Towards the Teraflops Capability for CFD[1]

Robert Schreiber[2] and Horst D. Simon[3]

Report RNR-92-016, May 1992

NAS Systems Division
Applied Research Branch
NASA Ames Research Center, Mail Stop T045-1
Moffett Field, CA 94035

Abstract. We are surveying current projects in the area of parallel supercomputers. The machines considered here will become commercially available in the 1990 - 1992 time frame. All are suitable for exploring the critical issues in applying parallel processors to large scale scientific computations, in particular CFD calculations. This report presents an overview of the surveyed machines, and a detailed analysis of the various architectural and technology approaches taken. Particular emphasis is placed on the feasibility of a Teraflops capability following the paths proposed by various developers. This a revision of an internal NAS document, which has been written originally in Spring of 1990, and revised for publication as a book chapter in Spring of 1992.

---

[1]to appear as Chapter 16 in "Parallel CFD - Implementations and Results Using Parallel Computers", pages 331 - 360, edited by Horst D. Simon, MIT Press, Cambridge, 1992.

[2]This author is an employee of the Research Institute for Advanced Computer Science (RIACS), NASA Ames Research Center, Moffett Field, CA 94035. This work was supported by the NAS Systems Division via Cooperative Agreement NCC 2-387 between NASA and the University Space Research Association (USRA).

[3]Numerical Aerodynamic Simulation (NAS) Systems Division, Mail Stop T045-1, NASA Ames Research Center, Moffett Field, CA 94035-1000 (simon@nas.nasa.gov). The author is an employee of Computer Sciences Corporation. This work was supported through NASA Contract NAS 2-12961.

1 Introduction

In the last several years a wide variety of parallel machines have become available for exploring the issues of using parallelism in scientific computing. Whereas most of the early ("zero-th generation") machines from 1983 to 1987 were rather experimental in nature, and served mainly for research investigations in areas such as algorithms, languages, operating systems for parallel computing, in 1988 and 1989 several members of a first generation of *parallel supercomputers* became available. We want to use the term "supercomputer" here because these parallel supercomputers such as the current CM-2 and Intel Touchstone Gamma machine are in their larger configurations comparable both in memory and peak computational speed to the performance of the most powerful conventional supercomputers, e.g., the CRAY Y-MP. However, it is well known that these machines are still very deficient in their systems aspects, for example in their ability to handle a large number of users. Today[3] we are at the threshold to a second generation of parallel supercomputers, which offer order of magnitude improvements in computational power over the previous generation as well as an improved software and user environment.

Because of their considerable potential computational power, parallel supercomputers are increasingly considered as an alternative to the more conventional supercomputers based on a small number of powerful vector processors. Even though many research issues concerning their effective use and their integration into a large scale production facility are still unresolved, parallel supercomputers are already used for production computing, although mostly in a single application mode.

The NAS Program Plan for the Numerical Aerodynamic Simulation (NAS) Systems Division at NASA Ames Research Center outlines an ambitious program leading to a sustained Teraflops computing capability by the year 2000 (for more details see [8]). An important component on the path to the Teraflops machine are "Highly Parallel Testbed Systems," which are to be installed at NAS both for parallel system research and for encouraging the gradual migration and/or new production of parallel applications on these

[3]Major portions of this chapter were written in 1990 and reflect the state of hardware development at that time. We believe that our analysis and conclusions are still valid today in late 1991. Actually some recent events such as the move towards HPF (High Performance Fortran) show that we were quite correct in our analysis.

testbed machines. Currently there are two first generation parallel supercomputers installed at NAS: a 32K processor Connection Machine 2 from Thinking Machines Corporation and a 128 processor Touchstone γ-prototype machine from Intel Scientific Computers. These machines satisfy the performance objectives for the first generation of parallel testbed systems in the NAS Program Plan (see Table 7.3 in [5]): a peak computing rate of about 10 Gflops, a sustained rate of about 1 Gflops, and of the order of 0.5 Gbytes of main memory.

The goals for the second generation testbed machine(s), which coincide with the second generation of parallel supercomputers are 10 Gflops sustained rate, 100 Gflops peak rate, and 8 Gbytes of main memory. In our survey we have examined in more detail several of the commercial parallel processors which will become available in the 1990 - 1992 time frame and evaluated their suitability for parallel CFD calculations.

There are several categories of high performance computing architectures, which we did *not* include in our survey. We did not include any university or other research machines (e.g. the Cedar project at the University of Illinois or new efforts by Chuck Seitz at Caltech), since consistent with the NAS program plan we were restricting ourselves to commercially available machines. We did not include the next generation of traditional vector supercomputers such as machines currently developed at Cray Research (C-90), Cray Computer (Cray-3), SSI, or NEC (NEC-SXX). These machines do offer some moderate parallelism (from 4 to 64 processors), but they are not scalable. Finally, we did not include various next generation machines, which are the logical evolution of current mini-supercomputers (e.g. Convex C3), current high performance workstations, or various machines based on the "let's put some transputers in a MacIntosh and call it supercomputer" approach. Even though offering moderate degrees of parallelism, most of these machines are intended for different applications, and are not likely to come even close to the performance goals for the second generation testbed machine.

After presenting some data about the machines in Section 2, we offer a first evaluation of the capabilities of the machines in Section 3. Section 4 presents the issues and tradeoffs which are to be considered when comparing the various architectural approaches. Here we attempt to draw some general conclusions about architectural and software issues independent from the particular machine implementations discussed in Section 4. Section 5 summarizes our findings and makes some recommendations.

Towards the Teraflops Capability for CFD

Robert Schreiber[1]
and
Horst D. Simon[2]
Mail Stop T045-1
NASA Ames Research Center
Moffett Field, CA 94035

May 1, 1992

Abstract

We are surveying current projects in the area of parallel supercomputers. The machines considered here will become commercially available in the 1990 - 1992 time frame. All are suitable for exploring the critical issues in applying parallel processors to large scale scientific computations, in particular CFD calculations. This chapter presents an overview of the surveyed machines, and a detailed analysis of the various architectural and technology approaches taken. Particular emphasis is placed on the feasibility of a Teraflops capability following the paths proposed by various developers.

Keywords: parallel processing, SIMD machines, MIMD machines, message passing, shared memory, performance evaluation, architecture evaluation

---

[1]This author is an employee of the Research Institute for Advanced Computer Science (RIACS), NASA Ames Research Center, Moffett Field, CA 94035. This work was supported by the NAS Systems Division via Cooperative Agreement NCC 2-387 between NASA and the University Space Research Association (USRA).

[2]This author is an employee of Computer Sciences Corporation. This work was supported by NASA Contract No. NAS2-12961.

2 Machine Characteristics

2.1 BBN

The BBN Butterfly is the oldest [3] among the machines considered here. Its history goes back to the late seventies, when the Butterfly development began, mainly as a switching network. The latest product in this series is the Butterfly TC2000. This project is a commercial development by BBN funded through a limited partnership. It is not related to the Monarch project, which was funded in its early stages by DARPA, but is apparently no longer actively pursued by BBN, and has currently no DARPA funding.

The Butterfly TC 2000 uses the new Motorola 88000 RISC chip for the processor nodes. The projected performance of the 88000 with a floating-point unit on the chip is about 10 Mflops for 64 bit computation. Since this is a scalar processor the LINPACK performance is a relatively high 7 Mflops. There will be 4 - 16 Mbyte memory per node.

The switch of the Butterfly TC 2000 is completely redesigned when compared with the earlier Butterfly GP1000. It will support 4 Gbytes global memory access, with a bandwidth of 38 MBytes/sec. An initial configuration will be 64 node system. A final extension to 512 processors and 16 Gbytes of memory is planned.

BBN had an ambitious follow on project to the TC2000 called Coral. However, BBN was unable to attract the necessary funding for continued support and closed down its advanced computing division in August of 1991.

2.2 Encore

Encore is a manufacturer of shared-memory multiprocessors using coherent caching, a bus-based architecture, and commodity microprocessors.

Under DARPA funding, a machine consisting of 8 Multimax systems with 20 Motorola 88100 processors each has been developed. A second level cache and bus maintains coherence of all caches across the whole machine. The cache policy is write deferred, which allows some writes to be entirely local, under the control of software. This reduces bus traffic by as much as 70% compared with the usual write through scheme. Delivery is in late 1990. Overall performance is 3000 MIPS sustained. Flops are about 1/10 MIPS for the 88100.

Table 1: BBN TC2000

Processor	
Number of processors	8–504
Chip or technology	88000 with 3 cache chips
Clock rate	20MHz
Architecture	RISC
Peak Mips	20
Peak Mflops 32 bit	20
64 bit	10
Peak Gflops total 64bit	5
Full System Price	$ 20M
IEEE arithmetic	yes
Memory	
Memory per processor	4–16Mbyte
Technology	CMOS 1M DRAM
Instruction Cache	32K
Data cache	16K
Total maximum memory	16Gbyte (?)
Price per Mbyte	$ 2500
Addressing virtual	32bit
Addressing physical	34bit
Peak access rate	38Mbytes /sec
Access rate per processor	
Latency for nonlocal data	2400nsecs = 48 clks
ECC	Parity
Interconnect	
Interconnect topology	Butterfly switch
	Packet routed with retry if blocked
Interconnect technology	gate arrays
	uses 8 X 8 X 8bit Xbars
I/O	
I/O architecture	VME/PE
Bandwidth total	2500 Mb/sec
Achievable	
Striping	No
Software	
Operating System	nX + pSOS
NFS	yes
Languages	own parallel Fortran, PCF like
	Linda

2.3 Evans and Sutherland

The Computer Division of Evans and Sutherland had announced the first two shipments of their first machine in October 1989. In November 1989 the parent company announced that the Computer Division is up for sale. Since no buyer was found the division has been closed by the end of the 1989. There are probably about six ES-1 machines, which have been produced, and which may find some use, but in all likelihood the inclusion of the ES-1 machine in this study appears to be only of historical interest (see Table 2).

2.4 IBM

IBM has developed a number of research parallel machines, including the RP3, the Victor, and the GF11. At present, a new research project for the construction of a scalable MIMD message passing machine known as Vulcan (formerly TF1) is in progress. None of these is a supported product of IBM. We include them because of IBM's importance and its recent decision to put significant resources into development of a new massively parallel system.

Victor is a conventional message passing multicomputer with a grid topology, based on transputer T800 chips. There are 256, in a 16 by 16 grid. The RP3 is a complex, physically distributed and logically shared memory multiprocessor. Although the design called for 512 PEs, only 64 were built. Neither is particularly fast.

The GF11 is more impressive. There are 330 working processors at 20 Mflops, and the design allows for 576. IBM users are getting 7 Gflops on QCD applications. It is tolerant of real-time failures. The memory is distributed: on each processor there are 256 32bit registers, 16 K words of SRAM, and 512 K words of DRAM. Thus a full system has 288 Mwords of DRAM. The processors are interconnected with a 3-stage Benes network of 24 by 24 crossbar switches, forming a Memphis switch. The switch may be reconfigured within one 200ns cycle to any of a pre-loaded list of 1024 switch settings. Many of the most useful data permutations may be included in this list, along with certain one-to-many mappings. The current configuration is able to move more than a gigaword per second (6.4 gigabytes) between processors. Ten of the switch ports are used for I/O to 10 Mbyte/sec disks. It burns 256Kwatt, which is about twice the power consumption of a Cray-1. The software environment is primitive at this time (assembly language).

Table 2: ES-1

Processor	
Number of processors	32–128
Chip or technology	full custom CMOS 100K gate
Clock rate	25 MHz
Architecture	RISC with dynamic overlap
Peak Mips	25
Peak Mflops 32 bit	25
64 bit	12.5
Peak Gflops total 64bit	1.6
Full System Price	$ 8.5M
IEEE arithmetic	yes
Memory	
Memory per processor	8 or 16 Mbytes
Technology	CMOS 1M DRAM
Instruction Cache	16K
Data cache	8K + 32K for context
Total maximum memory	2Gbyte
Price per Mbyte	8500
Addressing virtual	30bit
Addressing physical	32bit
Peak access rate per processor	50 Mbytes/sec
Latency for nonlocal data	
ECC	SECDED
Interconnect	
Interconnect topology	16 pe × 8 banks Xbar second level Xbar
Interconnect technology	
I/O	
I/O architecture	up to 8 IOP up to 8 channels each 25 Mb/sec-chan
Bandwidth total	1600 Mbyte/sec
Achievable	700 Mbyte/sec
Striping	yes; 12Mb/sec/channel
Software	
Operating System	Mach
NFS	yes
Languages	Fortran 77 with directives parallel Fortran, parallel C

Vulcan is the new name for TF1. It is a 32,768 processor MIMD message passing architecture based on the Intel i860 microprocessor. The peak aggregate performance of this architecture is $10 * 2^{18} = 2.56$ Teraflops. A second processor checks the arithmetic of the first. Memory per processor 8 Mbytes. Total memory in the full machine is 32 billion 64-bit words.

The message passing network consists of 8 independent switch planes, The machine is arranged as a 16 by 16 grid of racks. In every rack, on every switch plane, there is an H board, a V board and a P board. The H and V boards connect to the 16 other H and V boards in the same row or column of the grid and on the same switch plane. The P board connects the H to the V. I/O bandwidth is 5 gigabyte per sec per rack. The devices are 150 meg 3.5" winchesters. With 64,000 such drives the system capacity is 9 terabytes. The estimated cost of the hardware is about $120 million for the whole system. This is not a product, but if it were, sale price would likely be 3 - 4 times the hardware cost.

Little detail is known about the operating system. IBM will first implement a very low-level message passing system in the form of minimal kernel support and a linkable library. What higher level communication functions will be provided has not yet been decided.

2.5 Intel/Touchstone Project

Intel Scientific Computers (ISC)[4] delivered its first hypercube system in 1985 and has shipped more than 100 iPSC (Intel Personal Supercomputer) systems. In October 1987 ISC announced a major upgrade to their current hypercube, the iPSC/2 [1]. The iPSC/2 offers increased processor speed and a considerably improved communications network in comparison to the first generation iPSC. In Spring of 1989 Intel signed a major agreement with DARPA for the development of sequence of prototype machines. This is the so-called Touchstone project. NAS received in January 1990 one of the first Gamma machines [2, 8]. The Gamma prototype is built with the i860 chip and the communication network of the iPSC2. The next step in the Touchstone project is the Delta machine, which will be i860 based, but with a new communication network. Table 3 summarizes the last machine, inter-

[4]now called Intel Supercomputer Systems Division (SSD)

Table 3: Intel Touchstone σ machine

Processor	
Number of processors	2048
Chip or technology	i860XP
	custom CMOS; .75 μ
	4 procs and cache on single substrate
Clock rate	60–80 Mhz
Architecture	RISC
	2 instr/clock
Peak Mips	50
Peak Mflops 32 bit	100
64 bit	72
Peak Gflops total 64bit	150
Full System Price	?
IEEE arithmetic	yes
Memory	
Memory per processor	128 - 256 Mbytes
Technology	4Mb, 70 nsec CMOS DRAM
Shared Cache	256 K combined
Instruction Cache	16 K
Data cache	16 K
Total maximum memory	128 Gbyte
Price per Mbyte	8500
Addressing virtual	virtual
Peak access rate per processor	400 Mbytes/sec
Latency for nonlocal data	10–25 μsec
ECC	SECDED
Interconnect	
Interconnect topology	grid - 2D
Interconnect technology	own version of Caltech routing chips
I/O	
I/O architecture	separate I/O nodes
Disks	3.5 inch, 600 Mb
	12 μsec latency
Bandwidth total	
Achievable	
Striping	yes
Software	
Operating System	NX, reactive kernel
NFS	
Languages	Fortran 77, C, with message passing
	Linda

nally referred to as the Sigma prototype[5], that Intel will develop under the Touchstone project.

The chief new features are the use of the i860XP processor, which will have 72 Mflops double precision performance, roughly twice that of an i860. The machine will scale up to 2048 nodes. Memory per processor will be expandable up to 128 or 256 Mbytes per node. Router performance will be greatly enhanced through better software and an Intel implementation of the Caltech message routing chip. The 2D grid topology will still be used.

This machine offers very high performance per dollar, memory bandwidth per dollar, and memory per dollar.

2.6 Intel/iWARP

Intel is developing, in a completely separate part of the company, another high- performance, message-passing multicomputer, the iWARP. The iWARP is based on the hardware and software designs developed at CMU by the WARP Project, a DARPA funded research program headed by Prof. H.T. Kung. The principal aim of the project is a machine for signal processing applications, but there is really very little to distinguish it from one designed for other scientific applications.

The iWARP has some very interesting features(see Table 4):

1. The nodes are VLIW (Very Long Instruction Word) machines; instructions are 96 bits long. A compiler technique, software pipelining, developed by M. Lam at CMU, is used to schedule the use of parallel hardware resources in the node effectively.

2. A large fraction (30%) of the silicon is devoted to message passing support. As a result, the communication bandwidth of the iWARP chip is enormous: it has 4 fully bi-directional hardware channels that are byte wide and run at 40 Mhz. This gives the chip a total of 320 Mbytes/sec of communication bandwidth. We expect that the communication bandwidth of the overall system will be very high for regular (for example grid-type) communication and will probably also be very high for general communication. Moreover, the latency can be extremely low, allowing the use of very fine grained, systolic algorithms.

[5]now called Paragon

Table 4: iWARP

Processor	
Number of processors	1024
Chip or technology	CMOS 600K transistor
Clock rate	40Mhz
Architecture	VLIW
Peak Mips	100
Peak Mflops 32 bit	20
64 bit	10
Peak Gflops total 64bit	10
Full System Price	$ 5M
IEEE arithmetic	?
Memory	
Memory per processor	512 Kbyte
Technology	CMOS SRAM
Total maximum memory	512 Mbyte
Price per Mbyte	$10,000
Addressing virtual	physical
Peak access rate per processor	160 Mbyte/sec
Latency for nonlocal data	less than 10 cycles
ECC	?
Interconnect	
Interconnect topology	reconfigurable; 4 port/processor
Interconnect technology	integrated pathway unit
I/O	
I/O architecture	high bandwidth real time
Disks	
Bandwidth total	
Achievable	
Striping	
Software	
Operating System	Unix
NFS	
Languages	parallel Fortran and C with MIMD send/receive

3. Local memory is very fast SRAM. Only the iWARP and the Tera machines, among the highly parallel machines discussed here, use SRAM. (It is standard in the Crays, NEC, and other conventional supercomputers). The result is that cache is not needed, but the price of memory is high.

4. Message passing latency is extremely low, due to hardware support that provides flow control on a single word basis, and that also provides hardware routing.

2.7 Kendall Square Research

Kendall Square Research (KSR) is developing a is scalable, shared-memory MIMD supercomputer with the following salient features: up to 1020 CMOS custom processors with 40 Mflops peak performance; an innovative shared-memory system in which all memory is cache; a ring of rings interconnect system with most remote access handled with 100 cycle, or 5 μsecs latency; implementation of the processor is done with 12 custom ICs of 6 types; and a parallelizing Fortran compiler that restructures code to enhance locality of reference. First customer shipments were in late 1991. So far little has been published on the machine.

2.8 Maspar

Maspar was formed in 1988 with venture capital funding to develop a massively parallel, SIMD machine. Unlike other systems surveyed, the Maspar machine is a first generation parallel supercomputer: peak price is under $1 million, maximum memory is 256 Mbytes, peak 64bit performance is 1.34 billion adds and .44 billion multiplies per second. We included the Maspar machine because of certain very interesting hardware characteristics. The Maspar is a "workstation version" of a SIMD machine (see Table 5).

2.9 Myrias

The Myrias SPS-2 parallel computer system was developed by Myrias Research Corporation in Edmonton, Alberta. The system is Motorola 68020 based, with up to 4 Mbytes of memory per processor. The SPS-3 version to

Table 5: Maspar MP1

Processor	
Number of processors	16,384
Chip or technology	full custom CMOS
Clock rate	14 Mhz
Architecture	SIMD array; 4 bit wide
Peak Mips	1.4
Peak Mflops 32 bit	0.12 (+)
	0.074 (*)
64 bit	0.082 (+)
	0.027 (*)
Peak Gflops total 64bit	1.34
Full System Price	$ 0.75 M
IEEE arithmetic	DEC
Memory	
Memory per processor	16 Kbytes
Technology	1 Mbit DRAM
Shared cache	192 bytes
Total maximum memory	256 Mbyte
Price per Mbyte	3000
Addressing virtual	no
Peak access rate per processor	4bits / clk
	4bits / 9 clks
Latency for nonlocal data	router: 4500 clks / 32 bits = 315 μsec
	Xnet: 47 clks / 32 bits = 3.3 μsec
ECC	SECDED
Interconnect	
Interconnect topology	3 stage omega net
Interconnect technology	custom CMOS
	router chip
I/O	
I/O architecture	I/O cards
	up to 256 Mb each;
	connected to router
	229 Mb/sec bus; HSC
Disks	5.25 in 700 Mb
	4, 8, or 16 in pack with ECC and spare
Bandwidth total	1.5Mb/sec/drive
Achievable	24 Mb/sec/system
Striping	yes
Software	
Operating System	Ultrix
NFS	yes
Languages	Fortran 90 and parallel C

Table 6: Myrias SPS-3

Processor	
Number of processors	64 - 2048
Chip or technology	M68040
Clock rate	16.67Mhz
Architecture	CISC
Peak Mips	10.8 per PE
Peak Mflops 32 bit	10.8
64 bit	5.4
Peak Gflops total 64bit	0.33 (64 PE's)
Full System Price	$ 0.80 M (per 64 PE's)
IEEE arithmetic	yes
Memory	
Memory per processor	16 Mbytes
Technology	1 Mbit DRAM
Total maximum memory	1 Gbyte (per 64 PE's)
Price per Mbyte	$ 800
Addressing virtual	yes
Peak access rate	33 Mbytes/sec per processor
Latency for nonlocal data	variable
ECC	SECDED
Interconnect	
Interconnect topology	4 on-board processors connected via bus 16 boards connected via 2 backplanes into cage cages can be connected in any way
Interconnect technology	
I/O	
I/O architecture	one or more IOP per cage each IOP 20 Mbyte/sec
Disks	Maximum Strategy parallel disk arrays
Bandwidth total	20 Mb/sec per array
Achievable	20 Mb/sec
Striping	yes
Software	
Operating System	Unix
NFS	yes
Languages	PAMS(parallel appl. management system) parallel Fortran with "pardo"; parallel C

be available in July 1990 will be based on the 68040. With the exception of the processor upgrade, there will be no significant changes in the hardware. Myrias went out of business in 1991, and is included here only for historical interest.

In Table 6 the features of the SPS-3 system are listed. Unless indicated otherwise, the figures are for a 64 processor cage. Larger systems may consist of several cages, where one cage consists of 16 processor boards, and each board includes 4 processors, together with an interface to the backplane. The four on-board processors are connected via a bus. A card cage consists of 16 boards connected via a backplane, together with an I/O card containing up to four I/O processors.

From an application programmer's point of view, each of the processing elements appears to be directly connected to all others. The Myrias system supports virtual memory. Parallelism is utilized through a "pardo" (parallel do), which distributes independent instances of a loop over available processors. Myrias parallel Fortran and C both support the pardo construct.

2.10 NCUBE

The NCUBE2, the second generation of NCUBE machines, has been announced in the third quarter of 1989. The NCUBE2 processor is based on a custom chip with a peak of 3.2 Mflops in 32 bit mode and 2.4 Mflops for 64 bit arithmetic. The actual delivered speed is probably about 1.5 to 2 Mflops per node.

The hypercube interconnect scheme is retained, but there is full direct routing supported by the hardware. The message latency has been reduced from about 250 microseconds on the NCUBE1 machine to a range of about 2.5 microseconds. The bandwidth is about 2.5 Mbytes/sec per DMA channel.

Processor memory is available up to potentially 64 Mbytes per node (based on availability of 4 and 16 Mbit chips). In 1989/90 memory configurations of 1, 4, and 16 Mbytes/processor are available. The complete NCUBE2 system can be configured with up to 8192 processors. A 32,768 processor machine in 1992 appears feasible based on some not further disclosed packaging technology. A 256 node system with 4 Mbytes of memory per node would deliver 400 - 500 Mflops peak. Such a system would be available for about $ 1 million in 1990. Publicly NCUBE has not indicated any more detailed plans beyond the NCUBE2.

Table 7: NCUBE2

Processor	
Number of processors	8192
Chip or technology	1 micron CMOS
Clock rate	20 MHz
Architecture	CISC
Peak Mips	7.5
Peak Mflops 32 bit	3.2
64 bit	2.4
Peak Gflops total 64bit	20
Full System Price	$ 30 M(?)
IEEE arithmetic	yes(?)
Memory	
Memory per processor	up to 64 Mbyte
Technology	16 Mbit DRAM
Total maximum memory	130 Gbytes
Price per Mbyte	?
Addressing virtual	?
Peak access rate per processor	
Latency for nonlocal data	2.5 μsec
ECC	?
Interconnect	
Interconnect topology	hypercube
Interconnect technology	DMA
	2.5 Mbytes/sec/channel
I/O	
I/O architecture	
Disks	
Bandwidth total	
Achievable	
Striping	
Software	
Operating System	Unix like
NFS	
Languages	Parallel Fortran and C with message passing

2.11 Tera Computer Corporation

Tera was formed in 1988 to build a supercomputer based on a design developed by Burton Smith at the Supercomputer Research Center. Tera has received continued DARPA funding for their development work. The Tera machine will be a shared-memory multiprocessor in which processor nodes, memory nodes, I/O caches, and I/O processors are all intermingled in a three dimensional switching lattice, implemented with fast switch nodes. The processors are custom designed with a very short cycle time. In contrast to many other parallel machines, the switch bandwidth is very high. The relatively long latency for memory access is masked through the technique, pioneered by Smith in the HEP, of sharing the processor among multiple processes (instruction threads) on a single instruction basis, with hardware arbitration. A nice development is a dynamic hardware scheduling technique that executes only threads that are ready in a dataflow-like manner. Tera will also implement very fast synchronization as part of the memory system. Most of the details of the Tera system are not known to the public, although several presentations have been made at conferences.

2.12 Thinking Machines Corporation

Thinking Machines Corporation (TMC) has been very successful with their CM-2 [6, 8, 7] with about 35 machines installed. Recently DARPA announced continued funding of TMC's future development efforts for a 1 TFLOP machine. At the time we made this survey, TMC acknowledged the existence of a Connection Machine 5 development effort, which was accelerated through the new round of DARPA funding. However, TMC was only willing to discuss performance goals of the CM-5 in very general terms, without providing specific details of the architecture. Most of our discussion below is based on the CM-2.

The performance goals of the CM-5 are about 10 times the performance goals of the CM-2, upward compatibility with software, support of MIMD style multitasking, and multi-user network access. Much finer partitioning of the processors among tasks will be possible, giving the machine a MIMD architecture flavor. [6]

[6] Much more is now known of the CM-5: it is a Sparc-based MIMD system with custom, VLSI vector accelerators per node (reminiscent of the iPSC/VX) giving the node a peak

3 Comparison of the Machines

The surveyed machines are of four general types. Some (the BBN, Kendall Square, Myrias, and Tera) are MIMD multiprocessors, that share a unified memory space as do today's supercomputers (Cray, NEC, Fujitsu, Hitachi). As such, these machines are the most like those in use now. The second group (Intel and Ncube) are MIMD multicomputers. (They have also been called "message passing" machines, but we prefer the name multicomputer.) In these, each processor has its own memory and may not address the memory of another processor directly. Synchronization and communication are accomplished by messages, sent by one and received by one or several processors. The third group (Thinking Machines, Maspar) are SIMD machines that have a very large number of not very powerful processors, which operate in lock-step carrying out the same instruction. Finally the iWARP is a systolic machine, which does not fall into any of the above categories.

The somewhat confusing picture arising after studying several pages full of data becomes much clearer, when the machines are grouped according to category. In subsequent subsections a more detailed discussion of the strength and weaknesses of the surveyed machines are given.

3.1 MIMD shared-memory machines

There are six entrants in the MIMD shared-memory category: BBN, Encore, Evans & Sutherland, Kendall Square, Myrias, and Tera. As already mentioned above, the future of the computer division of E & S is uncertain. During our first visit with E & S we were told that there are plans for an "ES-2" machine. However, by the time a presentation on the new machine was about to take place, the company had announced its intentions to either

performance of 128 Mflops. Maximum configuration is 16K nodes (hence, in theory, it is a 2 Tflop machine) but at an astounding price. For ordinary supercomputer prices, 1024 node systems are possible. First shipments occurred in late 1991, but delivery of the vector units has been delayed; it now seems they will arrive at the end of 1992. The interconnect is, for the first time, a fat tree. (A fat tree is an complete binary tree in which the bandwidth of the edges increases geometrically as one moves up from the leaves toward the root. The use of fat tree interconnects was first advocated in the mid 80s by C. Leiserson of MIT, who proved some very powerful near-optimality theorems for them.) Node to node interconnect bandwidth seems to be in the 5-15 Mbyte/second range. The programming environment includes CM Fortran as well as a simple message-passing library.

sell or close down the computer division. In January 1990 the Computer Division has been effectively closed down. Although there is a possibility that the "ES-2" will be manufactured under a different name, the recent events have delayed this machine and there is no reason to pursue it further.

The BBN TC2000 looks like a very good first generation machine, which in its larger configuration is comparable in performance to the Intel Touchstone γ machine at NAS. BBN also has been more successful recently in placing its TC2000 in important scientific research laboratories. There are TC2000 machine at Argonne National Laboratories and at CERFACS. A 128 processor machine has been installed at Lawrence Livermore National Laboratories. The most impressive feature of the TC2000 is the switching speed, which results in a very low latency for nonlocal data. The claimed 2.4 μsec is better than almost any other machine in the survey. In spite of this feature, the TC2000 overall is a first generation parallel supercomputer. But since there are no apparent further upgrades or new developments available from BBN, even the high-end version of this machine will be outdated technology in the 1991-92 time frame that we are considering here.

The problem with the machines from Myrias Computer Corporation is that they apparently always lag the current microprocessor technology. Their current machine is based on the Motorola 68020, which is not up to the performance of the 88000 used in the TC2000, even though system prices are at the same level. The next generation will be based on the 68040, which is again no comparison to other machine based on the Intel i860 available in the same time frame. Plans for an SPS-4 have not been made in detail. Myrias claims that the admittedly weaker hardware platform is compensated by better software, in particular by their parallel tools, but this claim has not been verified. In particular there has been no demonstration that the "pardo" construct would lead to easy automatic parallelization of CFD codes. Thus we view the series of Myrias machines as not competitive in their performance with other machines available at the same time.

The Encore series of machines appears to be geared towards providing Giga and Teraops capabilities, but not necessarily the flops capabilities of interest in CFD applications. This appears to be a continuation of the current trend, with little scientific (i.e. floating point) applications use for the current Encore Multimax machine.

The KSR machine will be the most ambitious shared-memory system at the time of its introduction. We believe that KSR and later Tera will offer

very strong competition to the current, conventional vector supercomputer manufacturers, both domestic and Japanese. This is because they are building machines that in concept are quite similar to the standard supercomputers and are programmed in much the same way, but are scalable to large numbers of processor and memory nodes. There is, nevertheless, considerable technical risk involved in the KSR effort. The silicon and hardware is not yet all debugged. The performance of the memory system has been modeled and analyzed by KSR in detail, but there is as yet no actual measurement of its performance. The degree to which the compiler can restructure code to take best advantage of the memory system is the biggest question now: their performance estimates assume best-possible mappings of data and work to the processors.

Tera's machine is in several respects outside the scope of this study as originally intended. There is probably no scaled down version of this machine available, with the exception of a 16 processor prototype. Tera is planning to use an impressive array of new technology. The only possible weakness we see in Tera is its long term approach in a very short term oriented funding environment.

Among the shared-memory machines we thus find that Myrias is not competitive in terms of performance, that BBN currently has no appropriate offering for a second generation parallel supercomputer, that E&S will probably no longer be in the market, and that Encore is not addressing floating-point intensive applications.

Currently KSR and Tera appear to offer the computationally most powerful and competitive machines in the category of shared-memory MIMD machines, with some risks involved in both approaches.

3.2 MIMD distributed-memory machines

In addition to the machines with a detailed description in Section 2 (IBM Vulcan, Intel Touchstone, NCUBE) there are three additional distributed-memory machines of potential interest for which there are no detailed data in Section 2: Meiko, Suprenum, and the "Nosenchuck Machine". In these three cases we did not include further machine details for discussion because it was unlikely that these machines were of interest for CFD applications for various reasons. The Meiko Computing Surface offers currently a very flexible set of different configurations for parallel processing based on the

T800 transputer. Recently Meiko is also offering boards based on the i860 for compute intensive applications. Meiko has no clear plans to provide a very high-end machine for computationally intensive applications such as the ones described in Simon, *et al.* [8] that would be of interest for the next generation system.

Suprenum is a well-established German effort that is strongly supported by the European ESPRIT program. The current machine qualifies as a powerful first generation parallel supercomputer. Furthermore Suprenum claims a rich support environment for parallel processing, as well as support for many large scale applications. There are apparent plans for a second generation machine. However, at this point it was not clear if Suprenum would even distribute their machine in the U.S.[7]

Finally the Navier-Stokes Computer, also called the "Nosenchuck machine," is a prototype design that has been proposed and redesigned for years. Apparently there is currently some support for this machine at DARPA in the DST (Direct Simulation of Turbulence) program. A first review of the proposed machine by DARPA in December 1989 was not particularly positive, and it is not decided whether this machine would ever be built. [8] In order to scale this machine to the claimed performance level of the Tera machine, several unproven technologies need to be employed including a crossbar switch for more than 200 processors. Compared to the Tera machine the chances for success for the Navier-Stokes Computer is small, and it is not considered further in this study.

Even though the IBM Vulcan (TF1) project looks very impressive and of high potential interest to NAS, recent reorganizations and management changes at IBM make it increasingly unlikely that this machine ever will be built. After some initial enthusiasm this project seems to have fallen behind schedule and out of favor. In 1992, however, IBM has begun what seems to be a significant new effort to build a highly parallel machine.

This leaves the Intel Touchstone machine and the NCUBE2 as contenders in the distributed-memory MIMD category. The NCUBE2 machine, when built in its full configuration, is a strong production type second generation parallel supercomputer. NCUBE as a company has a long experience in the

[7]Suprenum closed down in 1991.

[8]This machine was actively marketed by Supercomputer Solutions of San Diego in 1990. The company closed down however towards the end of 1990.

scientific market, and some key accounts, for example at Shell. This good hold on the market should allow NCUBE to be at least moderately successful with their NCUBE2. But NCUBE has several disadvantages as a vendor compared to Intel. The only disadvantage directly related to technological issues concerns processor technology. NCUBE has the disadvantage of developing their own custom chips, whereas Intel Scientific Computers can leverage off the commodity market created by the mass production of Intel microprocessors. This has both the advantage of smaller development costs for Intel, as well as the availability of more software.

The second advantage Intel has compared to NCUBE, is the support through DARPA. This not only makes Intel Scientific Computers less vulnerable towards fluctuations in the market, but also provides access to a wide variety of DARPA sponsored research projects, which ISC is planning to integrate into their Touchstone plans. Currently both the NCUBE2 and the Touchstone γ machine are probably equal and are the leaders in the distributed-memory MIMD area. In a long term we believe that Intel has important competitive advantages and will have faster machines.

Cray Research is also investigating the development of a highly parallel MIMD machine. It appears that it will be a distributed-memory machine and that it will support high-level programming languages in addition to message passing. Cray's architecture has not yet been made public, although it is known that the machine will be based on the DEC alpha microprocessor.

As discussed above, the CM-5 also falls into this category.

3.3 SIMD machines

The only two entrants in the SIMD category are the Maspar MP-1 and the CM-5[9]. From its raw performance the MP-1 is a first-generation parallel supercomputer and thus should be compared to the CM-2 rather than the CM-5. While peak processor performance (measured in Gflops, for these systems) is the most often cited rough measure of a system's speed, it is well known that it is hard to achieve more than some fraction of this performance in real applications. The factors determining this fraction are the application's parallelism, its regularity, and its communication needs. With respect

[9] At the time of writing this chapter, the MIMD character of the CM-5 was not known to us.

Table 8: Comparison of Memory Bandwidth of SIMD Machines

Operation	CM2	MP1
Memory bandwidth	50 Gbytes / sec	13 Gbytes / sec
Memory bandwidth per unit cost	6.4 Kbytes / $-sec	17.3 Kbytes / $-sec
Memory -to- memory 64-bit multiply-add speed	1.5 Gflops	.665 Gflops
64-bit News grid move per 64-bit multiply-add	5	.13
64-bit router time per 64-bit multiply add	100	15

to the machine the relative costs of communication, the cost of synchronization, the vector start-up time, and finally the memory bandwidth are very important.

As was stated above, two critical factors (especially for parallel machines) that determine how much of a computer's peak performance can be achieved are the memory bandwidth and the cost of communication. Experience with the CM-2 has shown these to be two of its weaknesses. In Table 8 we compare the CM-2 and the MP1 from these viewpoints. Our purpose here is to show that a new generation of hardware for SIMD massively parallel machines — one that uses the best full custom VLSI technology to advantage — will be considerably more powerful than the already very impressive CM-2. We expect similar or even better results from Thinking Machines in the CM-5.

A number of other features make it even more likely that achieved performance will be a rather high fraction of peak performance of the Maspar machine. These are

1. Virtualization is handled by the compiler. This allows for some very important optimizations; in particular, the virtualizing loop may be omitted whenever there is no more than one active virtual processor per physical processor.

2. The X-net allows broadcasting along rows and columns of arrays. This is important in matrix computation, for example.

3. Integer computations, data movement, and branching are all much faster

than floating point, and so they should take an insignificant amount of time.

TMC now has an installed base of about 35 machines, and is thus (by this measure and ignoring IBM 3090's) the second largest supercomputer vendor in the U.S. (Cray is the largest.) These facts put TMC in an excellent position both financially (Cray is the largest.) and technically to complete the development of the CM-5. Furthermore the design of the CM-2 has shown that TMC can adapt successfully to the demands of the scientific user community. Extrapolating from the past we expect the CM-5 to address some of the major complaints about the CM-2 by offering faster communication, tighter integration of the floating-point units, more powerful front ends, and a true multi-user environment.

3.4 Systolic machines

Because the GF11 is a special purpose, one-of-a-kind machine, the Intel iWARP is the only machine considered in this category. It shows a very high level of performance among the machines available in 1990. The target applications for systolic computations are highly regular matrix and FFT type computations in areas such as signal and image processing. These will be the primary application areas for a machine such as the iWARP. However, the iWARP exhibits a high level of performance even as a general-purpose machine, and should not be categorically dismissed for computational aerosciences applications.

4 Analysis: Towards the Teraflops

It is clear from the tables in Section 2 and 3 that the rapid development of VLSI-based parallel machines is going to lead to Teraflops systems in the late 90's. There are strong MIMD shared-memory contenders (Tera and KSR) and multicomputers (Intel); we also expect that the DARPA-sponsored development at Thinking Machines will lead to SIMD massively parallel systems as well.

4.1 Shared-memory MIMD versus distributed-memory MIMD versus SIMD

The first question to be answered in determining the direction for supercomputing in CFD is one of architecture. We know that the von Neumann line of machines, the climax of which is the current vector multiheaded supercomputers (two evolutionary steps away from von Neumann already) cannot continue to evolve to meet our needs. For the future, there are several alternative branches and we have to decide which of them to follow.

The surveyed machines are of three general types. (For the purposes of this analysis we are not considering the systolic machines). Some (the BBN, Kendall Square, Myrias, and Tera) are MIMD multiprocessors, that share a unified memory space as do today's supercomputers (Cray, NEC, Fujitsu, Hitachi). As such, these machines are the most like those in use now. Promised performance in 1993 (from Tera) exceed the Y-MP by large factors. These machines clearly show that shared-memory architectures can be scaled up to thousands of processors. Of course, the latency for access to nonlocal memory is high on these machines as on all machines: 5μsecs is typical (whereas floating-point arithmetic takes a few tens of nanoseconds, at most).

The second group (e.g. Intel Touchstone and Ncube) are MIMD *multicomputers*. (They have also been called "message passing" machines, but we prefer the name multicomputer.) In these, each processor has its own memory and may not address the memory of another processor directly. Synchronization and communication are accomplished by messages, sent by one and received by one or several processors. Peak performances compare favorably with the shared-memory alternatives, but not by much: the difference, we feel, is due more to the use of very high performance stock microprocessors in the Intel machines. In both classes of machine hardware costs are roughly the same, with slightly less hardware devoted to interconnect in the multicomputers.

In early multicomputers, memory per node was inadequate. Large programs or large shared data structures that had to be copied in every node were therefore ruled out. The economics of hardware technology now readily permit several tens of megabytes per node (assuming that nodes are 64 bit processors) so that the amount of memory per processor is no longer a problem.

Latency for communication and synchronization is due essentially to the cost of the operating-system call needed to send or receive a message. In the current i386-based- machine latency is roughly 300 μsecs. Intel hopes that using the faster i860 and i860XP-based nodes, and re-implementing the code (using lower overhead, lower function alternative systems such as the Caltech Reactive Kernel, for instance) will reduce this to as little as 10 μsecs. Unlike the original, and indeed the current iPSC/860, these new systems will use a message routing subsystem connected as a grid in two dimensions and implemented by full custom, special purpose VLSI circuits. This has essentially eliminated hardware as a source of significant message passing latency. Bandwidth, however, is still hardware limited by the channel width (which is now 8 bits) to roughly 40 Mbytes/sec.

The fundamental difference between these two architectural species is that the shared-memory machines use hardware to generate messages on program demand, and the messages (words or cache lines) are a few tens of bytes long. The avoidance of a software layer to access remote memory greatly reduces the latency that can be achieved. On the multicomputers, the programmer has the burden of explicitly decomposing the data into its separate local data structures; this can enhance performance given the current state of compiler technology. It results in fewer messages with more information in each, thereby allowing for increased utilization of the network. It also makes programming these machines hard, especially when a computation is irregular or dynamic, for example in local mesh refinement, or in unsteady multiblock calculations with moving blocks.

The third group (Thinking Machines, Maspar) are SIMD machines that have a very large number of processors (65,536 and 16,384 respectively). Performance of nearly 2000 flops/dollar-second is best among current machines, and in most other measures of raw performance these machines shine. In fact, these data bear out the basic analysis of Hillis [4]. who claimed that the von Neumann design forces the main memory (the most costly hardware component in all these machines) to be largely underutilized.

A very interesting feature is the spectacular communication characteristics of the SIMD machines. Both can implement communication by nearest neighbors in a processor grid extremely well: latency for 64-bit data is less than the time for a single floating-point operation on the Maspar machine. Their routers, which implement random communication, have large bandwidth: for the Maspar it is 200 Mbytes/sec. That is not much compared to

the Y-MP (at 2560 Mbytes/sec) but the Y-MP costs about 25 times as much as the Maspar.

The chief drawback of these machines is that their processors are individually quite slow, so that less than completely parallel algorithms can do poorly. Also, the SIMD restriction leads to an inevitable loss of performance: while a few processors are working on enforcing boundary conditions, for example, the others have to wait. Finally, these machines (or rather efficient programs for them) tend to require the use of very large working storage. Thus, they are relatively inefficient in their use of memory. This may be a significant drawback in that it limits the largest problems that can be solved.

A significant advantage of these SIMD systems is that their synchronous hardware never has to wait in order to enforce synchronization on programmed events.

Our general assessment of these architectural classes is:

1. For simulations in which the grids are regular and are static, all three classes now have examples (the BBN TC2000, the Intel iPSC/2, and the CM2) that are comparable in performance to the Y-MP. The next generation of these machines will all be an order of magnitude more capable. All can be scaled to the Teraflops level.[10]

2. The choice will therefore hinge on the issues of how broad is the class of CFD problems that the machine can address efficiently and how difficult is the machine to program.

3. In these respects, we feel that the multicomputers are at a disadvantage. With respect to programmability, the SIMD machines are probably best, but research in compilers (for Fortran 90, possibly) that target the MIMD machines may change this.

4. The SIMD machines are ahead in several key areas of cost/performance today, and they may very well stay there, although our projections in the 1992 - 1993 time period do not show this.

5. In terms of the breadth of applicability the MIMD multiprocessors are most likely the best.

[10]More recent benchmarks have indicated that the performance of these machines is comparable to that of one or two processors of the Y-MP.

4.2 I/O structures in Highly Parallel Supercomputers

It appears that a ratio of 1 word per second of I/O bandwidth per 1000 floating-point operations per second is a reasonable balance. Thus, the Teraflops machine needs a gigaword per second of I/O bandwidth. This is likely to be achieved through the use of hundreds of small disks in a redundant array. The use of these highly parallel disk subsystems now seems to pervade the parallel machine area.

All manufacturers are also developing software to allow the programmer to access the parallel I/O in a relatively straightforward way. Single files are spread across the multiple disks automatically so that they function as one larger and faster virtual disk. Moreover, the MIMD multiprocessor vendors are all providing UNIX variants in which the file system code is multithreaded. In this way, several different threads of user code may make use of the file system simultaneously.

4.3 Programming Models for Parallel Supercomputers

In Section 4.1 we considered the architecture of highly parallel machines. A question of equal or greater importance is, "How will these highly parallel machines be programmed?" This question is one of semantics, not syntax. We may indeed call our language Fortran in 1999; but what will we be able to do in the language, and what will the compiler do for us?

On current supercomputers a combination of software and hardware relieves the programmer of some of the most onerous chores. The highest levels of the memory hierarchy (registers and cache) are hidden. (The SSD on these machines is not, however, and this is a significant difficulty). The details of scheduling the use of the hardware are of no concern.

This is not uniformly true of the highly parallel machines at present. The way that the relationship between the algorithm, the programmer, and the compiler and hardware will evolve is quite important and is also quite uncertain.

For the MIMD multicomputers, at the assembly language level we have one program per node with explicit use of messages to handle data sharing and synchronization. This model is currently the only one supported by the

manufacturers. (This pill usually comes with a C or Fortran flavored sugar coating). While an optimizing compiler shields the programmer from the peculiarities of the node architecture (and allows for portability between machines with different nodes) the programmer sees a machine with no unified memory space. Several alternatives are currently under study by university and commercial researchers:

1. Linda is a programming system that simulates in software an associative shared data space. It has been implemented on multicomputers. The chief questions are ones of efficiency.

2. Virtual shared memory can be simulated using software; an effort at Princeton is underway. Again, efficiency is the chief concern about the viability of this idea.

3. The compiler may undertake to partition the data and the work of an unpartitioned program, inserting message passing calls as needed. This is the most probable future direction for programming these machines. We do not now know how successful the compiler will be at the task of finding and exploiting locality of reference in order to reduce sufficiently the message traffic.

In shared-memory machines access to shared variables is tricky. Semaphores are needed to insure proper synchronization of writes and reads. A number of other synchronization mechanisms such as barriers are available to the programmer. Access to these synchronization tools can be expensive. So is access to nonlocal memory.

There are several large compiler and operating system research programs aimed at improving programmability in this environment. This is a hot research area, and we expect to see significant advances during the 90's. The commercial compiler vendors in the industry are also ready to develop and exploit this research.

Some current research directions in simplifying the programming of these machines are:

1. The development by the Parallel Computing Forum of a standard set of extensions to Fortran 77 to allow the programmer various ways to express parallelism in a program explicitly.

2. The development by machine and compiler vendors of Fortran compilers that automatically find and exploit parallelism at an outer loop level.

3. Operating systems that allow the amount of parallelism used in a job to vary as the characteristics of the computation vary.

4. Dynamic scheduling mechanisms that balance the load between processors at run time.

5. Compiler analysis of entire programs (interprocedural analysis) to allow for better optimization.

6. Automatic decomposition of programs into tasks that require relatively little communication (automatic blocking of algorithms).

The programming style for the SIMD machines is called "data parallelism". In essence, whole arrays are acted upon, elementwise, in parallel. Three levels of abstraction are possible:

1. All operations are done on arrays of one element per physical processor. This is the programmer's view at the assembly language level (microcode on the CM).

2. Operations are done on arrays of one element per virtual processor. Each virtual processor simulates V virtual processors. Thus, array sizes are a multiple of the machine size. The CM "assembly" language Paris works at this level (with the restriction that V is a power of two).

3. Arbitrary arrays and subarrays may be used as operands. Fortran 90 works at this level.

The third of these levels is the most appropriate for serious scientific computing and is, fortunately, soon to be available from all vendors.

A very important advantage of this programming model is reproducibility of results. Because there is a single thread of control, we get the same answer every time the program is run (assuming the same input data). This greatly eases the problem of debugging. The MIMD models on the other hand are nondeterministic. Thus, known bugs are hard to track down since they may

be ephemeral. And there is the insidious possibility that what works today may fail tomorrow.

This raises the possibility of the use of SIMD programming models such as Fortran 90 for the programming of MIMD machines. This is not at all unreasonable. All large scientific codes have a lot of exploitable data parallelism. Synchronization of the MIMD machines at an operation or statement level would be too costly, but it would not be necessary in general. The compiler could insert barriers only where needed to enforce correct flow of data (at the ends of loops, for example). This may well be the style of use of these machines, perhaps with some explicit MIMD constructs used to obtain a few very coarse grained parallel processes.

Let us summarize our thoughts on the programming environment:

1. Fortran 90 is a very promising approach to the programming of many, but not all, parallel supercomputing situations.

2. For more dynamic situations other, less restrictive MIMD languages will be used.

3. Vendor specific programming models and styles ought to be avoided.

4. The machines for which the software issues are most difficult are the multicomputers such as the Intel Touchstone.

5. The easiest machines to program are the SIMD machines, in some sense because they are the least flexible.

6. The data-parallel programming style can be supported on MIMD machines, but the converse is not true.

7. We feel that it is imperative that NAS track the development of the software environment for the shared-memory MIMD model of computing.

5 Summary and Recommendations

We can summarize the results of our survey in the following key conclusions:

1. The most important result from our survey is that there are several impressive efforts under way, which will lead to machines with a peak performance of a few hundred Gflops in the early to mid 90's. All these machines are built using existing technology or relatively low risk technology. They form an important stepping stone towards a peak Tflops capability in the late 90's, and thus make the goal of a sustained Tflops machine by the turn of the century realistic.

2. Progress is being made with four different architectural approaches: shared-memory MIMD, distributed-memory MIMD, SIMD, and systolic. Increased performance, and eventually a Tflops, capability is likely using *any* of these approaches.

3. Contrary to folklore, shared memory is not a significant performance disadvantage. Shared-memory MIMD systems will offer soon a variety of tools which make parallelism more accessible to the production user. Among the shared-memory MIMD alternatives available in 1991, the Kendall Square machine appears to be the most powerful. In fact, of all the machines available in that year, it is still the most attractive, at least on paper. In 1993, again on paper, the Tera machine looks outstanding.

4. In the MIMD distributed-memory area the Intel Touchstone σ machine is the clear performance leader in the 1992/93 time frame.

5. While their compute speeds are quite good, the chief strength of the massively parallel SIMD designs is their very high memory and communication bandwidth. This is illustrated well by the Maspar. They are also relatively easy to program.

6. Systolic machines will offer a fourth alternative. Their applicability to computational aerosciences should be investigated.

References

[1] A. Arlauskas. iPSC/2 system: A second generation hypercube. In Geoffrey Fox, editor, *The Third Conference on Hypercube Concurrent Computers and Applications*, pages 38 – 42. New York, NY, ACM, 1988.

[2] D. Bailey, E. Barszcz, R. Fatoohi, H. Simon, and S. Weeratunga. Performance results on the Intel touchstone gamma prototype. In David W. Walker and Quentin F. Stout, editors, *Proceedings of the Fifth Distributed Memory Computing Conference*, pages 1236 – 1246. Los Alamitos, CA, IEEE Computer Society Press, 1990.

[3] W. Crowther, J. Goodhue, R. Gurwitz, R. Rettberg, and R. Thomas. The Butterfly parallel processor. *IEEE Computer Architecture Tech. Newsletter*, pages 18 – 46, September 1985.

[4] W. D. Hillis. *The Connection Machine.* Cambridge, MA, MIT Press, 1985.

[5] NAS Systems Division, NASA Ames Research Center. *Numerical Aerodynamic Simulation Program Plan*, October 1988.

[6] R. Schreiber. An assessment of the Connection Machine. In H. D. Simon, editor, *Scientific Applications of the Connection Machine, 2nd edition*, pages 379 – 390. Singapore, World Scientific, 1992.

[7] H. D. Simon, editor. *Scientific Applications of the Connection Machine*, Singapore, World Scientific, 1989.

[8] Horst D. Simon, William Van Dalsem, and Leonardo Dagum. Parallel CFD: Current Status and Future Requirements. In Horst D. Simon, editor, *Parallel CFD - Implementations and Results Using Parallel Computers*, pages 1 – 28. Cambridge, MA, MIT Press, 1992.