An Introduction to Parallel Computing

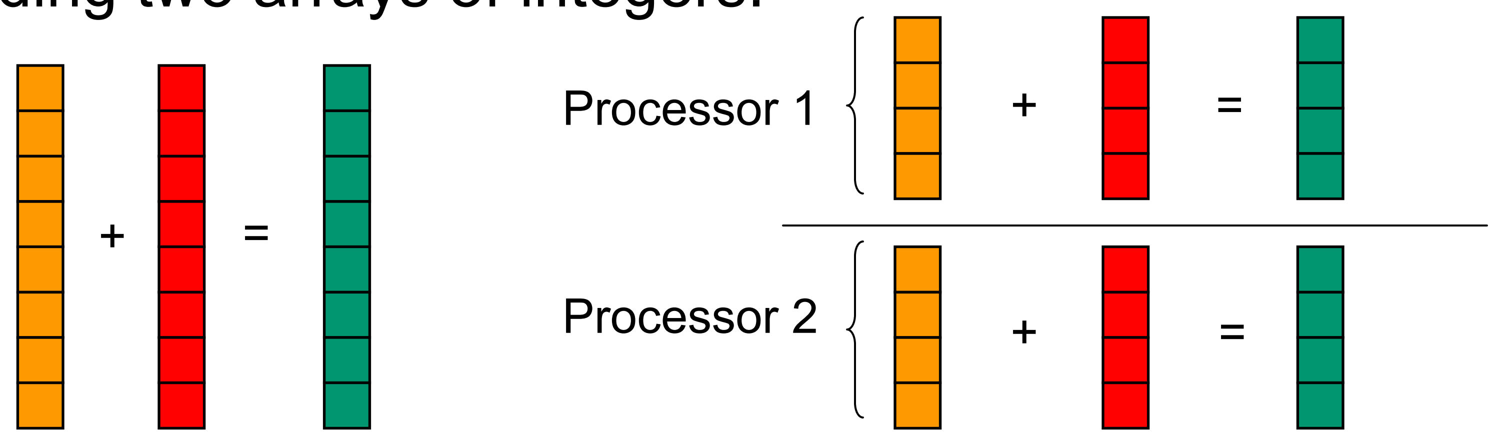
Edgar Gabriel
Department of Computer Science
University of Houston
gabriel@cs.uh.edu

Why Parallel Computing?

- To solve larger problems
  - many applications need significantly more memory than a regular PC can provide/handle
- To solve problems faster
  - despite of many advances in computer hardware technology, many applications are running slower and slower
    - e.g. databases having to handle more and more data
    - e.g. large simulations working on even more accurate solutions

Parallel Programming

- Exploit concurrency
  - Internet: Client and server are independent, interacting applications
  - Searching an element: distribute the search database onto multiple processors
  - Adding two arrays of integers:



Parallel Programming (II)

- Scalar product:

$$s = \sum_{i=0}^{N-1} a[i] * b[i]$$

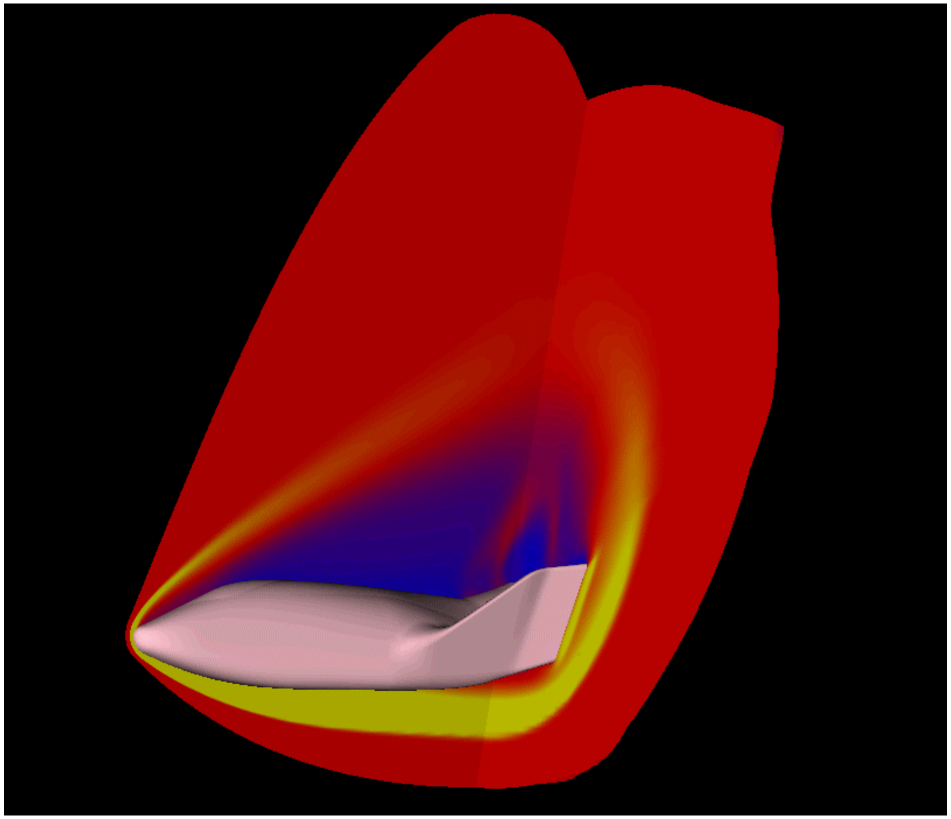
- Parallel algorithm

$$s = \sum_{i=0}^{N/2-1} (a[i] * b[i]) + \sum_{i=N/2}^{N-1} (a[i] * b[i])$$

$$= \underbrace{\sum_{i=0}^{N/2-1} (a_{local}[i] * b_{local}[i])}_{rank=0} \quad + \quad \underbrace{\sum_{i=0}^{N/2-1} (a_{local}[i] * b_{local}[i])}_{rank=1}$$

– requires communication between the processes

Flow around a re-entry vehicle



HPC in movies

- Special effects are highly compute intensive
- Example: Lord of the Rings
  - company: Weta Digitals
  - 3200 processor cluster
  - a single scene contains:
    - per second 24 frames
    - per frame: 4996 x 3112 points with 32- or 64 bit color encoding
    - each object means a separate compute cycle
- Number of computer-added special effects in movies:
  - Jurassic Park (I): 75
  - Lord of the Rings (I): 540
  - each of the following episodes of Lord of the Rings doubled the number of special effects
  - last episode of Star-Wars: 2000-2500

What is a Parallel Computer?

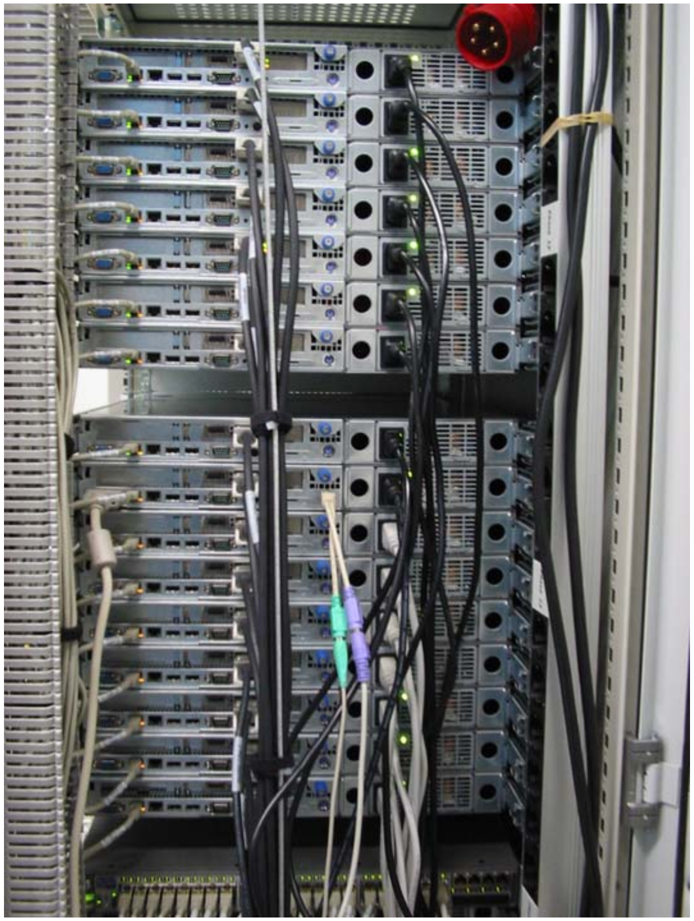
What is a Parallel Computer?

PC cluster in 2000

- 4U height per node
- Nodes:
  - 1-4 processors
  - 1-4 GB memory
- Network interconnects:
  - Myrinet (~ 200MB/s, 7µs)
  - Fast Ethernet (~ 12MB/s, 1ms)

(1U = 1.75” = 4.45 cm)

What is a Parallel Computer?

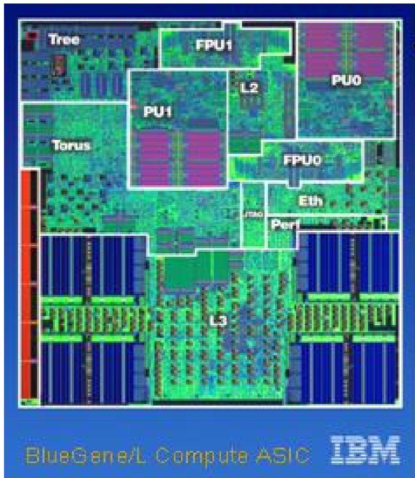


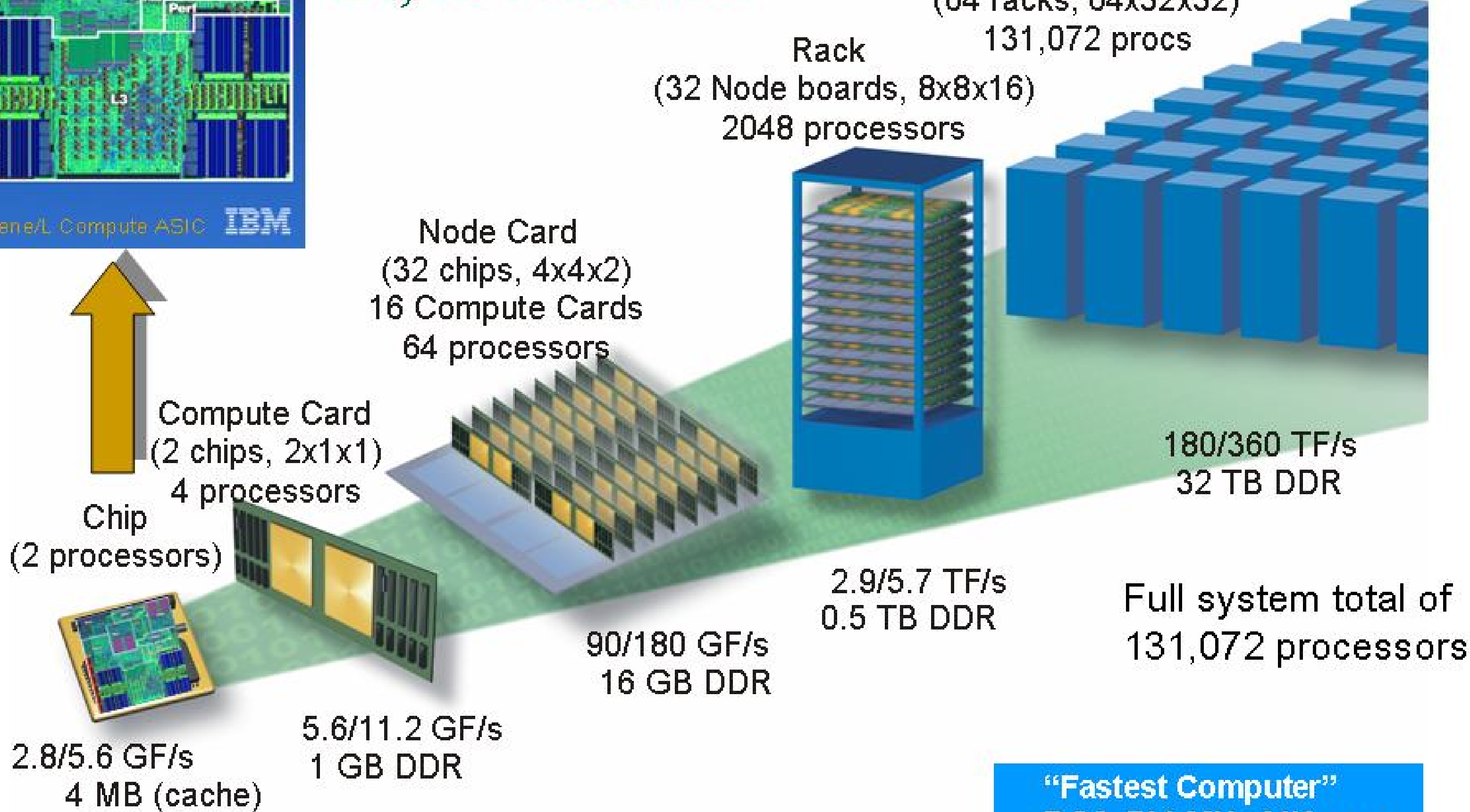
PC Cluster in 2005

- 1U height
- Nodes:
    - 1-8 Processors
    - 1-32 GB main memory
- Network interconnects
    - Infiniband (~ 1 GB/s, 5 μs)
    - Gigabit Ethernet (~80 MB/s, 50 μs)

IBM BlueGene/L

131,072 Processors

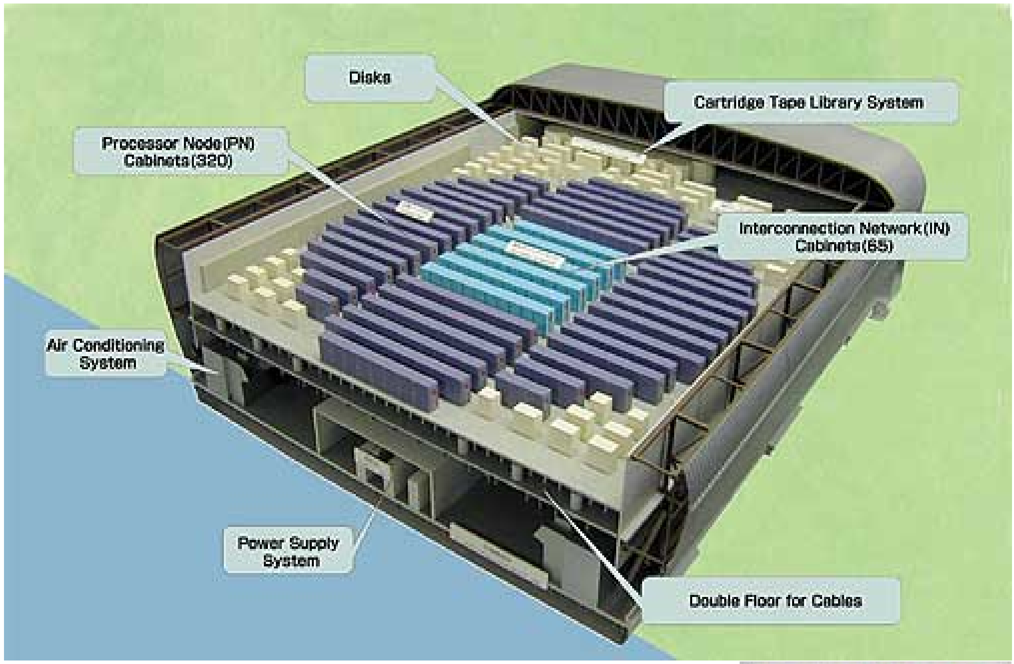




"Fastest Computer"
BG/L 700 MHz 64K proc
32 racks
Peak: 184 Tflop/s
Linpack: 135 Tflop/s
73% of peak

Earth Simulator

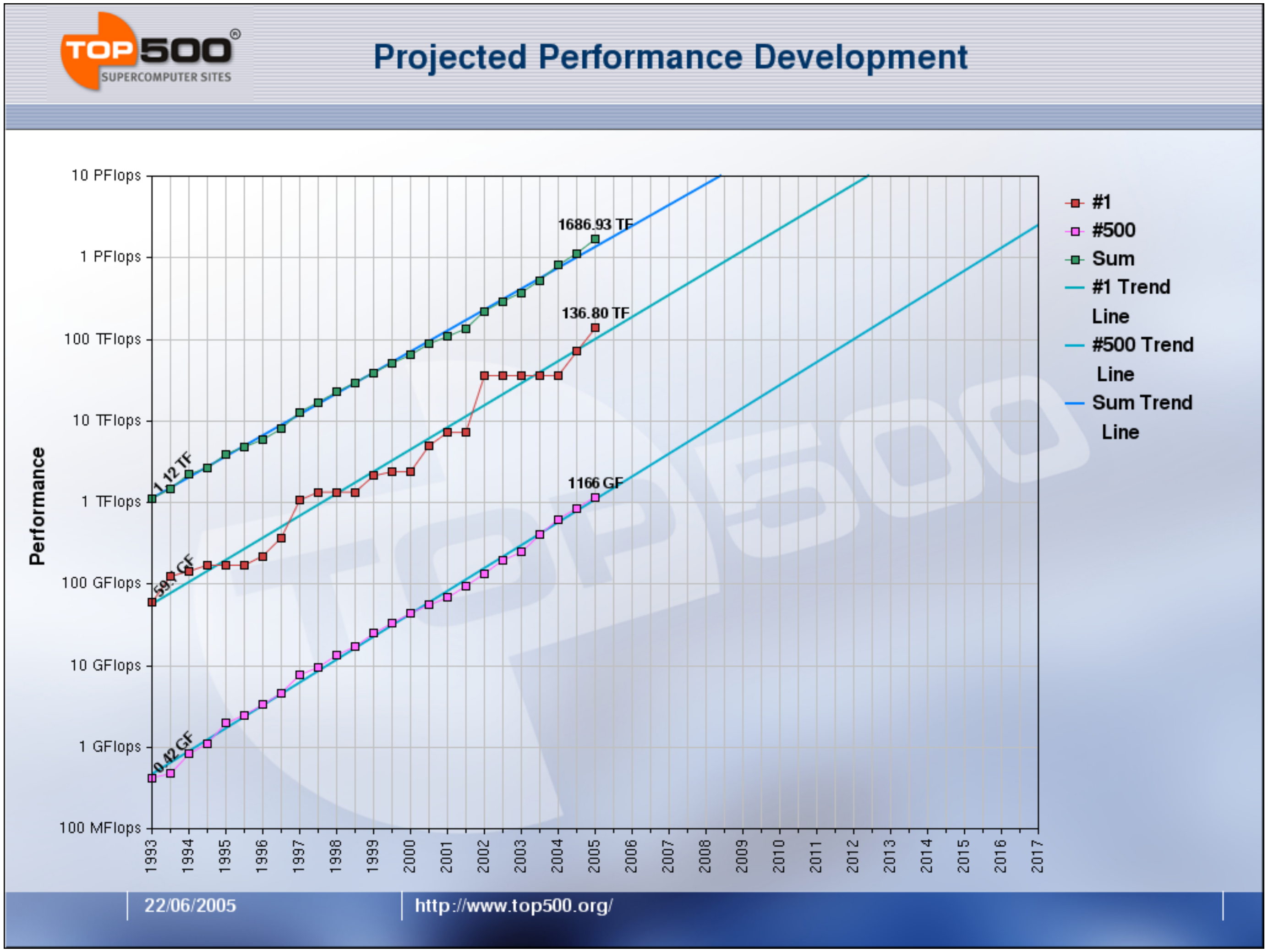
- Target: Achievement of high-speed numerical simulations with processing speed of 1000 times higher than that of the most frequently used supercomputers in 1996. (www.es.jamstec.go.jp)
  - 640 nodes
  - 8 processors/node
  - 40 TFLOPS peak
  - 10 TByte memory



Top 500 List (www.top500.org)

Rank	Site Country/Year	Computer / Processors Manufacturer	Computer Family Model	Inst. type Installation Area	Rmax Rpeak	Nmax nhalf
1	DOE/NNSA/LLNL United States/2005	*BlueGene/L* **eServer Blue Gene Solution** / 65536 IBM	**IBM BlueGene/L** eServer Blue Gene Solution	**Research**	136800 183500	1277951
2	IBM Thomas J. Watson Research Center United States/2005	*BGW* **eServer Blue Gene Solution** / 40960 IBM	**IBM BlueGene/L** eServer Blue Gene Solution	**Research** Information Processing Service	91290 114688	983039
3	NASA/Ames Research Center/NAS United States/2004	*Columbia* **SGI Altix 1.5 GHz, Voltaire Infiniband** / 10160 SGI	**SGI Altix** SGI Altix 3700	**Research**	51870 60960	1290240
4	The Earth Simulator Center Japan/2002	**Earth-Simulator** / 5120 NEC	**NEC Vector** SX6	**Research**	35860 40960	1075200 266240
5	Barcelona Supercomputer Center Spain/2005	*MareNostrum* **JS20 Cluster, PPC 970, 2.2 GHz, Myrinet** / 4800 IBM	**IBM Cluster** JS20 CLuster	**Academic**	27910 42144	977816
6	ASTRON/University Groningen Netherlands/2005	**eServer Blue Gene Solution** / 12288 IBM	**IBM BlueGene/L** eServer Blue Gene Solution	**Academic**	27450 34406.4	516095
7	Lawrence Livermore National Laboratory United States/2004	*Thunder* **Intel Itanium2 Tiger4 1.4GHz - Quadrics** / 4096 California Digital Corporation	**NOW - Intel Itanium** Itanium2 Tiger4 Cluster	**Research**	19940 22938	975000 110000
8	Computational Biology Research Center, AIST Japan/2005	*Blue Protein* **eServer Blue Gene Solution** / 8192 IBM	**IBM BlueGene/L** eServer Blue Gene Solution	**Research**	18200 22937.6	442367
9	Ecole Polytechnique Federale de Lausanne Switzerland/2005	**eServer Blue Gene Solution** / 8192 IBM	**IBM BlueGene/L** eServer Blue Gene Solution	**Academic**	18200 22937.6	442367
10	Sandia National Laboratories United States/2005	**Red Storm, Cray XT3, 2.0 GHz** / 5000 Cray Inc.	**Cray XT3** Cray XT3	**Research**	15250 20000	
11	Oak Ridge National Laboratory United States/2005	**Cray XT3, 2.4 GHz** / 3748 Cray Inc.	**Cray XT3** Cray XT3	**Research**	14170 17990	

Top 500 List


TOP500 SUPERCOMPUTER SITES
Projected Performance Development
10 PFlops
1 PFlops
100 TFlops
10 TFlops
1 TFlops
100 GFlops
10 GFlops
1 GFlops
100 MFlops
Performance
1686.93 TF
136.80 TF
1166 GF
1.12 TF
59.7 GF
0.42 GF
1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017
#1
#500
Sum
#1 Trend Line
#500 Trend Line
Sum Trend Line
22/06/2005
http://www.top500.org/

Recommended Literature

- Timothy G. Mattson, Beverly A. Sanders, Berna L. Massingill "*Patterns for Parallel Programming*" Software Pattern Series, Addison Wessley, 2005.
- Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar: "*Introduction to Parallel Computing*", Pearson Education, 2003.
- Jack Dongarra, Ian Foster, Geoffrey Fox, William Gropp, Ken Kennedy, Linda Torczon, Andy White "*Sourcebook of Parallel Computing*", Morgan Kaufmann Publishers, 2003.
- Michael J. Quinn: "*Parallel Programming in C with MPI and OpenMP*", McGrawHill, 2004.
- L. Ridgeway Scott, Terry Clark, Babak Bagheri: "*Scientific Parallel Computing*", Princeton University Press, 2005.

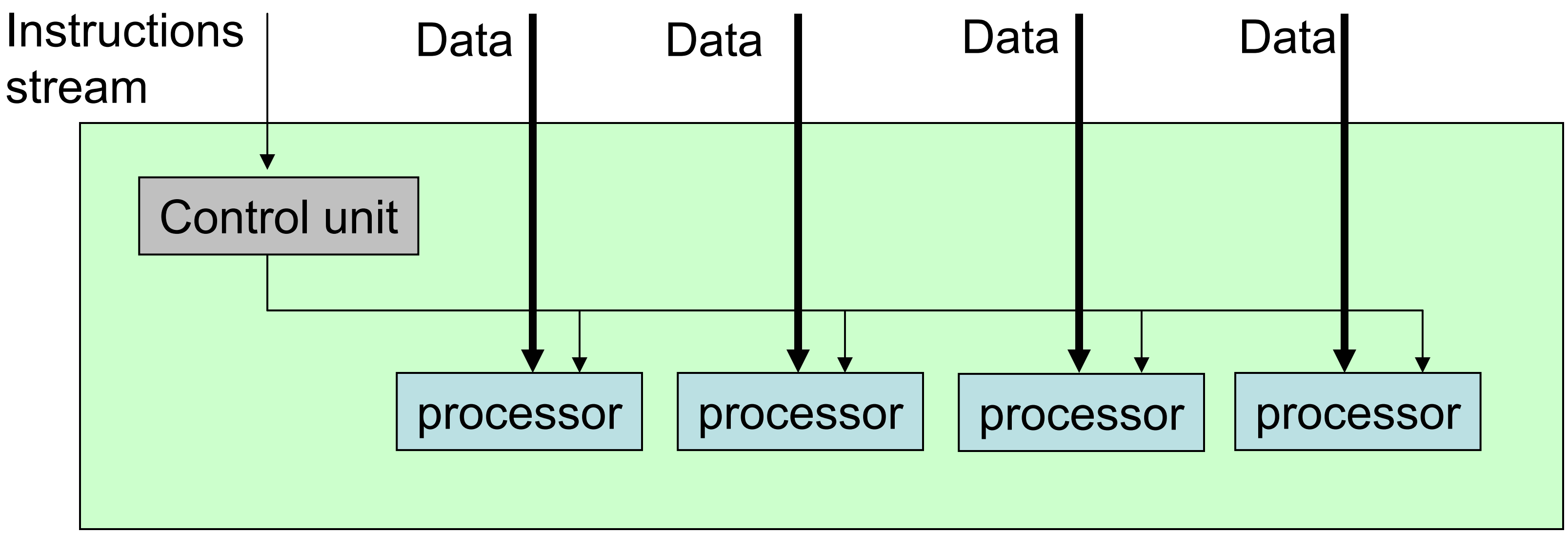
Classification of Parallel Architectures

Flynn's Taxonomy

- SISD: Single instruction single data
  - Classical von Neumann architecture
- SIMD: Single instruction multiple data
- MISD: Multiple instructions single data
  - Non existent, just listed for completeness
- MIMD: Multiple instructions multiple data
  - Most common and general parallel machine

Single Instruction Multiple Data

- Also known as Array-processors
- A single instruction stream is broadcasted to multiple processors, each having its own data stream
  - Still used in graphics cards today



Multiple Instructions Multiple Data (I)

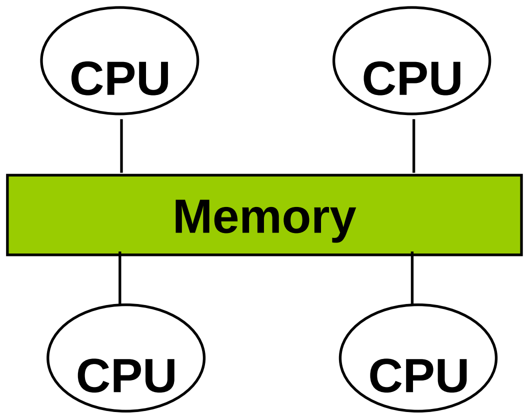
- Each processor has its own instruction stream and input data
- Very general case
  - every other scenario can be mapped to MIMD
- Further breakdown of MIMD usually based on the memory organization
  - Shared memory systems
  - Distributed memory systems

Shared memory systems (I)

- All processes have access to the same address space
  - E.g. PC with more than one processor
- Data exchange between processes by writing/reading shared variables
  - Shared memory systems are easy to program
  - Current standard in scientific programming: OpenMP
- Two versions of shared memory systems available today
  - Symmetric multiprocessors (SMP)
  - Non-uniform memory access (NUMA) architectures

Symmetric multi-processors (SMPs)

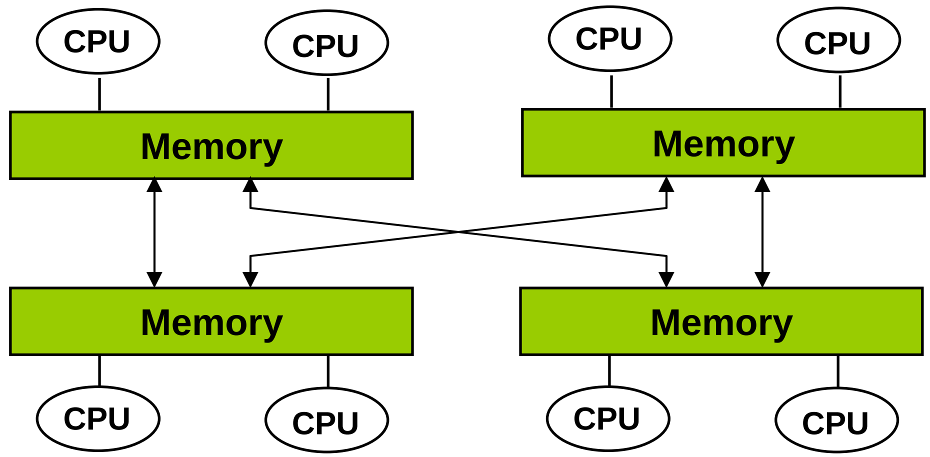
- All processors share the same physical main memory



- Memory bandwidth per processor is limiting factor for this type of architecture
- Typical size: 2-32 processors

NUMA architectures (I)

- Some memory is closer to a certain processor than other memory
  - The whole memory is still addressable from all processors
  - Depending on what data item a processor retrieves, the access time might vary strongly

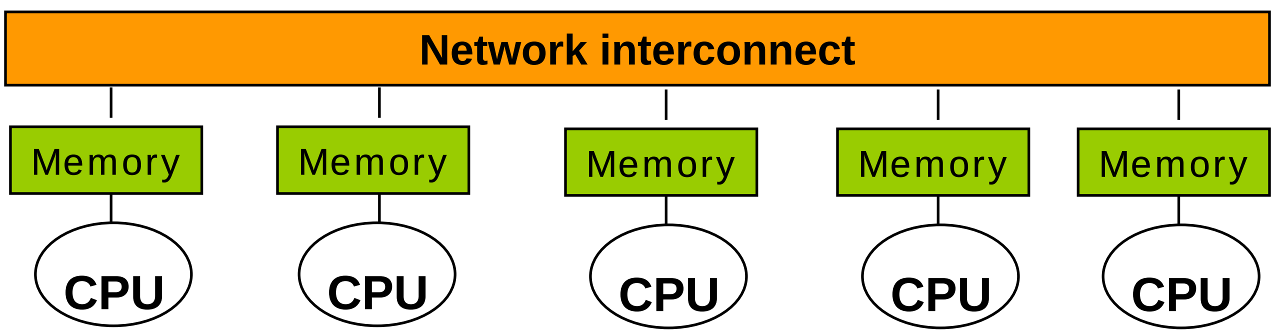


NUMA architectures (II)

- Reduces the memory bottleneck compared to SMPs
- More difficult to program efficiently
  - E.g. first touch policy: data item will be located in the memory of the processor which uses a data item first
- To reduce effects of non-uniform memory access, caches are often used
  - ccNUMA: cache-coherent non-uniform memory access architectures
- Largest example as of today: SGI Origin with 512 processors

Distributed memory machines (I)

- Each processor has its own address space
- Communication between processes by explicit data exchange
  - Sockets
  - Message passing
  - Remote procedure call / remote method invocation



Distributed memory machines (II)

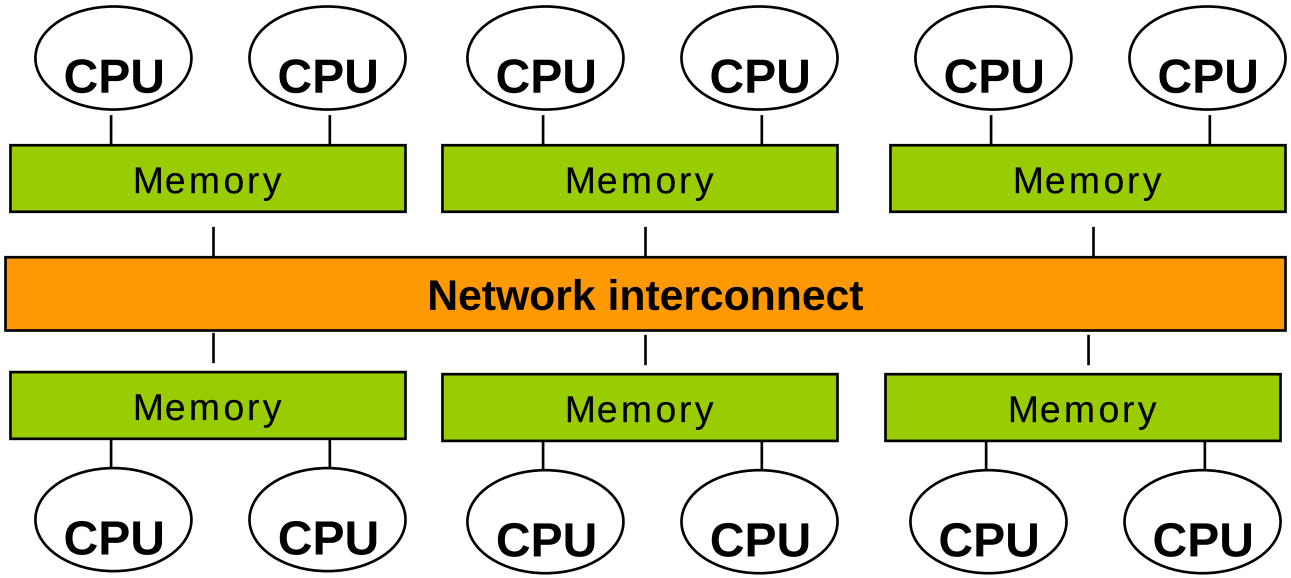
- Performance of a distributed memory machine strongly depends on the quality of the network interconnect and the topology of the network interconnect
  - Of-the-shelf technology: e.g. fast-Ethernet, gigabit-Ethernet
  - Specialized interconnects: Myrinet, Infiniband, Quadrics, 10G Ethernet …

Distributed memory machines (III)

- Two classes of distributed memory machines:
  - Massively parallel processing systems (MPPs)
    - Tightly coupled environment
    - Single system image (specialized OS)
  - Clusters
    - Of-the-shelf hardware and software components such as
      - Intel P4, AMD Opteron etc.
      - Standard operating systems such as LINUX, Windows, BSD UNIX

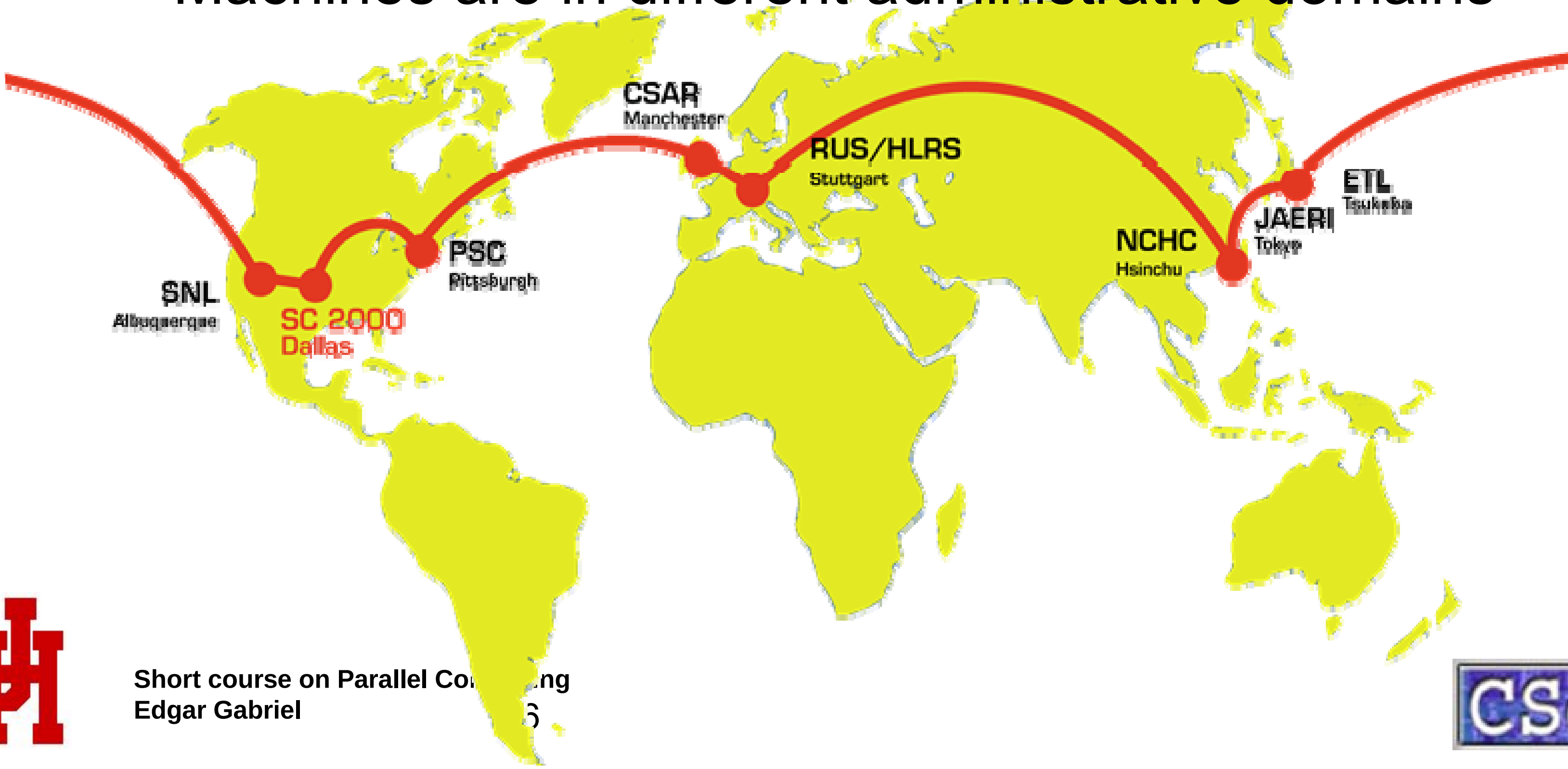
Hybrid systems

- E.g. clusters of multi-processor nodes



Grids

- ‘Evaluation’ of distributed memory machines and distributed computing
- Several (parallel) machines connected by wide-area links (typically the internet)
  – Machines are in different administrative domains



Network topologies (I)

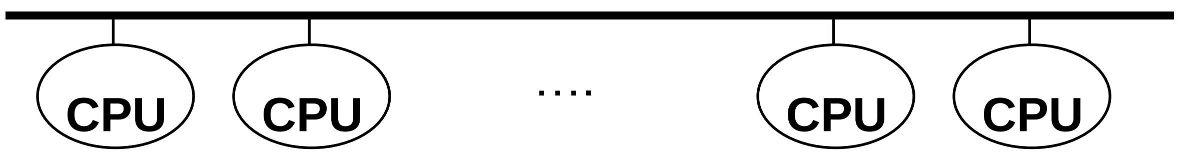
- Important metrics:
  - Latency:
    - minimal time to send a very short message from one processor to another
    - Unit: ms, μs
  - Bandwidth:
    - amount of data which can be transferred from one processor to another in a certain time frame
    - Units: Bytes/sec, KB/s, MB/s, GB/s
      Bits/sec, Kb/s, Mb/s, Gb/s,
      baud

Network topologies (II)

Metric	Description	Optimal parameter
Link	A direct connection between two processors	
Path	A route between two processors	As many as possible
Distance	Minimum length of a path between two processors	Small
Diameter	Maximum distance in a network	Small
Degree	Number of links that connect to a processor	Small (costs) / Large (redundancy)
Connectivity	Minimum number of links that have to be cut to separate the network	Large (reliability)
Increment	Number of procs to be added to keep the properties of a topology	Small (costs)
Complexity	Number of links required to create a network topology	Small (costs)

Bus-Based Network

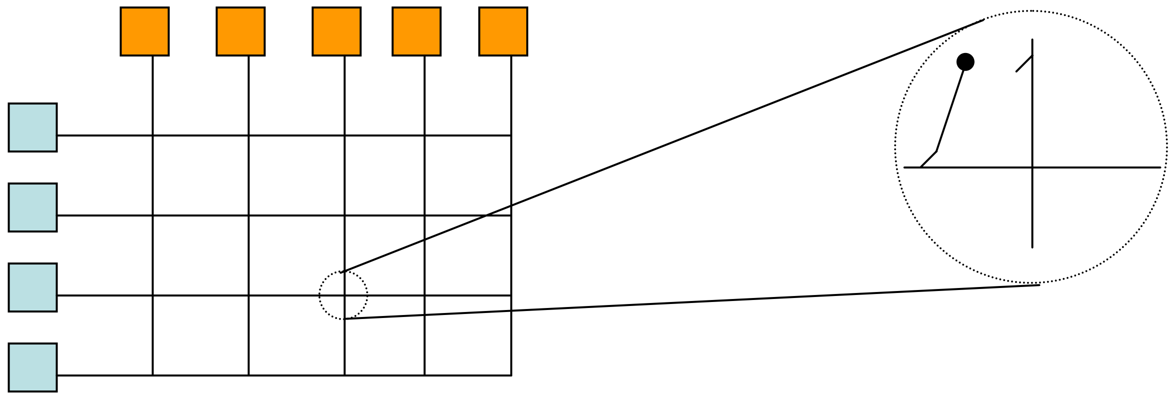
- All nodes are connected to the same (shared) communication medium
- Only one communication at a time possible
  - Does not scale



- Examples: Ethernet, SCSI, Token Ring, Memory bus
- Main advantages:
  - Cheap
  - Simple broadcast

Crossbar Networks (I)

- A grid of switches connecting *n*x*m* ports



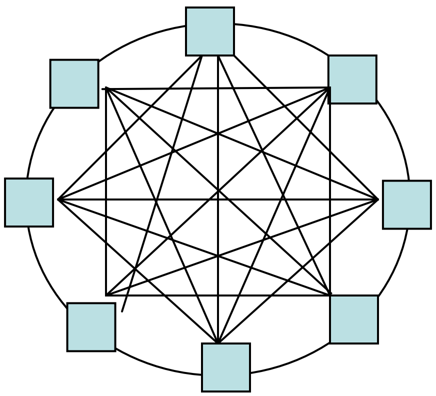
- a connection from one process to another does not prevent communication between other process pairs
- Aggregated Bandwidth of a crossbar: sum of the bandwidth of all possible simultaneous connections

Directly connected networks

- A direct connection between two processors exists
- Relevant topologies
  - Ring
  - Star
  - Fully connected
  - Meshes
  - Toruses
  - Tree based networks
  - Hypercubes

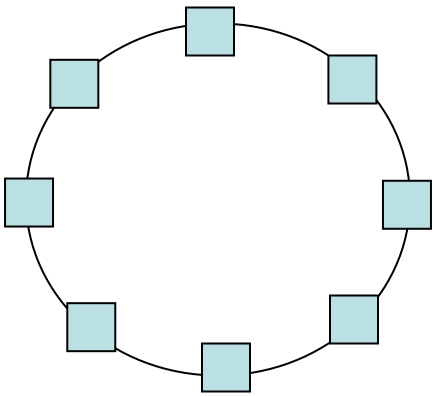
fully connected network

- Every node is connected directly with every other node
  - Distance: 1
  - Diameter: 1
  - Degree: N-1
  - Connectivity: N-1
  - Increment: 1
  - Complexity: N*(N-1)/2
- Positive:
  - Fast: one *hop* to each node
  - Fault-tolerant
- Negative:
  - Does not scale / expensive
  - Technically difficult!



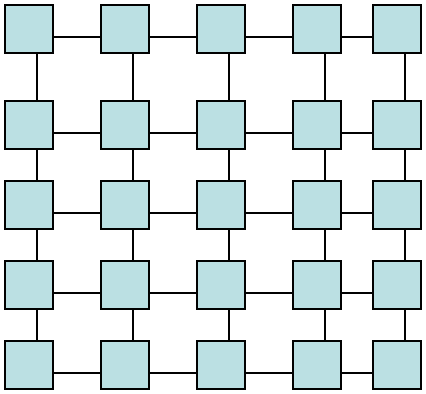
Ring network

- N: Number of processor connected by the network
  - Distance: 1: N/2
  - Diameter: N/2
  - Degree: 2
  - Connectivity: 2
  - Increment: 1
  - Complexity: N-1



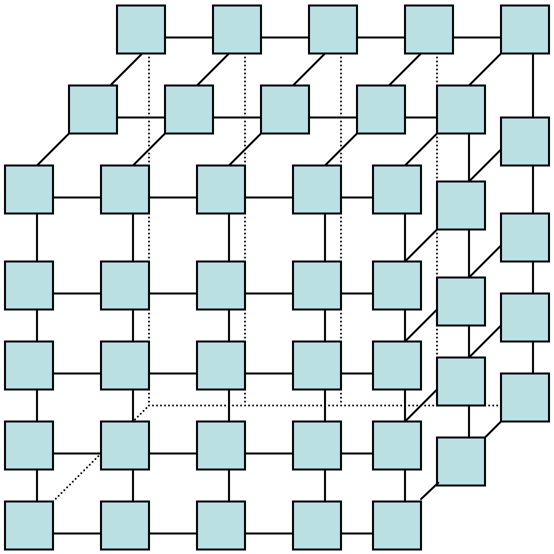
Meshes (I)

- E.g. 2-D mesh
  - Distance: 1:~ $2\sqrt{N}$
  - Diameter: ~ $2\sqrt{N}$
  - Degree: 2-4
  - Connectivity: 2
  - Increment: ~ $\sqrt{N}$
  - Complexity: ~2N



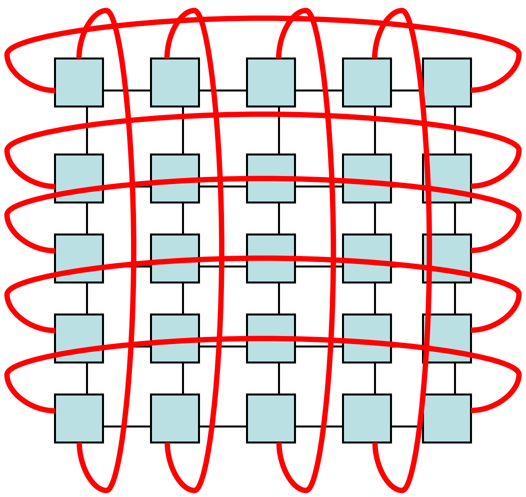
Meshes (II)

- E.g. 3-D mesh
  - Distance: 1:~ $3\sqrt[3]{N}$
  - Diameter: ~ $3\sqrt[3]{N}$
  - Degree: 3-6
  - Connectivity: 3
  - Increment: ~
  - Complexity: ~ $\sqrt[3]{N}$



Toruses (I)

- E.g. 2-D Torus
  - Distance: 1:~ $\sqrt{N}$
  - Diameter: ~ $\sqrt{N}$
  - Degree: 4
  - Connectivity: 4
  - Increment: ~ $\sqrt{N}$
  - Complexity: ~2N



Tree-based networks

- Leafs are computational nodes, Intermediate nodes in the tree are switches
- Fat tree: binary tree which increases the number of communication links between higher level switching elements to avoid contention
  - Distance: $1:2\log_2(N)$
  - Diameter: $2\log_2(N)$
  - Degree: 1
  - Connectivity: 1
  - Increment: N
  - Complexity: ~2N