Lecture 6 - Parallel Computer Architectures

- **Computer program performance relies on**
  - the processor speed and
  - the ability of the memory system to feed data to the processor
- **A memory system takes in a request for a word of memory and returns a bock of data of size "b" containing the requested word after "l" nanoseconds**
  - "l" is referred to as the ***latency*** of the memory (time-delay to fetch data)
  - The rate at which the data can be sent from memory to the processor determines the ***bandwidth*** of the memory system

Parallel Computer Architectures

- **Some Units:**

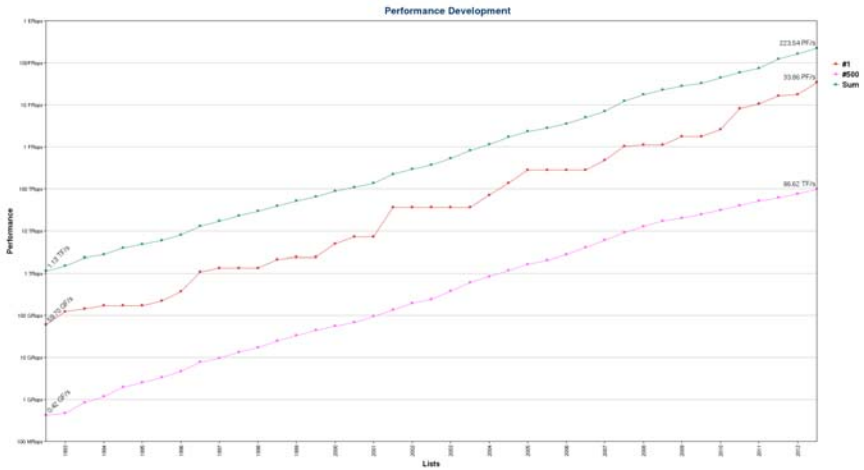
1Mflop= 10^6 flops/sec **1Mbyte=10^6 bytes**

1Gflop= 10^9 flops/sec **1Gbyte=10^9 bytes**

1Tflop= 10^{12} flops/sec **1Tbyte=10^{12} bytes**

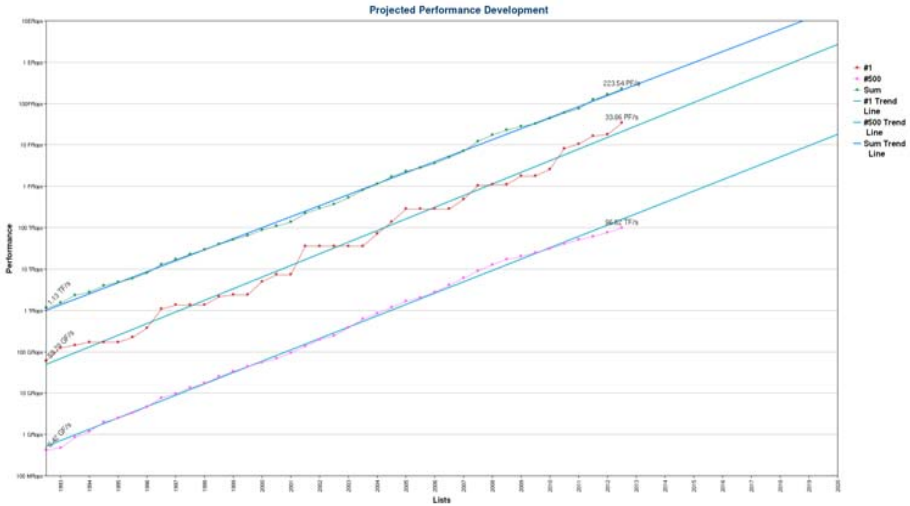
1Pflop= 10^{15} flops/sec **1Pbyte=10^{15} bytes**

Current Supercomputer Performance



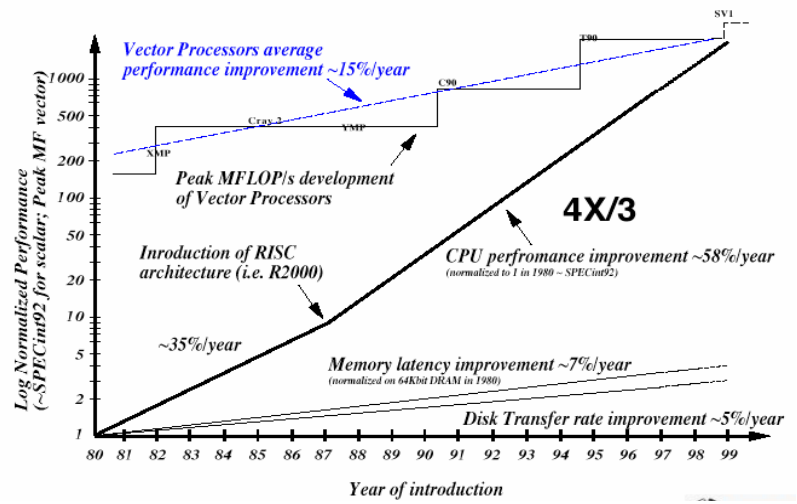
www.top500.org

Projected Supercomputer Performance

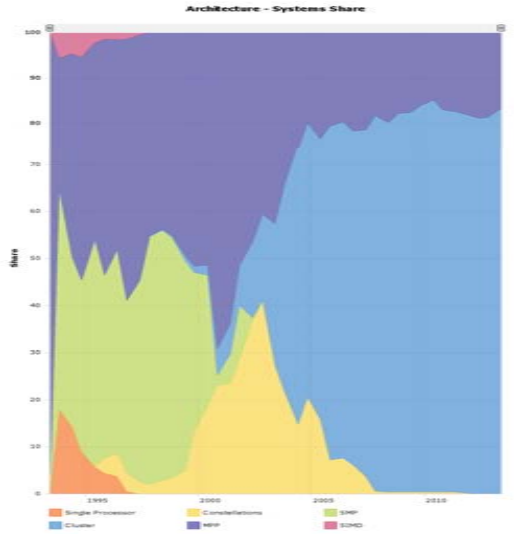


www.top500.org

Past Microprocessor Performance



Computer System



www.top500.org

Parallel Computers

- **A parallel computer is a collection of CPUs that cooperate to solve a problem.**
- **It can solve a problem faster or just solve a bigger problem.**
  - How large is the collection?
  - How is the memory organized?
  - How do they communicate and transfer information?

Categorization of Parallel Architectures

- **Control mechanism:**
  - Instruction (program) stream and data stream
- **Process granularity**
  - Decomposition (low, medium, fine-grain)
- **Address space organization**
- **Interconnection network**
  - Static (routing path of messages between processors is fixed)
  - Dynamic (routing path is flexible depending on congestion)

Control Mechanism (Flynn's Taxonomy)

- **SISD: Single Instruction stream Single Data stream**
  - Traditional single processor computers
- **SIMD: Single Instruction stream Multiple Data stream**
  - Massively parallel computers
- **MIMD: Multiple Instruction stream Multiple Data stream**
  - Most multi-processor computers where processors can work separately or together
- **MISD: Multiple Instruction stream Single Data stream**
  - Never commercially successful

SIMD

- **Multiple processing elements are under the supervision of a control unit**
  - Thinking Machine CM-2, MasPar MP-2, Quadrics, GPUs
- **SIMD extensions are now present in commercial microprocessors (MMX or Katmai in Intel x86, 3DNow in AMD K6 and Athlon, Altivec in Motorola G4)**
- **Example:**

```
DO I = 1,1000
  C(I) = A(I) + B(I)
END DO
```

  - In this example, the various iterations are independent of each other so they can be executed independently. Thus the same instruction (add) can be broadcast with the appropriate data to 1000 CPU's and performed in parallel.

MIMD

- **Each processing element is capable of executing a different program independent of the other processors**
- **Most multiprocessor can be classified in this category)**


MIMD

- **MIMD computers can be "shared-memory" or "distributed-memory"**
- **Example:**

```
DO I = 1,1000
  C(I) = A(I) + B(I)
END DO
```

  - For shared-memory computers, this loop could be internally broken up into n sub-loops, each performed on a different processor. All processors have access to the arrays A, B, and C.
  - For distributed-memory computers, this loop could be broken up by the programmer with the sub-sets of A and B sent to each processor using messages, and then receiving the sum back in another message. The sum of the sub-sets of C are then accumulated.

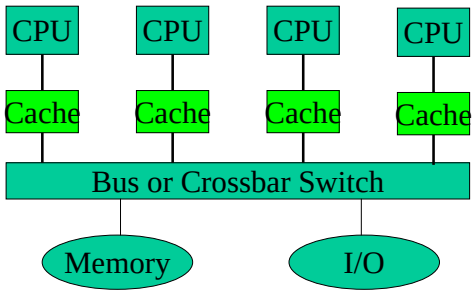
Process Granularity

- **Coarse grain: Cray C90, Fujitsu**
  - Decomposition is minimal. Small number of blocks with large amounts of data are processed in parallel.
- **Medium grain: IBM SP2/3, CM-5**
  - Decomposition is medium to large. Medium to large number of blocks with medium amounts of data are processed in parallel.
- **Fine grain: CM-2, Quadrics, GPUs**
  - Decomposition is very large (perhaps on a by-point or by-cell basis). Very large number of blocks (or units) with very small amounts of data are processed in parallel.

Types of Computer Platforms

- **Single (shared or common) address space**
  - Uniform Memory Address: SMP (UMA)
    - The time by a processor to access any word in memory (local or global) is identical.
  - Non Uniform Memory Address (NUMA)
    - The time by a processor to access certain words in memory are longer than others. Processors that have high-speed local memory in addition to cache fall into this category.
- **Distributed address space - Message passing**
  - Memory is distributed so that each processor has its own exclusive address space. Processing nodes can consist of single processors or shared-address-space multi-processors.

SMP-UMA Architecture

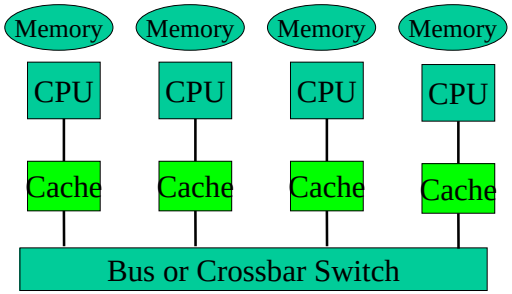


Cache => low-latency high-bandwidth storage may be included

Note that processors do not have local memory

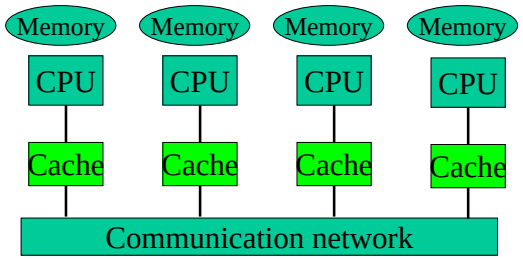
- **SMP uses shared system resources (memory, I/O) that can be accessed equally from all the processors**
- **Cache coherence is maintained.**
  - The presence of cache or local memory leads to multiple copies of a memory word being manipulated by 2 or more processors at the same time.
  - *Invalidate* and *update* protocols can be used to maintain consistency of data across all memory systems

SMP-NUMA Architecture



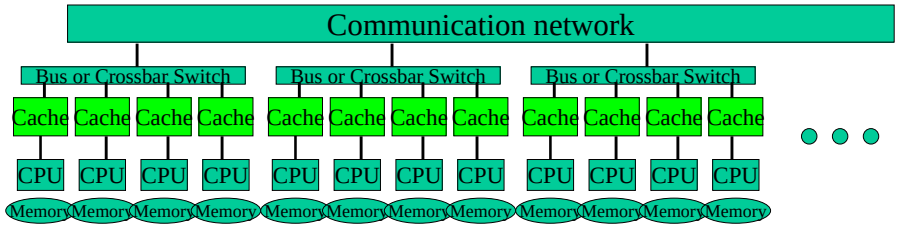
- **Shared address space with non-uniform access. Central memory in addition to local memory.**
- **Memory latency varies whether you access local or remote memory**
- **Cache coherence is maintained using a hardware or software protocol**
- **SGI Origin 2000 and Sun Ultra HPC are examples**

Message-Passing Distributed Memory (MDM)



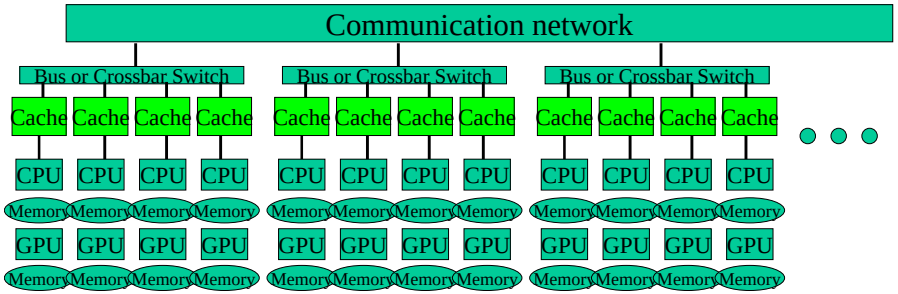
- **Local address space (distributed memory).**
- **No cache coherence. Each processor responsible for its own cache coherence.**
- **Bus or cross-bar switch of SMP is replaced with a communication network (switch or Ethernet)**
- **Beowulf cluster is an example where each "CPU" might represent multiple "cores"**

Distributed Shared Memory (DSM)



- **Shared address space (shared memory at the individual processor level).**
- **Local address space (distributed memory at the system level)**
- **Cache coherence at the individual processor level is maintained using a hardware or software protocol**
- **No cache coherence at system level. Each processor responsible for its own cache coherence.**
- **Multi-node Beowulf cluster is an example where each node contains multiple cores (CPU)**

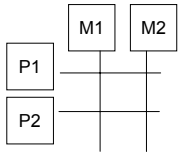
Multi-Processor Distributed Shared Memory (MPDSM)



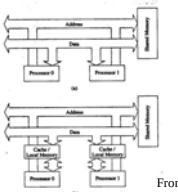
- **Shared address space (shared memory at the individual processor level).**
- **Local address space (distributed memory at the system level)**
- **Cache coherence at the individual processor level is maintained using a hardware or software protocol**
- **No cache coherence at system level. Each processor responsible for its own cache coherence.**
- **Multi-core CPU/GPU Beowulf cluster or cluster of NVIDIA Tesla computers is an example**

Dynamic Interconnections

- **Crossbar Switching :**
  - Most expensive and extensive interconnection.

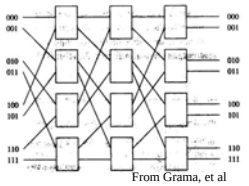


- **Bus connected :**
  - Processors are connected to memory through a common datapath

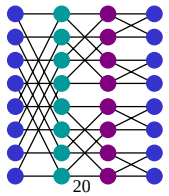


From Grama, et al

- **Multistage interconnection:**
  - Butterfly,Omega network, perfect shuffle, etc



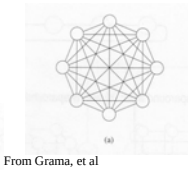
From Grama, et al



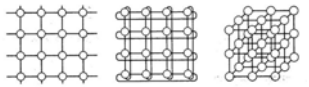
Butterfly

Static Interconnection Network

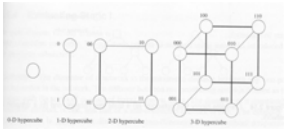
- **Complete interconnection**
- **Star interconnection**
- **Linear array**
- **Mesh: 2D/3D mesh, 2D/3D torus**
- **Tree and fat tree network**
- **Hypercube network**



From Grama, et al



From Grama, et al



From Grama, et al

Characteristic of Static Network

- **Diameter: maximum distance between any two processors in the network**

D=1	complete connection
D=N-1	linear array
D=N/2	ring
D=2(√N -1)	2D mesh
D=2 (√(N/2))	2D torus
D=log N	hypercube

N is the number of processors in network

- **Connectivity: measure of the multiplicity of paths between any two processors in the network**
  - High connectivity is desirable because it lowers contention for communication resources

Characteristic of Static Network

- **Bisection width: minimum number of communications links that have to be removed to partition the network in half.**
- **Channel rate: peak rate at which a single wire can deliver bits**
- **Channel bandwidth: product of channel rate and channel width**
- **Bisection bandwidth: product of bisection width and channel bandwidth.**

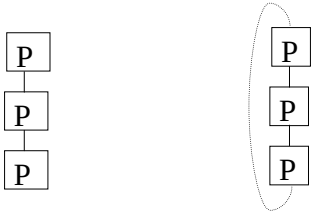
Network Characteristics

- **Some characteristics of each network:**

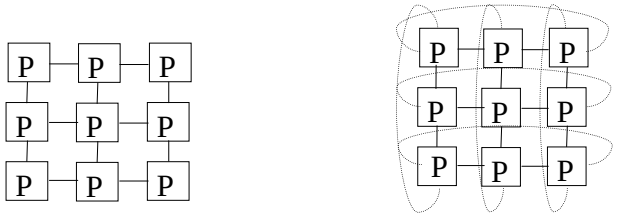
Network	Diameter	Bisection Width	Arc Connectivity	Cost (No. of links)
Completely-connected	1	$p^2/4$	$p-1$	$p(p-1)/2$
Star	2	1	1	$p-1$
Complete binary tree	$2\log((p+1)/2)$	1	1	$p-1$
Linear array	$p-1$	1	1	$p-1$
2-D mesh, no wraparound	$2(\sqrt{p}-1)$	$\sqrt{p}$	2	$2(p-\sqrt{p})$
2-D wraparound mesh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

From Grama, et al

Linear Array, Ring, Mesh, Torus

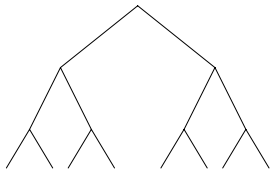


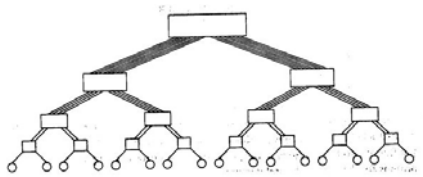
Processors are arranged as a d-dimensional grid or torus



Tree, Fat-Tree

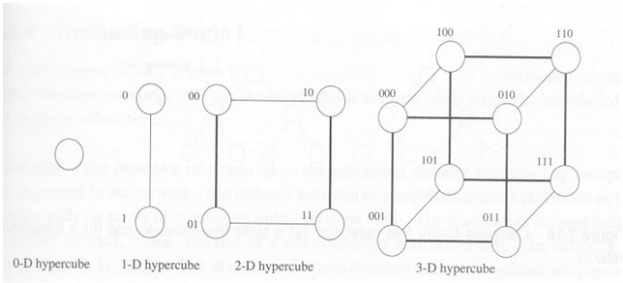
- **Tree network: there is only one path between any pair of processors.**
- **Fat tree network: increase the number of communication links close to the root**





From Grama, et al

Hypercubes



From Grama, et al