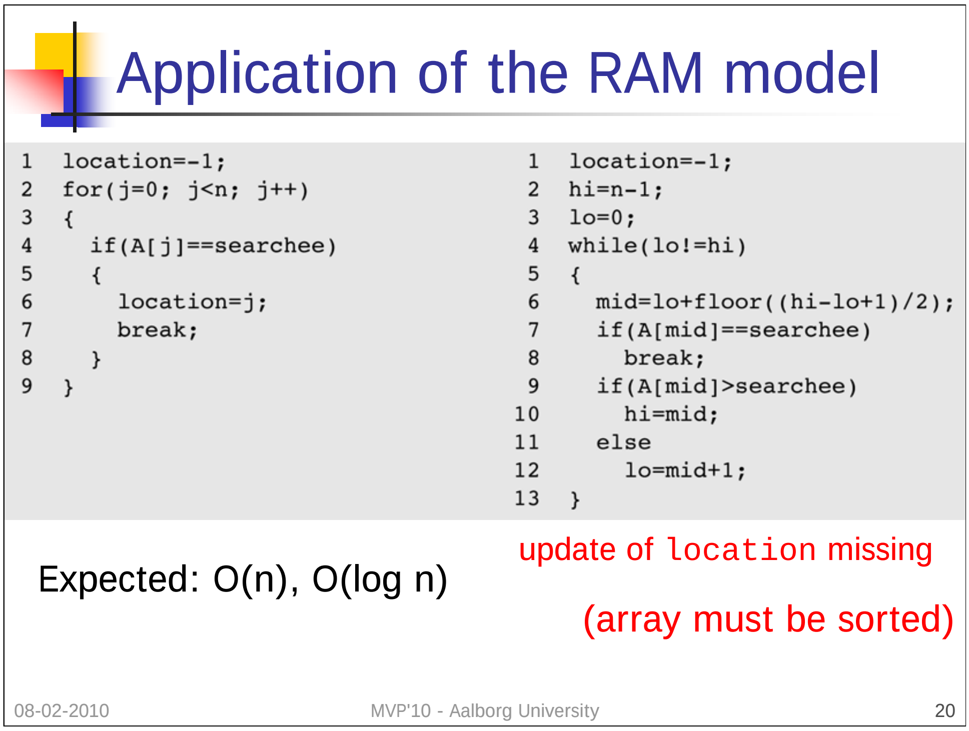
Lessons

- Very different architectures.
  - SMP
  - Distributed
- But we want one meaningful model.
- Hints:
  - local accesses - cheap
  - non-local accesses - expensive



RAM model

- Sequential execution unit with unbounded memory.
  - every operation takes 1 unit of time
- Limited
  - ok for algorithms – reason on complexity
  - unrealistic



Application of the RAM model

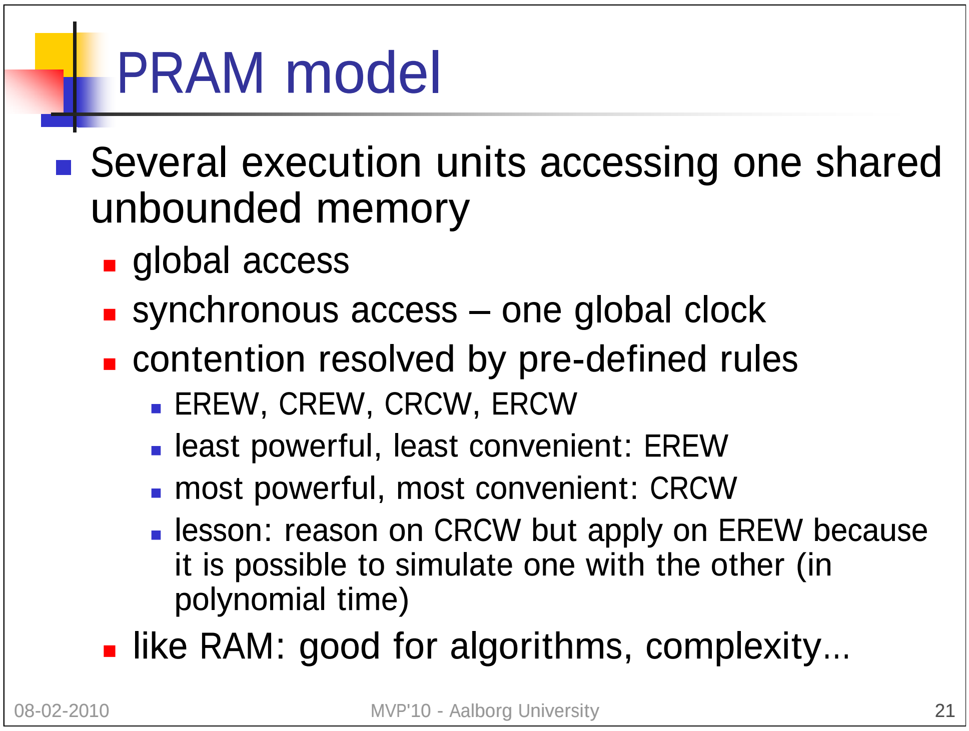
```
location=-1;
for(j=0; j<n; j++)
{
  if(A[j]==searchee)
  {
    location=j;
    break;
  }
}
```

```
location=-1;
hi=n-1;
lo=0;
while(lo!=hi)
{
  mid=lo+floor((hi-lo+1)/2);
  if(A[mid]==searchee)
    break;
  if(A[mid]>searchee)
    hi=mid;
  else
    lo=mid+1;
}
```

Expected: O(n), O(log n)

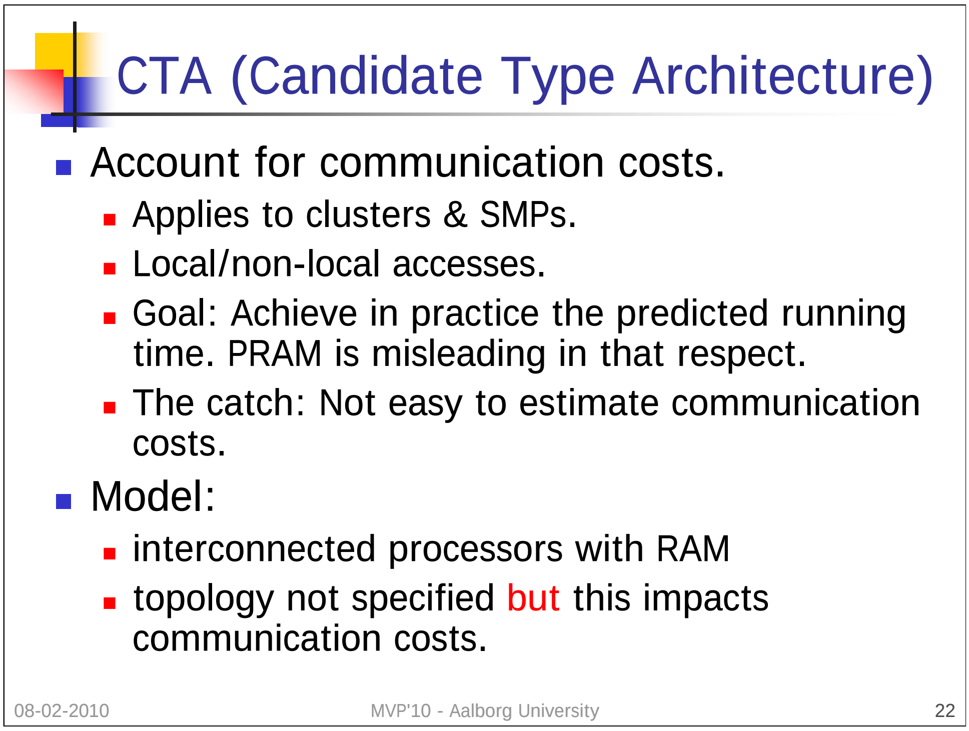
update of `location` missing

(array must be sorted)



PRAM model

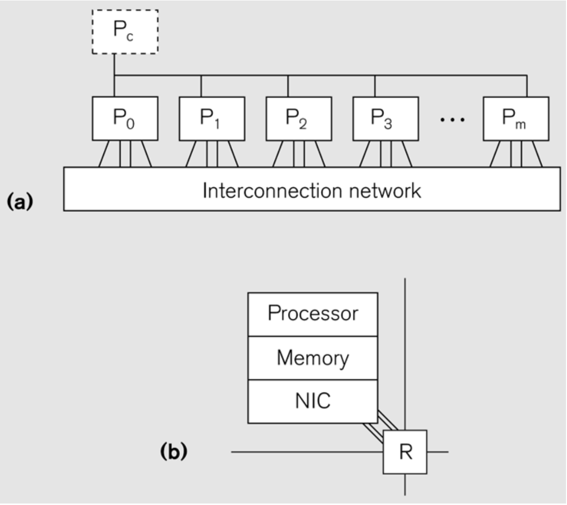
- Several execution units accessing one shared unbounded memory
  - global access
  - synchronous access – one global clock
  - contention resolved by pre-defined rules
    - EREW, CREW, CRCW, ERCW
    - least powerful, least convenient: EREW
    - most powerful, most convenient: CRCW
    - lesson: reason on CRCW but apply on EREW because it is possible to simulate one with the other (in polynomial time)
  - like RAM: good for algorithms, complexity...



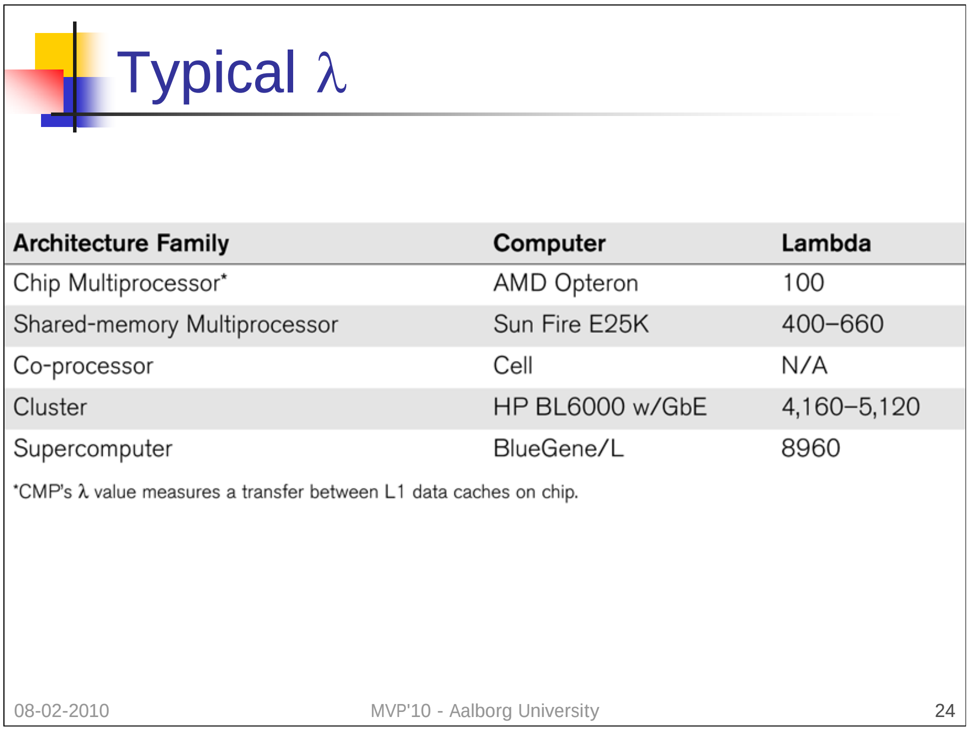
CTA (Candidate Type Architecture)

- Account for communication costs.
  - Applies to clusters & SMPs.
  - Local/non-local accesses.
  - Goal: Achieve in practice the predicted running time. PRAM is misleading in that respect.
  - The catch: Not easy to estimate communication costs.
- Model:
  - interconnected processors with RAM
  - topology not specified but this impacts communication costs.

CTA



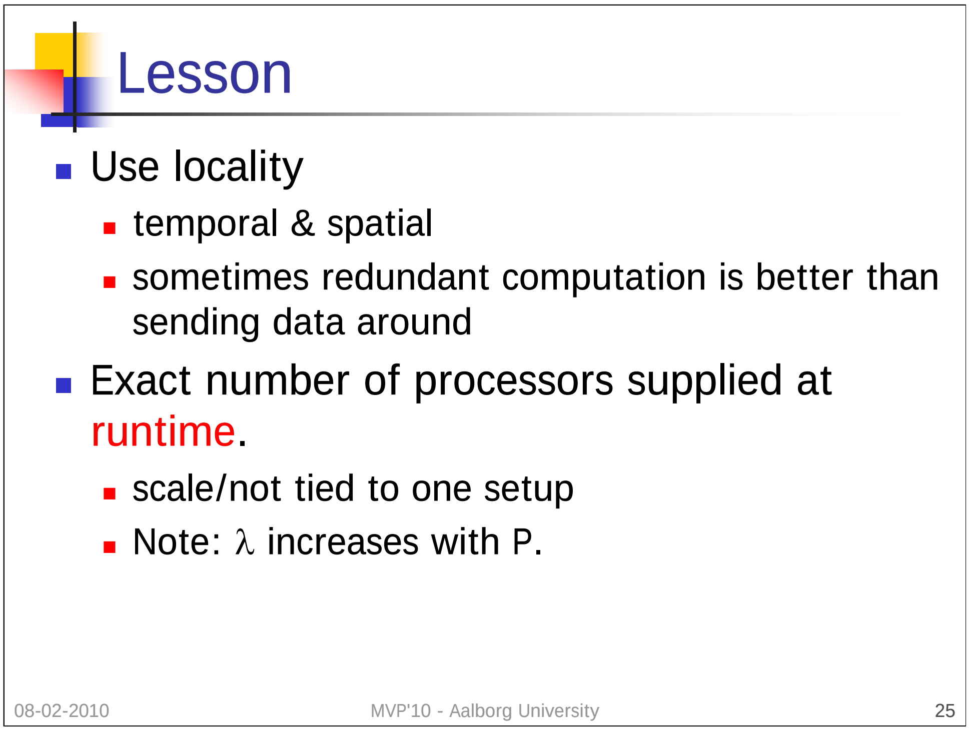
SMP
Cluster
Cell
…
Memory latency specified in function of the real architecture.
Non-local: λ.



Typical λ

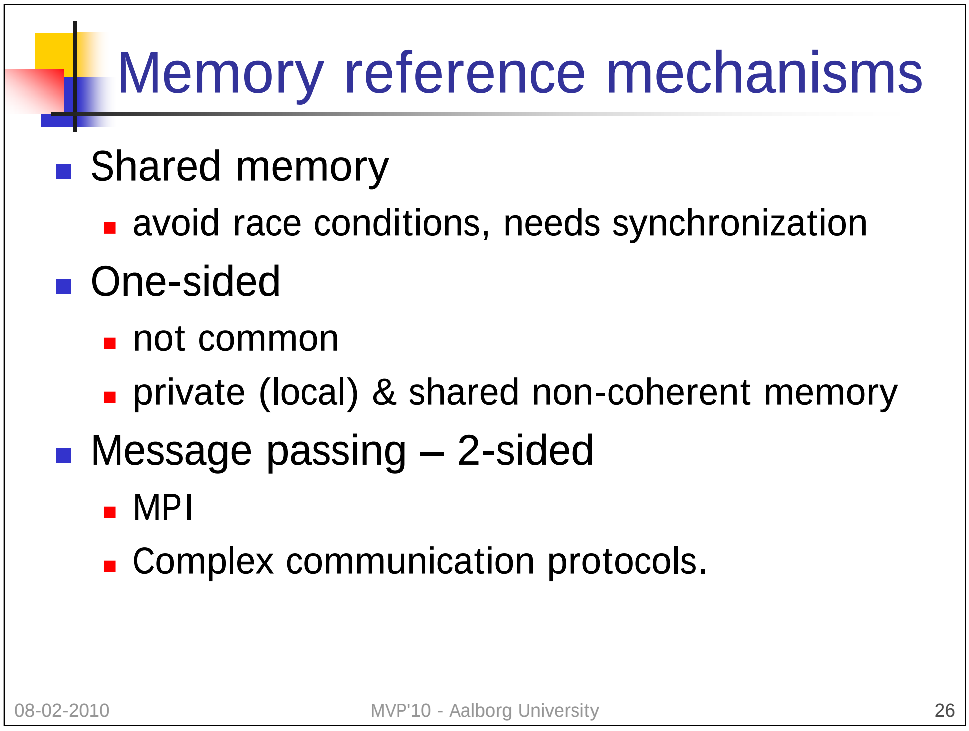
Architecture Family	Computer	Lambda
Chip Multiprocessor*	AMD Opteron	100
Shared-memory Multiprocessor	Sun Fire E25K	400–660
Co-processor	Cell	N/A
Cluster	HP BL6000 w/GbE	4,160–5,120
Supercomputer	BlueGene/L	8960

*CMP's λ value measures a transfer between L1 data caches on chip.



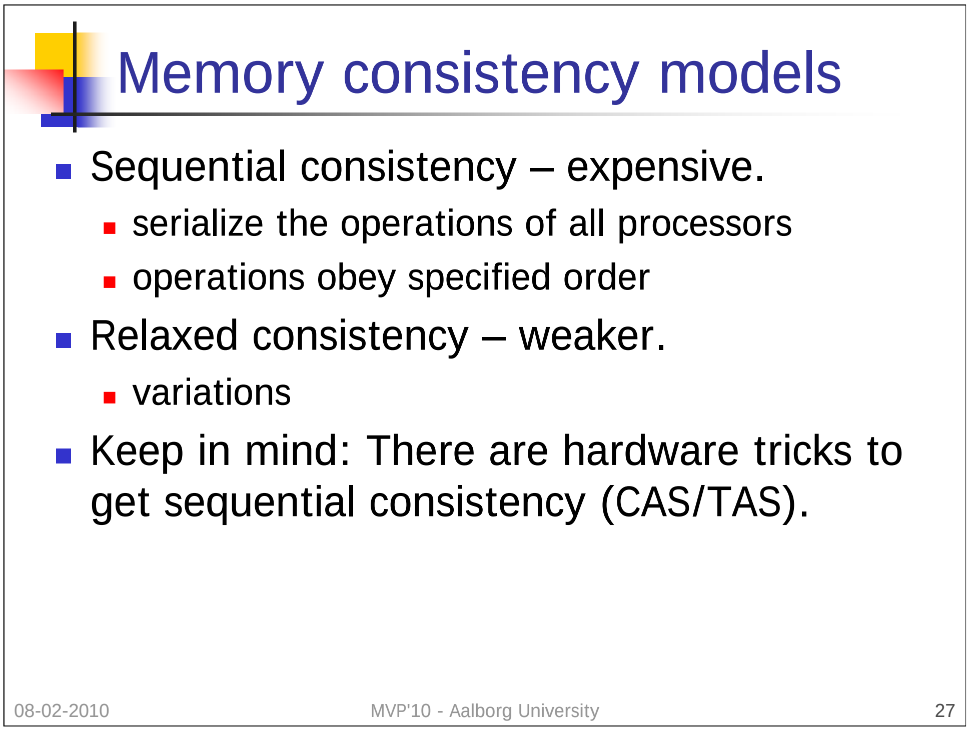
Lesson

- Use locality
  - temporal & spatial
  - sometimes redundant computation is better than sending data around
- Exact number of processors supplied at runtime.
  - scale/not tied to one setup
  - Note: λ increases with P.



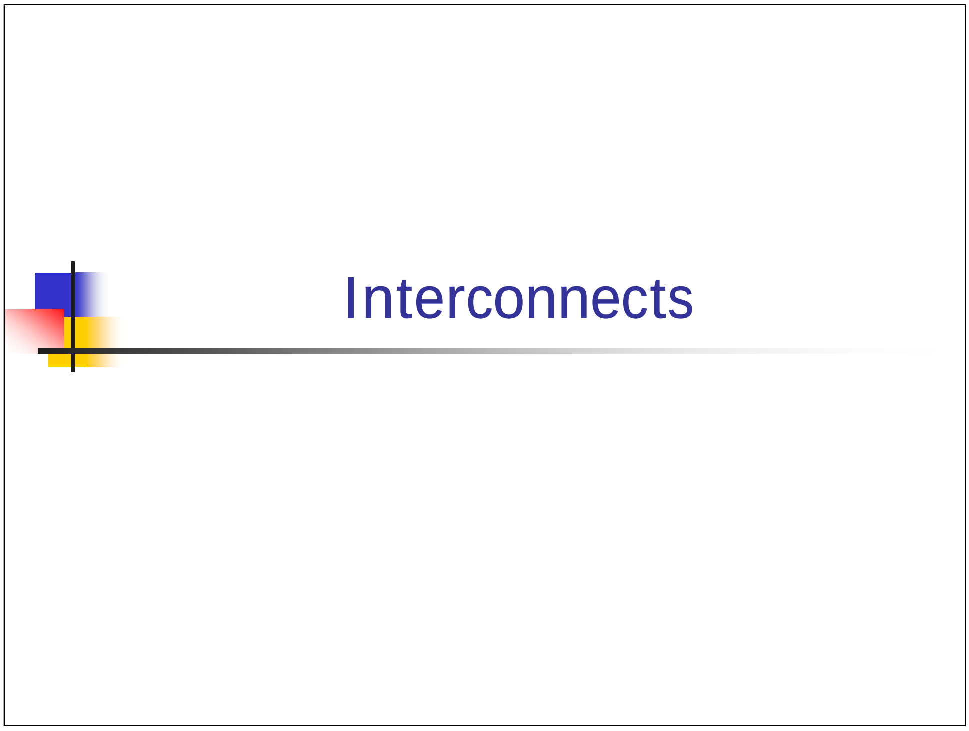
Memory reference mechanisms

- Shared memory
  - avoid race conditions, needs synchronization
- One-sided
  - not common
  - private (local) & shared non-coherent memory
- Message passing – 2-sided
  - MPI
  - Complex communication protocols.

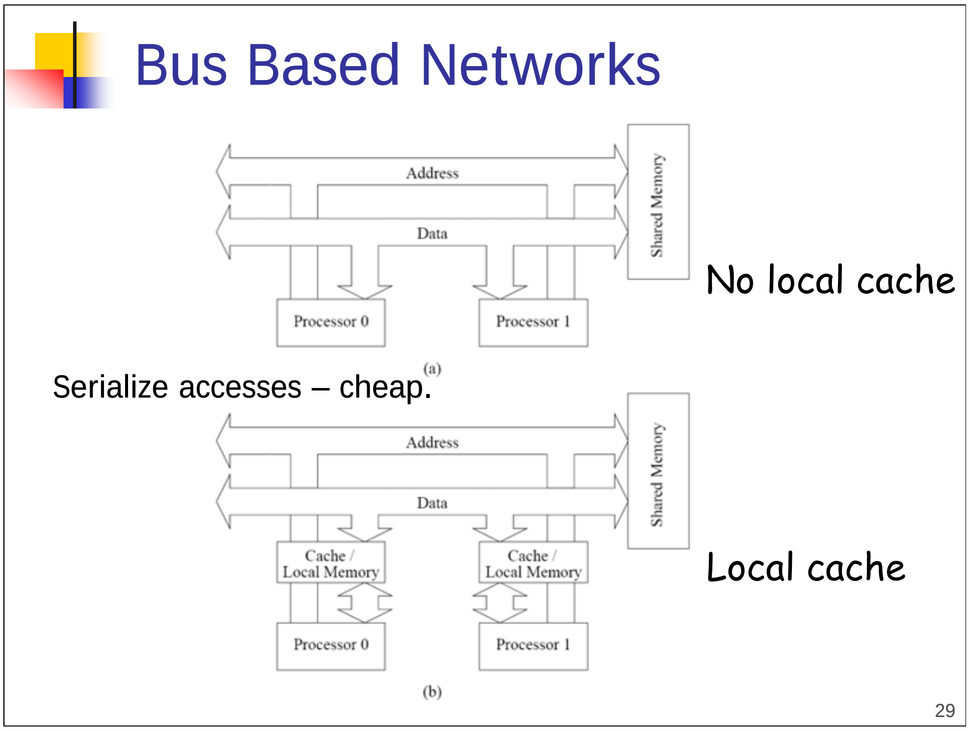


Memory consistency models

- Sequential consistency – expensive.
  - serialize the operations of all processors
  - operations obey specified order
- Relaxed consistency – weaker.
  - variations
- Keep in mind: There are hardware tricks to get sequential consistency (CAS/TAS).



Interconnects



Bus Based Networks

No local cache

Serialize accesses – cheap.

Local cache

Good:

•Cost scales linearly with the number of nodes.

•The distance between all the nodes is constant.

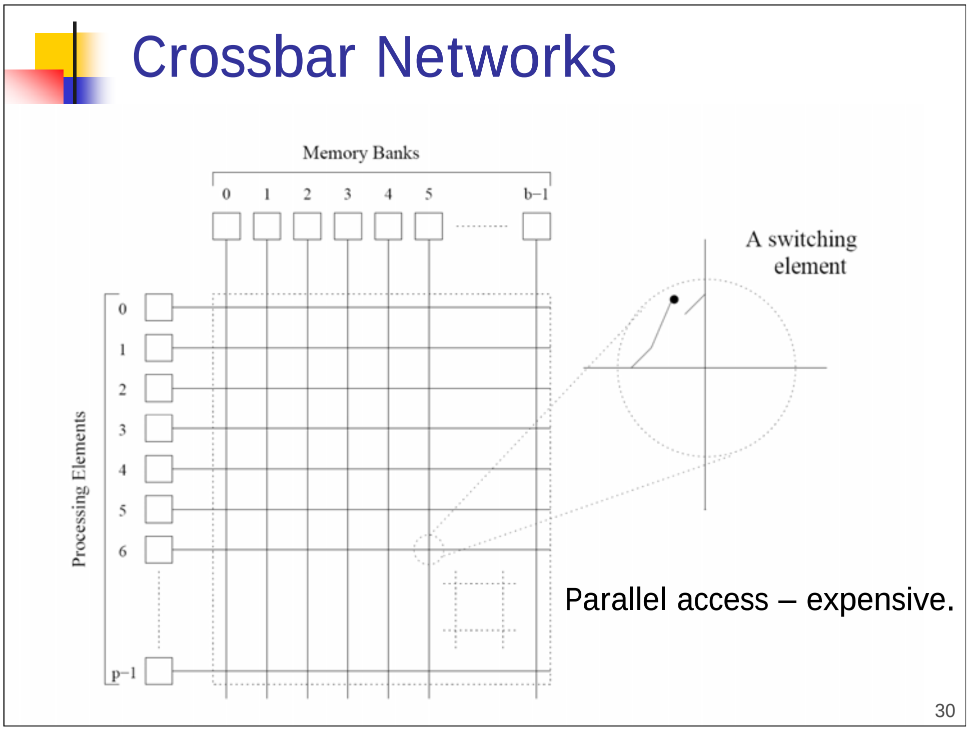
•It is ideal for broadcasting.

Bad:

•Shared bandwidth between all the nodes -> bottleneck in performance.

In practice bus based only for small SMP (Intel). Caches are only a trick to reduce bandwidth consumption on the bus (not to reduce bandwidth as stated in the book).

Both for processors & memory.



Crossbar Networks

Parallel access – expensive.

30

Grid to connect *p* processors to *b* memory banks. Non blocking in the sense that a connection (routing) does not block the connection of any other processing node, in contrast to multistage networks.

Good: scalable in performance (non blocking).

Bad: number of switches = *p*b*, not scalable in cost.

Omega networks

Multi-stage network – compromise cost/performance.
N nodes – log n stages.

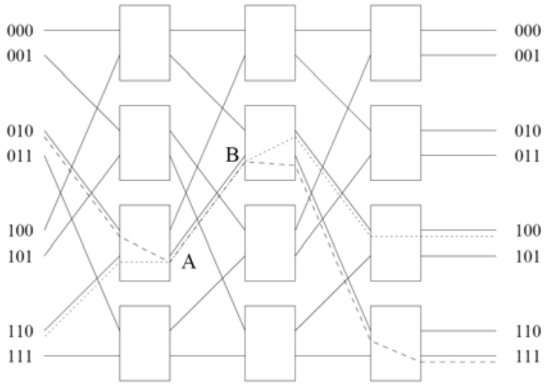
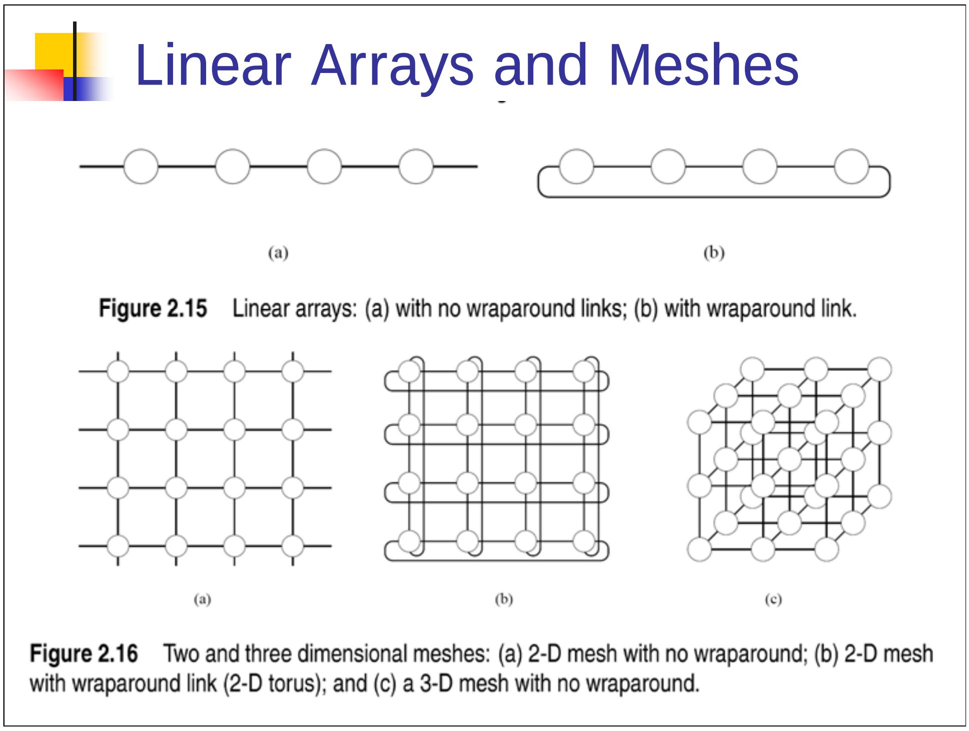


Figure 2.13 An example of blocking in omega network: one of the messages (010 to 111 or 110 to 100) is blocked at link AB.



Linear Arrays and Meshes

Figure 2.15 Linear arrays: (a) with no wraparound links; (b) with wraparound link.

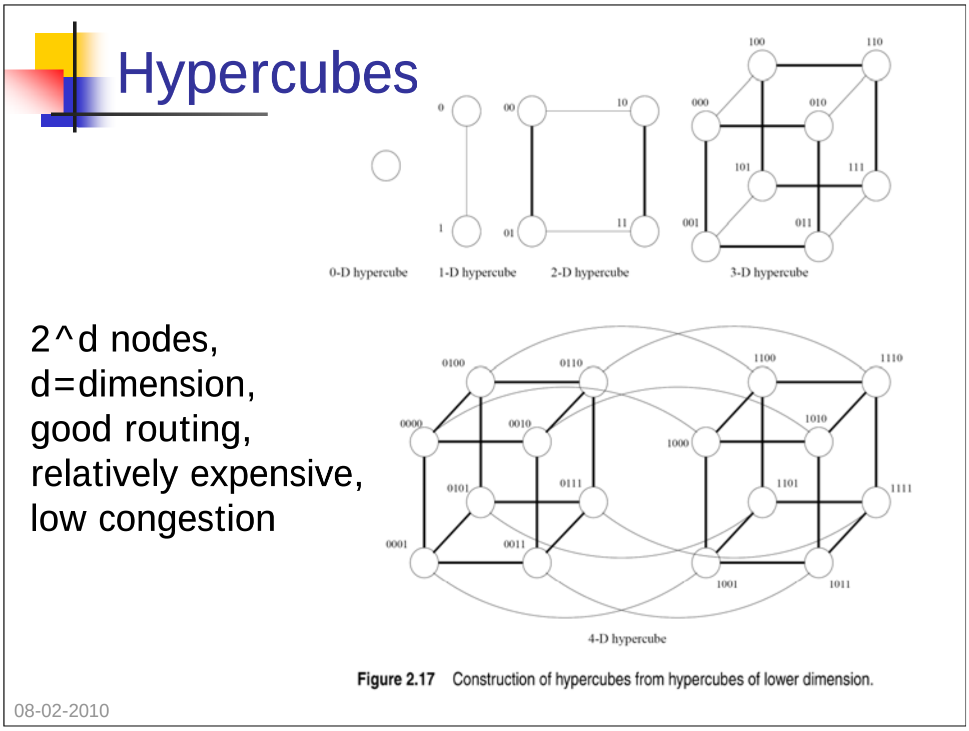
Figure 2.16 Two and three dimensional meshes: (a) 2-D mesh with no wraparound; (b) 2-D mesh with wraparound link (2-D torus); and (c) a 3-D mesh with no wraparound.

Wrap around changes the number of neighbors and distance for some nodes.

Linear array: each node has 2 neighbors (except start & end). It becomes a ring (or 1-D torus) with wraparound.

2-D mesh has ***p*** processors so the dimension is given by *sqrt(p)*. Every node (except on the border) has 4 neighbors. Attractive from a wiring point of view. Adding wraparound links gives a 2-D torus.

3-D, similarly. Every time we add a dimension, we add 2 neighbors. 3-D meshes good for physical simulations because they correspond to the modeled problem and the way processing is distributed.



Hypercubes

2^d nodes,
d=dimension,
good routing,
relatively expensive,
low congestion

Figure 2.17 Construction of hypercubes from hypercubes of lower dimension.

08-02-2010

Fat trees

More bandwidth where it is needed.

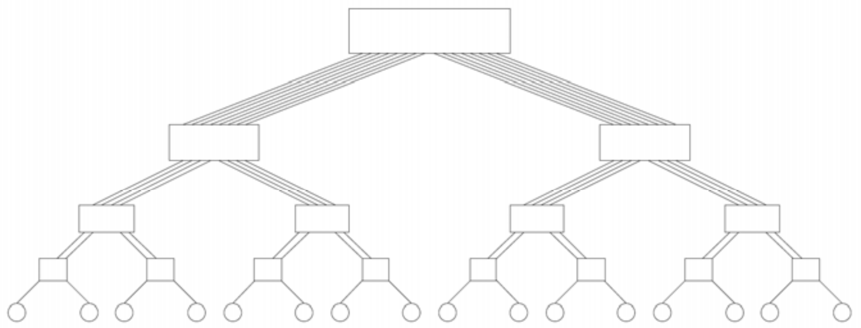
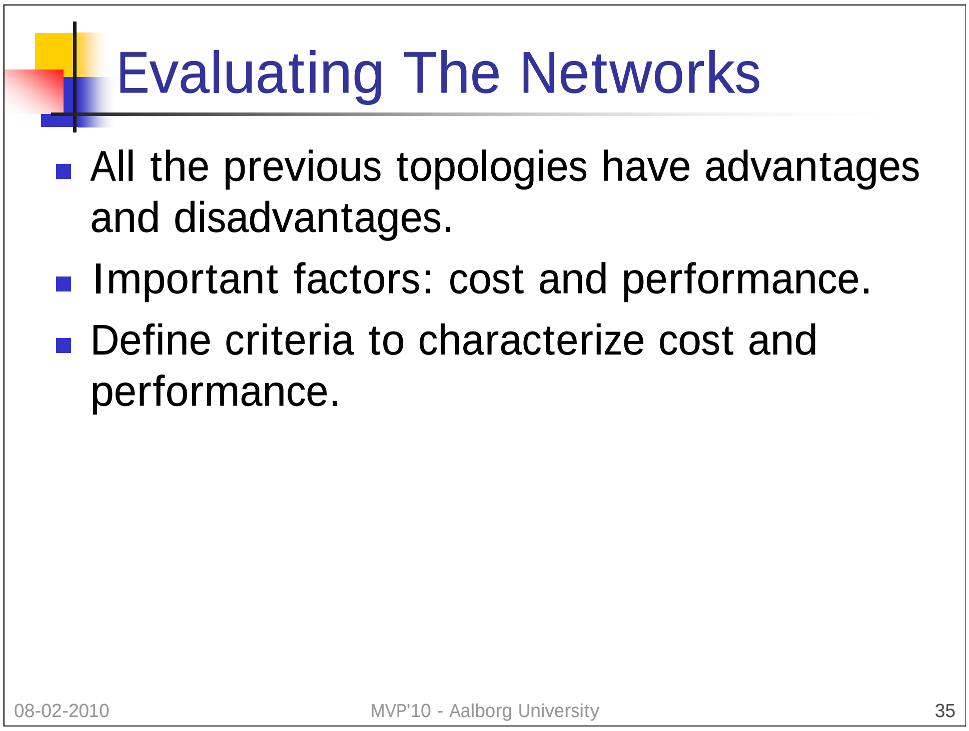


Figure 2.19 A fat tree network of 16 processing nodes.

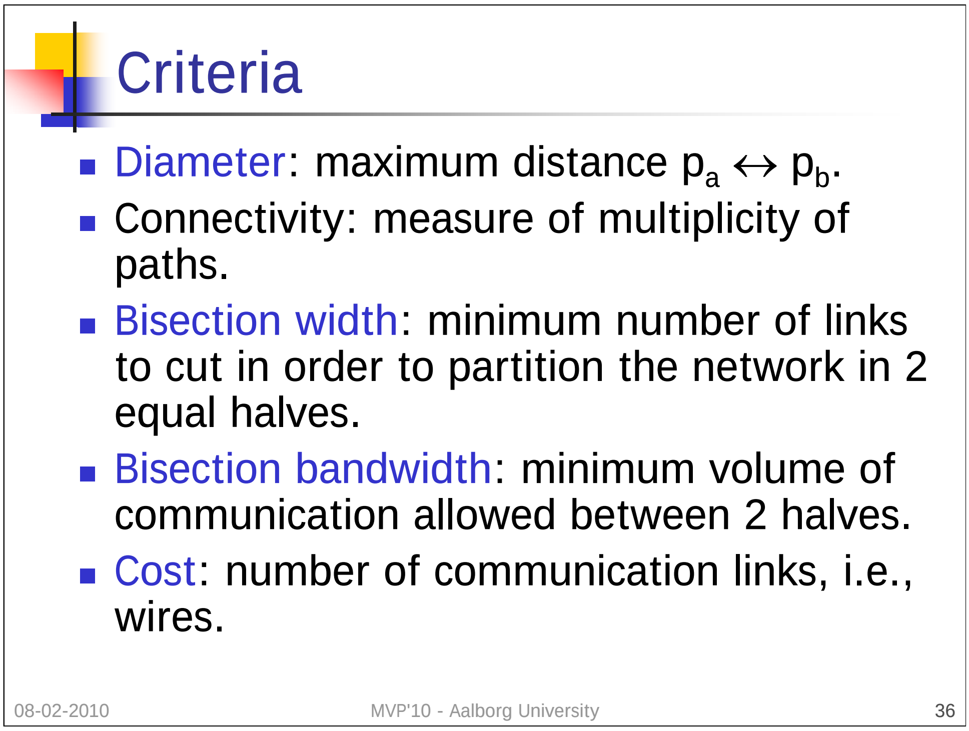


Evaluating The Networks

- All the previous topologies have advantages and disadvantages.
- Important factors: cost and performance.
- Define criteria to characterize cost and performance.

08-02-2010 MVP'10 - Aalborg University 35

Your turn: Give suggestions on measure criteria.



Criteria

- Diameter: maximum distance $p_a \leftrightarrow p_b$.
- Connectivity: measure of multiplicity of paths.
- Bisection width: minimum number of links to cut in order to partition the network in 2 equal halves.
- Bisection bandwidth: minimum volume of communication allowed between 2 halves.
- Cost: number of communication links, i.e., wires.

08-02-2010 MVP'10 - Aalborg University 36

Distance = shortest path between 2 nodes.
Diameter: How far 2 nodes may be.

•Completely connected: 1.

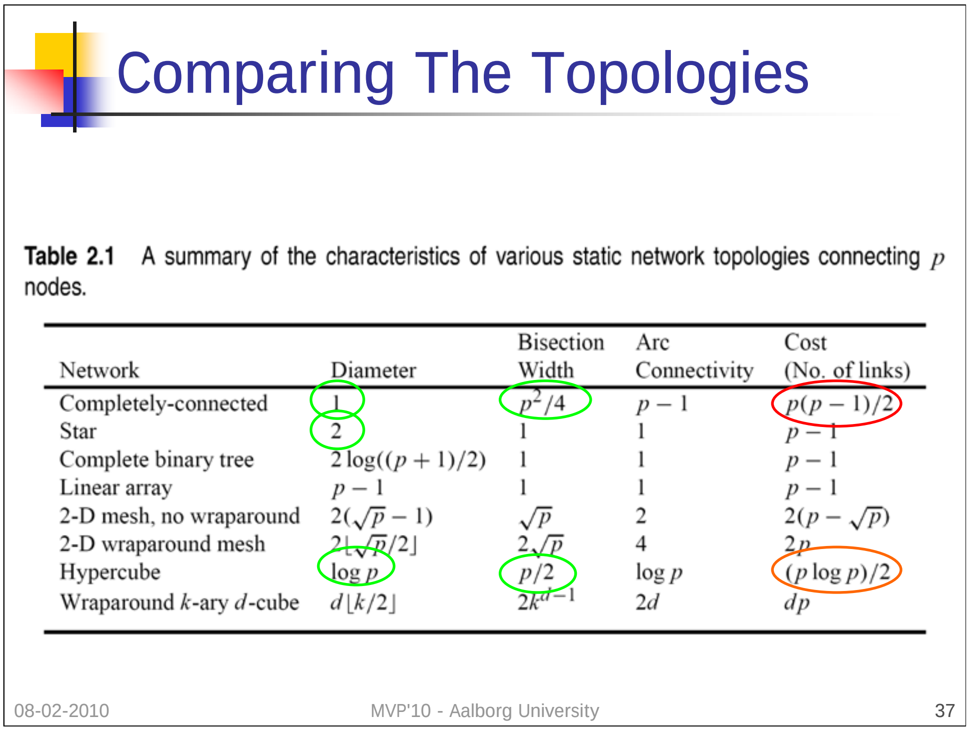
•Star connected: 2.

•Ring: floor(p/2).

•2-D mesh without wraparound: 2(dim-1). With wraparound: 2*floor(dim/2).
Note: dim = sqrt(*p*).

•Hypercube: dim (=log *p*).

•Complete binary tree: height=h, $p=2^{h+1}-1$, h =log($(p+1)/2$), travel $2h$.



Comparing The Topologies

Table 2.1 A summary of the characteristics of various static network topologies connecting p nodes.

Network	Diameter	Bisection Width	Arc Connectivity	Cost (No. of links)
Completely-connected	1	$p^2/4$	$p-1$	$p(p-1)/2$
Star	2	1	1	$p-1$
Complete binary tree	$2\log((p+1)/2)$	1	1	$p-1$
Linear array	$p-1$	1	1	$p-1$
2-D mesh, no wraparound	$2(\sqrt{p}-1)$	$\sqrt{p}$	2	$2(p-\sqrt{p})$
2-D wraparound mesh	$2\lfloor\sqrt{p}/2\rfloor$	$2\sqrt{p}$	4	$2p$
Hypercube	$\log p$	$p/2$	$\log p$	$(p\log p)/2$
Wraparound k-ary d-cube	$d\lfloor k/2\rfloor$	$2k^{d-1}$	$2d$	dp