Parallel Computing Survey

- Basic Definitions
- Illustrative Examples
- Computer models
- Speedup
- Engineering principles
- Multiple processors
- Parallel programming models
- Evolution of Performances
- References:
  - PP Chapter 1,2,3
  - http://www.top500.org

Parallel Computing

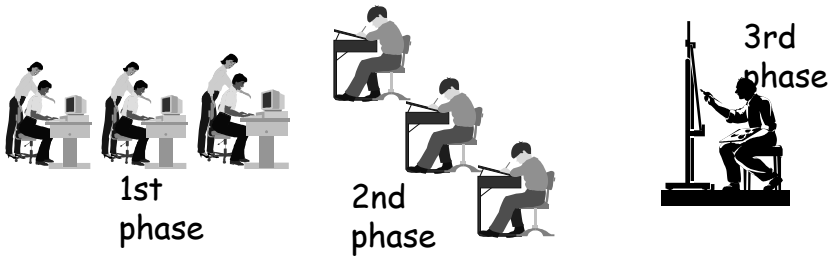
- Parallel computer + parallel program = parallel computing
- A parallel computer is a set of processors that are able to work cooperatively to solve a common computational problem.
  - parallel supercomputers with hundreds or thousands of processors,
  - networks of workstations,
  - multiple-processor workstations, and
  - embedded systems.
- A parallel program is a code being able to be executed by parallel computers.

Parallel Terminology

- There are many different expressions used in the literature for similar principles.
- Processor, Computer, Node, etc.
- Parallel execution, Concurrent execution, Distributed execution
- Multicomputer, Multiprocessor, Parallel processor, Distributed processor
- Communication channel, Link,
- We will use parallel computer when communication by messages is used and parallel processor if shared memory is used for communication.

Illustrative Examp.1-Writing Projects

- Writing projects (some communication in the first phase, no communication during work - all writers are independent, and no communication on grading).



- Finally, several projects have been done in the time needed to finish a single project plus some overhead because of initial personalised communication.

Illustrative Examp.2 - Rowing a Boat

- Synchronisation -> helmsman & oarsmen, Communication -> oarsman- oarsman, On-line corrections -> oarsmen and helmsman

- The boat is moving faster if the crew is larger but not proportionally and only for some extent. To much people could even sink the boat.

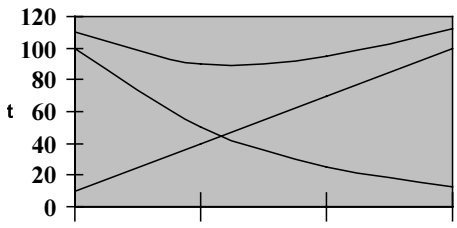
Fundamental Problems to be Solved

- Problem decomposition,
- Functional and data dependencies,
- Synchronisation, Communication,
- Partitioning,
- Visualisation of results,
- Performance measurements.

Parallel Applications

- Numerical simulations of complex systems such as weather, climate, mechanical devices, electronic circuits,
- Control of manufacturing processes, and chemical reactions,
- Commercial applications for large amounts of data,
- Video servers (real-time video).
- The need for faster computers is driven by the demands of both data-intensive applications in commerce and computation-intensive applications in science and engineering.

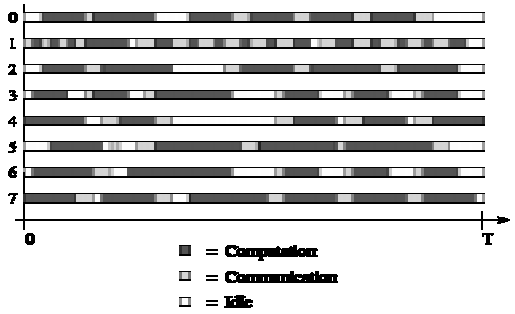
Computation + Communication



Total execution time =
Communication time
+ Idle time
+ Calculation time

- execution time =
  computation + communication + idle time
- some optimal number of processors p exist for a given problem size N,
- increase processor performances (technology),
- shorten communication and idle time (algorithm).

Computation + Communication (cont.)

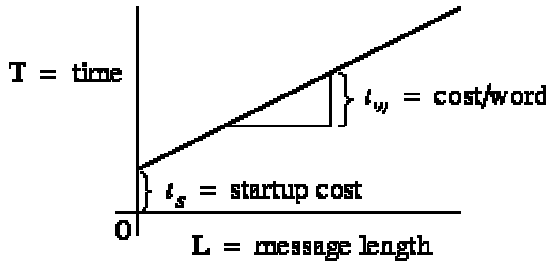


Activity plot during execution of a parallel program on eight processors. Each processor spends its time computing, communicating, or idling.

Overlapping Computation with Communication

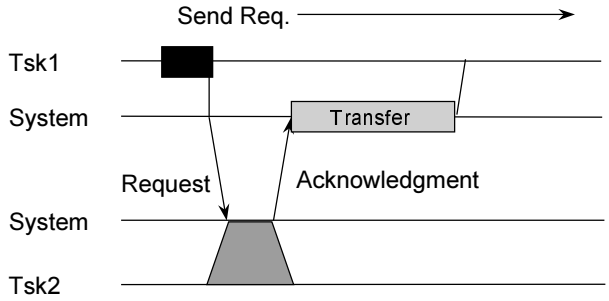
- A worker that has sent a request to the input task waits for the parameters to arrive.
- In some cases, efficiency can be improved by pre-fetching, that is, requesting the next parameter before it is needed.
- The worker can then perform computation while its request is being processed by the input task.
- A part of Computation and Communication can be executed in parallel - overlapped in time.
- In such a way idle time can be reduced.

Communication time - Model



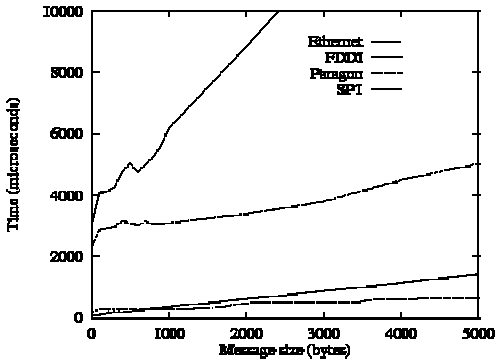
-Time T versus message length L is given, the slope of the line corresponds to the cost per word transferred tw (1/tw is bandwidth). T(0) is the message startup cost ts.
-This idealised model of communication performance is adequate for many purposes but does break down in some situations (congestion of communication channels).

Communication time (exm.)



- After a send request a fixed period is spend to service (ts). Then data start to transfer (tw).

Communication time - Reality



Round-trip time for a single message between two processors as a function of message length on Ethernet-connected workstations, FDDI-connected workstations, Intel Paragon, and IBM SP1.

Speedup and Efficiency

- Execution time of a program on a single processor is T1.
- Execution time of the same parallelized program on p (possibly equally loaded) processors is Tp.
- Tp is composed of probably shorter execution time, some communication time, and some idle time.
- Speedup S = T1/Tp > 1,
- Efficiency E = S/p < 1.
- Ideally, S=p and E=1, typically, speedup increases slower than p and efficiency decreases.
- Isoefficiency function tells us how the amount of computation must scale with P to keep E constant. Algorithm with isoef. fun. of O(P) is highly scalable.

Performance Analysis

- We estimate the total computation time for a calculation domain of dimension f(N) by analysing the code and the number of floating point operations.

$$T\_comp = t\_c * f(N)$$

- The communication time depends on the number of messages M(N,P) per calculation step and on the amount of data D(N,P) to be transferred. Clearly both parameters depend on the number of processors.

$$T\_comm = P * M(N,P) * D(N,P)$$

- If idle time is 0, because of well balanced computation the execution time of a parallelized program is:

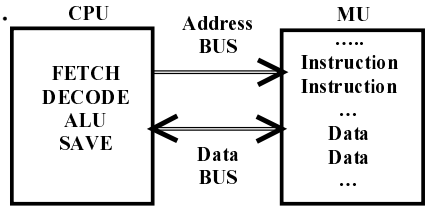
$$Tp = (T\_comp + Tcomm) / P$$

How to increase computer performances ?

- By increasing the data parallelism in a processor (parallel busses, etc.)
- By using multiple functional units and pipelining,
- By using vectorized processors able to operate on several vectorized items at the same time,
- By using multiple computing and memory units,
- By increasing the system clock.
- ???

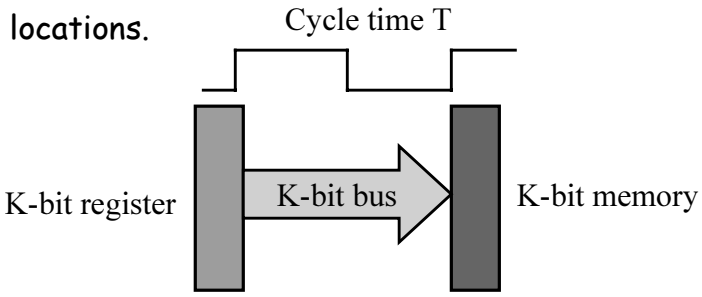
Single Computer Model (Von Neuman)

- Control unit (CU) - fetch and decode instructions,
- Processing Unit (PU) - execute instructions,
- CU + PU = CPU,
- Random Access Memory (RAM) Unit (MU) - save data and instructions,
- CPU read instructions and data from MU and write results back to the MU.



Data Parallelism

- Parallel bits ⇒ bytes ⇒ words ⇒ etc.
- Registers, data buses, address busses,
- Arithmetic units,
- Memory locations.

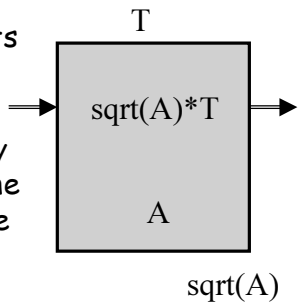


Physical Limits

- Speed of light and electromagnetic field
  $v = 3*10^8$ m/s,
- $f = 10^9$ Hz = 1 GHz $\Rightarrow$ $T = 10^{-9}$ s = 1ns $\Rightarrow$ Distance
  $d = v*T = 3*10^8$ [m/s] $*10^{-9}$ [s] = 30 cm,
- For 100 GHz we get the distance of 3mm,
- Two objects at the distance of 3mm, with an ideal optical communication channel can not communicate faster than by 100GHz synchronous clock (clock delays should be in accordance with data delays).
- We can put the clock source in the middle of both objects, but the problem might appear for other objects...

Potential Problem - AT^2 Complexity(*)

- internal concurrency in a chip, operating simultaneously on 64 bits of two numbers that are to be multiplied,
- transitive computation in which any output may depend on any input, the chip area A and the computing time T are related as $A\,T^2$=const.
- on an area of n^2A we may build a single component that needs time T/n, or, use n^2 parallel components each on area A with the time T/n^2.
- Parallel n^2-component system can be n-times faster.



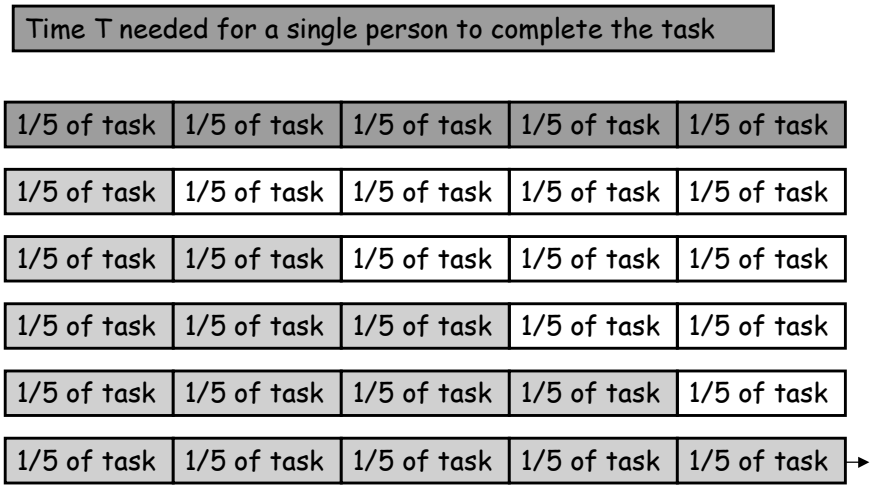
$AT^2 = (n^2A)(T/n)^2$

$n^2(A(T/n^2)^2)$

Pipelining

- Suppose that for finishing a complex repetitive task (e.g., car production) time *T* is needed for one person.
- If the task can be divided into *k* equally time-demanding subtasks, *k* workers can work in parallel, each its own subtask.
- The resources needed for the implementation of our task can be exploited totally with k workers.
- The first car will come from the pipe in time *T*, but the second car will be out already after *T+T/k*, and third after an additional delay of *T/k*.

Pipelining (cont.)



- What will happen if subtasks have diff. time complexity?

Multiple Functional Units

- A central processing unit CPU has a task to execute instructions,
- Subtasks: Fetch form memory, decode, calculate, save results,
- More arithmetic units with Adders, Dividers, etc.,
- Pipelining can be implemented in these functional units,
- Several functional units can be present to be able to execute several operations in the same time slot.

Vector Computers

- Sophisticated Von Neuman design.
- More identical functional units (VPU, IP/ALU, FPU, I/OP) on the same bus that connect memory (MU)
- Vector data manipulated in parallel; vector registers; no cache memory; mask registers to enable operation on a selected subset of elements.
- Bottleneck - the bandwidth between memory and the VPU. For example, the vector operation A=B+C, where vector operands are represented by multiple (i.e.:64) data items, can be performed in a single instruction time, with 2x64 memory accesses.
- Japan is the leading manufacturer of Vector Processors.

Multiple Computing and Memory Units

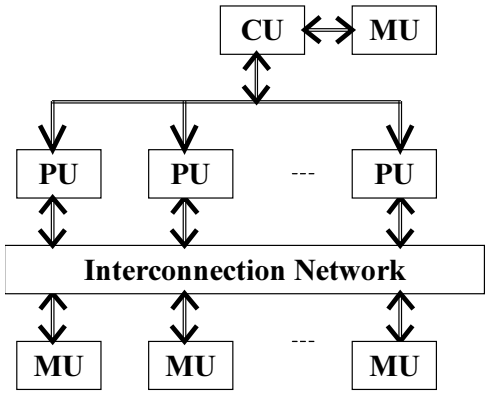
- On the highest level we may use more single Von Neuman computers (nodes), able to communicate by an interconnection network.
- Communication channels for messages
  - distributed memory computer - parallel computer,
  - Message Passing Processor (MPP),
- Common data locations in RAM, accessible by parallel buses
  - shared memory computer - multiprocessor,
  - Shared Memory Processor (SMP).

Flyn's Classification

- SISD (Single Instruction Single Data) - Von Neuman computer
- SIMD (Single Instruction Multiple Data) - Distributed and shared memory computers (i.e.: Connection Machine, vector processors).
- MISD (Multiple Instruction Single Data) - no existing prototype??
- MIMD (Multiple Instruction Multiple Data) - Distributed and shared memory computers (i.e.: Network of processors, Computer clusters, Symmetric processors, Parallel Random Access Machine (PRAM)-theoretical model)

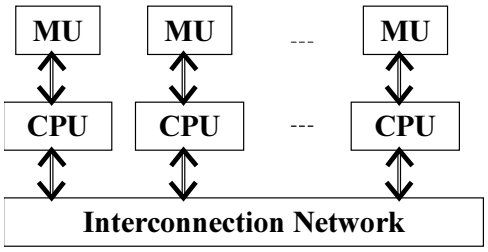
SIMD - processor-arrays

- distributed memory SIMD (single CU multiple PU and MU); no synchronisation, fine-grain parallelism, specialised
- vector computers can be seen as shared memory SIMD (single CU, multiple PU and single MU); more general

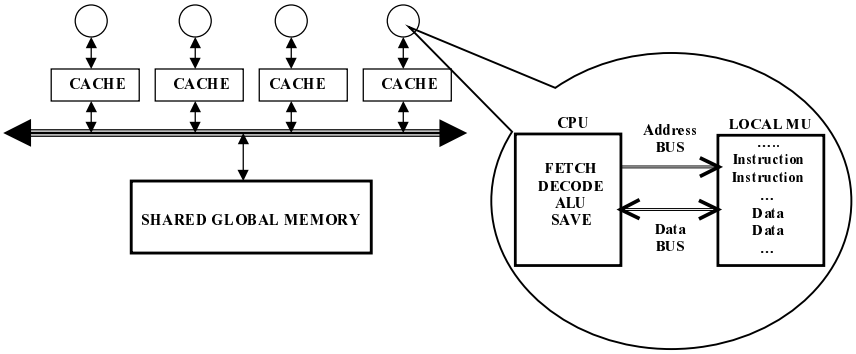


MIMD - multiple CU, PU and MU

- MIMD (multiple CU, PU and MU & interconnection network)
- coarse-grain parallelism
- fastest growing family of high-performance computers.



Shared Memory MIMD Computer- tightly coupled



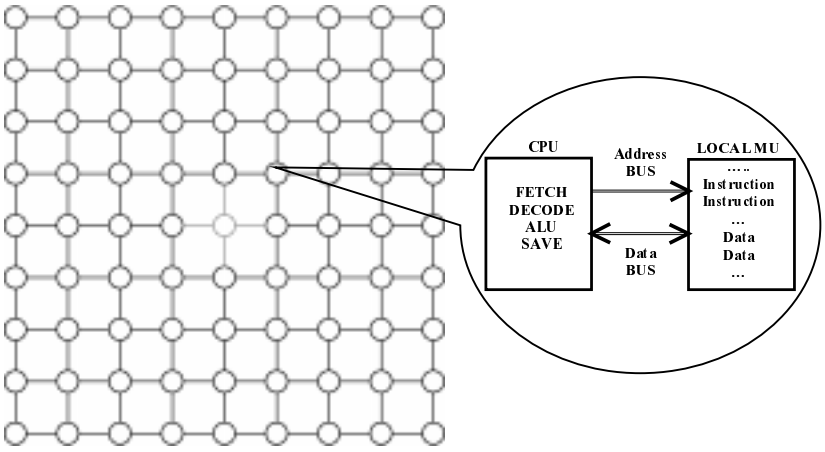
Shared Memory MIMD Computer- tightly coupled (cont.)

- can execute multiple tasks in the same time
- communication and synchronisation through the shared memory locations
- interconnection network should enable ideally a full connected system (one-to-one, one-to-all, all-to-all)
  - bus $O(N)$
  - multi-stage dynamic interconnection networks $O(N.\log N)$,
  - cross-bar $O(N^2)$
- memory should be extremely fast to enable effective global communication and to follow the processors speed.

Properties of SM-MIMD

- Shared Memory Processors (SMP)
- Advantages
  - no need for explicit communication mechanisms
  - the data distribution is completely transparent to the user
  - simpler and more compact program code
- Disadvantages
  - the connection of the CPUs to each other and to the memory (caches can alleviate the bandwidth problem)
  - speed of the memory
  - in practice the number of processors limited to 10-20

Distributed Memory MIMD Computer- loosely coupled



Distributed Memory MIMD Computer- loosely coupled (cont.)

- can execute multiple tasks in the same time
- data communication and synchronisation through messages
- interconnection network should enable routing of messages
- communication hardware (routers, queues) should work in parallel with the node processor
- local memory should be balanced with a number of directly connected processors and the speed of a node processor

Properties of DM-MIMD Message Passing Processors (MPP)

- Advantages
  - communication bandwidth scales up automatically with the number of processors
  - no need for very fast memory
  - unlimited number of processors
- Disadvantages
  - global communication is much slower (synchronisation slower too)
  - user has to distribute the data over the processors
  - data exchange between processors has to be performed explicitly
  - more complex program code (actual speed-up << theoretical speed-up)

Network of Processors

- If the network part represent the foundation of the system, that connect usually heterogeneous computers, we talk about Network of Workstations (NOW).
- Internet computing (millions of connected computers), Grid Computing,
- Cluster Of Workstations (COW).
- Constellations (clusters of SMP or MPP)
- Distributed computing (reliability, security, heterogeneity etc.),

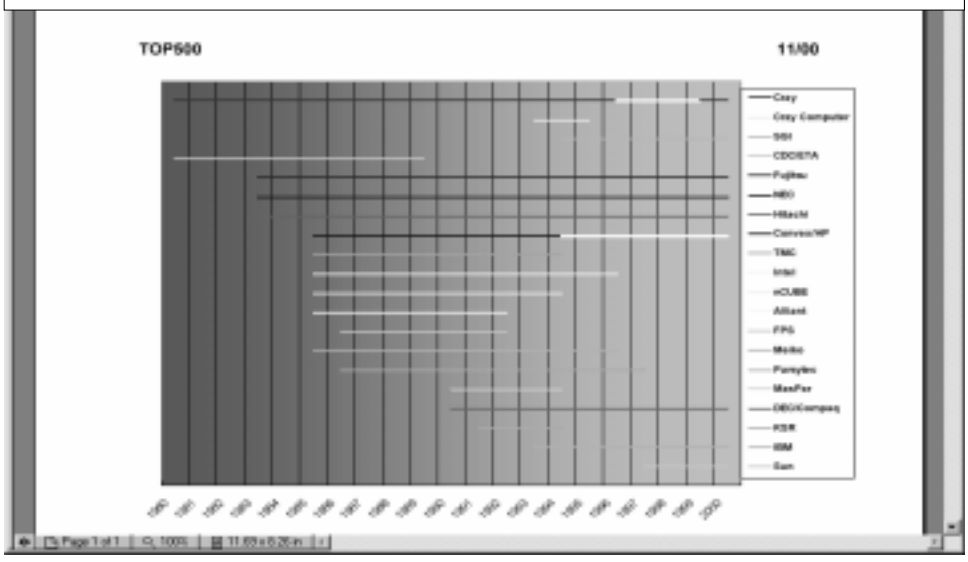
Performance Measures

- Floating Point Operation is an arithmetic operation on 64 bit wide data.
- Number of Floating Point Operations per second (R): Mflop/s = 10^6, Gflop/s = 10^9, Tflop/s = 10^{12}
- Clock of the Central Processing Unit (MHz = 10^6/s, GHz = 10^9/s),
- Memory (Cache, Bulk, Back-up) (Mbit=Mb, Mbyte=MB, Mword=Mw).
- A problem has order of magnitude f(N)=O(g(N)) if there is a positive constant c such that $|f(N)| \leq c*|g(N)|$ (e.g., for $2/3*N^2+N$ operations the order of magnitude is $O(N^2)$, because terms of lower order than N^2 become insignificant if N becomes large.

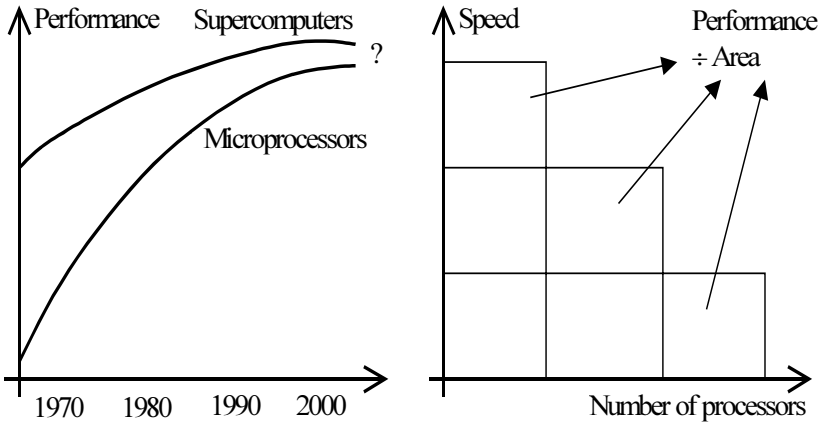
LINPAC Benchmark

- The solution problem used in the LINPACK Benchmark is to solve a dense system of linear equations (e.g.: 1000 equations).
- The algorithm used in solving the system of equations in the benchmark procedure is LU factorisation with partial pivoting with an approximate number of flops: 2/3 n^3 + O(n^2)
- By measuring the actual performance for different problem sizes n, a user can get:
  - Rmax - maximal LINPACK performance achieved
  - Nmax - problemsize for achieving Rmax
  - N1/2 - problemsize for achieving Rmax/2
  - Rpeak - Theoretical peak performance

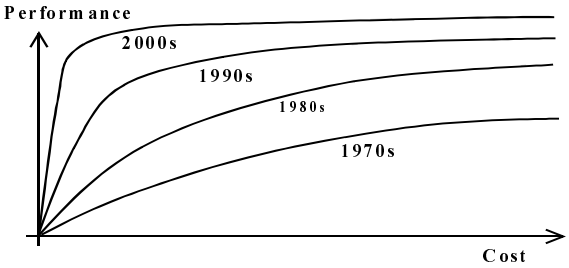
Manufacturers - 1980/00



Improving Parallel Performance

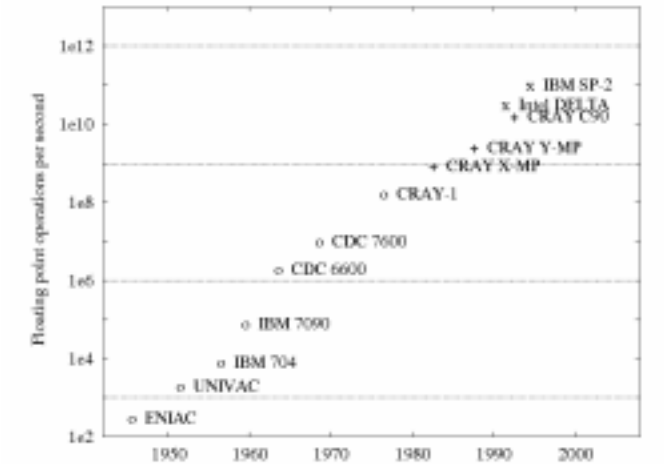


Cost Versus Performance

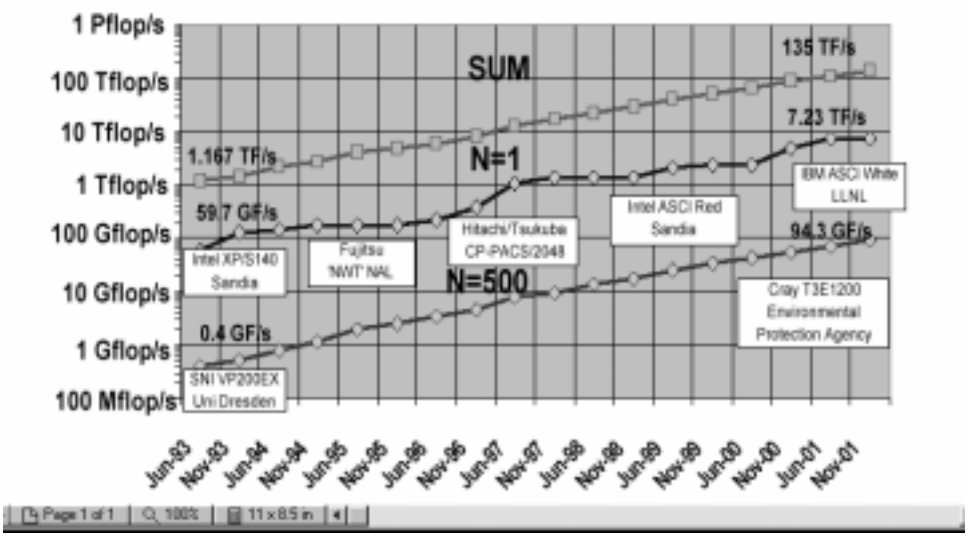


- At the lower end performance increases linearly with cost (ordinary market).
- Beyond a certain point, curves start to saturate (technological limits).
- The transition point become sharper with time (advances in technology & parallel computers).

Peak Performance 1950-2000



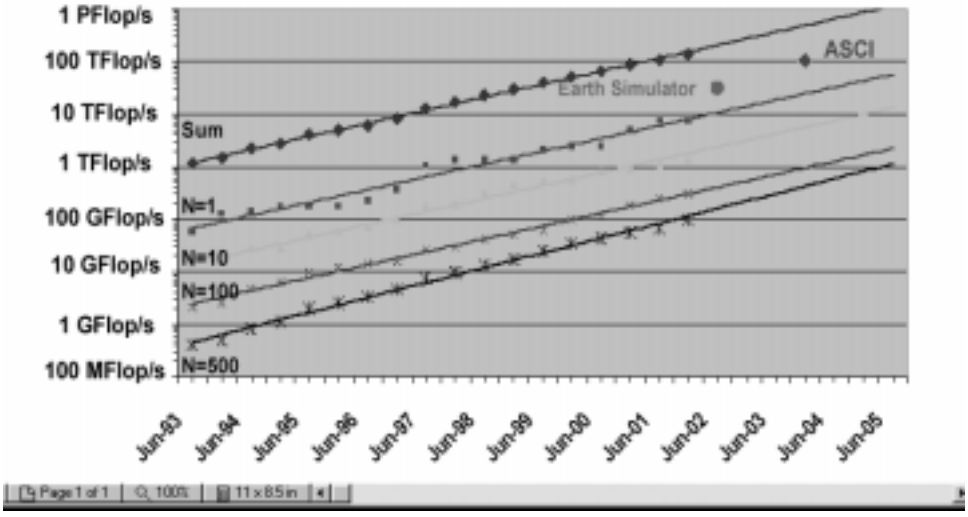
Performance - 1993/00



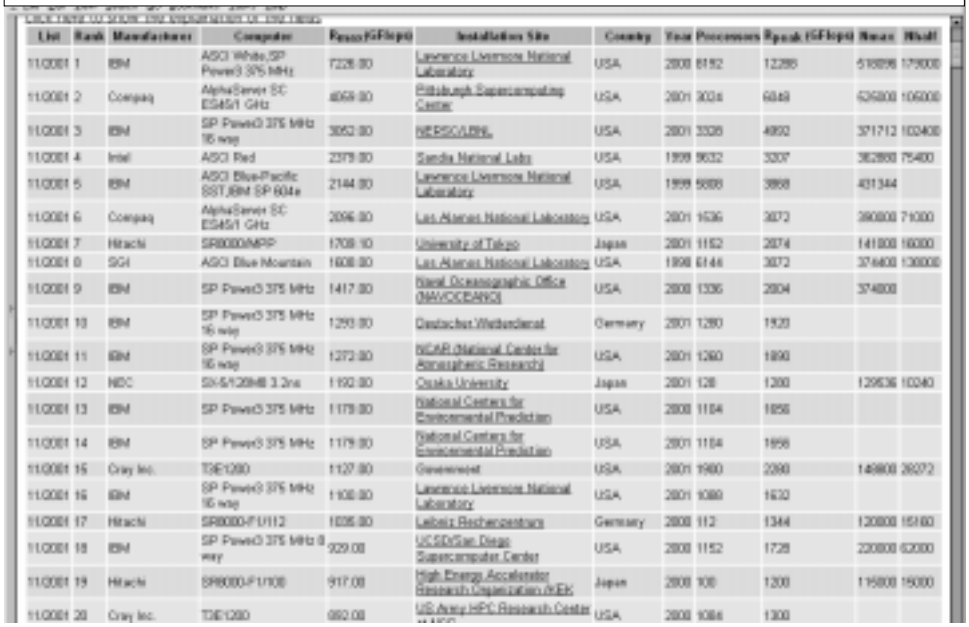
Moore's Law

- We can learn from the previous graphs that the computer performance increased exponentially.
- From 10^4 in 1955 to 10^{11} in 1995 (factor of 10^7 in 40 years)
- From 10^{11} in 1995 to $7.2*10^{12}$ in 2002 (factor of 60 in 7 years).
- $(R_2-R_1)=2^{(t2-t1)/Td}$ (Td=time for double performance; log2=0.3)
- Td=(40/7)*log2 = 1.7 Td=(7/log(60))*log2 = 1.2
- Moore's law states that the computer performance doubles each one and half year.
- The performance of storage media is for an order of magnitude less intensive.

Performance development - 1993/02 -> 05



Top20 from Top500



List	Rank	Manufacturer	Computer	Rmax [GFlop/s]	Installation Site	Country	Year	Processors	Rpeak [GFlop/s]	Nmax	Nhalf
11.2001	1	IBM	ASCI White,SP Power3 375 MHz	7226.00	Lawrence Livermore National Laboratory	USA	2000	8192	12288	518096	179000
11.2001	2	Compaq	AlphaServer SC ES45/1 GHz	4059.00	Pittsburgh Supercomputing Center	USA	2001	3016	6048	626800	106000
11.2001	3	IBM	SP Power3 375 MHz 16 way	3052.00	NERSC/LBNL	USA	2001	3328	4992	371712	102400
11.2001	4	Intel	ASCI Red	2379.00	Sandia National Labs	USA	1999	9632	3207	362880	75400
11.2001	5	IBM	ASCI Blue-Pacific SST,IBM SP 604e	2144.00	Lawrence Livermore National Laboratory	USA	1999	5808	3868	431344	
11.2001	6	Compaq	AlphaServer SC ES45/1 GHz	2096.00	Los Alamos National Laboratory	USA	2001	1536	3072	380800	71000
11.2001	7	Hitachi	SR8000/MPP	1709.10	University of Tokyo	Japan	2001	1152	2074	141800	16000
11.2001	8	SGI	ASCI Blue Mountain	1608.00	Los Alamos National Laboratory	USA	1998	6144	3072	374400	130000
11.2001	9	IBM	SP Power3 375 MHz	1417.00	Naval Oceanographic Office (NAVOCEANO)	USA	2000	1336	2004	374800	
11.2001	10	IBM	SP Power3 375 MHz 16 way	1293.00	Deutscher Wetterdienst	Germany	2001	1280	1920		
11.2001	11	IBM	SP Power3 375 MHz 16 way	1272.00	NCAR (National Center for Atmospheric Research)	USA	2001	1260	1890		
11.2001	12	NEC	SX-5/128M8 3.2ns	1192.00	Osaka University	Japan	2001	128	1280	129536	10240
11.2001	13	IBM	SP Power3 375 MHz	1179.00	National Centers for Environmental Prediction	USA	2000	1104	1656		
11.2001	14	IBM	SP Power3 375 MHz	1179.00	National Centers for Environmental Prediction	USA	2001	1104	1656		
11.2001	15	Cray Inc.	T3E1200	1127.00	Government	USA	2001	1980	2380	149800	28272
11.2001	16	IBM	SP Power3 375 MHz 16 way	1106.00	Lawrence Livermore National Laboratory	USA	2001	1080	1620		
11.2001	17	Hitachi	SR8000-F1/112	1035.00	Leibniz Rechenzentrum	Germany	2000	112	1344	120000	15100
11.2001	18	IBM	SP Power3 375 MHz 8 way	929.00	UCSD/San Diego Supercomputer Center	USA	2000	1152	1728	220000	62000
11.2001	19	Hitachi	SR8000-F1/100	917.00	High Energy Accelerator Research Organization /KEK	Japan	2000	100	1200	116800	19000
11.2001	20	Cray Inc.	T3E1200	892.00	US Army HPC Research Center (ARSC)	USA	2000	1084	1300		

Highlights from the Top10

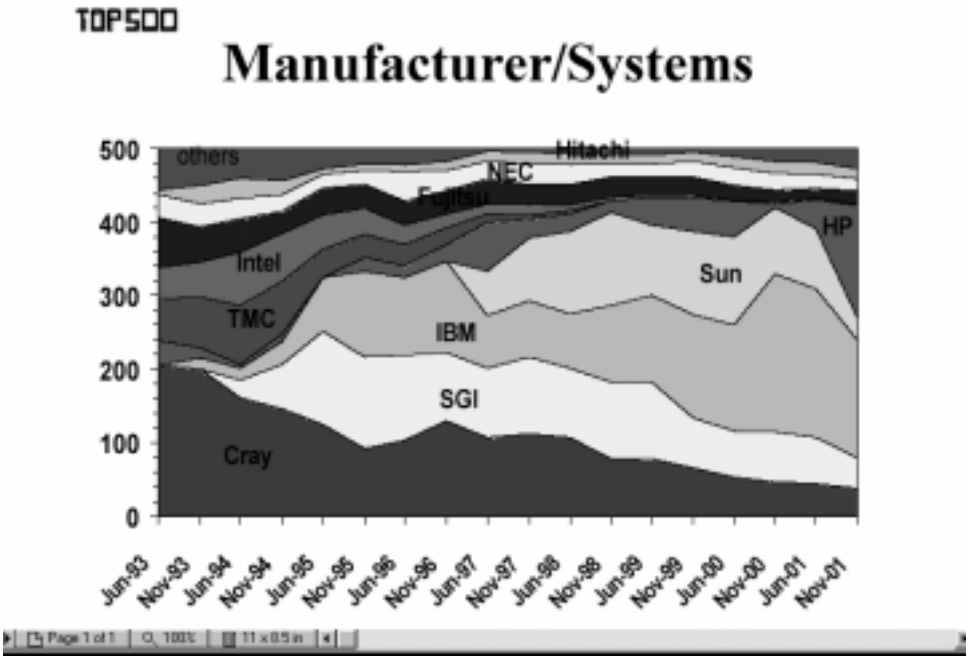
- Accelerated Strategic Computing Initiative – ASCI White is #1 with 7.2 TF/s on the Linpack.
- 16 systems have Linpack performance above 1 TFlop/s.
- 30 systems have peak performance above 1 TFlop/s.
- 1.29 Tflop/s is the entry point for the Top 10.
- Smallest number of processor in Top500 is 8.
- All vector based systems are of Japanese origin.
- Leading manufactures: IBM, Hewlett-Packard, SGI, Cray Inc., Sun.

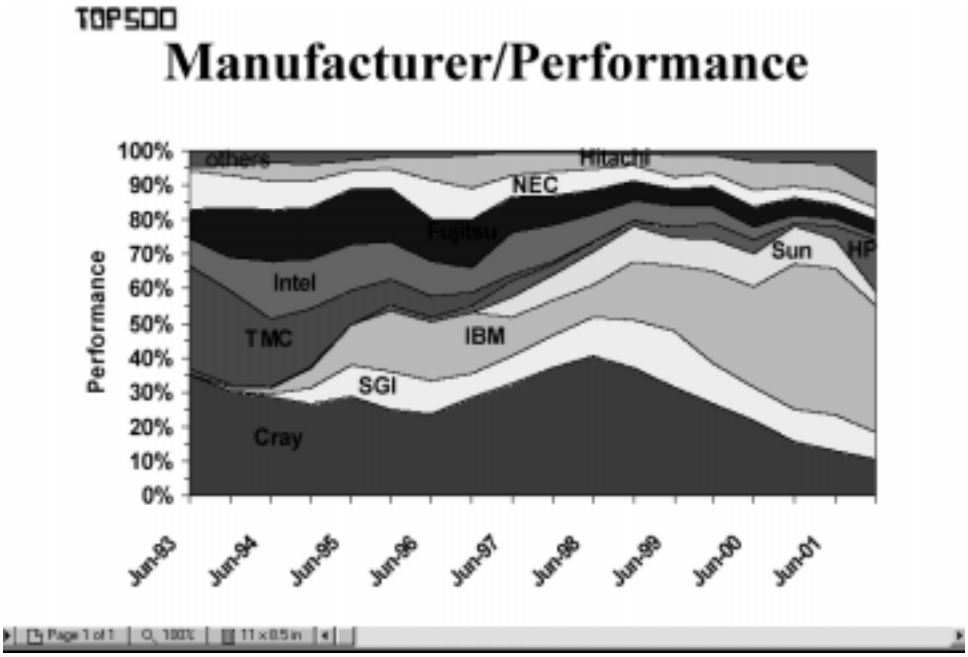
Austria on TOP500 -November 2000-01

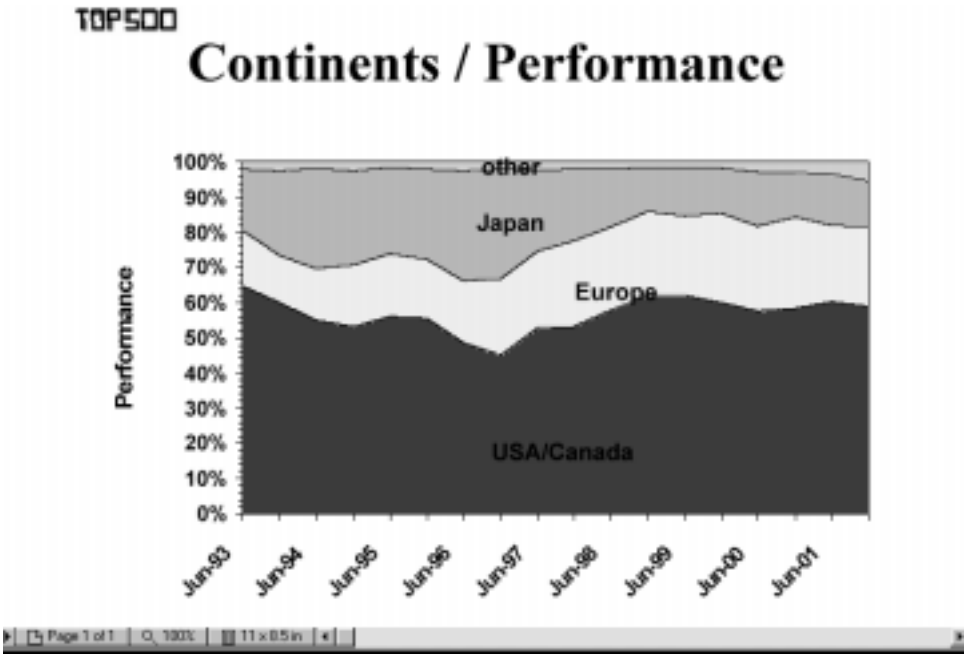
List	Rank	Manufacturer	Computer	R_{max} (GFlops)	Installation Site	Country	Year	Installation Type	Processors
November 2000	53	IBM	SP Power3 222 MHz	334.00	European Patent Office	Austria	2000	Government	562
November 2000	58	IBM	SP Power3 375 MHz	325.00	BM F LV STAB LRBS	Austria	2000	Government	318
November 2000	117	IBM	SP Power3 375 MHz	154.00	Bayer AG	Austria	2000	Industry	150
November 2000	209	IBM	SP Power3 375 MHz	100.00	Telekom Austria	Austria	2000	Industry	96
November 2000	400	IBM	SP PC604e 332 MHz	57.60	AI Informatics GmbH (AII)	Austria	2000	Industry	136

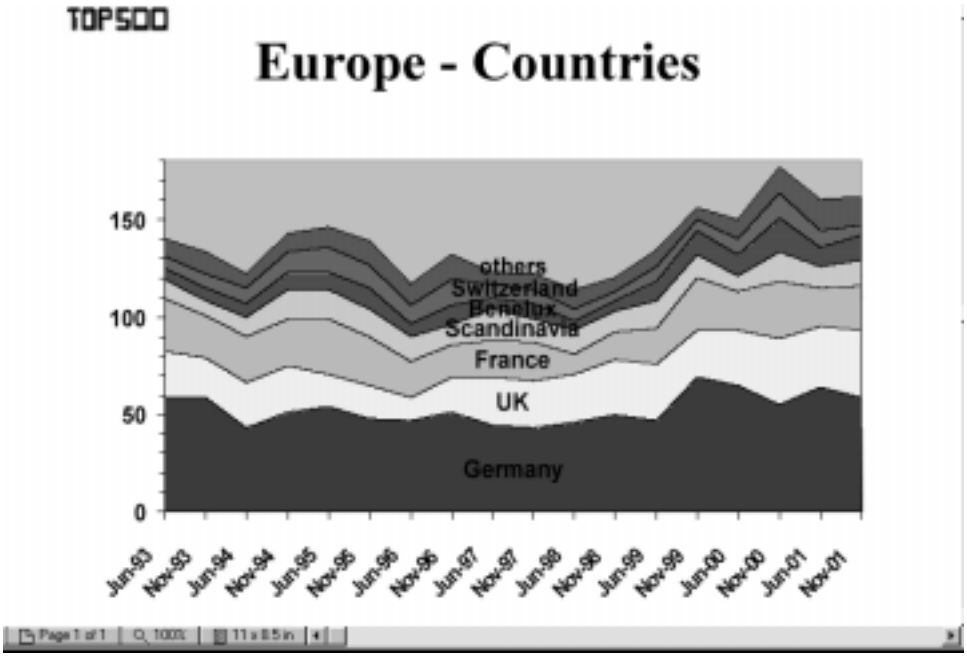
Document: Done

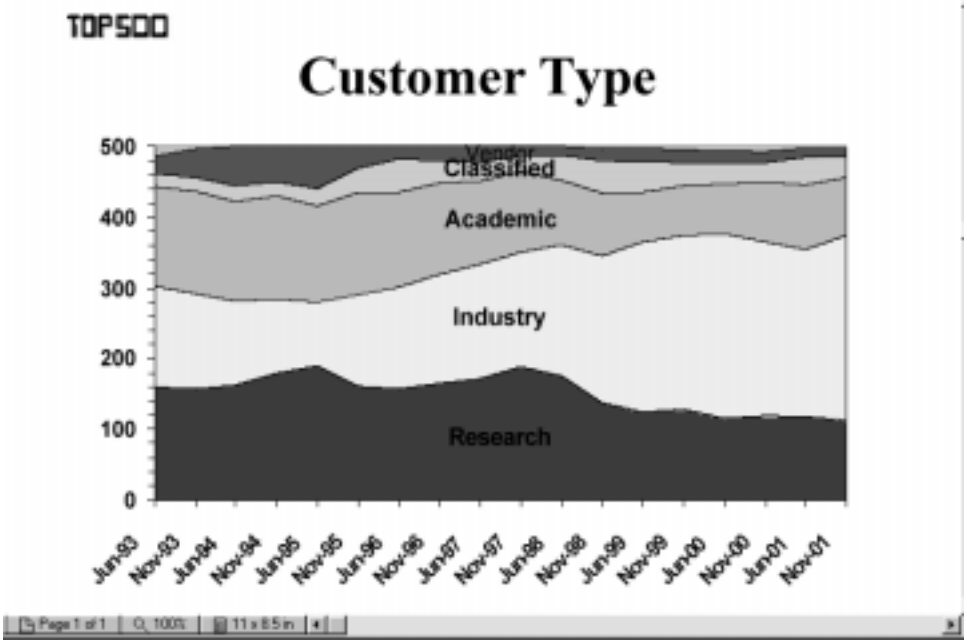
Nov./01	356	IBM	SP Power3 375 MHz	112.00	Banco Vitalico	Austria	2001	Industry	108
Nov./01	402	IBM	SP Power3 375 MHz	100.00	Telecom	Austria	2001	Industry	96

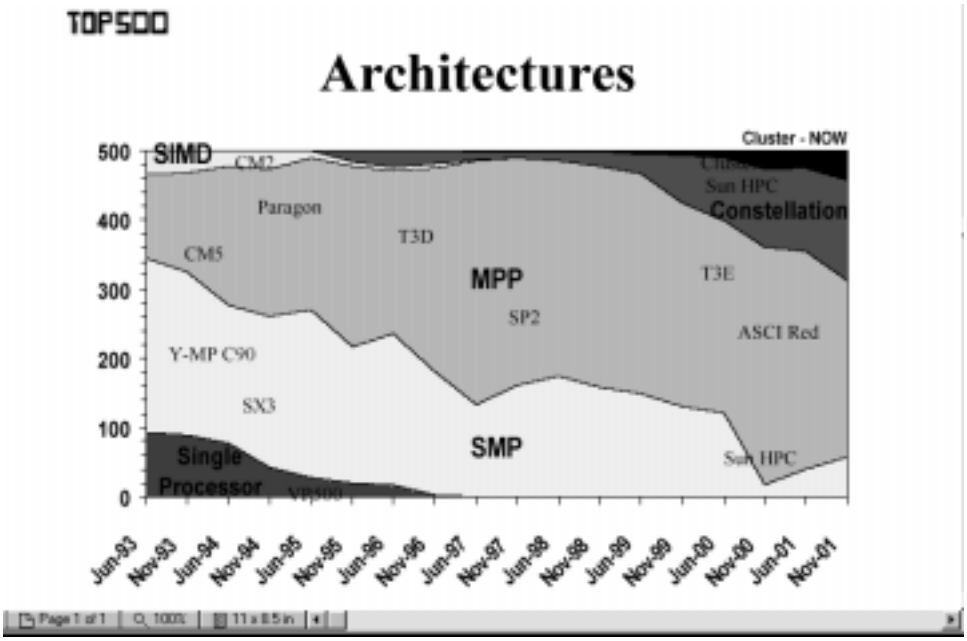


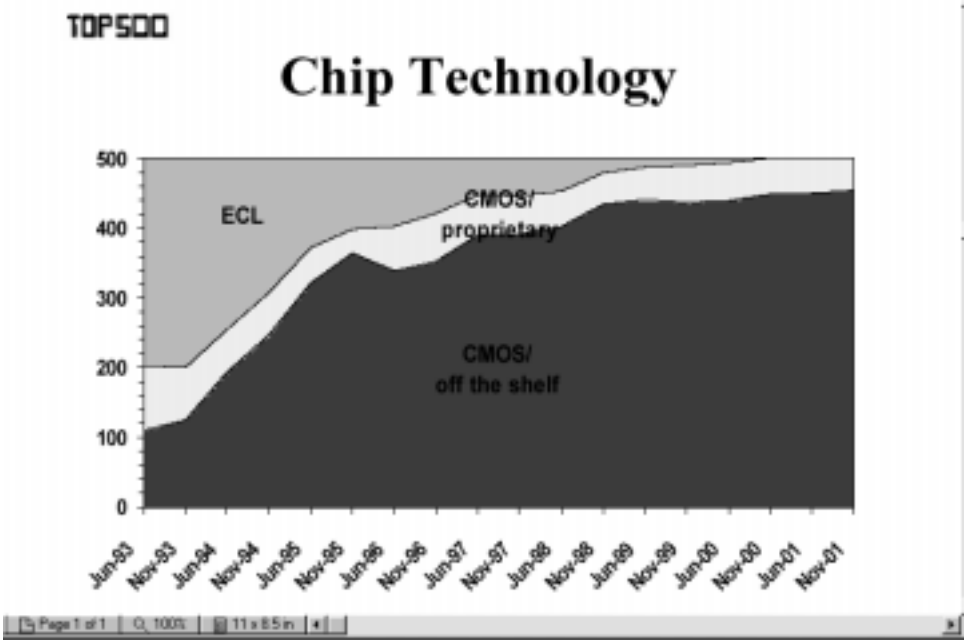

TOP500
Manufacturer/Performance
Performance
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
others
Hitachi
NEC
Fujitsu
Sun
HP
Intel
TMC
IBM
SGI
Cray
Jun-93
Jun-94
Jun-95
Jun-96
Jun-97
Jun-98
Jun-99
Jun-00
Jun-01


TOP500
Continents / Performance
Performance
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
other
Japan
Europe
USA/Canada
Jun-93
Jun-94
Jun-95
Jun-96
Jun-97
Jun-98
Jun-99
Jun-00
Jun-01


TOP500
Europe - Countries
150
100
50
0
others
Switzerland
Benelux
Scandinavia
France
UK
Germany
Jun-93
Nov-93
Jun-94
Nov-94
Jun-95
Nov-95
Jun-96
Nov-96
Jun-97
Nov-97
Jun-98
Nov-98
Jun-99
Nov-99
Jun-00
Nov-00
Jun-01
Nov-01


TOP500
Customer Type
500
400
300
200
100
0
Vendor
Classified
Academic
Industry
Research
Jun-93
Nov-93
Jun-94
Nov-94
Jun-95
Nov-95
Jun-96
Nov-96
Jun-97
Nov-97
Jun-98
Nov-98
Jun-99
Nov-99
Jun-00
Nov-00
Jun-01
Nov-01


TOP500
Architectures
Cluster - NOW
SIMD
CM2
Paragon
T3D
CM5
MPP
T3E
SP2
ASCI Red
Y-MP C90
SX3
SMP
Single Processor
Sun HPC
Constellation
500
400
300
200
100
0
Jun-93
Nov-93
Jun-94
Nov-94
Jun-95
Nov-95
Jun-96
Nov-96
Jun-97
Nov-97
Jun-98
Nov-98
Jun-99
Nov-99
Jun-00
Nov-00
Jun-01
Nov-01


TOP500
Chip Technology
ECL
CMOS/
proprietary
CMOS/
off the shelf
500
400
300
200
100
0
Jun-93
Nov-93
Jun-94
Nov-94
Jun-95
Nov-95
Jun-96
Nov-96
Jun-97
Nov-97
Jun-98
Nov-98
Jun-99
Nov-99
Jun-00
Nov-00
Jun-01
Nov-01