Analysis and modeling of

Computational Performance

Execution time

for distributed memory programs

Execution time for parallel programs

- Execution time T_i for i-th thread/process
  - simplified model with explicitly indicated times for computations (including memory accesses), communication, system overhead and idle time
  - $T_i < T_i^{comp} + T_i^{comm} + T_i^{syst} + T_i^{idle}$
- The actual execution time depends on the component times, as well as the degree to which component times overlap
  - one of optimization goals is to maximize the overlap between computation and communication times, as well as minimizing and hiding system overhead
- Total execution time
  - from the beginning of the first thread, till the end of the last thread
  - after suitably complementing with idle time:
    - $T_{||} = \max_i(T_i)$
    - $T_{||} = \sum_i T_i /p$

Execution time for parallel programs

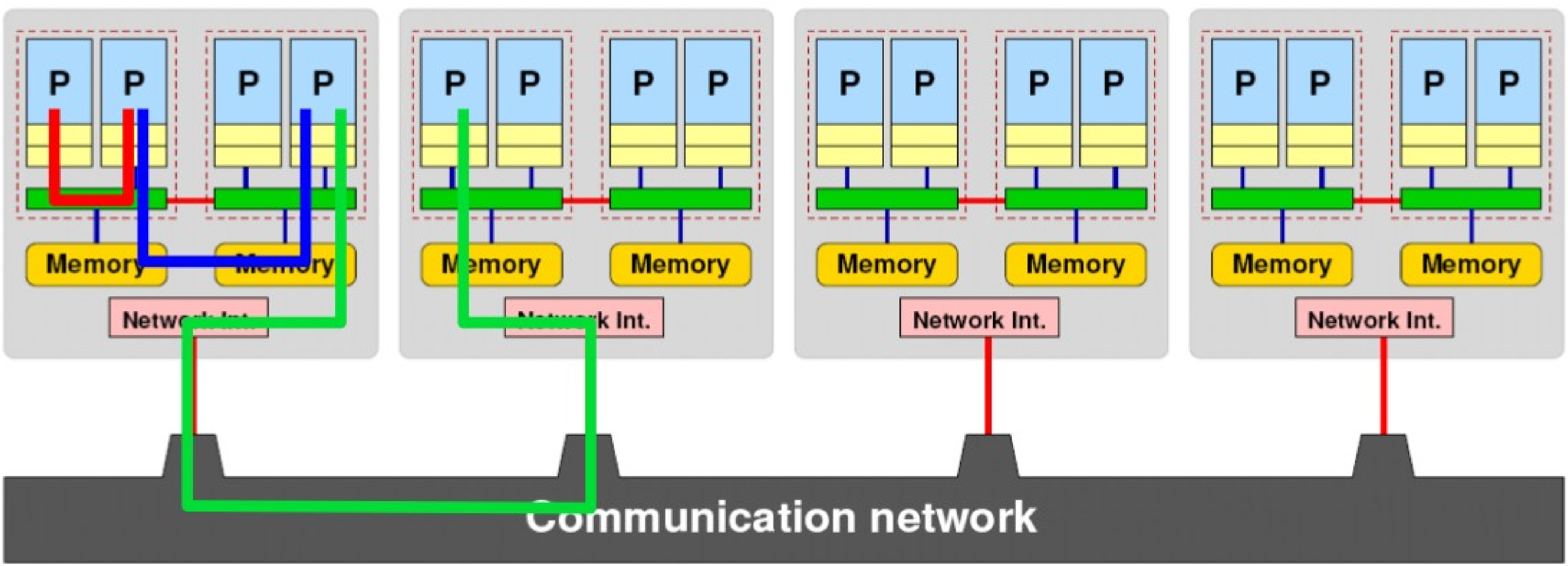
- Time for computations - T_i^{comp}
- In the general context of parallel computations, that include also distributed memory processing, T_i^{comp} concerns operations performed by processors and memory accesses
- The total computations time for all threads/processes can be compared with the time of sequential (i.e. single thread) execution
  - $pT_{||} = \sum_i T_i^{comp} = T^{seq} + T^{ovh} = T_{||}(1) + T^{ovh}$
    T^{ovh} – the overhead introduced by the parallel execution of the program
- The notions of parallel speed-up and parallel efficiency can be associated with the parallel overhead
  - $S(p) = T_{||}(1) / T_{||}(p) = pT_{||}(1) / (T_{||}(1) + T^{ovh}) < p$
  - $E(p) = S(p)/p = T_{||}(1) / (T_{||}(1) + T^{ovh}) < 1$
  - the larger overhead time the lower parallel performance

Execution time for parallel programs

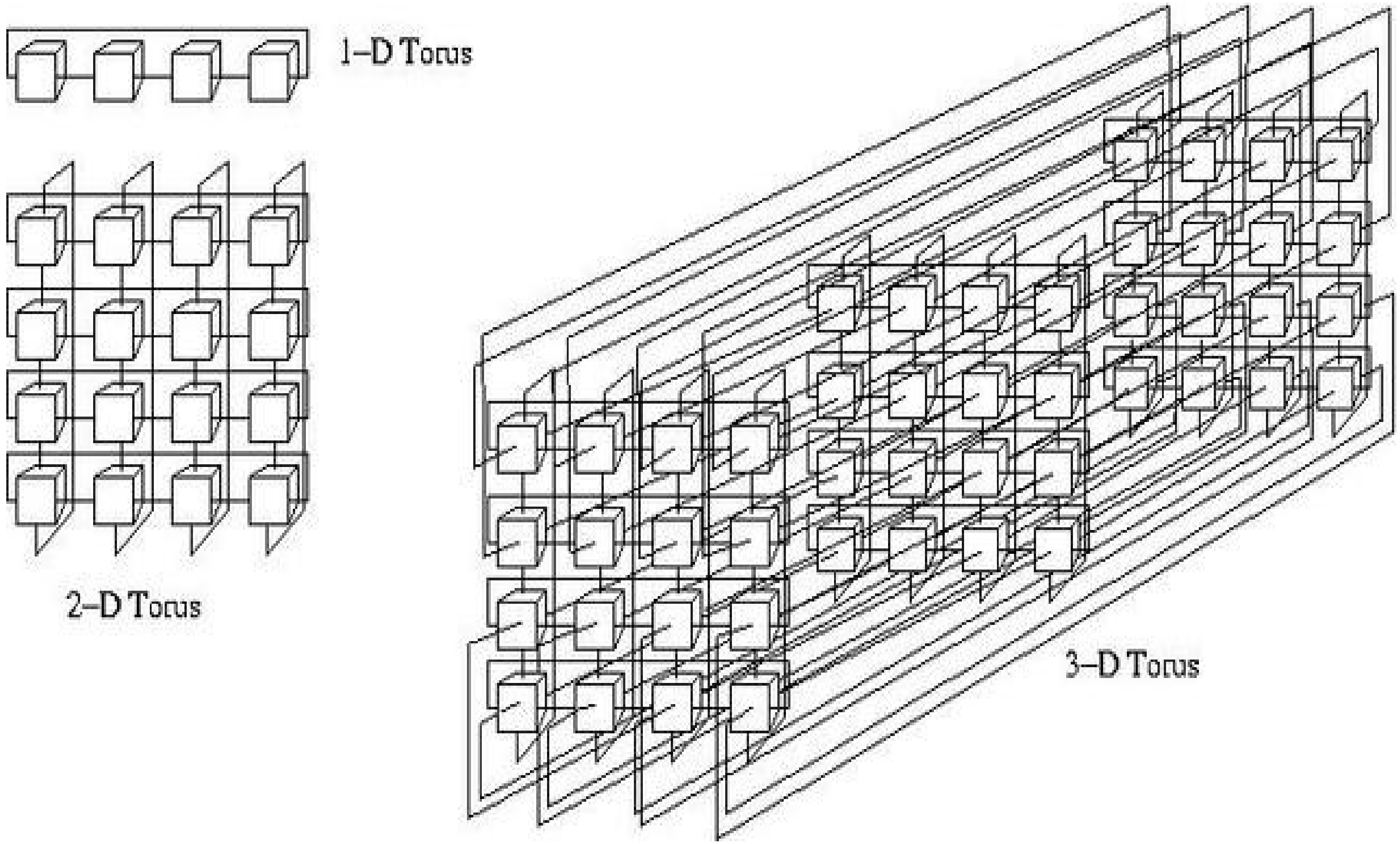
- Time for communication - T_i^{comm}
- Communication time can be modelled using assumptions concerning the technology of message passing and the topology of interconnection network
  - the simplest model for store-and-forward switching technology gives the time for sending *m* bytes from one computational node to another node, separated by *l* hops:
    - $T_i^{comm} = t_s + l*(m*t_w + t_h)$
      - with: t_s – startup time, t_w – time for sending a single byte, t_h – time for single hop switching
  - for cut-through technology, the simple model gives:
    - $T_i^{comm} = t_s + m*t_w + l*t_h$
  - for networks with cut-through technology and small single hop times a still more simplified model can be used:
    - $T_i^{comm} = t_s + m*t_w$

Execution time for parallel programs

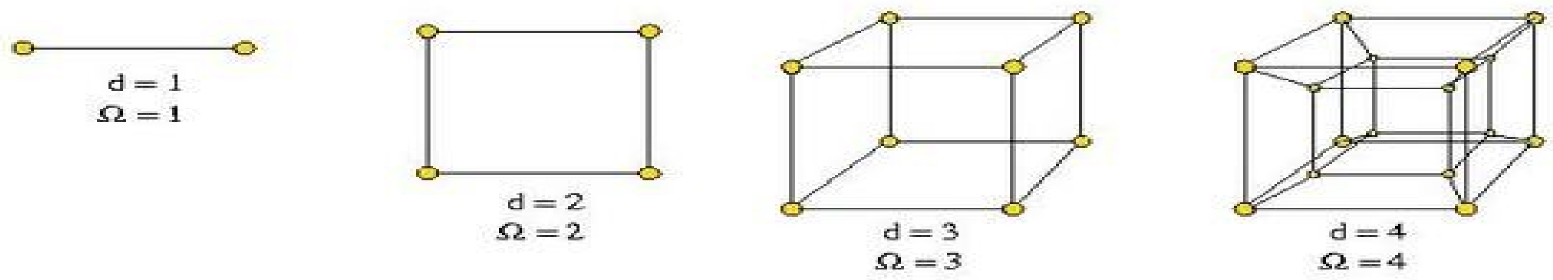
- Time for communication - T_i^{comm}
- Parameters for modelling communication time can be taken from technical specifications or measured for example configurations
- For today's complex hardware environments simple models of communication time may give inaccurate results



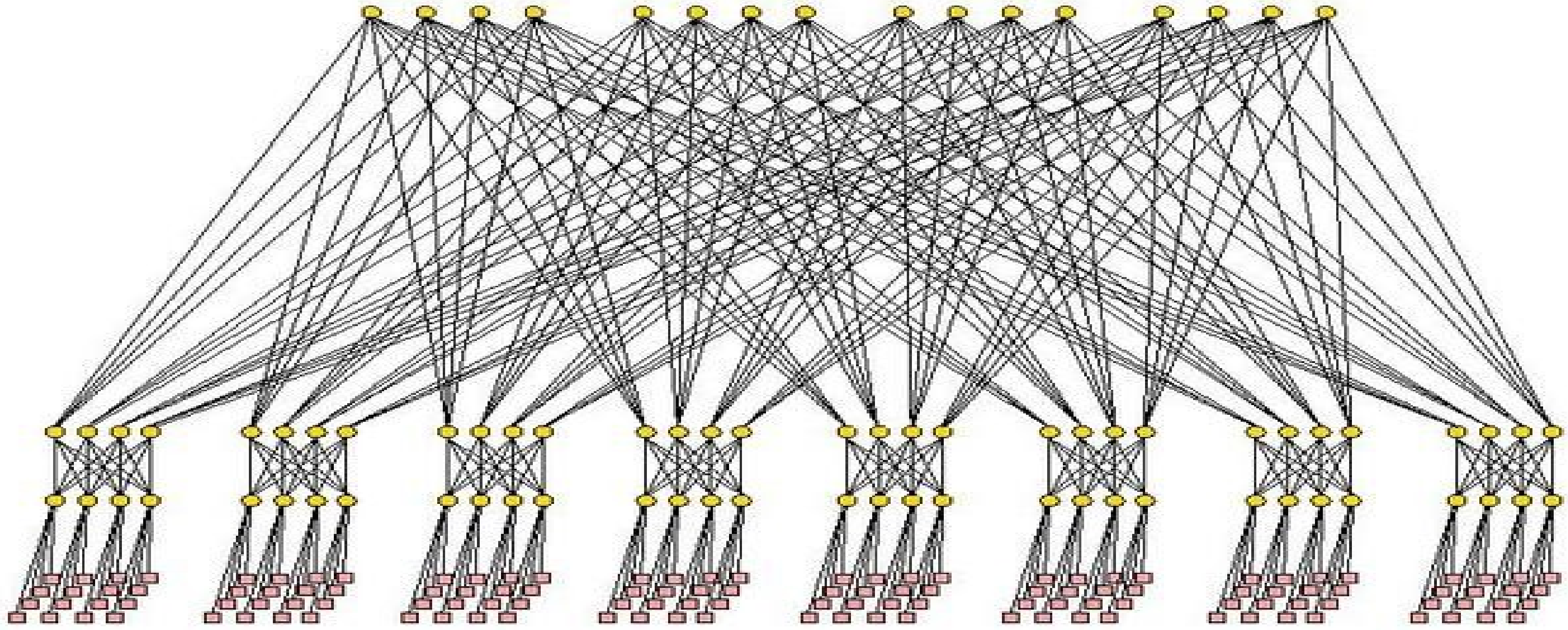
Interconnection network topologies



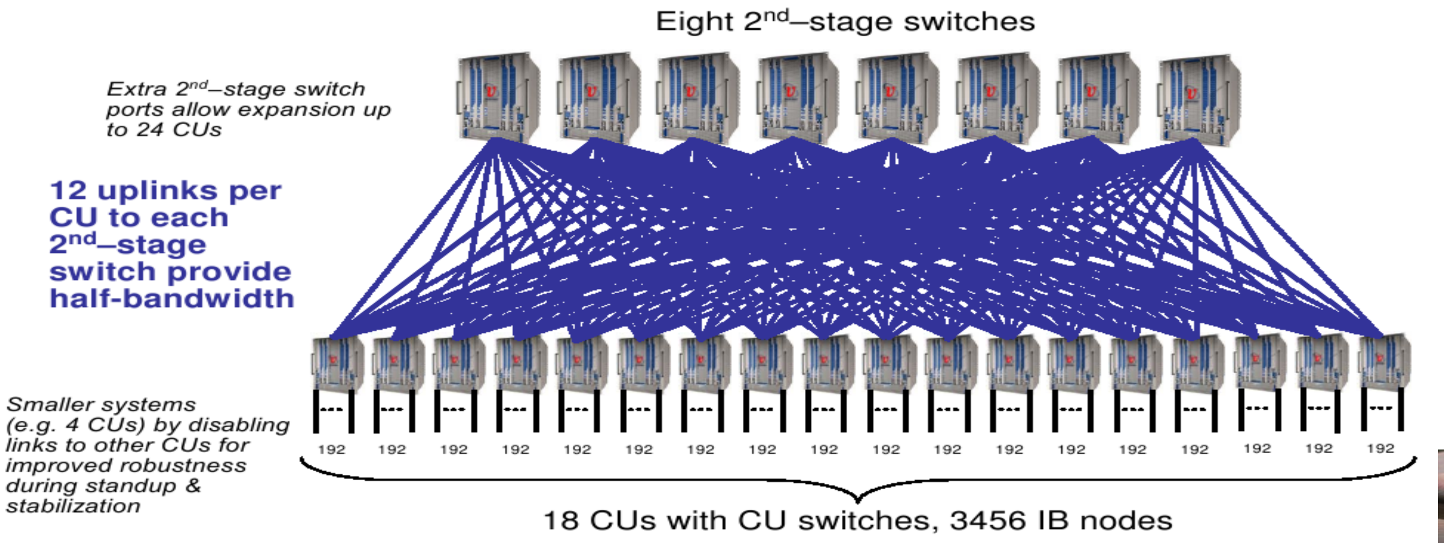
Interconnection network topologies

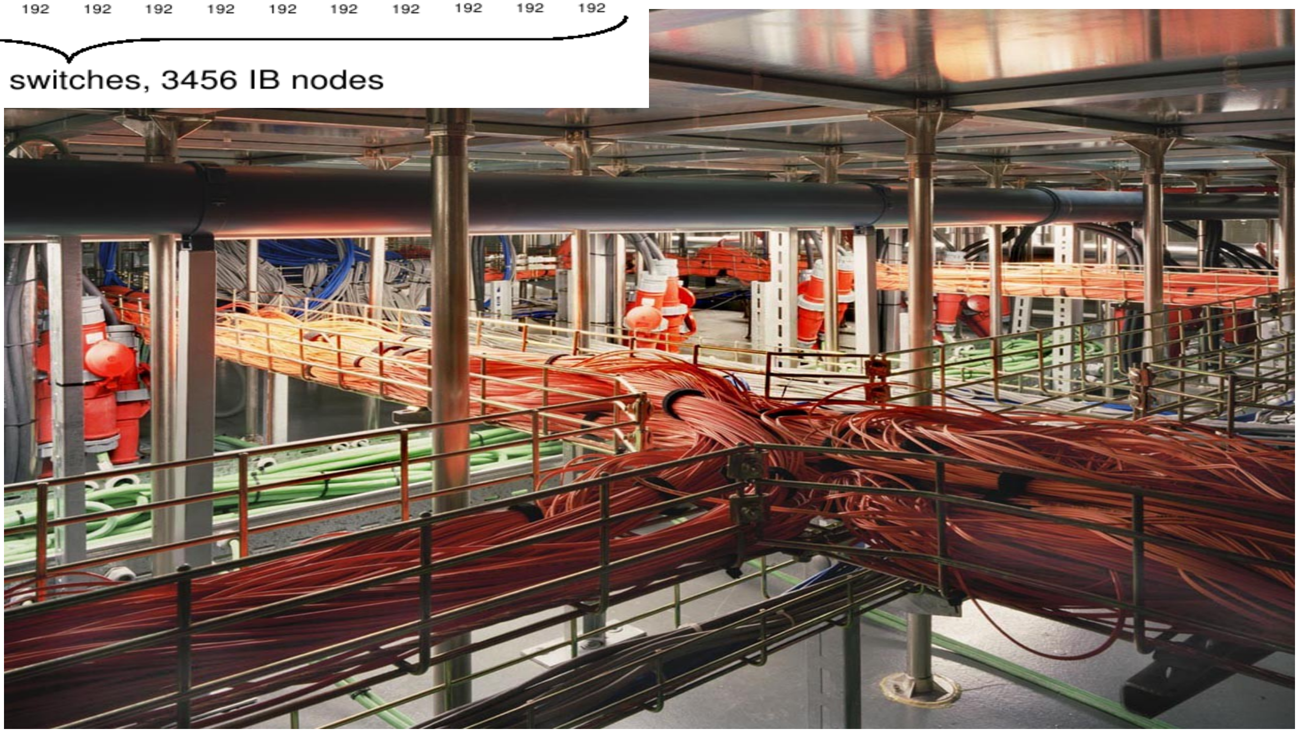


(a) Hypercubes, dimension 1-4.


(b) A 128-way fat tree.

Interconnection network topologies





The role of network parameters

- Loosely coupled versus tightly coupled computations

Interconnection network topologies

	I	II	III	IV
fully connected newtork	1	p-1	(p^2)/4	p(p-1)/2
ring	p/2	2	2	p
2D mesh	2(√p - 1)	2	√p	2(p-√p)
2D torus	2(√p / 2)	4	2√p	2p
binary tree	2log(p/2+1/2)	1	1	p-1
hypercube	log p	log p	p/2	p(log p)/2

	Ring	2D Mesh	2D Torus	3D Mesh	3D Torus	Fat Tree	CCC
Average Hop Count	16	7	4	14	3	9	5.4
Maximum Hop Count	32	14	8	7	6	11	10
Average Latency	16	10.6	8	12	9	38	10.8
Bisection BW	8	16	32	32	43	21	21
Effective BW: Uniform	16	32	64	64	64	42	42
Effective BW: Hot-spot	1	1	1	1	1	1	1
Effective BW: Bit Complement	8	16	32	32	43	21	21
Effective BW: NEWS	22.4	64	64	64	64	64	64
Effective BW: Transpose	7	28	56	56	64	36.8	36.8
Effective BW: Perfect-Shuffle	8	32	64	64	64	42	42

Table 5: Comparison of performance of topologies with 64 endnodes

Collective communication times

- Communication times for collective operations on hypercube
  - *One-to-all broadcast:* $T_{B\_HC} = (t_s + mt_w) \log(p)$
  - *All-to_one reduction:* $T_{R\_HC} = (t_s + mt_w) \log(p)$
  - *Allgather (all-to-all broadcast):* $T_{AG\_HC} = t_s \log p + mt_w (p - 1)$
  - *All-to-all reduction:* $T_{AR\_HC} = (t_s + t_w) \log p$
  - *Gather and scatter:* $T_{G(S)\_HC} = t_s \log p + mt_w (p - 1)$
  - *All-to-all (full exchange):* $T_{AA\_HC} = (t_s + \frac{1}{2} pmt_w) \log(p)$
- For other topologies communication times are the same or longer, e.g.:
  - *Allgather for 2D torus:* $T_{AG\_2T} = 2t_s (\sqrt{p} - 1) + mt_w (p - 1)$
  - *All-to-all for 2D torus:* $T_{AA\_2T} = (2t_s + pmt_w) (\sqrt{p} - 1)$