

Parallel computer architectures: state of the art and trends

Domenico Laforenza

CNUCE-Istituto del CNR, Reparto Calcolo Parallelo, Via S. Maria, 36, I-56100 Pisa, Italy

Received September 12, 1990/Accepted November 13, 1990

Summary. An increasing number of parallel architectures is becoming available for numerically intensive applications. Many chemical problems need intensive calculations due to the complexity of the underlying physical models. Very often these applications show an intrinsic parallelism and therefore can be easily adapted to parallel machines. In the future, in addition to the classical numerically intensive applications, the use of these machines will be extended to a more general purpose use (e.g. data base machines, advanced graphics, AI and expert systems applications, etc.). The principal aim of this paper is to show the state of the art of the commercially available parallel architectures and related trends. A comparison of the main features of shared and distributed memory systems will be presented. The characteristics of coarse and fine grained architectures will be discussed. The analysis will include not only the large-scale machines (usually called “supercomputers”), but also smaller machines (e.g. minisuper and multi-computers) having a very favourable price/performance ratio.

Key words: Parallel computer architectures – Instruction streams – Data streams – Computer memory – Processor granularity

1. Introduction

High-performance computing systems are those at the forefront of the computing field in terms of computational power, storage capability, input/output bandwidth and software [5]. These systems include high-speed general-purpose vector and pipeline machines, special-purpose and experimental systems and scalable parallel architectures.

It will be helpful to begin by defining what we mean by “supercomputing”. Although a canonical definition of “supercomputer” does not exist, the term is usually associated with “the most powerful computer available at any given time”. The adjective “powerful” is defined in terms of: execution rate, memory capacity, and precision [1, 2, 3, 4, 8]. In general, a supercomputer offers speed and capacity significantly greater than the most widely available machines built primarily for commercial use.

In the last two decades supercomputing has become an important complement to experimental and theoretical research and it is a dynamic and expanding field that can be considered strategic for science and engineering. In fact, supercomputing may increase productivity in scientific research and industrial design and manufacturing.

Today, the computer systems may achieve impressive performances and the most advanced computer models can deliver a peak performance of hundreds of MIPS (millions of instructions per second) and more than a thousand MFLOPS (millions of floating point operations per second).

These high performances are mostly achieved using one of the following different architectural/technological approaches [1, 2, 6, 7, 19, 25]:

- a *single* processor may be *pipelined* in such a way that a high performance is obtained overlapping the different phases of a single instruction execution (*pipelining* approach);
- *multiple* processors may be used to execute the *same* program concurrently over a set of different data (*data parallelism*);
- *multiple processors* may be used to execute concurrently different programs, or different parts of the same program (*control parallelism*);
- *multiple execution units* may be used within the same processor to execute concurrently different instructions belonging to the same program (*VLIW* approach);
- a *computational model* differing from the classical control flow model can be used to implement a high performance processor (*Dataflow* or *Demand-driven* approach);

All these approaches allow the machine to deliver a high performance because during program execution some kind of parallelism is exploited.

2. Parallel architectures: a classification of design

A wide variety of different parallel architectures has been proposed. A fair number of them has been implemented, at least as prototypes. However, a large number of them was never implemented (paper machines).

A much-referenced and useful taxonomy of computer architectures was given by Flynn. In 1972 Flynn [19] proposed a classification based on the distinctions between single or multiple data stream and single or multiple instruction streams. That taxonomy attempts to bring some order in this apparently confused situation; according to Flynn's taxonomy the architectures can be divided into four categories:

- SISD (Single Instruction stream/Single Data stream):
 - conventional serial Von Neumann computer (uniprocessor);
 - one stream of instructions, each operating on a single data;
 - one arithmetic instruction initiates one arithmetic operation;
 - belong to this class serial scalar computer.
- SIMD (Single Instruction stream/Multiple Data stream):
 - a single stream of instructions, each operating on multiple data;
 - the processors perform the same instruction at every machine cycle, but each operating on different data;

- belong to this class all machines with vector instructions;
 - regardless whether vector processing is realised by pipelining or by building arrays of processors.
- Examples: (pipelined) Cray-1, Fujitsu VP, NEC SX, Hitachi S820-80, etc.; (array processors) Illiac IV, ICL DAP, TM Connection Machine, etc.
- MISD (Multiple Instruction stream/Single Data stream):
 - several instructions operating on the same data item simultaneously;
 - this is an empty class.
 - MIMD (Multiple Instruction stream/Multiple Data stream)
 - several instructions operating on different data item simultaneously;
 - each processor can execute different portions of the same program or completely different programs;
 - belong to this class many multiprocessor and multicomputer systems.

Examples: Cray X-MP, Intel iPSC, Cray Y-MP, NCUBE 2, IBM 3090 Multiprocessors, Alliant FX, Convex C, etc.

3. Some different ways to exploit parallelism

3.1. *Pipelining approach*

The pipelining approach exploits fine grain parallelism at the subinstruction level. Actually, pipelined computers overlap either the fetch, decode and execute phases of any single instruction, or the different phases of a single instruction execution [1, 2, 25].

Sometimes, both the former and the latter form of phase overlapping are present in a single computer system. Very often, such pipelined computers are supplied with some kind of vector feature that allows vector operations on vector registers. The performance values delivered by these machines are in general very high, reaching to some hundred MIPS or MFLOPS. Most of the parallelism exploited in such pipelined architectures is extracted from standard sequential programs by means of some kind of optimizing and vectorizing compiler.

Pipelined computers supplied with vector computing facilities may need some minimum amount of implicit parallelism to be present in the programs that have to be executed in order to fill up the vector pipelines. In general, it is very hard to reach peak performance values claimed by vendors.

3.2. *SIMD approach*

The SIMD approach exploits parallelism at an instruction or process level, by allowing multiple processing agents to perform the same operation (or process) over different sets of input data.

Very different machines fall into this class of computer systems, such as systolic arrays [12], the TM Connection Machine [26] or vector coprocessors. As the parallelism is exploited at data level, i.e. in the concurrent execution of equal programs over different input data sets, SIMD computers are well suited for the execution of a restricted class of algorithms.

3.3. MIMD approach

The MIMD approach exploits parallelism at a *coarse grain* level, as it allows multiple execution units to run concurrently a set of programs or independent parts of their. The MIMD approach to high performance is probably the best candidate to exploit VLSI potentialities [10, 18]. However, it has to be noticed that MIMD machines deliver good performance speedups only if they are programmed properly. In particular, there exists an optimal process grain for the programs that have to be run on a MIMD machine.

If a set of processes which have a greater or smaller grain than the optimum one is run over the MIMD machine, then the speedup will not be in general neither linear in nor proportional to the number of processors present in the machine configuration.

3.4. VLIW approach

VLIW (Very Long Instruction Word) approach exploits *fine-grain* parallelism between the execution of different instructions belonging to the same program, using parallel processors which are able to execute far more than a single operation in each processor cycle [23]. As in the pipelined machines case, the VLIW approach allows impressive performances to be achieved, also due to the particular kind of optimizing compilers used in the translation of high level language programs into machine code.

3.5. Dataflow and reduction approaches

Finally, both dataflow and reduction computer systems exploit parallelism at the instruction level [24]. Dataflow computer systems use different functional units to perform multiple instructions at a time, under the supervision of a dataflow control unit.

Data driven computer systems are very similar to dataflow machines, but for the control unit organization, which operates under a reduction computational rule. Dataflow computers have been demonstrated to be well suitable for the execution of scientific codes, and are usually programmed using single-assignment functional-like programming languages. At the moment the impact of these machines on the market is poor and the major part of them are confined in the research environments. Prototype examples of dataflow machines are: SIGMA-1 (Electrotechnical Laboratory, Japan), Manchester dataflow machine (University of Manchester, U.K.), MIT dataflow machine (MIT, USA), etc.

4. Parallel architectures: other classifications

Other classifications are also possible. These may lead to further subdivisions of above discussed classes into specific subdomains.

4.1. Memory management classification

A classification related to the way the technology is presently developing, divides parallel machines, in particular MIMD systems, into two further subclasses: *shared* and *distributed* memory architectures [1, 2, 4, 8].

Shared-memory architectures are composed of a varying number of processors and memory modules connected by means of a high-speed interconnecting network, such as a cross-bar switch, a bus, or another efficient routing network. All processors share all memory modules and have the ability of executing different instructions on each of the processors using different data streams. One limitation of the shared-memory approach is that it may be difficult or expensive to make a memory that can serve a large number of processors simultaneously. Machines like: Cray X-MP, Cray 2, Cray Y-MP, IBM 3090 Multiprocessors, Alliant FX, Convex C, Sequent Balance, etc., fall into this class.

Distributed-memory architectures are composed of a varying number of processing nodes, each containing one or more processors, local memory, and communication interfaces to other nodes. These architectures are scalable, have no memory shared among the processing nodes, exchange data through their network connections, and execute independent (multiple) instruction streams using different data streams. The most popular architecture in this class is the hypercube. Machines like: Intel iPSC, NCUBE 2, TM Connection Machine, Meiko Computing Surface, etc., fall into this class.

4.2. Granularity classification

A classification in terms of “size” and “number” of processors available in a parallel machine is usually expressed in terms of “granularity” [16]. In this sense, the architectures are divided into:

- fine-grained machines (large number of small processors – hundreds or thousands);
- coarse-grained machines (small number of powerful processors – typically from two to 16).

The boundaries of this definition change with time and it is not always clear where the dividing line comes in between. For example, the power per processing element for any given degree of parallelism will increase with VLSI capabilities.

Another way of classifying commercial parallel architectures is to consider these products from the industry point of view. In this way, the parallel machines available on the market are divided into two separate clusters:

- Farms
- Cubes.

The characteristics of a typical “farm” are:

- small number of processors (generally, no more than eight);
 - processor performance equivalent at least to that of a minicomputer;
 - processor-to-processor communication via shared memory;
 - compilers recognizing the opportunity for parallelism;
 - run existing software in a parallel mode with a minimum of modification.
- Examples: Cray X-MP, Cray Y-MP, Cray-2, NEC SX-3, IBM 3090 Multiprocessors, Alliant FX/8, Convex Cx, Sequent Balance 21000, Suprenum, etc.

The characteristics of a typical “cube” are:

- large number of small processors (generally a power of two: 128, 1024, etc.);
- each processor has its own local memory and communicates over a network;
- network topology: mesh, ring, hypercube, etc.;
- the computational network is driven by a separate host computer (mini or workstation);
- the software (in general) has to be specially developed;
- to make efficient simultaneous use of a large number of processor occurs to rethink the parallel algorithm.

Examples: Intel iPSC, NCUBE, FPS T-series, Meiko Computing Surface, TM Connection Machine, etc.

Although the previous classifications give helpful coarse divisions, by examining real systems we find immediately that the situation is more complicated, with some architectures exhibiting aspects of more than one category [4, 6, 16].

Many of today’s machines have actually a hybrid design. For example, the Cray X-MP has up to four processors and can be considered a MIMD-architecture, but each processor uses pipelining (SIMD) for vectorization. A CRAY X-MP can be classified as “shared-memory” architecture, but also “coarse-grain” system or “farm” machine. Other examples are represented by BBN TC2000 (Butterfly) and IBM RP3.

5. A more “market-oriented” classification

For more practical purposes, more “market-oriented” classifications have also been proposed [3, 4, 9]. Obviously these classifications become more and more arbitrary as the complete spectrum of high-performance computing grows. For this reason it is possible that classes that once had relatively clear boundaries have blurred together. Many models, in fact, include a large range of options that may extend across more than one of these classes.

5.1. *Supergraphics workstations (single user systems)*

The supercomputer market has been expanded by the introduction of supercomputing workstations; the machines, from a growing number of vendors, combine substantial computing power with high-quality visualization features. For that, vendors of minisupers and highly-parallel systems are increasingly building strong graphics capabilities into their systems. We may include in this class, a desktop or other compact systems, typically priced in the \$50,000 to \$150,000 range which offer strong visual capabilities as well as more computational power than ordinary workstations.

Examples include models from Alliant Computer Systems Corp., Hewlett-Packard Corp. (Apollo), AT&T Pixel Machines, Digital Equipment Corp., Silicon Graphics Inc., Stardent Computer Inc., Stellar Computer Inc. and Sun Microsystems Inc.

5.2. Minisupercomputers

By “minisuper” one often means a computer which costs 0.1–1 M\$ and which has more than 10 MFLOPS peak performance per processor. In general, a system offering a significant fraction of supercomputer power (typically in the 100 to 500 MFLOPS peak performance range) at prices ranging from \$200,000 to \$2 million or so, with a typical price around \$500,000. The prices are proportional to the number of processors, amount of memory and similar differences that provide added performance.

Minisupers are used widely as production machines and are also used as teaching/research platforms.

A growing number of minisupers, including Alliant, Convex, Encore and Sequent are multiprocessor machines. Especially in terms of software requirements, they have much in common with highly parallel systems. Some of the vendors are: Alliant Computer Systems Corp., CONVEX Computer Corp., ELXSI Corp., Encore Computer Corp., FPS Computing, MIPS Computer Systems, Inc., Multiflow Computer Inc. (no longer in business), Pyramid Technology, Sequent Computer Systems and Supertek Computers Inc. (a firm recently acquired by Cray).

An alternative in this class is the “attached” (array) processor. Companies such as FPS Computing, Star Technology and CSPI are actively marketing these “add-on” products in an effort to attract current supercomputer users interested in having a better price/performance ratio on specific problems.

5.3. Vector mainframes

Since the early 1980s, some *general purpose* mainframes capable of enhancing their computational capabilities by using particular vector processors were introduced in the market. These vector features allow machines produced for general purpose applications to offer enhanced numerical capabilities. In some cases, the ability to attach vector features is extended to more than one processor in multiprocessing mode.

These solutions were adopted to avoid problems and costs related to the installation of a big supercomputer (needs of: front-end systems, software environment duplication, site preparation requirements: i.e., well-engineered cooling systems, appropriate floor structures, etc.).

Companies currently offering such vector-processing capabilities include IBM (IBM 3090/VF), Control Data (CDC 180/995E), Hitachi (Hitachi IAP S-8, NAS AS/91XO), Unisys (Unisys 1190/ISP), Digital (VAX 9000), etc.

5.4. Near-supercomputers

A new market niche, distinguished from top-end supercomputers primarily by lower cost and performance (in the 200 to 500 claimed MFLOPS range) close to that usual for supercomputers of the mid-1980s rather than the leading-edge supercomputers of 1990.

The price range is \$1.5 to \$5 million with the larger versions of the IBM 3090s extending up to \$10 million or more. Current occupants include the ETA-IOQ (out of business), CRAY X-MP/14se and single-processor CRAY X-MPs.

IBM 3090s with vector features differ in a number of respects from other machines in this category, including lower clock speeds and an emphasis on scalar capabilities: they have more power than minisupers, yet are not at the top-end supercomputer level. The lower-level Fujitsu models fall also into this category.

Some of the most powerful minisupers and highly-parallel systems offer performance near this level.

5.5. Top-end supercomputers

Models that meet the classical definition of supercomputer. These machines provide more than 500 MFLOPS peak performance, and the latest have potential output of a GigaFLOPS or above (TeraFLOPS). Top-end supercomputers are used primarily as production systems, supporting research in a wide range of disciplines.

This class includes multiprocessor versions of the CRAY X-MP, CRAY-2 and CRAY Y-MP, the most powerful options from Fujitsu, Hitachi and NEC, and ETA-IOF and ETA-IOG (ETA is no longer in business). The most powerful IBM 3090s, with six processors and vector features, perform at the lower boundary of this category; in March 1989, Cray Research broadened its product line in this category by introducing single-, double- and four processor versions of the Y-MP, with a broad range of memory options. Forthcoming machines in this category, already announced, will include the 16-processor CRAY Y-MP (1992), the Fujitsu VI-2000 (1990: multiprocessor in 1991), 64-processor CRAY 4 (1993), a 64-processor machine from Supercomputer Systems, Inc., currently code-named the S-1 and produced with support from IBM (1992).

5.6. Highly parallel systems

These are typically highly scalable systems with a number of processors ranging from two to several thousands. Hypercube architectures dominate this field. Three features distinguish these machines from other parallel systems: distributed memory instead of a shared common memory, message based operating systems instead of shared variable operating systems, a large network of processors (thousands instead of a dozen or so).

Highly parallel systems are beginning to gain acceptance as "production" systems and are widely used in teaching programs, especially in the many universities exploring the theory and practice of parallel processing.

The range of prices and performance is equally broad. The largest Connection Machine approaches supercomputer levels in both prices and potential performance. Several other vendors offer models whose maximum is of the order of dozens, hundreds or thousands processors with minimum prices well under \$100,000 and maximum from \$3 to \$4 million.

Manufacturers include: Active Memory Technology, Inc. (from 1,000 to 4,000 processors), BBN Advanced Computers Inc. (from 1 to 512 processors), Flexible Computer Corporation, Integrated Parallel Systems (from 500 to 4,000 processors), Intel Scientific Computers (hypercube: Intel microprocessors are the basis for nodes; from 16 to 128 processors), International Parallel Machines, Inc. (from 1 to 33 processors), Meiko Scientific Corp. (transputer-based:

Table 1. List of some high-performance computing systems

(*)	Computer	Max No. process.	Clock cycle (ns)	Peak perform (MFLOPS)
s	CRAY-1 (1976)	1	12.5	160
tes	CRAY X-MP (1985)	4	8.5	940
tes	CRAY-2 (1985)	4	4.1	1,941
tes	CRAY Y-MP (1988)	8	6.0	2,700
tes	CRAY 3 (1990)	16	2.0	16,000
tes	CRAY 4 (1992)	64	1.0	128,000
s	CDC 205 (1982) (4-pipes)	1	20.0	400
tes	ETA-10E (1987)	4	10.5	1,700
ns	ETA-10P (1987)	2	24.0	375
tes	FUJITSU VP-200 (1984)	1	7.0	857
tes	FUJITSU VP-2000 (1990)	1	4.0	4,000
tes	HITACHI S-810/20 (1983)	1	14.0	620
tes	HITACHI S-820/80 (1988)	1	4.0	3,000
s	NEC SX2-100 (1980)	1	6.0	285
tes	NEC SX-3 (1990)	4	2.9	22,000
vm	IBM 3090/180E VF (1988)	1	17.2	116
tes	IBM 3090/600 (J-JH)	6	14.5	828
ms	ALLIANT FX/8 (1985)	8	170.0	94
ms	ALLIANT FX/80 (1987)	8	85.0	188
ms	CONVEX C-1 (1984)	1	100.0	20
ms	CONVEX C-240 (1987)	4	40.0	200
ms	SCS-40 (1986)	1	45.0	44
sgw	ARDENT TITAN (1988)	4	62.5	64
ms	FPS-500 (1988)	2/4	30.0	33 + 133
sgw	STELLAR GS2000 (1989)	4	50.0	80
hps	INTEL iPSC/2 (1988)	128	16.0 MHz	1,280
hps	INTEL iPSC/860 (1989)	128	40.0 MHz	7,600
hps	NCUBE 2 (1989)	8192	20.0 MHz	27,000
hps	Thinking Machines CM-2 FP	65536	8.0 MHz	28,000

(*) Computer class: tes = top-end supercomputer; s = supercomputer; ns = near-supercomputer; ms = minisupercomputer; vm = vector mainframe; sgw = supergraphics workstation; hps = highly-parallel system

reconfigurable; the largest system currently operating has 1,024 nodes), Microway Ltd. (transputer-based), NCUBE Corp. (hypercube: custom microprocessors are the basis for nodes; from 16 to 8,192 processors), PARSYTEC Inc. (transputer-based parallel processors), Thinking Machines Corp. (Connection Machine has up to 64,000 nodes).

6. Conclusions: parallel architectures trends

For a decade, technology advances always and primarily constituted a major driving force for the developments in computer architecture. In particular, supercomputer power has increased primarily through higher clock speed, more efficient chips, larger memories, and efficient vector calculations.

To enhance the speed of the computer, the components must be packed very

tightly. Too closely packaging, however does not allow an efficient heat dissipation. Sophisticated cooling systems are now being used and studies on some special chip materials capable of generating less heat are in progress (substances like gallium arsenide, Josephson junctions, HEMT, etc.) [1, 2, 8]. The speed of any circuit is, however, bound to that of the electric signal. In the long term such a problem may be bypassed by using optical and biological/molecular devices. Optical circuits (based upon devices that switch light rather than electricity) although already available are still far from being assembled into a commercial high speed computer. At the same time, there is not yet clear consensus about the viability of biological/molecular computing, although this technology is an exciting research area.

As a result, the only practical way of getting high computing speed is to make use of multiple processors architectures: several or very many processors operate on a problem at the same time. The commercial availability of parallel computer systems is expanding rapidly and, in the short term (about 3–5 years), it seems highly probable that all transaction processing systems (parallel systems that exploit the natural parallel structure of UNIX, i.e.: SEQUENT Balance, ENCORE Multimax, etc.) will become an industrial standard for mainframes, minicomputers, workstations and even personal computers; in this way the parallel computers will become the accepted commercial norm [8, 13, 16].

An obstacle to parallel restructure existing codes is the loss of validation provided by the usage.

Researchers, in fact, need performance and are willing to pay extra effort to achieve goals otherwise unachievable. As example, several applications have been written using LISP or C language to get the best performance out of Connection Machine or the hypercubes [17].

In order to develop suitable approaches to the use of novel highly parallel architectures, the gap between the universities, public research centers and industrial environments interested to large problems (big codes) should be reduced. Universities and public institutions must introduce parallel architectures inside their organizations and must offer special training in using them; the role of the universities to develop parallel software will be critical. Nowadays, good examples in this sense are: ACRF-Argonne National Laboratory (Argonne, Illinois, USA), Caltech Concurrent Computation Program (Caltech, Pasadena, USA), Edinburgh Concurrent Supercomputer project (University of Edinburgh, UK) and in a smaller scale subprojects of the "Progetto Finalizzato: Sistemi Informatici e Calcolo Parallelo" of the Italian National Research Council (CNR) [11, 20, 21, 22].

Figure 1 shows the possible organization of a computing center hosting among other computer facilities a highly-parallel machine. The presence of a highly parallel system raises an interesting question: will the access to this machine remain restricted to a small set of specialized users or will it be possible, in the future, to offer parallel computing as a general-purpose service? [11, 21].

In fact, it is interesting to ask which roles will parallel machines play with the whole range of computing needs of large organizations and corporations. We can define these roles as: general-purpose parallel computing.

At the moment it is realistic to think that the very-large-distributed-MIMD machine (i.e. hypercubes with thousands of nodes) are going to be used just for big problems, where the scientists need of performances in the 20,000 to 30,000 MFLOPS range [17].

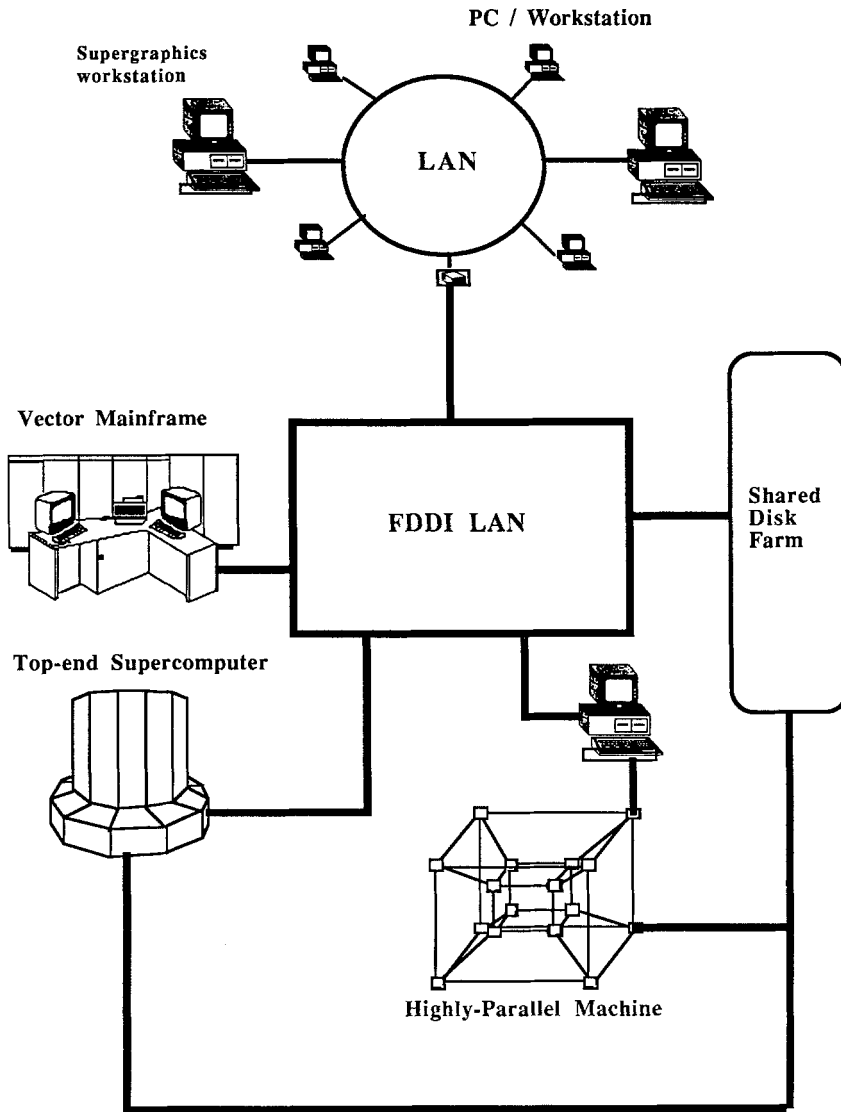


Fig. 1. A possible computing center hosting a highly parallel machine.

It is clear that, to permit a wider use of these systems, we need that the parallel architectures will lose their special-purpose tag and provide the solution to scalable, general-purpose computing performance [10, 17].

Moreover, in order to protect the user's parallel software investments and to make the growth of a significant third-party parallel software industry possible, it is essential to be able to guarantee portability and standard languages across a range of parallel hardware.

The principal question is: "In which manner is it possible to keep separate the hardware and the parallel software concerns?"

Many researchers believe that such a separation of parallel hardware and software issues is possible, and some of them (i.e. MIT, Argonne National Lab., Caltech, Southampton, etc.) proposed a primitive set of mechanisms that attempts to separate issues of programming models from issues of machine organization [14].

Others are currently involved in proving that it is possible to emulate shared-memory models of computation on distributed memory machines with only a constant inefficiency factor given sufficient parallelism [15].

The computer scientists are studying an abstraction capable of unifying the apparently different distributed memory and shared memory architectures and allow a uniform programming model and standard languages to be supported.

A good way to approach the conclusion of this paper is to cite an interesting panel discussion held recently in Los Angeles, during a conference concerning "Parallel Computational Fluid Dynamics" [17]. The panel coordinator asked the panellists: "Are highly parallel systems ready for prime time?".

The best answer was: we do not know whether it is prime time for massive parallelism, but it is a very exciting time. It is a synergistic, evolving development, and we are at a very exciting phase. I think the next years will have profound impact for the entire decade to come. In any event the prime time for parallelism is closer than we think.

Acknowledgements. This work was supported by Finalized Project "Informatic Systems and Parallel Computing" (Subproject No. 8 - Parallel Computing Support Initiative) of the Italian National Research Council (CNR).

References

1. Hockney RW, Jesshope CR (1988) *Parallel Computers 2*, Adam Hilger
2. Hwang K, Briggs FA (1985) *Computer architecture and parallel processing*. McGraw-Hill, New York
3. Karin S, Parker Smith N (1987) *The supercomputer era*. Hartcourt Brace Jovanovich Publ
4. Dongarra JJ (1989) Overview of current high-performance computers, *Proceedings in Supercomputing Vol 1*, p 3
5. The Federal High Performance Computing Program (September 1989) Executive Office of the President, Office of Science and Technology Policy
6. Laforenza D (Gennaio 1986) *Elaborazione Parallela e Calcolo Vettoriale*, Informatica Oggi Gr. Editoriale Jackson, Anno 6-N. 13, pp. 42-71
7. Danelutto M (1990) A massively parallel architecture using VLIW for fine grain parallelism exploitation, PD Thesis TD-5/90, Dipartimento di Informatica, Università di Pisa
8. Treleaven PC (1988) Parallel architecture overview, *Parallel Computing* 8:59. North-Holland
9. Choosing a High-Performance Computing System (April 1989) *Supercomputing Review*
10. Fox GC et al. (1988) *Solving problems on concurrent processors. Vol I*. Prentice Hall
11. Fox GC (September, 1989) Parallel computing comes of age: supercomputer level parallel computations at Caltech, *Concurrency: practice and experience. Vol 1(1)*. Wiley, New York
12. Kung HT (1989) in: Elliott RJ, Hoare CAR (eds) *Scientific Applications of Multiprocessors*. Prentice Hall
13. Hey AJG (1990) Supercomputing with transputers - past, present and future, *Proceedings of 1990 Int Conf Supercomputing*, ACM Press, Amsterdam, p 479
14. Dally WJ, Wills DS (1989) *Universal mechanisms for concurrency*, MIT research report
15. Valiant LG (1990) *Bulk-Synchrony: A bridging model for parallel computation*. To be published in the *Proceedings of DMCCS*, Charleston

16. Johnson T, Durham T (1986) Parallel processing: the challenge of new computer architectures. OVUM, London
17. Perspectives: Are Highly-Parallel Systems Ready for Prime Time? (1990) The International Journal of Supercomputer Applications, Vol 4(1) Spring 1990, p 88. Massachusetts Institute of Technology
18. Fox GC, Messina PC (October, 1987) Advanced computer architectures. Scientific American
19. Flynn MJ (September, 1971) Some computer organizations and their effectiveness, IEEE Trans on Comp
20. Progetto Finalizzato "Sistemi Informatici e Calcolo Parallelo" (1989) Progetto Esecutivo, Consiglio Nazionale delle Rcerche, Roma
21. Edinburgh Parallel Computing Centre (April, 1990), Newsletter Number 10
22. The Advanced Computing Research Facility (ACRF) (1989) General documentation, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Illinois
23. Fisher JA (July, 1984) The VLIW machine: A multiprocessor for compiling scientific code, IEEE Computer
24. Treleaven PH, Brownbridge DR, Hopkins RH (March, 1982) Data-driven and demand-driven computer architecture. ACM Computing Surveys, Vol 14(1)
25. Baiardi F, Tomasi A, Vanneschi M (1987) Architettura dei sistemi di elaborazione, Franco Angeli Libri, Milano
26. Boghosian BM (1990) Computational physics on the connection machine, Computers in Physics, Vol 4(1)