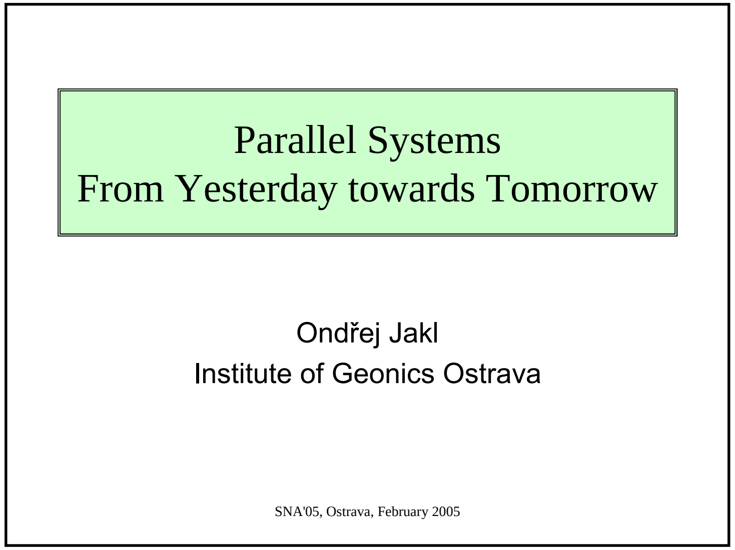
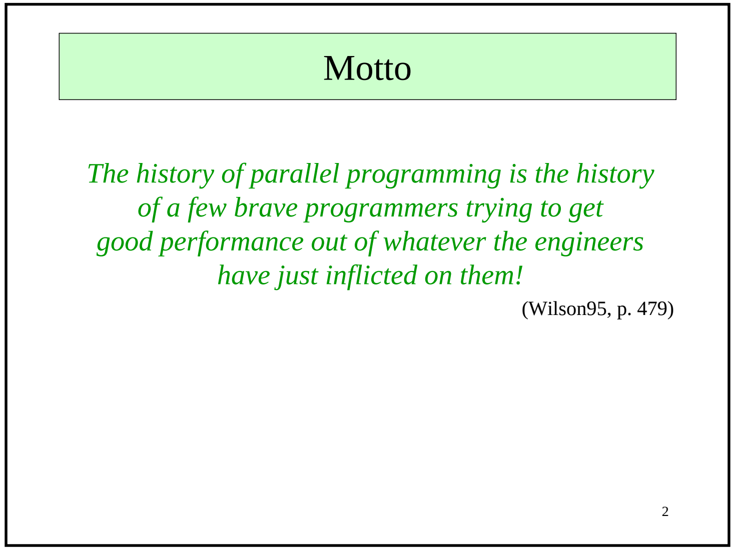


Parallel Systems
From Yesterday towards Tomorrow

Ondřej Jakl
Institute of Geonics Ostrava

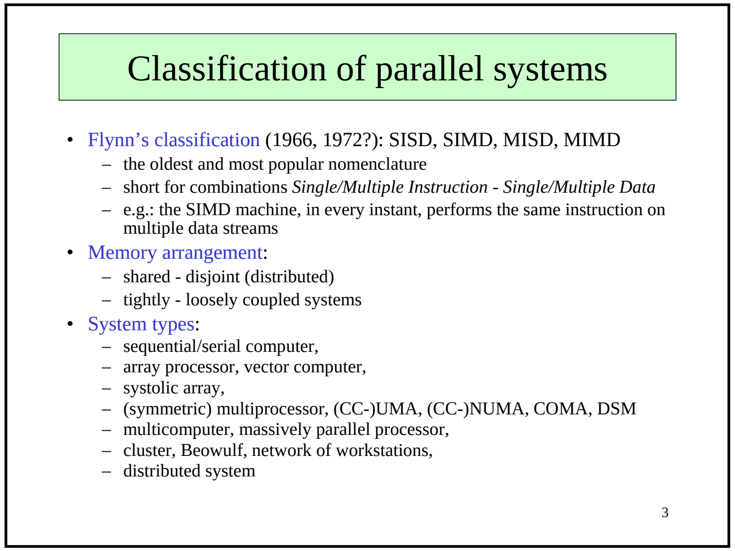
SNA'05, Ostrava, February 2005



Motto

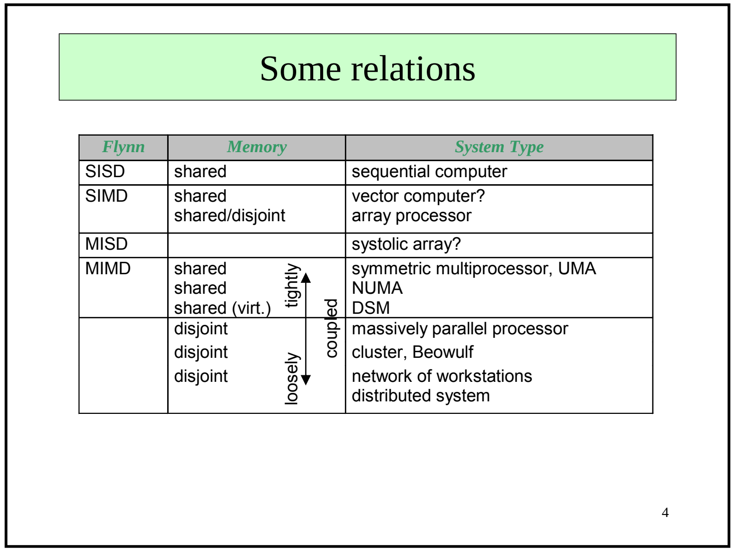
The history of parallel programming is the history of a few brave programmers trying to get good performance out of whatever the engineers have just inflicted on them!

(Wilson95, p. 479)



Classification of parallel systems

- Flynn's classification (1966, 1972?): SISD, SIMD, MISD, MIMD
  - the oldest and most popular nomenclature
  - short for combinations *Single/Multiple Instruction - Single/Multiple Data*
  - e.g.: the SIMD machine, in every instant, performs the same instruction on multiple data streams
- Memory arrangement:
  - shared - disjoint (distributed)
  - tightly - loosely coupled systems
- System types:
  - sequential/serial computer,
  - array processor, vector computer,
  - systolic array,
  - (symmetric) multiprocessor, (CC-)UMA, (CC-)NUMA, COMA, DSM
  - multicomputer, massively parallel processor,
  - cluster, Beowulf, network of workstations,
  - distributed system



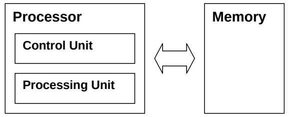
Some relations

Flynn	*Memory*	*System Type*
SISD	shared	sequential computer
SIMD	shared shared/disjoint	vector computer? array processor
MISD		systolic array?
MIMD	shared shared shared (virt.)	symmetric multiprocessor, UMA NUMA DSM
	disjoint disjoint disjoint	massively parallel processor cluster, Beowulf network of workstations distributed system

tightly ↕ loosely
coupled

SISD = sequential computer

- Sequential (serial) machine: one instruction processes one piece of data at a time
- One processor with a control unit and a processing (arithmetic, logic) unit interfaced with memory – von Neumann's model
- Pipelining, superscalar processing (several processing units) possible
  - as long as everything can be regarded as a single processor the architecture remains SISD
- Parallel architectures are generalizations of this model



Processor

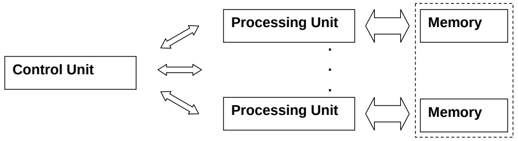
Control Unit

Processing Unit

Memory

SIMD = array processor

- Computer which has many identical interconnected processing units (PU) (processors) under the supervision of a single control unit (CU)
- CU transmits the same instruction to all PU
  - PU operate synchronously
- Each PU works on data from its own memory (on distinct data streams)
  - some systems also provide a shared global memory for communications
  - PU must be allowed to complete its instruction before the next instruction is taken for execution



Control Unit

Processing Unit

Memory

Processing Unit

Memory

1964

ILLIAC

1974

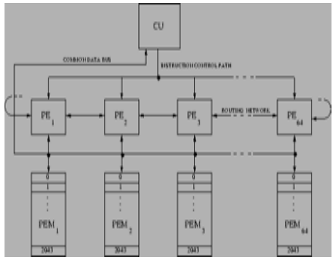
1984

1994

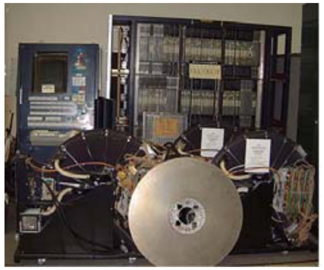
2004

History lesson I: ILLIAC-IV

- 1964 (1967?) ARPA signed contract with University of Illinois
  - project leader Daniel Slotnick
- 1976 (1972?) its first successful application
- Not commercially available – just one exemplar built (31 mil. USD)
- 1982 decommissioned
- One central unit (CU) controlled 64 processing elements (PE)
  - a quarter of the planned machine
  - small 1 MB memory (64x16 KB)
- Actual performance 15 MFLOPS
  - planned 1 GFLOPS
- Global bus & 2D mesh between PE
- Super fast disk system (500 Mbit/s)
  - compensated for the small memory
  - made I-IV fastest until the mid 1980's at problems with large data processed
- Software: low-level Fortran-like CDF

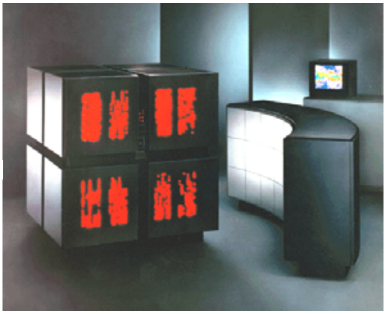


ILLIAC-IV – more pictures



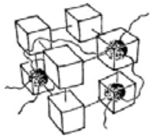
History lesson II: Connection Machine

1964
ILLIAC
1974
1984
CM-1
1994
2004



- 1981: first description of the CM architecture by Danny Hillis (MIT)
- 1983: Thinking Machines Corp.
  - DARPA support
  - cornered the market “on sex appeal in high-performance computing”
- 1986: CM-1 introduced
- Richard Feynman played a critical role in developing CM
- about 80 CM installations
- 1996 TMC abandoned hardware development

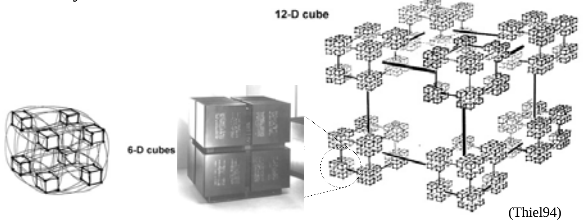
CM-1



- The CM-1 Connection Machine was designed to provide massive parallelism for the solution of AI problems
  - simulation of intelligence and life
    - ILLIAC-IV designed primarily for the solution of highly numeric problems
  - importance not so much on the processors themselves, but rather on the nature and mutability of the connections between them
- 65535 very simple 1-bit processors
  - private 4k memory
  - single bit ADD, AND, OR, MOVE, and SWAP operations to and from memory or one of the single bit flag registers available
- Every processor was connected to a central/control unit called the “microcontroller” which issues identical “nanoinstructions” to all of them to do the work
- Data Vault disk array

CM-1’s interconnect

- Processors connected to form a 256x256 mesh
- For faster routing between distant processors, clumps of 16 processors were also interconnected by a packet switched network configured as a 12-dimensional hypercube
- Each processor within the clump is linked to two others in a linear array
- Extremely fast and flexible



(Thiel94)

CM-2, etc.

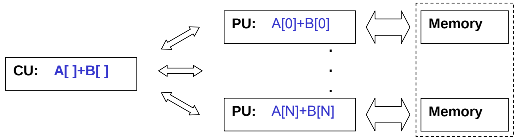
- Great fixed-point speed – 32 bit integer addition has a peak rate close to 2000 MOPS.
- CM Fortran, LISP* and C* with parallel constructs
- CM-1 was not very efficient in floating-point calculations
  - important for the commercial success

- CM-2 (1987), CM-200
  - faster versions of the same computer architecture
  - memory increased to 64K or 256K per processor
  - one special floating-point accelerator for each 32 1-bit processors added
    - corresponding to the 32-bit width of one floating-point variable
  - 1989: Gordon Bell Prize for absolute performance 6 GFLOPS
- CM-5 (1993) custom-built processors abandoned
  - standard microprocessors (SPARC MIPS RISC)
  - SIMD principle abandoned
  - appeared in the movie “Jurassic Park”

Vector computer ∈ SIMD?

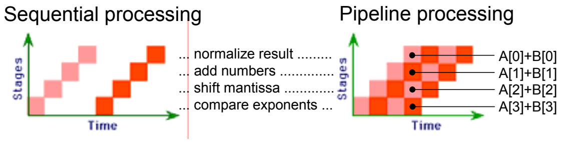
Naive conception of a vector computer

Adding two real arrays A, B



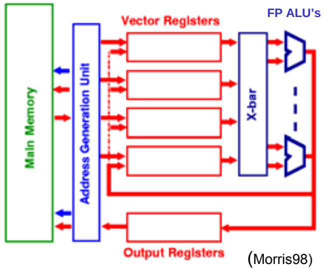
Pipeline processing

- A single data stream feeds the processor
- Processor itself provides a multi-stage processing
  - at each stage the data is operated upon by a different part of the computation required for one complex machine code instruction
  - the total time taken to process the data, although constant for one data item, is reduced for a number of data items
- Example: Adding two real arrays A[ ], B[ ]


Sequential processing
Pipeline processing
Stages
Time
... normalize result ....
... add numbers ....
... shift mantissa ....
... compare exponents ...
Stages
Time
A[0]+B[0]
A[1]+B[1]
A[2]+B[2]
A[3]+B[3]

Vector computer

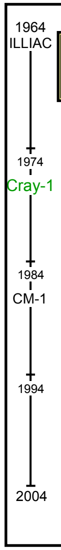
- Built to handle large scientific and engineering calculations
- Heavily pipelined architecture for efficient operations on vectors and matrices
- Vector registers: FIFO queues capable of holding ~100 FP values
- Special instructions for operations on vectors of numbers, e.g.
  - load a vector register from memory
  - perform operation on elements in the vector registers
  - store data in the vector registers back into memory
- Vectorization transparent thanks to parallelizing (Fortran) compilers



(Morris98)

Vector computer ∈ SIMD?

- *Vector computers are parallel, in the sense that they execute many instructions at the same time, but each instruction on any piece of data is performed in sequence in respect of the piece of data concerned.* (Barry 1996)
- *A problem arises when Flynn's taxonomy is applied to vector supercomputers like the Cray-1. In Hockney and Jesshope's book [4] the Cray-1 is categorised as SIMD machine because it has vector units. However, Hwang and Briggs in their book[5] categorise the Cray-1 as a SISD because there are no multiple processing elements. Which of these classifications is correct comes down to whether the vector units are regarded as processing a single or multiple data stream. This is open to interpretation, and leads to problems.* (Wasel 1994)
- (Mazke 2004) considers pipelined vector processor as MISD.
- *The vector processors fit the term array processor in its general sense. They are, however, such an important sub-category that they retain their own identity and are referred to as vector computers rather than being lumped in with array processors.* (Wasel 1994)



History lesson III: Cray-1

- The first Cray computer (1976)
  - the first supercomputer
  - the “world's most expensive loveseat”
- by Seymour Cray:

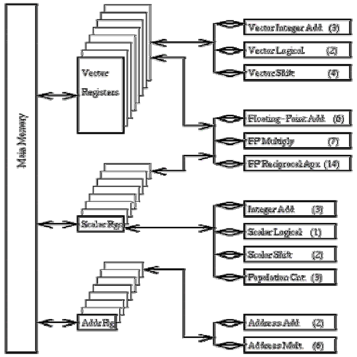
”In all of the machines that I've designed, cost has been very much a secondary consideration. Figure out how to build it as fast as possible, completely disregarding the cost of construction.”

- Besides being a vector computer, it was the fastest scalar machine of its period
- 133 (160?) MFLOPS peak
- A hand-crafted machine – took months to build
- At least 16 systems produced

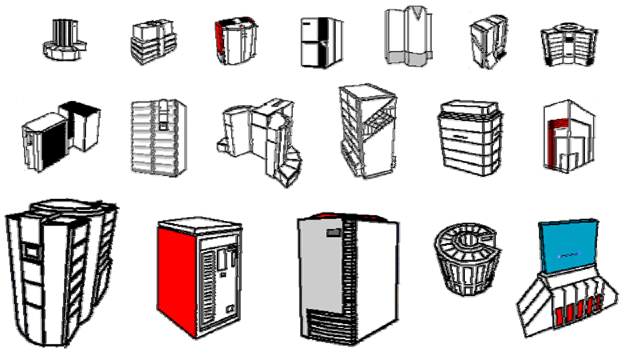
Cray-1

- Vector registers: FIFO queues capable of holding 64 single precision (64-bit) elements
- Vector pipelines filled from the vector elements in the vector registers
  - reduces the time to fill the pipelines for vector arithmetic operations
  - vector registers can even be filled while the pipelines are performing some other operation
  - 12 different pipelines (functional units)
    - for integer or logical operations on vectors
    - for floating-point operations using scalars or vectors
    - for integer or logical operations on scalars
    - for address calculations
- The first machine to use chaining
  - vector results may be put back into a vector register or they may be piped directly into another pipeline for an additional vector operation
- Limited by one memory read and write operation per clock cycle

Cray-1 inside

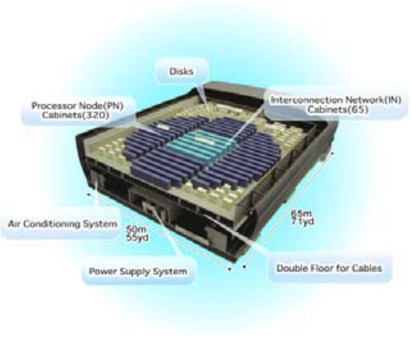


Crays from outside

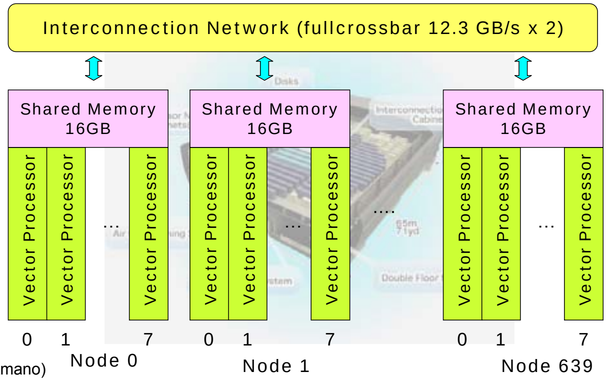


Today: Earth Simulator

1964
ILLIAC
1974
Cray-1
1984
CM-1
1994
ES
2004



ES as a vector processor


Interconnection Network (fullcrossbar 12.3 GB/s x 2)

Shared Memory 16GB
Vector Processor
Vector Processor
…
Vector Processor
0 1 7
Node 0

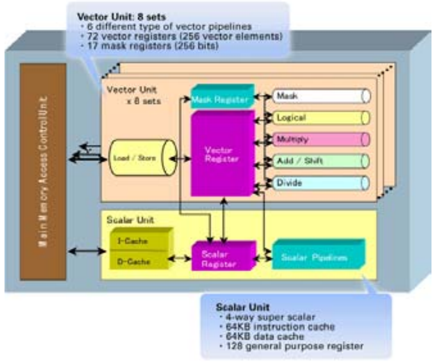
Shared Memory 16GB
Vector Processor
Vector Processor
…
Vector Processor
0 1 7
Node 1

….

Shared Memory 16GB
Vector Processor
Vector Processor
…
Vector Processor
0 1 7
Node 639

(Amano)

ES – arithmetic processor


Vector Unit: 8 sets
- 6 different type of vector pipelines
- 72 vector registers (256 vector elements)
- 17 mask registers (256 bits)

Scalar Unit
- 4-way super scalar
- 64KB instruction cache
- 64KB data cache
- 128 general purpose register

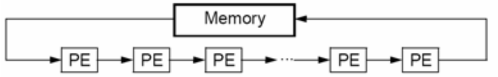
Decline of the SIMD/array processors

No longer considered perspective for general-purpose parallel computers

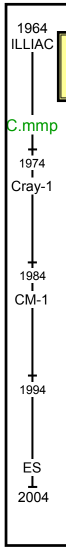
- Most problems do not map into the strict data-parallel solution
- Inefficient parallel execution of (nested) IF-THEN-ELSE or CASE statements
- Most naturally single-user systems only
- Entry-level systems too expensive
  - difficult to scale down the price/performance ratio of the necessary high-bandwidth interconnects
- Built using custom processors, which are not competitive (price, performance) with commodity CPUs
- Original motivation – relatively high cost of control units – is no longer valid

MISD?

- A collection of processing elements, all of which execute independent streams of instructions on the same data stream.
- There are two ways in which this can be done:
  – the same data item can be fed to many processing elements each executing their own stream of instructions
  – the first processing element could pass on its results to the second processing element and so on, thus forming a macro-pipeline
- Without literal architectural implementation
- Some authors identify systolic arrays as a possible example of this form
  – a mesh-like network of processors that rhythmically compute and pass data through the system; basic configuration

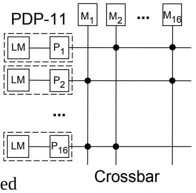


  – realization: iWarp machines (1990, Carnegie Mellon and Intel)



History lesson IV: C.mmp

- C.mmp (Carnegie multi-mini-processor)
- 1970 - 1977, William Wulf et. al, Carnegie Mellon University
- Experimental setup for research on parallel computer architectures
  – built out of off-the-shelf components
- 16 DEC PDP-11's connected together through a 16 x 16 crossbar switch to 16 memory modules
  – allowing 16 memory references to take place at once (on different ports)
- Could be extensively reconfigured
  – MIMD mode: normal mode of operation
  – SIMD mode: all the processors are coordinated by a single master controller
  – MISD mode: the processors are arranged in a chain with a single stream of data passing through all of them
- Novel HYDRA operating system



MIMD

- Multiple Instruction - Multiple Data
- Several independent processors capable of executing separate programs (asynchronously)
- Avoids most problems of SIMD, e.g.
  – can use commodity processors
  – naturally supports multiple users
  – is efficient in conditionally executed parallel code
- Almost all current interest in parallel computers centres about the MIMD concept
- Flynn's taxonomy too coarse – akin to dividing all contemporary computers into just the two categories, parallel and sequential
- Coarse subdivision based on memory organization:
  – shared / centralized ⇒ MIMD-SM, multiprocessor, tightly coupled
  – disjoint / distributed ⇒ MIMD-DM, multicomputer, loosely coupled

MIMD-SM – multiprocessors

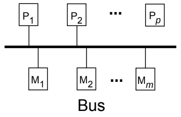
- Multiple-processor computer with shared memory
- Single address space
  – the same address on different CPUs refers to the same memory location
  – data sharing possible
- Single copy of the operating system

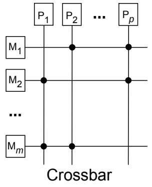
- Shared memory hardware is becoming commonplace
  – typically: workstation 1 - 4 processors, server 4 - 64 processors
- Supported by all modern operating systems
  – including Linux and Windows
- Excellent at providing high throughput for a multiprocessing load
  – within limits, scales almost linearly with number of processors

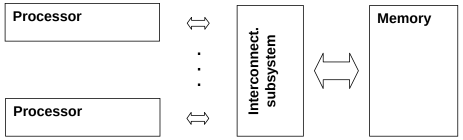
- Basic division:
  – symmetric /asymmetric processors
  – uniform / non-uniform memory access

Symmetric multiprocessor (SMP)

- Identical (equivalent) processors that share memory
  - share the workload equally
  - access the memory equally at equal speeds
  - operate independently
  - communicate via write/reads to memory
- Interconnecting subsystem:
  - bus: less expensive, but restricted throughput
  - crossbar: better performance at higher cost

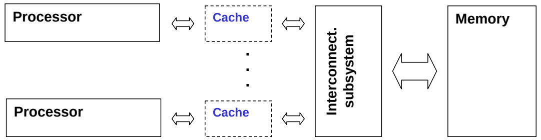




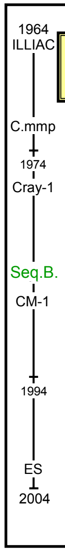


SMP = (CC-)UMA

- UMA – Uniform Memory Access
- In practice, there is a bandwidth bottleneck of the interconnect
  - scalability limited to hundreds of processors
    - tens in case of bus-based systems
- Processors may have some local memory (cache)
  - difficult to maintain cache consistency



- CC-UMA – Cache Coherent UMA
  - if one processor updates a location in shared memory, all the other processors learn about the update
  - accomplished at the hardware level, expensive



History lesson V: Sequent

- 1984: Sequent Balance 8000 SMP
  - the first commercially successful parallel machine
  - up to 20 National Semiconductor NS32016 processors
    - each with a small cache connected to a common memory
  - modified version of BSD Unix they called DYNIX
  - each of their inexpensive processors dedicated to a particular process
  - a series of libraries that could be used to develop applications using more than one processor at a time
  - designed to compete with the DEC VAX 11/780
  - sold well to banks, the government, other commercial enterprises, and universities interested in parallel computing
- 1987: Sequent Symmetry:
  - Intel 80386-based, 2 - 30 processors
- Another pioneers in MIMD: Pyramid, Encore, Alliant, AT&T

Today: IBM xSeries 455 [Natan]

Processors	1-4-way Itanium 2 [4]
Memory	up to 56 GB [8]
L4 cache	64 MB

64 bit hardware (IA-64)
EPIC - Explicitly Parallel Instruction Computing

Intel Itanium 2 processor “Madison” (2004)

Processor speed	1.3, 1.4, 1.5 GHz [1.3]
L1/L2/L3 cache	32 kB / 256 kB / 3, 4, 6 MB [3]
Pipeline stages	8
Frontside bus bandwidth	6.4 GB/s (128-bit wide @ 400 MHz)
Registers	328
Integer units	6
Branch units	3
FP units	2
SIMD units	1

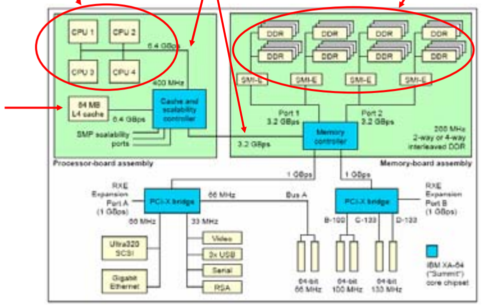
Natan as (CC-)UMA

Processors
4 x Itanium 2

Bus 6.4 / 3.2 GB/s

Memory banks

Cache
L4 64 MB



NUMA & CC-NUMA

- NUMA – Non-Uniform Memory Access
- Aims at surpassing the scalability limits of the UMA architecture due to memory bandwidth bottleneck
- Memory physically shared, but access to different portions of the memory may require significantly different times
  - direct access via global address space
- Many different ways to realize
  - often by physically linking two or more SMPs (= nodes)
    - local memory access is the fastest, access across link is slower
    - hardware includes support circuitry to deal with remote accesses
- Cache coherency with NUMA (CC-NUMA) is de facto standard
  - directory-based protocols for cache coherency (no snooping possible)

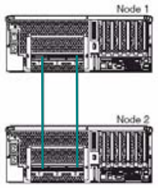
Natan as (CC-)NUMA

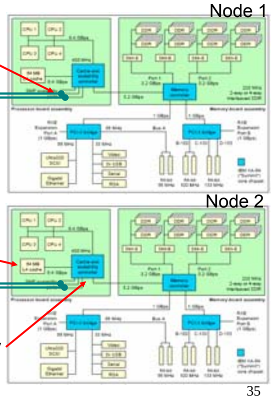
xSeries 455
multinode configuration:

2/4 nodes – 8/16 processors

single copy of the OS
(Linux, Win Server)

Natan: 2-node NUMA





SMP scalability ports

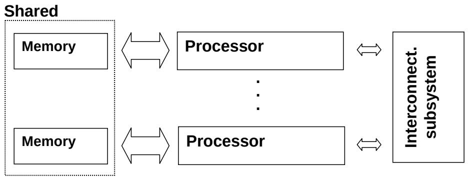
Scalability cables
2 x 3.2 GB/s

Cache

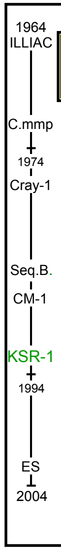
Cache & scalability controller

Distributed shared memory (DSM)

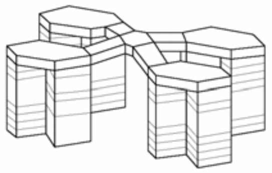
- Memory physically distributed among the processors, but the system gives the illusion that it is shared
  - concept of virtual shared memory
- Structure close to MIMD-DM systems
  - message passing hidden in the remote memory access
- Adds more hw scalability for the shared variable programming model



History lesson VI: KSR



- Kendall Square Research Corp.
  - start-up since 1986
  - Henry Burkhardt, Steve Frank
- 1992 KSR-1 (next slide)
  - rank 168 in TOP500 (256 CPUs)
- 1994? KSR-2
  - rank 87 in TOP500
- 1995 KSR stopped production

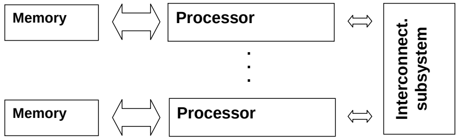


KSR-1

- Proprietary 64-bit processors
  - 40 MFLOPS per node
  - 32 MB local memory (called a local cache)
  - up to 1088 processors in a two-level unidirectional communication ring (34 x 32 CPUs)
    - 43.8 GFLOPS peak
    - KSR-2: 5000 processors, 80 MFLOPS per node
- ALLCACHE engine
  - unique implementation of virtual shared memory
  - data not found in the local cache are routed automatically from the node that has it
  - classified also as COMA (Cache Only Memory Architecture)
    - each address becomes a name without direct physical relevance
  - cache coherency automatically maintained
    - ideas developed at Swedish Institute of Computer Science (SICS)

MIMD-DM – multicomputers

- Multiple-processor computer with distributed memory
- Disjoint local address spaces
  - the same address on different CPUs refers to different memory locations
  - no cache coherence problems
  - message passing necessary for the processors to interact
- Nodes can be autonomous computers
  - with a separate copy of the operating system
- Easier to scale than MIMD-SM
  - in practice to thousands nodes with specialized interconnecting hardware



MIMD-DM types

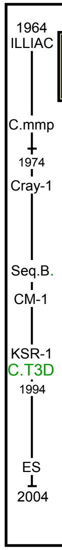
- Basic types:
  - massively parallel processor
  - cluster
  - network of workstations

Massively parallel processor (MPP)

- The most tightly coupled MIMD-DM
- “Flagships” of the leading computer vendors
  - exterior design, publicity, support, etc.
  - price corresponds to the uniqueness of MPP
- Up to thousands processor nodes
  - commodity microprocessors killed off custom CPUs
- Custom switching networks to provide low-latency, high-bandwidth access between processor nodes
  - good balance between speed of the processors and speed of the interconnection subsystem
- Ideal environment for parallel processing
  - homogenous collection of powerful processor nodes
  - very fast interprocess communication
  - shielded from external impacts

ASCI White / IBM SP

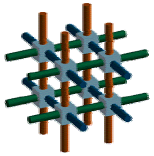
History lesson VII: Cray T3D/E



- By CRI (Cray Research Institute)
- Cray T3D – Cray’s first MPP (1993)
  - captured MPP market leadership from early MPP companies such as Thinking Machines and MasPar
  - exceptionally robust, reliable, sharable and easy-to-administer
- Cray T3E – its successor (1995)
  - the world's best selling MPP system
  - Cray T3E-1200: the first supercomputer to sustain one TFLOPS on a real-world application
- Cray XT3 – third generation (2004)
  - AMD Opteron processors

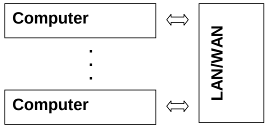
Cray T3D

- 2 DEC 21044 (21064?) Alpha commodity CPUs per node
  - 150 MFLOPS peak
  - 64 MB local memory per node
  - systems up to 2048 CPUs (never built)
- Interconnect: 3D torus (hence T3D’s name)
  - each computing node interconnects in 3 bi-directional dimensions with its nearest neighbours
  - 300 MB/s, very low latency
- Although the memory of the T3D is physically distributed, it is one globally addressable address space
  - considered as NUMA by some authors (Amano)
- No I/O capability - attached to and hosted by a YMP or C90 front-end



Network of workstations (NOW)

- Sometimes referred to as distributed system
- Set of computers connected by a (local area) network
  - very loosely coupled system
- Often heterogeneous nodes
  - different hardware, operating system, etc.
- Uses LANs/WANs for communication
- Many features similar to massively parallel processor
- Issues: reliability, security, etc.



Cluster (COW)

- Specialized, tightly-coupled NOW:
- Possible roles:
  - high availability
  - load balancing
  - high performance
- Commodity clusters:

assembled from commodity, off the self (COTS) components

- High-performance interconnect(s) (Fast/Giga Ethernet, Myrinet, etc.)
- Interactive access restricted/excluded
- Identical/homogeneous nodes
- Nodes intended to cooperate
- Nodes without peripheral units (e.g. displays)
- OS tuned to optimize throughput
- ...

45

1964
ILLIAC
C.mmp
1974
Cray-1
Seq.B.
CM-1
KSR-1
C.T3D
Beow.
ES
2004

History lesson VIII: Beowulf

- PC based cluster system
  - designed as a cost-effective alternative to large supercomputers
  - Donald Becker and Thomas Sterling, CESDIS*), 1994
- 16 personal computers (Intel 486DX4 processors)
- “Channel bonded” Ethernet 10 Mbit/s (drivers by Backer)
  - network traffic striped across two or more Ethernets
  - processors were too fast for a single Ethernet
- Instant success: Recognized as a new genre within the HPC community
  - prevalence of computers for home & office, new cost-effective components
  - availability of fully assembled subsystems (processors, motherboards, disks, NIC’s)
  - mass market competition: prices down, reliability up
  - open source software (Linux OS, GNU compilers, MPI, PVM)
  - obtaining high performance, even from vendor provided parallel platforms, is hard work and requires researchers to adopt a do-it-yourself attitude
  - increased reliance on computational science which demands HPC

*) Center of Excellence in Space Data and Information Sciences, a NASA contractor

46

Beowulf class cluster computers

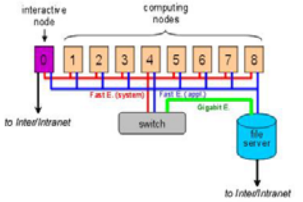
- Dedicated nodes and networks, serve no other purpose
  - usually identical computing nodes
  - usually one special front-end node
- Commodity computers, relatively inexpensive, as nodes
- Networks also commodity entities
  - at least they must interconnect through a standard bus (e.g. PCI)
  - to differentiate from MPP where the network and CPUs are custom-integrated at very high cost
- The nodes all run open source software
  - usually Linux as OS
- The resulting cluster is used for HPC – computational cluster
  - usually just one computation at a time
  - there are also high-availability, load-balancing clusters, ...

47

The THEA PC cluster

Built: 2001-2002, Institute of Geonics Ostrava
Price: 570 000 Kč, i.e. ~18 000 EUR

LINPACK highly parallel computing benchmark:
6.2 GFLOPS (8 nodes)



Computing nodes:
- Athlon 1.4 GHz
- 768 MB SDRAM
- 20 GB, 7200rev./min.
- 2x Fast Ethernet

Interactive node:
- 3x Fast Ethernet
- otherwise identical with the computing

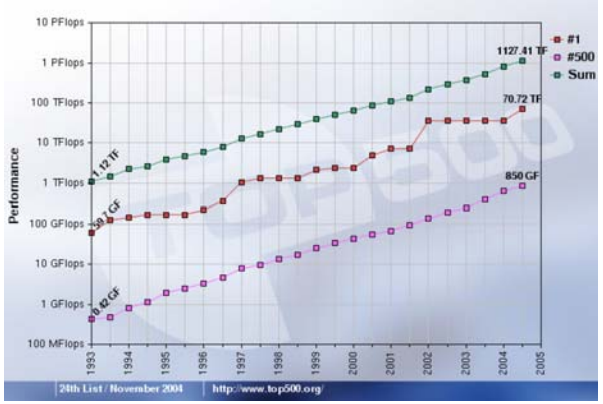
File server:
- 2x AMD Athlon MP 1900+
- 1 GB DDR RAM
- 3x 36 GB, 15 000 rev./min.
- RAID controller

Switch:
- Cisco Catalyst 2950T
- 24 ports Fast Ethernet 100 Mbit/s
- 2 uplinks Gigabit Ethernet 1000 Mbit/s

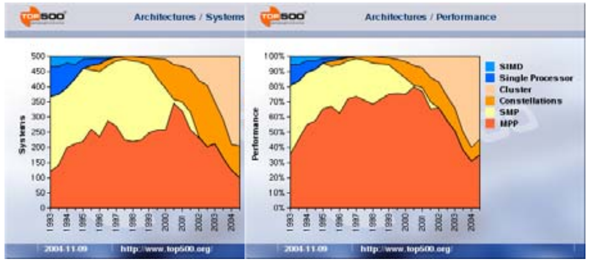
Software:
- Linux (Debian), XFS
- Portland Group C/C++ and Fortran compilers
- PVM, MPICH, LAM/MPI
- PETSc, BLAS, ATLAS, ...
- Cfengine

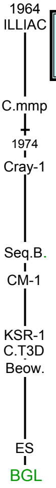
48

TOP500 performance development



TOP500 architectures development



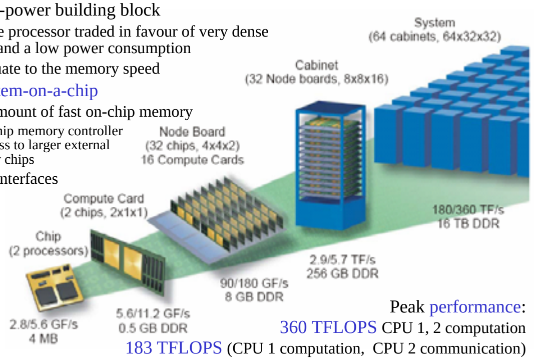


Future lesson: BlueGene/L

- A next-generation very-massively-parallel computing system by IBM
- 70.72 TFLOPS – No. 1 in TOP500 11/2004
- Designed for research and development in computational science
  - IBM's BlueGene initiative (1999) to build a petaflop scale machine
- Extremely high compute-density, attractive cost
- Relatively modest power and cooling requirements
  - 1/100 the physical size of the ES
  - 1/28 the power per computation

BlueGene/L processors

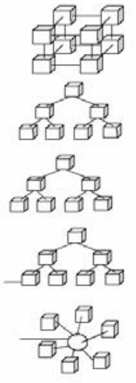
- Dual PowerPC 440/700 MHz (modified PPC400)
- Compact, low-power building block
  - speed of the processor traded in favour of very dense packaging and a low power consumption
  - more adequate to the memory speed
- Complete system-on-a-chip
  - a modest amount of fast on-chip memory
    - an on-chip memory controller for access to larger external memory chips
  - 5 network interfaces



Peak performance:
360 TFLOPS CPU 1, 2 computation
183 TFLOPS (CPU 1 computation, CPU 2 communication)

BlueGene/L interconnects

- Five interconnecting subsystems
  - complementary high-speed low-latency networks
  - two of interest for inter-processor communication
- 3D torus network
  - simple 3-dimensional nearest neighbour interconnect
  - for most general point-to-point communication patterns
  - hardware bandwidth 175 MB/s per link
- Tree network
  - for fast global operations (collective communication patterns like broadcasting, reduction operations, etc.)
  - hardware bandwidth 350 MB/s per link
- Other networks
  - global barrier and interrupt network
  - Gigabit Ethernet network for connection to other systems
  - Gigabit Ethernet network for machine control



Conclusions

I find digital computers of the present day to be very complicated and rather poorly defined. As a result, it is usually impractical to reason logically about their behaviour. Sometimes, the only way of finding out what they will do is by experiment. Such experiments are certainly not mathematics. Unfortunately, they are not even science, because it is impossible to generalise from their results or to publish them for the benefit of other scientists.

(Speech given by Tony Hoare at the Boston Computer Museum on the occasion of BYTE [magazine's] 10th Anniversary celebration)

Acknowledgements

- Academy of Sci. of the Czech Republic, project No. S3086102
- National Research Program of the Czech Republic, project No. 1ET400300415